Cryptography
CS 555

Topic 30: El-Gamal Encryption

Recap

- CPA/CCA Security for Public Key Crypto
- Key Encapsulation Mechanism

A Quick Remark about Groups

- Let $\mathbb{G}$ be a group with order $\mathrm{m} = |\mathbb{G}|$ with a binary operation $\circ$ (over G) and let $\mathrm{g}, \mathrm{h} \in \mathbb{G}$ be given and consider sampling $\mathrm{k} \in \mathbb{G}$ uniformly at random then we have

$$\Pr_{\mathrm{k} \leftarrow \mathbb{G}}[k = g] = \frac{1}{m}$$

Question: What is $\Pr_{\mathrm{k} \leftarrow \mathbb{G}}[k \circ h = g] = \frac{1}{m}$?

Answer:

$$\Pr_{\mathrm{k} \leftarrow \mathbb{G}}[k \circ h = g] = \Pr_{\mathrm{k} \leftarrow \mathbb{G}}[k = g \circ h^{-1}] = \frac{1}{m}$$

A Quick Remark about Groups

Lemma 11.15: Let $\mathbb{G}$ be a group with order $\mathrm{m} = |\mathbb{G}|$ with a binary operation ∘ (over G) then for any pair $\mathrm{g}, \mathrm{h} \in \mathbb{G}$ we have

$$\Pr_{\mathrm{k} \leftarrow \mathbb{G}}[k \circ h = g] = \frac{1}{m}$$

Remark: This lemma gives us a way to construct perfectly secret private-key crypto scheme. How?

El-Gamal Encryption

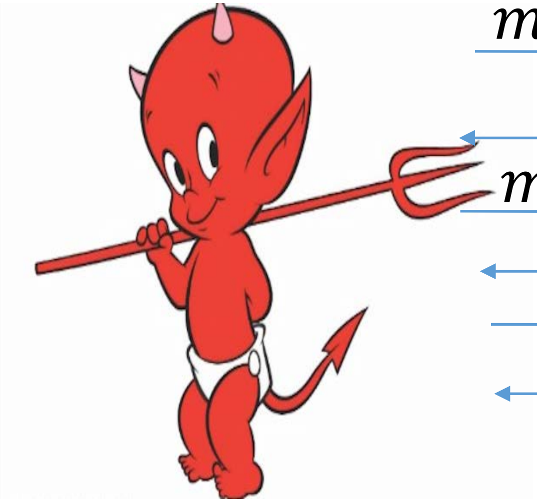
- Key Generation ($\text{Gen}(1^n)$):
  1. Run $\mathcal{G}(1^n)$ to obtain a cyclic group $\mathbb{G}$ of order q (with $\|q\| = n$) and a generator g such that $< \mathrm{g} >= \mathbb{G}$.
  2. Choose a random $\mathrm{x} \in \mathbb{Z}_q$ and set $h = g^x$
  3. Public Key: $\mathrm{pk} = \langle \mathbb{G}, q, g, h \rangle$
  4. Private Key: $\mathrm{sk} = \langle \mathbb{G}, q, g, x \rangle$
- $\mathrm{Enc}_{\mathrm{pk}}(m) = \langle g^y, m \cdot h^y \rangle$ for a random $\mathrm{y} \in \mathbb{Z}_q$
- $\mathrm{Dec}_{\mathrm{sk}}(c = (c_1, c_2)) = c_2 c_1^{-x}$

El-Gamal Encryption

- $\mathrm{Enc}_{\mathrm{pk}}(m) = \langle g^y, m \cdot h^y \rangle$ for a random y $\in \mathbb{Z}_q$
- $\mathrm{Dec}_{\mathrm{sk}}(c = (c_1, c_2)) = c_2 c_1^{-x}$

$$\begin{aligned}
\mathrm{Dec}_{\mathrm{sk}}(g^y, m \cdot h^y) &= m \cdot h^y (g^y)^{-x} \\
&= m \cdot h^y (g^y)^{-x} \\
&= m \cdot (g^x)^y (g^y)^{-x} \\
&= m \cdot g^{xy} g^{-xy} \\
&= m
\end{aligned}$$

CPA-Security ($\mathrm{PubK}_{A,\Pi}^{\mathrm{LR-cpa}}(n)$)



Public Key: pk

m_0^1, m_1^1

$\boldsymbol{c_1 = \mathrm{Enc}_{pk}(m_b^1)}$

m_0^2, m_1^2

$\boldsymbol{c_2 = \mathrm{Enc}_{pk}(m_b^2)}$

m_0^3, m_1^3

$\boldsymbol{c_3 = \mathrm{Enc}_{pk}(m_b^3)}$

⋮

b'

Random bit b

(pk,sk) = Gen(.)

$$\forall PPT\ A\ \exists \mu\ (\text{negligible})\ \text{s.t}$$

$$\Pr\left[\mathrm{PubK}_{A,\Pi}^{\mathrm{LR-cpa}}(n) = 1\right] \le \frac{1}{2} + \mu(n)$$

El-Gamal Encryption

- $\mathrm{Enc}_{\mathrm{pk}}(m) = \langle g^y, m \cdot h^y \rangle$ for a random y $\in \mathbb{Z}_q$
- $\mathrm{Dec}_{\mathrm{sk}}(c = (c_1, c_2)) = c_2 c_1^{-x}$

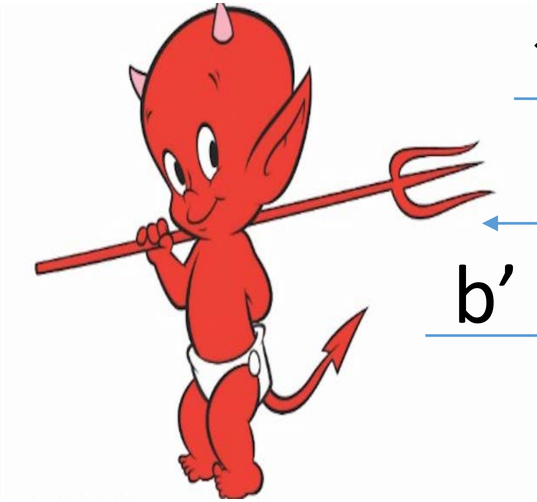
Theorem 11.18: Let $\Pi = (Gen, Enc, Dec)$ be the El-Gamal Encryption scheme (above) then if DDH is hard relative to $\mathcal{G}$ then Π is CPA-Secure.

Proof: Recall that CPA-security and eavesdropping security are equivalent for public key crypto. It suffices to show that for all PPT A there is a negligible function **negl** such that

$$\Pr\left[\mathrm{PubK}_{\mathrm{A},\Pi}^{\mathrm{eav}}(\mathrm{n}) = 1\right] \le \frac{1}{2} + \mathbf{negl(n)}$$

Eavesdropping Security ($\mathrm{PubK}_{A,\Pi}^{\mathrm{eav}}(n)$)

Public Key: pk



m_0 , m_1

$\boldsymbol{c_1 = \mathrm{Enc}_{pk}(m_b)}$

b’

Random bit b

(pk,sk) = Gen(.)

$$\forall PPT\ A\ \exists \mu\ (\text{negligible})\ \text{s.t}$$

$$\Pr\left[\mathrm{PubK}_{A,\Pi}^{\mathrm{eav}}(n) = 1\right] \leq \frac{1}{2} + \mu(n)$$

El-Gamal Encryption

Theorem 11.18: Let $\Pi = (Gen, Enc, Dec)$ be the El-Gamal Encryption scheme (above) then if DDH is hard relative to $\mathcal{G}$ then Π is CPA-Secure.

Proof: First introduce an `encryption scheme' $\widetilde{\Pi}$ in which $\widetilde{\mathrm{Enc}}_{\mathrm{pk}}(m) = \langle g^y, m \cdot g^z \rangle$ for random $\mathrm{y}, \mathrm{z} \in \mathbb{Z}_q$ (there is actually no way to do decryption, but the experiment $\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\widetilde{\Pi}}(\mathrm{n})$ is still well defined). In fact, (using Lemma 11.15)

$$\Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\widetilde{\Pi}}(\mathrm{n}) = 1\right]$$

$$= \frac{1}{2}\Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\widetilde{\Pi}}(\mathrm{n}) = 1 | b = 1\right] + \frac{1}{2}\left(1 - \Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\widetilde{\Pi}}(\mathrm{n}) = 1 | b = 0\right]\right)$$

$$= \frac{1}{2} + \frac{1}{2}\Pr_{\mathrm{y,z} \leftarrow \mathbb{Z}_q}[A(\langle g^y, m \cdot g^z \rangle) = 1] - \frac{1}{2}\Pr_{\mathrm{y,z} \leftarrow \mathbb{Z}_q}[A(\langle g^y, g^z \rangle) = 1]$$

$$= \frac{1}{2}$$

El-Gamal Encryption

Theorem 11.18: Let $\Pi = (Gen, Enc, Dec)$ be the El-Gamal Encryption scheme (above) then if DDH is hard relative to $\mathcal{G}$ then Π is CPA-Secure.

Proof: We just showed that

$$\Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\widetilde{\Pi}}(\mathrm{n}) = 1\right] = \frac{1}{2}$$

Therefore, it suffices to show that

$$\left|\Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\Pi}(\mathrm{n}) = 1\right] - \Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\widetilde{\Pi}}(\mathrm{n}) = 1\right]\right| \leq \mathbf{negl}(n)$$

This, will follow from DDH assumption.

El-Gamal Encryption

Theorem 11.18: Let $\Pi = (Gen, Enc, Dec)$ be the El-Gamal Encryption scheme (above) then if DDH is hard relative to $\mathcal{G}$ then Π is CPA-Secure.

Proof: We can build $B(g^x, g^y, Z)$ to break DDH assumption if Π is not CPA-Secure. Simulate eavesdropping attacker A

1. Send attacker public key $\mathrm{pk} = \langle \mathbb{G}, q, g, h = g^x \rangle$
2. Receive m_0,m_1 from A.
3. Send A the ciphertext $\langle g^y, m_b \cdot Z \rangle$.
4. Output 1 if and only if attacker outputs b'=b.

$$\left|\Pr\left[B(g^x, g^y, Z) = 1 \middle| Z = g^{xy}\right] - \Pr\left[B(g^x, g^y, Z) = 1 \middle| Z = g^z\right]\right|$$
$$= \left|\Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\Pi}(\mathrm{n}) = 1\right] - \Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\widetilde{\Pi}}(\mathrm{n}) = 1\right]\right|$$
$$= \left|\Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\Pi}(\mathrm{n}) = 1\right] - {}^{1}/_{2}\right|$$

El-Gamal Encryption

- $\mathrm{Enc}_{\mathrm{pk}}(m) = \langle g^y, m \cdot h^y \rangle$ for a random y $\in \mathbb{Z}_q$ and $h = g^x$,
- $\mathrm{Dec}_{\mathrm{sk}}(c = (c_1, c_2)) = c_2 c_1^{-x}$

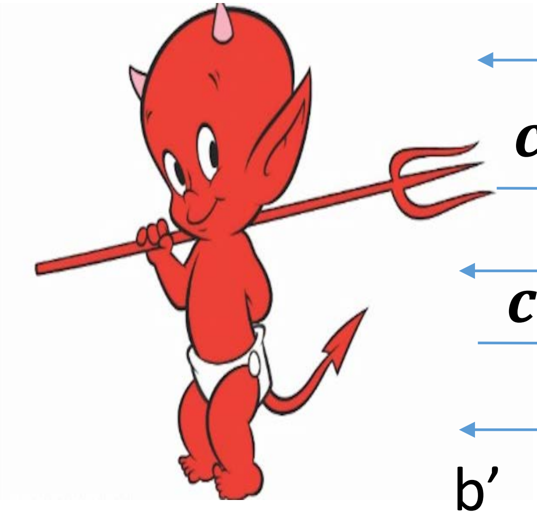
Fact: El-Gamal Encryption is malleable.

$$\mathrm{c} = \mathrm{Enc}_{\mathrm{pk}}(m) = \langle g^y, m \cdot h^y \rangle$$
$$\mathrm{c}' = \mathrm{Encpk}(m) = \langle g^y, 2 \cdot m \cdot h^y \rangle$$
$$\mathrm{Dec}_{\mathrm{sk}}(c') = 2 \cdot m \cdot h^y \cdot g^{-xy} = 2m$$

Key Encapsulation Mechanism (KEM)

- Three Algorithms
  - $\mathrm{Gen}(1^n, R)$ (Key-generation algorithm)
    - Input: Random Bits R
    - Output: $(\boldsymbol{pk}, \boldsymbol{sk}) \in \boldsymbol{\mathcal{K}}$
  - $\mathrm{Encaps}_{\mathrm{pk}}(1^n, R)$
    - Input: security parameter, random bits R
    - Output: Symmetric key $\mathrm{k} \in \{0,1\}^{\ell(n)}$ and a ciphertext c
  - $\mathrm{Decaps}_{\mathrm{sk}}(c)$ (Deterministic algorithm)
    - Input: Secret key $\mathrm{sk} \in \mathcal{K}$ and a ciphertex c
    - Output: a symmetric key$\{0,1\}^{\ell(n)}$ or ⊥ (fail)
- **Invariant**: $\mathrm{Decaps}_{\mathrm{sk}}$(c)=k whenever (c,k) = $\mathrm{Encaps}_{\mathrm{pk}}(1^n, R)$

KEM CCA-Security ($\text{KEM}^{\text{cca}}_{\text{A},\Pi}(\text{n})$)



$(\boldsymbol{pk, c, kb})$

⋮

$\boldsymbol{c_1 \neq c}$

$\textbf{Decaps}_{\boldsymbol{sk}}(\boldsymbol{c_1})$

$\boldsymbol{c_2 \neq c}$

$\textbf{Decaps}_{\boldsymbol{sk}}(\boldsymbol{c_2})$

b'

⋮

Random bit b

(pk,sk) = Gen(.)

$\boldsymbol{(c, k_0) = \textbf{Encaps}_{pk}(.)}$

$\boldsymbol{k_1 \leftarrow \{0,1\}^n}$

$$\forall PPT\ A\ \exists \mu\ (\text{negligible})\ \text{s.t}$$

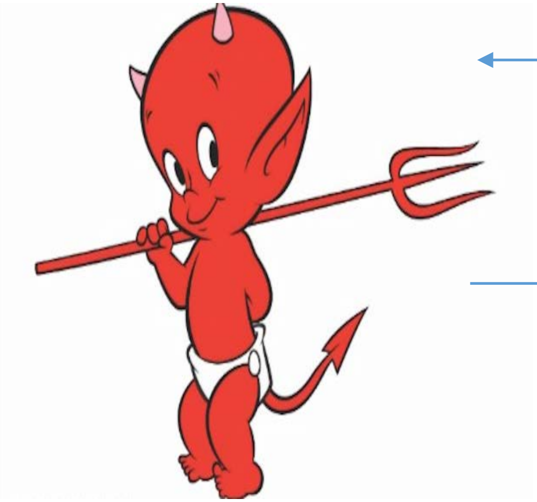
$$\Pr\left[\text{KEM}^{\text{cca}}_{\text{A},\Pi} = 1\right] \leq \frac{1}{2} + \mu(n)$$

Recall: Last Lecture

- CCA-Secure KEM from RSA in Random Oracle Model
- What if we want security proof in the standard model?
- Answer: DDH yields a CPA-Secure KEM in standard model

KEM CPA-Security $(\mathrm{KEM}^{\mathrm{cpa}}_{\mathrm{A},\Pi}(\mathrm{n}))$

$(\boldsymbol{pk}, \boldsymbol{c}, \boldsymbol{kb})$



b’

Random bit b

(pk,sk) = Gen(.)

$\forall PPT\ A\ \exists \mu\ (\text{negligible})\ \mathrm{s.t}$

$$\Pr\left[\mathrm{KEM}^{\mathrm{cpa}}_{\mathrm{A},\Pi} = 1\right] \le \frac{1}{2} + \mu(n)$$

$(\boldsymbol{c}, \boldsymbol{k_0}) = \mathbf{Encaps}_{\boldsymbol{pk}}(.)$

$\boldsymbol{k_1} \leftarrow \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{n}}$

CCA-Secure Encryption from CPA-Secure KEM

$$\mathbf{Enc_{pk}}(\boldsymbol{m}; \boldsymbol{R}) = \langle \boldsymbol{c}, \mathbf{Enc_k^*}(\boldsymbol{m}) \rangle$$

Where

- $(\boldsymbol{c}, \boldsymbol{k}) \leftarrow \mathbf{Encaps_{pk}}(\mathbf{1}^{\boldsymbol{n}}; \boldsymbol{R})$,
- $\mathbf{Enc_k^*}$ is a eavesdropping-secure symmetric key encryption algorithm
- $\mathbf{Encaps_{pk}}$ is a CPA-Secure KEM.

Theorem 11.12: $\mathbf{Enc_{pk}}$ is CCA-Secure public key encryption scheme.

CPA-Secure KEM with El-Gamal

- $\mathrm{Gen}(1^n, R)$ (Key-generation algorithm)
  1. Run $\mathcal{G}(1^n)$ to obtain a cyclic group $\mathbb{G}$ of order q (with $\|q\| = 2n$) and a generator g such that $< \mathrm{g} >= \mathbb{G}$.
  2. Choose a random $\mathrm{x} \in \mathbb{Z}_q$ and set $h = g^x$
  3. Public Key: $\mathrm{pk} = \langle \mathbb{G}, q, g, h \rangle$
  4. Private Key: $\mathrm{sk} = \langle \mathbb{G}, q, g, x \rangle$
- $\mathrm{Encaps}_{\mathrm{pk}}(1^n, R)$
  - Pick random $\mathrm{y} \in \mathbb{Z}_q$
  - Output: $\langle g^y, k = \mathrm{LeastSigNBits}(h^y) \rangle$
- $\mathrm{Decaps}_{\mathrm{sk}}(c)$ (Deterministic algorithm)
  - Output: $k = \mathrm{LeastSigNBits}(c^x)$

CPA-Secure KEM with El-Gamal

- $\mathrm{Gen}(1^n, R)$ (Key-generation algorithm)
  1. Run $\mathcal{G}(1^n)$ to obtain a cyclic group $\mathbb{G}$ of order q (with $\|q\| = 2n$) and a generator g such that $<\mathrm{g}>= \mathbb{G}$.
  2. Choose a random x $\in \mathbb{Z}_q$ and set $h = g^x$
  3. Public Key: $\mathrm{pk} = \langle \mathbb{G}, q, g, h \rangle$
  4. Private Key: $\mathrm{sk} = \langle \mathbb{G}, q, g, x \rangle$
- $\mathrm{Encaps}_{\mathrm{pk}}(1^n, R)$
  - Pick random y $\in \mathbb{Z}_q$
  - Output: $\langle g^y, k = \mathrm{LeastSigNBits}(h^y) \rangle$
- $\mathrm{Decaps}_{\mathrm{sk}}(c)$ (Deterministic algorithm)
  - Output: $k = \mathrm{LeastSigNBits}(c^x)$

$$\mathrm{Decaps}_{\mathrm{sk}}(g^y) = \mathrm{LeastSigNBits}(g^{xy}) = \mathrm{LeastSigNBits}(h^y) = k$$

CPA-Secure KEM with El-Gamal

- $\text{Gen}(1^n, R)$ (Key-generation algorithm)
  1. Run $\mathcal{G}(1^n)$ to obtain a cyclic group $\mathbb{G}$ of order q (with $\|q\| = 2n$) and a generator g such that $<\text{g}>=\mathbb{G}$.
  2. Choose a random $\text{x} \in \mathbb{Z}_q$ and set $h = g^x$
  3. Public Key: $\text{pk} = \langle \mathbb{G}, q, g, h \rangle$
  4. Private Key: $\text{sk} = \langle \mathbb{G}, q, g, x \rangle$
- $\text{Encaps}_{\text{pk}}(1^n, R)$
  - Pick random $\text{y} \in \mathbb{Z}_q$
  - Output: $\langle g^y, k = \text{LeastSigNBits}(h^y) \rangle$
- $\text{Decaps}_{\text{sk}}(c)$ (Deterministic algorithm)
  - Output: $k = \text{LeastSigNBits}(c^x)$

Theorem 11.20: If DDH is hard relative to $\mathcal{G}$ then (Gen,Encaps,Decaps) is a CPA-Secure KEM

CPA-Secure KEM with El-Gamal

- $\text{Gen}(1^n, R)$ (Key-generation algorithm)
  1. Run $\mathcal{G}(1^n)$ to obtain a cyclic group $\mathbb{G}$ of order q (with $\|q\| = 2n$) and a generator g such that $< \text{g} >= \mathbb{G}$.
  2. Choose a random $\text{x} \in \mathbb{Z}_q$ and set $h = g^x$
  3. Public Key: $\text{pk} = \langle \mathbb{G}, q, g, h \rangle$
  4. Private Key: $\text{sk} = \langle \mathbb{G}, q, g, x \rangle$
- $\text{Encaps}_{\text{pk}}(1^n, R)$
  - Pick random $\text{y} \in \mathbb{Z}_q$
  - Output: $\langle g^y, k = \text{LeastSigNBits}(h^y) \rangle$
- $\text{Decaps}_{\text{sk}}(c)$ (Deterministic algorithm)
  - Output: $k = \text{LeastSigNBits}(c^x)$

Remark: If CDH is hard relative to $\mathcal{G}$ then (Gen,Encaps,Decaps) and we replace LeastSigNBits with a random oracle H then this is a CPA-Secure KEM

(…also CCA-secure under a slightly stronger assumption called gap-CDH)

CCA-Secure Variant in Random Oracle Model

- Key Generation ($\text{Gen}(1^n)$):
  1. Run $\mathcal{G}(1^n)$ to obtain a cyclic group $\mathbb{G}$ of order q (with $\|q\| = n$) and a generator g such that $< \mathrm{g} >= \mathbb{G}$.
  2. Choose a random $\mathrm{x} \in \mathbb{Z}_q$ and set $h = g^x$
  3. Public Key: $\mathrm{pk} = \langle \mathbb{G}, q, g, h \rangle$
  4. Private Key: $\mathrm{sk} = \langle \mathbb{G}, q, g, x \rangle$
- $\mathrm{Enc}_{\mathrm{pk}}(m) = \langle g^y, c', Mac_{K_M}(c') \rangle$ for a random $\mathrm{y} \in \mathbb{Z}_q$ and $K_E \| K_M = H(h^y)$ and $c' = \mathrm{Enc}'_{\mathrm{K_E}}(m)$
- $\mathrm{Dec}_{\mathrm{sk}}(\langle c, c', t \rangle)$

1. $K_E \| K_M = H(c^x)$

2. If $\mathrm{Vrfy}_{\mathrm{K_M}}(c', t) \neq 1$ or $c \notin \mathbb{G}$ output $\perp$; otherwise output $\mathrm{Dec}'_{\mathrm{K_E}}(c', t)$

CCA-Secure Variant in Random Oracle Model

Theorem: If $\mathrm{Enc}'_{\mathrm{K_E}}$ is CPA-secure, $\mathrm{Mac}_{\mathrm{K_M}}$ is a strong MAC and a problem called gap-CDH is hard then this a CCA-secure public key encryption scheme in the random oracle model.

- $\mathrm{Enc}_{\mathrm{pk}}(m) = \langle g^y, c', \mathrm{Mac}_{\mathrm{K_M}}(c') \rangle$ for a random y $\in \mathbb{Z}_q$ and $K_E \| K_M = H(h^y)$ and $c' = \mathrm{Enc}'_{\mathrm{K_E}}(m)$
- $\mathrm{Dec}_{\mathrm{sk}}(\langle c, c', t \rangle)$

1. $K_E \| K_M = H(c^x)$
2. If $\mathrm{Vrfy}_{\mathrm{K_M}}(c', t) \neq 1$ or $c \notin \mathbb{G}$ output $\perp$; otherwise output $\mathrm{Dec}'_{\mathrm{K_E}}(c', t)$

CCA-Secure Variant in Random Oracle Model

Remark: The CCA-Secure variant is used in practice in the ISO/IEC 18033-2 standard for public-key encryption.

- Diffie-Hellman Integrated Encryption Scheme (DHIES)
- Elliptic Curve Integrated Encryption Scheme (ECIES)
- $\mathrm{Enc}_{\mathrm{pk}}(m) = \langle g^y, c', \mathrm{Mac}_{\mathrm{K_M}}(c') \rangle$ for a random y $\in \mathbb{Z}_q$ and $K_E \| K_M = H(h^y)$ and $c' = \mathrm{Enc}'_{\mathrm{K_E}}(m)$
- $\mathrm{Dec}_{\mathrm{sk}}(\langle c, c', t \rangle)$

1. $K_E \| K_M = H(c^x)$
2. If $\mathrm{Vrfy}_{\mathrm{K_M}}(c', t) \neq 1$ or $c \notin \mathbb{G}$ output $\perp$; otherwise output $\mathrm{Dec}'_{\mathrm{K_E}}(c', t)$

Next Class: RSA Attacks + Fixes

- Read Katz and Lindell: 11.5