

Cryptography

Lecture 22

Arkady Yerukhimovich

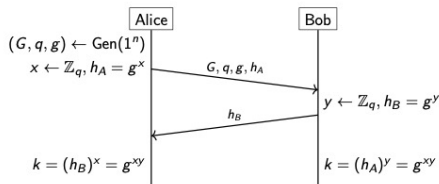
November 13, 2024

- 1 Lecture 21 Review
- 2 RSA Encryption Scheme (Chapter 11.5)
- 3 Hybrid Encryption and CCA Security

Lecture 21 Review

- Diffie-Hellman Key Exchange
- Public-key revolution
- From key exchange to public-key encryption

From Key Exchange to Public-Key Encryption



- 1 To encrypt $m \in G$ using g^{xy} , compute $m \cdot g^{xy}$.
This is essentially a multiplicative OTP
- 2 To make encryption non-interactive, A sets pk to be her first message $pk_A = (G, q, g, h_A)$
- 3 To encrypt a message m , B has to complete the KE and use the resulting key to encrypt
 - Choose $y \leftarrow \mathbb{Z}_q$
 - Compute $g^{xy} \cdot m$
 - To enable A to decrypt, include B 's message $h_B = g^y$ in c
- 4 To decrypt, A computes $(h_B)^x = g^{xy}$ and uses this to unmask m

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$
- $\text{Enc}_{pk}(m)$:
 - Given $pk = (G, q, g, h)$ and message $m \in G$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$
- $\text{Enc}_{pk}(m)$:
 - Given $pk = (G, q, g, h)$ and message $m \in G$
 - $y \leftarrow \mathbb{Z}_q$, compute $c = (g^y, h^y \cdot m)$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$
- $\text{Enc}_{pk}(m)$:
 - Given $pk = (G, q, g, h)$ and message $m \in G$
 - $y \leftarrow \mathbb{Z}_q$, compute $c = (g^y, h^y \cdot m)$
- $\text{Dec}_{sk}(c)$:
 - Given $sk = x$ and $c = (c_1, c_2)$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$
- $\text{Enc}_{pk}(m)$:
 - Given $pk = (G, q, g, h)$ and message $m \in G$
 - $y \leftarrow \mathbb{Z}_q$, compute $c = (g^y, h^y \cdot m)$
- $\text{Dec}_{sk}(c)$:
 - Given $sk = x$ and $c = (c_1, c_2)$
 - Compute $\hat{m} = c_2 / c_1^x$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$
- $\text{Enc}_{pk}(m)$:
 - Given $pk = (G, q, g, h)$ and message $m \in G$
 - $y \leftarrow \mathbb{Z}_q$, compute $c = (g^y, h^y \cdot m)$
- $\text{Dec}_{sk}(c)$:
 - Given $sk = x$ and $c = (c_1, c_2)$
 - Compute $\hat{m} = c_2 / c_1^x$

Correctness:

$$\hat{m} = c_2 / c_1^x = \frac{h^y \cdot m}{(g^y)^x} = \frac{(g^x)^y \cdot m}{(g^y)^x} = m$$

El Gamal Example

$$p = 2q + 1 \quad \text{where } q \text{ is prime}$$

- Let G be the group of quadratic residues mod 11

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined
 - Let g be $2^2 = 4 \bmod 11$, this is a generator because $|G|$ is prime

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined
 - Let g be $2^2 = 4 \bmod 11$, this is a generator because $|G|$ is prime
- Gen:
 - Choose $sk = 3 \in \mathbb{Z}_5$

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined
 - Let g be $2^2 = 4 \bmod 11$, this is a generator because $|G|$ is prime
- Gen:
 - Choose $sk = 3 \in \mathbb{Z}_5$
 - Output $pk = (G, q, g, g^{sk}) = (11, 5, 4, [4^3 \bmod 11]) = (11, 5, 4, \underline{9})$

$$4^3 = \underline{(4 \cdot 4)} \cdot 4 = 5 \cdot 4 = 9 \text{ a.d. } 11$$

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined
 - Let g be $2^2 = 4 \bmod 11$, this is a generator because $|G|$ is prime
- Gen:
 - Choose $sk = 3 \in \mathbb{Z}_5$
 - Output $pk = (G, q, g, g^{sk}) = (11, 5, 4, [4^3 \bmod 11]) = (11, 5, 4, 9)$
- $\text{Enc}_{pk}(m = 3)$: Note that $3 = 5^2 \bmod 11$

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined
 - Let g be $2^2 = 4 \bmod 11$, this is a generator because $|G|$ is prime
- Gen:
 - Choose $sk = 3 \in \mathbb{Z}_5$
 - Output $pk = (G, q, g, g^{sk}) = (11, 5, 4, [4^3 \bmod 11]) = (11, 5, 4, 9)$
- $\text{Enc}_{pk}(m = 3)$: Note that $3 = 5^2 \bmod 11$
 - Choose $y = 2 \in \mathbb{Z}_q$

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined
 - Let g be $2^2 = 4 \bmod 11$, this is a generator because $|G|$ is prime
- Gen:
 - Choose $sk = 3 \in \mathbb{Z}_5$
 - Output $pk = (G, q, g, g^{sk}) = (11, 5, 4, [4^3 \bmod 11]) = (11, 5, 4, 9)$
- $\text{Enc}_{pk}(m = 3)$: Note that $3 = 5^2 \bmod 11$
 - Choose $y = 2 \in \mathbb{Z}_q$
 - $c = ([4^2 \bmod 11], [9^2 \cdot 3 \bmod 11]) = (5, 1)$

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined
 - Let g be $2^2 = 4 \bmod 11$, this is a generator because $|G|$ is prime
- Gen:
 - Choose $sk = 3 \in \mathbb{Z}_5$
 - Output $pk = (G, q, g, g^{sk}) = (11, 5, 4, [4^3 \bmod 11]) = (11, 5, 4, 9)$
- $\text{Enc}_{pk}(m = 3)$: Note that $3 = 5^2 \bmod 11$
 - Choose $y = 2 \in \mathbb{Z}_q$
 - $c = ([4^2 \bmod 11], [9^2 \cdot 3 \bmod 11]) = (5, 1)$
- $\text{Dec}_{sk}(c_1, c_2)$:
 - $m = [1 \cdot 5^{-3} \bmod 11]$

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined
 - Let g be $2^2 = 4 \bmod 11$, this is a generator because $|G|$ is prime
- Gen:
 - Choose $sk = 3 \in \mathbb{Z}_5$
 - Output $pk = (G, q, g, g^{sk}) = (11, 5, 4, [4^3 \bmod 11]) = (11, 5, 4, 9)$
- $\text{Enc}_{pk}(m = 3)$: Note that $3 = 5^2 \bmod 11$
 - Choose $y = 2 \in \mathbb{Z}_q$
 - $c = ([4^2 \bmod 11], [9^2 \cdot 3 \bmod 11]) = (5, 1)$
- $\text{Dec}_{sk}(c_1, c_2)$:
 - $m = [1 \cdot 5^{-3} \bmod 11]$
 - Observe that $[5^{-1} \bmod 11] = 9$, and $9^3 = (-2)^3 = -8 = 3 \bmod 11$

$\frac{c}{c^{sk}}$

El Gamal Example

- Let G be the group of quadratic residues mod 11
 - Note that for $q = 5$, $p = 2q + 1 = 11$, so G is well defined
 - Let g be $2^2 = 4 \bmod 11$, this is a generator because $|G|$ is prime
- Gen:
 - Choose $sk = 3 \in \mathbb{Z}_5$
 - Output $pk = (G, q, g, g^{sk}) = (11, 5, 4, [4^3 \bmod 11]) = (11, 5, 4, 9)$
- $\text{Enc}_{pk}(m = 3)$: Note that $3 = 5^2 \bmod 11$
 - Choose $y = 2 \in \mathbb{Z}_q$
 - $c = ([4^2 \bmod 11], [9^2 \cdot 3 \bmod 11]) = (5, 1)$
- $\text{Dec}_{sk}(c_1, c_2)$:
 - $m = [1 \cdot 5^{-3} \bmod 11]$
 - Observe that $[5^{-1} \bmod 11] = 9$, and $9^3 = (-2)^3 = -8 = 3 \bmod 11$
 - So, we recover $m = [1 \cdot 3 \bmod 11] = 3$

Defining CPA-Secure Public-Key Encryption

Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ be an encryption scheme. Consider the following game between an adversary \mathcal{A} and a challenger:

$\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n)$ (sk, pk)

- The challenger chooses $k \leftarrow \text{Gen}(1^n)$ Give pk to \mathcal{A}
- $\mathcal{A}^{\text{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $b \leftarrow \{0, 1\}$, computes $c \leftarrow \text{Enc}_k(m_b)$ and gives c to \mathcal{A}
- $\mathcal{A}^{\text{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n) = 1$ (i.e., \mathcal{A} wins) if $b' = b$.

Definition: An encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with message space \mathcal{M} is CPA-secure if for all PPT \mathcal{A} it holds that

$$\Pr[\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n) = 1] \leq 1/2 + \text{negl}(n)$$

Defining CPA-Secure Public-Key Encryption

Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ be a public-key encryption scheme. Consider the following game between an adversary \mathcal{A} and a challenger:

$\text{PubK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n)$

- The challenger chooses $(pk, sk) \leftarrow \text{Gen}(1^n)$ and gives pk to \mathcal{A} .
- $\mathcal{A}^{\text{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $b \leftarrow \{0, 1\}$, computes $c \leftarrow \text{Enc}_k(m_b)$ and gives c to \mathcal{A} .
- $\mathcal{A}^{\text{Enc}_k(\cdot)}(1^n)$ outputs a guess bit b' .
- We say that $\text{PubK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n) = 1$ (i.e., \mathcal{A} wins) if $b' = b$.

Definition: A public-key encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with message space \mathcal{M} is CPA-secure if for all PPT \mathcal{A} it holds that

$$\Pr[\text{PubK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n) = 1] \leq 1/2 + \text{negl}(n)$$

Proof of CPA Security of El Gamal

Define the following encryption scheme $\tilde{\Pi}$:

Define the following encryption scheme $\tilde{\Pi}$:

$\tilde{\Pi}$ Encryption Scheme

- $\widetilde{\text{Gen}}(1^n)$: Same as in El Gamal

Define the following encryption scheme $\tilde{\Pi}$:

$\tilde{\Pi}$ Encryption Scheme

- $\widetilde{\text{Gen}}(1^n)$: Same as in El Gamal
- $\widetilde{\text{Enc}}(pk, m)$: Choose $y, z \leftarrow \mathbb{Z}_q$, output $c = (g^y, g^z \cdot m)$

Proof of CPA Security of El Gamal

Define the following encryption scheme $\tilde{\Pi}$:

$\tilde{\Pi}$ Encryption Scheme

- $\widetilde{\text{Gen}}(1^n)$: Same as in El Gamal
- $\widetilde{\text{Enc}}(pk, m)$: Choose $y, z \leftarrow \mathbb{Z}_q$, output $c = (g^y, g^z \cdot m)$
- $\widetilde{\text{Dec}}(sk, c)$: No efficient decryption procedure

Proof of CPA Security of El Gamal

Define the following encryption scheme $\tilde{\Pi}$:

$\tilde{\Pi}$ Encryption Scheme

- $\widetilde{\text{Gen}}(1^n)$: Same as in El Gamal
- $\widetilde{\text{Enc}}(pk, m)$: Choose $y, z \leftarrow \mathbb{Z}_q$, output $c = (g^y, g^z \cdot m)$
- $\widetilde{\text{Dec}}(sk, c)$: No efficient decryption procedure

Security: $\Pr[\text{PubK}_{\mathcal{A}, \tilde{\Pi}}^{\text{cpa}}(n) = 1] = 1/2$ because g^z is random group element

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

$\mathcal{A}_r(G, q, g, h_1 = g^x, h_2 = g^y, h_3 = g^{xy} \text{ or } g^z)$

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

$\mathcal{A}_r(G, q, g, h_1 = g^x, h_2 = g^y, h_3 = g^{xy} \text{ or } g^z)$

- Set $pk = (G, q, g, h_1)$ and run $\mathcal{A}_c(pk)$ to get $(m_0, m_1) \in G^2$

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

$\mathcal{A}_r(G, q, g, h_1 = g^x, h_2 = g^y, h_3 = g^{xy} \text{ or } g^z)$

- Set $pk = (G, q, g, h_1)$ and run $\mathcal{A}_c(pk)$ to get $(m_0, m_1) \in G^2$
- Choose $b \leftarrow \{0, 1\}$ and set $c = (h_2, h_3 \cdot m_b)$

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

$\mathcal{A}_r(G, q, g, h_1 = g^x, h_2 = g^y, h_3 = g^{xy} \text{ or } g^z)$

- Set $pk = (G, q, g, h_1)$ and run $\mathcal{A}_c(pk)$ to get $(m_0, m_1) \in G^2$
- Choose $b \leftarrow \{0, 1\}$ and set $c = (h_2, h_3 \cdot m_b)$
- Give c to \mathcal{A}_c and get b' , if $b' = b$ output 0 (DH)

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

$\mathcal{A}_r(G, q, g, h_1 = g^x, h_2 = g^y, h_3 = g^{xy} \text{ or } g^z)$

- Set $pk = (G, q, g, h_1)$ and run $\mathcal{A}_c(pk)$ to get $(m_0, m_1) \in G^2$
- Choose $b \leftarrow \{0, 1\}$ and set $c = (h_2, h_3 \cdot m_b)$
- Give c to \mathcal{A}_c and get b' , if $b' = b$ output 0 (DH)
- Case 1: If $h_3 = g^z$ for $z \leftarrow \mathbb{Z}_q$ then $c = (g^y, g^z \cdot m_b)$

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

$\mathcal{A}_r(G, q, g, h_1 = g^x, h_2 = g^y, h_3 = g^{xy} \text{ or } g^z)$

- Set $pk = (G, q, g, h_1)$ and run $\mathcal{A}_c(pk)$ to get $(m_0, m_1) \in G^2$
- Choose $b \leftarrow \{0, 1\}$ and set $c = (h_2, h_3 \cdot m_b)$
- Give c to \mathcal{A}_c and get b' , if $b' = b$ output 0 (DH)

- Case 1: If $h_3 = g^z$ for $z \leftarrow \mathbb{Z}_q$ then $c = (g^y, g^z \cdot m_b)$, so

$$\Pr[\mathcal{A}_r(G, q, g, g^x, g^y, g^z) = 0] = \Pr[\text{PubK}_{\mathcal{A}_c, \tilde{\Pi}}^{cpa}(n) = 1] = 1/2$$

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

$\mathcal{A}_r(G, q, g, h_1 = g^x, h_2 = g^y, h_3 = g^{xy} \text{ or } g^z)$

- Set $pk = (G, q, g, h_1)$ and run $\mathcal{A}_c(pk)$ to get $(m_0, m_1) \in G^2$
- Choose $b \leftarrow \{0, 1\}$ and set $c = (h_2, h_3 \cdot m_b)$
- Give c to \mathcal{A}_c and get b' , if $b' = b$ output 0 (DH)

- Case 1: If $h_3 = g^z$ for $z \leftarrow \mathbb{Z}_q$ then $c = (g^y, g^z \cdot m_b)$, so

$$\Pr[\mathcal{A}_r(G, q, g, g^x, g^y, g^z) = 0] = \Pr[\text{PubK}_{\mathcal{A}_c, \tilde{\Pi}}^{cpa}(n) = 1] = 1/2$$

- Case 2: If $h_3 = g^{xy}$ then $c = (g^y, g^{xy} \cdot m_b)$, so

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

$\mathcal{A}_r(G, q, g, h_1 = g^x, h_2 = g^y, h_3 = g^{xy} \text{ or } g^z)$

- Set $pk = (G, q, g, h_1)$ and run $\mathcal{A}_c(pk)$ to get $(m_0, m_1) \in G^2$
- Choose $b \leftarrow \{0, 1\}$ and set $c = (h_2, h_3 \cdot m_b)$
- Give c to \mathcal{A}_c and get b' , if $b' = b$ output 0 (DH)

- Case 1: If $h_3 = g^z$ for $z \leftarrow \mathbb{Z}_q$ then $c = (g^y, g^z \cdot m_b)$, so

$$\Pr[\mathcal{A}_r(G, q, g, g^x, g^y, g^z) = 0] = \Pr[\text{PubK}_{\mathcal{A}_c, \tilde{\Pi}}^{cpa}(n) = 1] = 1/2$$

- Case 2: If $h_3 = g^{xy}$ then $c = (g^y, g^{xy} \cdot m_b)$, so

$$\Pr[\mathcal{A}_r(G, q, g, g^x, g^y, g^{xy}) = 0] = \Pr[\text{PubK}_{\mathcal{A}_c, \Pi}^{cpa}(n) = 1] > 1/2 + 1/p(n)$$

Proof of CPA Security of El Gamal

Assume \mathcal{A}_c breaking CPA-security of El Gamal, build \mathcal{A}_r breaking DDH

$\mathcal{A}_r(G, q, g, h_1 = g^x, h_2 = g^y, h_3 = g^{xy} \text{ or } g^z)$

- Set $pk = (G, q, g, h_1)$ and run $\mathcal{A}_c(pk)$ to get $(m_0, m_1) \in G^2$
- Choose $b \leftarrow \{0, 1\}$ and set $c = (h_2, h_3 \cdot m_b)$
- Give c to \mathcal{A}_c and get b' , if $b' = b$ output 0 (DH)

- Case 1: If $h_3 = g^z$ for $z \leftarrow \mathbb{Z}_q$ then $c = (g^y, g^z \cdot m_b)$, so

$$\Pr[\mathcal{A}_r(G, q, g, g^x, g^y, g^z) = 0] = \Pr[\text{PubK}_{\mathcal{A}_c, \tilde{\Pi}}^{cpa}(n) = 1] = 1/2$$

- Case 2: If $h_3 = g^{xy}$ then $c = (g^y, g^{xy} \cdot m_b)$, so

$$\Pr[\mathcal{A}_r(G, q, g, g^x, g^y, g^{xy}) = 0] = \Pr[\text{PubK}_{\mathcal{A}_c, \Pi}^{cpa}(n) = 1] > 1/2 + 1/p(n)$$

Contradiction

\mathcal{A}_r can break the DDH assumption with non-negligible advantage

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$
- $\text{Enc}_{pk}(m)$:
 - Given $pk = (G, q, g, h)$ and message $m \in G$
 - $y \leftarrow \mathbb{Z}_q$, compute $c = (g^y, h^y \cdot m)$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$
- $\text{Enc}_{pk}(m)$:
 - Given $pk = (G, q, g, h)$ and message $m \in G$
 - $y \leftarrow \mathbb{Z}_q$, compute $c = (g^y, h^y \cdot m)$
- $\text{Dec}_{sk}(c)$:
 - Given $sk = x$ and $c = (c_1, c_2)$
 - Compute $\hat{m} = c_2 / c_1^x$

El Gamal Encryption

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$
- $\text{Enc}_{pk}(m)$:
 - Given $pk = (G, q, g, h)$ and message $m \in G$
 - $y \leftarrow \mathbb{Z}_q$, compute $c = (g^y, h^y \cdot m)$
- $\text{Dec}_{sk}(c)$:
 - Given $sk = x$ and $c = (c_1, c_2)$
 - Compute $\hat{m} = c_2 / c_1^x$

Correctness:

$$\hat{m} = c_2 / c_1^x = \frac{h^y \cdot m}{(g^y)^x} = \frac{(g^x)^y \cdot m}{(g^y)^x} = m$$

Security: El Gamal is CPA-secure based on DDH

- 1 Lecture 21 Review
- 2 RSA Encryption Scheme (Chapter 11.5)
- 3 Hybrid Encryption and CCA Security

Plain RSA

Recall:

- GenRSA(1^n): Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA

Recall:

- GenRSA(1^n): Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA Encryption

Plain RSA

Recall:

- $\text{GenRSA}(1^n)$: Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA Encryption

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$

Plain RSA

Recall:

- $\text{GenRSA}(1^n)$: Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA Encryption

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$: For $m \in \mathbb{Z}_N^*$, compute $c = [m^e \bmod N]$

Plain RSA

Recall:

- $\text{GenRSA}(1^n)$: Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA Encryption

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$: For $m \in \mathbb{Z}_N^*$, compute $c = [m^e \bmod N]$
- $\text{Dec}_{sk}(c)$: Compute $m = [c^d \bmod N]$

Plain RSA

Recall:

- $\text{GenRSA}(1^n)$: Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA Encryption

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$: For $m \in \mathbb{Z}_N^*$, compute $c = [m^e \bmod N]$
- $\text{Dec}_{sk}(c)$: Compute $m = [c^d \bmod N]$

Example

- $p = 3, q = 7, N = 21, \phi(N) = 12$

Plain RSA

Recall:

- $\text{GenRSA}(1^n)$: Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA Encryption

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$: For $m \in \mathbb{Z}_N^*$, compute $c = [m^e \bmod N]$
- $\text{Dec}_{sk}(c)$: Compute $m = [c^d \bmod N]$

Example

- $p = 3, q = 7, N = 21, \phi(N) = 12$
- Suppose $e = 5$, then $d = 5^{-1} \bmod 12 = 5$

Plain RSA

Recall:

- $\text{GenRSA}(1^n)$: Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA Encryption

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$: For $m \in \mathbb{Z}_N^*$, compute $c = [m^e \bmod N]$
- $\text{Dec}_{sk}(c)$: Compute $m = [c^d \bmod N]$

Example

- $p = 3, q = 7, N = 21, \phi(N) = 12$
- Suppose $e = 5$, then $d = 5^{-1} \bmod 12 = 5$
- $\text{Enc}_{(21,5)}(11) = [11^5 \bmod 21] = [(16)(16)(11) \bmod 21] = 2$

Plain RSA

Recall:

- $\text{GenRSA}(1^n)$: Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA Encryption

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$: For $m \in \mathbb{Z}_N^*$, compute $c = [m^e \bmod N]$
- $\text{Dec}_{sk}(c)$: Compute $m = [c^d \bmod N]$

Example

- $p = 3, q = 7, N = 21, \phi(N) = 12$
- Suppose $e = 5$, then $d = 5^{-1} \bmod 12 = 5$
- $\text{Enc}_{(21,5)}(11) = [11^5 \bmod 21] = [(16)(16)(11) \bmod 21] = 2$
- $\text{Dec}_{(21,5)}(2) = [2^5 \bmod 21] = 11$

Plain RSA

Recall:

- GenRSA(1^n): Choose $N = pq$, e, d s.t. $ed = 1 \bmod \phi(N)$
- RSA Assumption: Given $y = x^e \bmod N$, hard to find x

Plain RSA Encryption

- Gen(1^n): $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- Enc $_{pk}(m)$: For $m \in \mathbb{Z}_N^*$, compute $c = [m^e \bmod N]$
- Dec $_{sk}(c)$: Compute $m = [c^d \bmod N]$

Example

- $p = 3, q = 7, N = 21, \phi(N) = 12$ $11^5 = 11^2 \cdot 11^2 \cdot 11$
- Suppose $e = 5$, then $d = 5^{-1} \bmod 12 = 5$ $11^2 = 121$
- Enc $_{(21,5)}(11) = [11^5 \bmod 21] = [(16)(16)(11) \bmod 21] = 2$
- Dec $_{(21,5)}(2) = [2^5 \bmod 21] = 11$

Encryptions So Far

El Gamal Encryption: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with $\mathcal{M} = G$

- $\text{Gen}(1^n)$:
 - $(G, q, g) \leftarrow \text{Gen}(1^n)$
 - $x \leftarrow \mathbb{Z}_q, h = g^x$
 - $pk = (G, q, g, h)$ and $sk = x$
- $\text{Enc}_{pk}(m)$:
 - Given $pk = (G, q, g, h)$ and message $m \in G$
 - $y \leftarrow \mathbb{Z}_q$, compute $c = (g^y, h^y \cdot m)$
- $\text{Dec}_{sk}(c)$:
 - Given $sk = x$ and $c = (c_1, c_2)$
 - Compute $\hat{m} = c_2 / c_1^x$

Plain RSA Encryption

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$: For $m \in \mathbb{Z}_N^*$, compute $c = [m^e \bmod N]$
- $\text{Dec}_{sk}(c)$: Compute $m = [c^d \bmod N]$

Security of Plain RSA

Security of Plain RSA

Plain RSA is not CPA secure

Enc is deterministic

Security of Plain RSA

Plain RSA is not CPA secure

Enc is deterministic

- 1 Common factors in public keys

$$N_1 = p_1 \cdot q_1$$

$$N_2 = p_2 \cdot q_2$$

$$p_1 = p_2$$

$$\gcd(N_1, N_2) = p$$

Security of Plain RSA

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!

Security of Plain RSA

Plain RSA is not CPA secure

Enc is deterministic

- 1 Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e
 - To make encryption fast, RSA often uses small exponent e (e.g., $e = 3$)

Security of Plain RSA

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e
 - To make encryption fast, RSA often uses small exponent e (e.g., $e = 3$)
 - If $m < N^{1/e}$, then m^e does not wrap around mod N

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e
 - To make encryption fast, RSA often uses small exponent e (e.g., $e = 3$)
 - If $m < N^{1/e}$, then m^e does not wrap around mod N
 - Easy to find e^{th} roots over the integers

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e
 - To make encryption fast, RSA often uses small exponent e (e.g., $e = 3$)
 - If $m < N^{1/e}$, then m^e does not wrap around mod N
 - Easy to find e^{th} roots over the integers
 - If $e = 3$ and $\|N\| = 1024$ bits, can decrypt 300-bit m

Security of Plain RSA

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e
 - To make encryption fast, RSA often uses small exponent e (e.g., $e = 3$)
 - If $m < N^{1/e}$, then m^e does not wrap around mod N
 - Easy to find e^{th} roots over the integers
 - If $e = 3$ and $||N|| = 1024$ bits, can decrypt 300-bit m
- ③ Same m sent to multiple receivers

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e
 - To make encryption fast, RSA often uses small exponent e (e.g., $e = 3$)
 - If $m < N^{1/e}$, then m^e does not wrap around mod N
 - Easy to find e^{th} roots over the integers
 - If $e = 3$ and $||N|| = 1024$ bits, can decrypt 300-bit m
- ③ Same m sent to multiple receivers
 - Suppose we encrypt m to 3 public keys N_1, N_2, N_3 using $e = 3$

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e
 - To make encryption fast, RSA often uses small exponent e (e.g., $e = 3$)
 - If $m < N^{1/e}$, then m^e does not wrap around mod N
 - Easy to find e^{th} roots over the integers
 - If $e = 3$ and $||N|| = 1024$ bits, can decrypt 300-bit m
- ③ Same m sent to multiple receivers
 - Suppose we encrypt m to 3 public keys N_1, N_2, N_3 using $e = 3$
 - $c_1 = [m^3 \bmod N_1]$, $c_2 = [m^3 \bmod N_2]$, $c_3 = [m^3 \bmod N_3]$

Security of Plain RSA

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e
 - To make encryption fast, RSA often uses small exponent e (e.g., $e = 3$)
 - If $m < N^{1/e}$, then m^e does not wrap around mod N
 - Easy to find e^{th} roots over the integers
 - If $e = 3$ and $||N|| = 1024$ bits, can decrypt 300-bit m
- ③ Same m sent to multiple receivers
 - Suppose we encrypt m to 3 public keys N_1, N_2, N_3 using $e = 3$
 - $c_1 = [m^3 \bmod N_1]$, $c_2 = [m^3 \bmod N_2]$, $c_3 = [m^3 \bmod N_3]$
 - By CRT, there exists $\hat{c} < N_1 \cdot N_2 \cdot N_3$ s.t. $\hat{c} = c_1 \bmod N_1$,
 $\hat{c} = c_2 \bmod N_2$, $\hat{c} = c_3 \bmod N_3$

Security of Plain RSA

Plain RSA is not CPA secure

Enc is deterministic

- ① Common factors in public keys
 - If public keys N_i and N_j have $\gcd(N_i, N_j) > 1$, then can factor both!
 - This attack works even on secure variants of RSA
 - Has been detected in the wild
- ② Short m using small e
 - To make encryption fast, RSA often uses small exponent e (e.g., $e = 3$)
 - If $m < N^{1/e}$, then m^e does not wrap around mod N
 - Easy to find e^{th} roots over the integers
 - If $e = 3$ and $||N|| = 1024$ bits, can decrypt 300-bit m
- ③ Same m sent to multiple receivers
 - Suppose we encrypt m to 3 public keys N_1, N_2, N_3 using $e = 3$
 - $c_1 = [m^3 \bmod N_1]$, $c_2 = [m^3 \bmod N_2]$, $c_3 = [m^3 \bmod N_3]$
 - By CRT, there exists $\hat{c} < N_1 \cdot N_2 \cdot N_3$ s.t. $\hat{c} = c_1 \bmod N_1$,
 $\hat{c} = c_2 \bmod N_2$, $\hat{c} = c_3 \bmod N_3$
 - But, $m^3 < N_1 \cdot N_2 \cdot N_3$, so can take e^{th} root over integers

Padded RSA Encryption Scheme

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$:
 - Given $m \in \{0, 1\}^{\|N\| - \ell(n) - 2}$ – somewhat shorter message

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$:
 - Given $m \in \{0, 1\}^{\|N\| - \ell(n) - 2}$ – somewhat shorter message
 - Choose $r \leftarrow \{0, 1\}^{\ell(n)}$, and set $\hat{m} = r || m \in \mathbb{Z}_N^*$

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$:
 - Given $m \in \{0, 1\}^{\|N\| - \ell(n) - 2}$ – somewhat shorter message
 - Choose $r \leftarrow \{0, 1\}^{\ell(n)}$, and set $\hat{m} = r || m \in \mathbb{Z}_N^*$
 - Output $c = [\hat{m}^e \bmod N]$

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$:
 - Given $m \in \{0, 1\}^{\|N\| - \ell(n) - 2}$ – somewhat shorter message
 - Choose $r \leftarrow \{0, 1\}^{\ell(n)}$, and set $\hat{m} = r || m \in \mathbb{Z}_N^*$
 - Output $c = [\hat{m}^e \bmod N]$
- $\text{Dec}_{sk}(c)$:
 - Set $\hat{m} = [c^d \bmod N]$

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$:
 - Given $m \in \{0, 1\}^{\|N\| - \ell(n) - 2}$ – somewhat shorter message
 - Choose $r \leftarrow \{0, 1\}^{\ell(n)}$, and set $\hat{m} = r || m \in \mathbb{Z}_N^*$
 - Output $c = [\hat{m}^e \bmod N]$
- $\text{Dec}_{sk}(c)$:
 - Set $\hat{m} = [c^d \bmod N]$
 - Output $\|N\| - \ell(n) - 2$ least significant bits

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$:
 - Given $m \in \{0, 1\}^{\|N\| - \ell(n) - 2}$ – somewhat shorter message
 - Choose $r \leftarrow \{0, 1\}^{\ell(n)}$, and set $\hat{m} = r || m \in \mathbb{Z}_N^*$
 - Output $c = [\hat{m}^e \bmod N]$
- $\text{Dec}_{sk}(c)$:
 - Set $\hat{m} = [c^d \bmod N]$
 - Output $\|N\| - \ell(n) - 2$ least significant bits

Security:

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$:
 - Given $m \in \{0, 1\}^{\|N\| - \ell(n) - 2}$ – somewhat shorter message
 - Choose $r \leftarrow \{0, 1\}^{\ell(n)}$, and set $\hat{m} = r || m \in \mathbb{Z}_N^*$
 - Output $c = [\hat{m}^e \bmod N]$
- $\text{Dec}_{sk}(c)$:
 - Set $\hat{m} = [c^d \bmod N]$
 - Output $\|N\| - \ell(n) - 2$ least significant bits

Security:

- Avoids small message and multiple receivers attacks through random padding

Padded RSA Encryption Scheme

Let $\ell(n) \leq 2n - 4$

- $\text{Gen}(1^n)$: $(N, e, d) \leftarrow \text{GenRSA}(1^n)$, $pk = (N, e)$, $sk = (N, d)$
- $\text{Enc}_{pk}(m)$:
 - Given $m \in \{0, 1\}^{\|N\| - \ell(n) - 2}$ – somewhat shorter message
 - Choose $r \leftarrow \{0, 1\}^{\ell(n)}$, and set $\hat{m} = r || m \in \mathbb{Z}_N^*$
 - Output $c = [\hat{m}^e \bmod N]$
- $\text{Dec}_{sk}(c)$:
 - Set $\hat{m} = [c^d \bmod N]$
 - Output $\|N\| - \ell(n) - 2$ least significant bits

Security:

- Avoids small message and multiple receivers attacks through random padding
- Conjectured to be CPA secure, though not known how to prove this

- 1 Lecture 21 Review
- 2 RSA Encryption Scheme (Chapter 11.5)
- 3 Hybrid Encryption and CCA Security

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key
Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key
Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key
Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Suppose A wants to send many encrypted messages to B

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key
Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Suppose A wants to send many encrypted messages to B

- A chooses $k \leftarrow \{0, 1\}^n$

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key
Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Suppose A wants to send many encrypted messages to B

- A chooses $k \leftarrow \{0, 1\}^n$
- A computes $c = \text{Enc}_{pk_B}(k)$ and sends c to B

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key
Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Suppose A wants to send many encrypted messages to B

- A chooses $k \leftarrow \{0, 1\}^n$
- A computes $c = \text{Enc}_{pk_B}(k)$ and sends c to B
- B uses sk_B to decrypt c and learn k

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key
Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Suppose A wants to send many encrypted messages to B

- A chooses $k \leftarrow \{0, 1\}^n$
- A computes $c = \text{Enc}_{pk_B}(k)$ and sends c to B
- B uses sk_B to decrypt c and learn k
- A and B use private-key encryption with key k to encrypt the messages

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key
Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Suppose A wants to send many encrypted messages to B

- A chooses $k \leftarrow \{0, 1\}^n$
- A computes $c = \text{Enc}_{pk_B}(k)$ and sends c to B
- B uses sk_B to decrypt c and learn k
- A and B use private-key encryption with key k to encrypt the messages

Observations:

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key encryption. Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Suppose A wants to send many encrypted messages to B

- A chooses $k \leftarrow \{0, 1\}^n$
- A computes $c = \text{Enc}_{pk_B}(k)$ and sends c to B
- B uses sk_B to decrypt c and learn k
- A and B use private-key encryption with key k to encrypt the messages

Observations:

- Only one slow public-key encryption to send many messages

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key encryption. Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Suppose A wants to send many encrypted messages to B

- A chooses $k \leftarrow \{0, 1\}^n$
- A computes $c = \text{Enc}_{pk_B}(k)$ and sends c to B
- B uses sk_B to decrypt c and learn k
- A and B use private-key encryption with key k to encrypt the messages

Observations:

- Only one slow public-key encryption to send many messages
- Can have symmetric key chosen during encryption
 - This is known as key-encapsulation mechanism (KEM)

Hybrid Encryption

Public-key encryption is 2-3 orders of magnitude slower than private-key encryption. Question: Can we achieve functionality of public-key encryption at cost of private-key encryption?

Hybrid Encryption

Suppose A wants to send many encrypted messages to B

- A chooses $k \leftarrow \{0, 1\}^n$
- A computes $c = \text{Enc}_{pk_B}(k)$ and sends c to B
- B uses sk_B to decrypt c and learn k
- A and B use private-key encryption with key k to encrypt the messages

Observations:

- Only one slow public-key encryption to send many messages
- Can have symmetric key chosen during encryption
 - This is known as key-encapsulation mechanism (KEM)
- This is how most public-key encryption is used in practice

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

CCA (In)Security of El Gamal:

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

CCA (In)Security of El Gamal:

- \mathcal{A} gets $c = (g^y, h^y \cdot m)$

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

CCA (In)Security of El Gamal:

- \mathcal{A} gets $c = (g^y, h^y \cdot m)$
- Compute $c' = (g^y, 2h^y \cdot m) = (g^y, h^y \cdot 2m)$

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

CCA (In)Security of El Gamal:

- \mathcal{A} gets $c = (g^y, h^y \cdot m)$
- Compute $c' = (g^y, 2h^y \cdot m) = (g^y, h^y \cdot 2m)$
- \mathcal{A} asks $\text{Dec}_{sk}(c')$ to learn $2m$, can recover m from this

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

CCA (In)Security of El Gamal:

- \mathcal{A} gets $c = (g^y, h^y \cdot m)$
- Compute $c' = (g^y, 2h^y \cdot m) = (g^y, h^y \cdot 2m)$
- \mathcal{A} asks $\text{Dec}_{sk}(c')$ to learn $2m$, can recover m from this

CCA (In)Security of Padded RSA:

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

CCA (In)Security of El Gamal:

- \mathcal{A} gets $c = (g^y, h^y \cdot m)$
- Compute $c' = (g^y, 2h^y \cdot m) = (g^y, h^y \cdot 2m)$
- \mathcal{A} asks $\text{Dec}_{sk}(c')$ to learn $2m$, can recover m from this

CCA (In)Security of Padded RSA:

- \mathcal{A} gets $c = [\hat{m}^e \bmod N]$

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

CCA (In)Security of El Gamal:

- \mathcal{A} gets $c = (g^y, h^y \cdot m)$
- Compute $c' = (g^y, 2h^y \cdot m) = (g^y, h^y \cdot 2m)$
- \mathcal{A} asks $\text{Dec}_{sk}(c')$ to learn $2m$, can recover m from this

CCA (In)Security of Padded RSA:

- \mathcal{A} gets $c = [\hat{m}^e \bmod N]$
- \mathcal{A} computes $c' = [2^e \cdot \hat{m}^e \bmod N] = [(2\hat{m})^e \bmod N]$

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

CCA (In)Security of El Gamal:

- \mathcal{A} gets $c = (g^y, h^y \cdot m)$
- Compute $c' = (g^y, 2h^y \cdot m) = (g^y, h^y \cdot 2m)$
- \mathcal{A} asks $\text{Dec}_{sk}(c')$ to learn $2m$, can recover m from this

CCA (In)Security of Padded RSA:

- \mathcal{A} gets $c = [\hat{m}^e \bmod N]$
- \mathcal{A} computes $c' = [2^e \cdot \hat{m}^e \bmod N] = [(2\hat{m})^e \bmod N]$
- \mathcal{A} asks $\text{Dec}_{sk}(c')$ to learn $[2\hat{m} \bmod N]$, can recover m from this

What About CCA Security?

Similar to the case for private-key encryption, we can define CCA-security by giving \mathcal{A} access to a decryption oracle $\text{Dec}_{sk}(\cdot)$

CCA (In)Security of El Gamal:

- \mathcal{A} gets $c = (g^y, h^y \cdot m)$
- Compute $c' = (g^y, 2h^y \cdot m) = (g^y, h^y \cdot 2m)$
- \mathcal{A} asks $\text{Dec}_{sk}(c')$ to learn $2m$, can recover m from this

CCA (In)Security of Padded RSA:

- \mathcal{A} gets $c = [\hat{m}^e \bmod N]$
- \mathcal{A} computes $c' = [2^e \cdot \hat{m}^e \bmod N] = [(2\hat{m})^e \bmod N]$
- \mathcal{A} asks $\text{Dec}_{sk}(c')$ to learn $[2\hat{m} \bmod N]$, can recover m from this

Malleability

Both El Gamal and RSA are not CCA-secure because they are *malleable*.

Building CCA-Secure Public-Key Encryption – Try 1

Recall that Encrypt-then-authenticate was CCA-secure in the secret-key model

Question

Can we build Encrypt-then-sign using PKE and digital signatures?

Building CCA-Secure Public-Key Encryption – Try 1

Recall that Encrypt-then-authenticate was CCA-secure in the secret-key model

Question

Can we build Encrypt-then-sign using PKE and digital signatures?

The problem:

- Unfortunately, there is a private/public key mismatch

Building CCA-Secure Public-Key Encryption – Try 1

Recall that Encrypt-then-authenticate was CCA-secure in the secret-key model

Question

Can we build Encrypt-then-sign using PKE and digital signatures?

The problem:

- Unfortunately, there is a private/public key mismatch
- We want encryption to be done by anyone, but signing key is private

Building CCA-Secure Public-Key Encryption – Try 1

Recall that Encrypt-then-authenticate was CCA-secure in the secret-key model

Question

Can we build Encrypt-then-sign using PKE and digital signatures?

The problem:

- Unfortunately, there is a private/public key mismatch
- We want encryption to be done by anyone, but signing key is private
- Without signing key, public cannot encrypt

Building CCA-Secure Public-Key Encryption

- Making CCA-secure public-key encryption is fairly complicated

Building CCA-Secure Public-Key Encryption

- Making CCA-secure public-key encryption is fairly complicated
- A standardized scheme, RSA-OAEP, can be proven secure in the random oracle model

Building CCA-Secure Public-Key Encryption

- Making CCA-secure public-key encryption is fairly complicated
- A standardized scheme, RSA-OAEP, can be proven secure in the random oracle model
- Not known whether you can turn CPA-secure encryption into CCA-secure encryption generically