

# Lecture 8: Public-Key Encryption

# The Setting

- Alice and Bob don't share any secret

# The Setting

- Alice and Bob don't share any secret
- Alice wants to send a private message  $m$  to Bob

# The Setting

- Alice and Bob don't share any secret
- Alice wants to send a private message  $m$  to Bob
- Goals:

# The Setting

- Alice and Bob don't share any secret
- Alice wants to send a private message  $m$  to Bob
- Goals:
  - **Public key:** Encryption and decryption keys are different. Encryption key can be “public”

# The Setting

- Alice and Bob don't share any secret
- Alice wants to send a private message  $m$  to Bob
- Goals:
  - **Public key:** Encryption and decryption keys are different. Encryption key can be “public”
  - **Correctness:** Alice can compute an encryption  $c$  of  $m$  using  $pk$ . Bob can decrypt  $m$  from  $c$  correctly using  $sk$

# The Setting

- Alice and Bob don't share any secret
- Alice wants to send a private message  $m$  to Bob
- Goals:
  - **Public key:** Encryption and decryption keys are different. Encryption key can be “public”
  - **Correctness:** Alice can compute an encryption  $c$  of  $m$  using  $pk$ . Bob can decrypt  $m$  from  $c$  correctly using  $sk$
  - **Security:** No eavesdropper can distinguish between encryptions of  $m$  and  $m'$  (even using  $pk$ )

# Definition

- **Syntax:**

- $\text{Gen}(1^n) \rightarrow (pk, sk)$
- $\text{Enc}(pk, m) \rightarrow c$
- $\text{Dec}(sk, c) \rightarrow m'$  or  $\perp$

All algorithms are polynomial time

- **Correctness:** For every  $m$ ,  $\text{Dec}(sk, \text{Enc}(pk, m)) = m$ , where  $(pk, sk) \leftarrow \text{Gen}(1^n)$
- **Security:** ?



## Definition ((Weak) Indistinguishability Security)

A public-key encryption scheme  $(\text{Gen}, \text{Enc}, \text{Dec})$  is weakly indistinguishably secure under chosen plaintext attack (weak IND-CPA) if for all n.u. PPT adversaries  $\mathcal{A}$ , there exists a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr \left[ \begin{array}{l} (pk, sk) \xleftarrow{\$} \text{Gen}(1^n), \\ (m_0, m_1) \leftarrow \mathcal{A}(1^n), \\ b \xleftarrow{\$} \{0, 1\} \end{array} : \mathcal{A}(pk, \text{Enc}(pk, m_b)) = b \right] \leq \frac{1}{2} + \mu(n)$$

- 1 Think: Semantic security style definition?

## Definition ((Weak) Indistinguishability Security)

A public-key encryption scheme  $(\text{Gen}, \text{Enc}, \text{Dec})$  is weakly indistinguishably secure under chosen plaintext attack (weak IND-CPA) if for all n.u. PPT adversaries  $\mathcal{A}$ , there exists a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr \left[ \begin{array}{l} (pk, sk) \xleftarrow{\$} \text{Gen}(1^n), \\ (m_0, m_1) \leftarrow \mathcal{A}(1^n), \\ b \xleftarrow{\$} \{0, 1\} \end{array} : \mathcal{A}(pk, \text{Enc}(pk, m_b)) = b \right] \leq \frac{1}{2} + \mu(n)$$

- 1 Think: Semantic security style definition?
- 2 Think Equivalence of above definition and semantic security

## Security (contd.)

A stronger definition:

### Definition (Indistinguishability Security)

A public-key encryption scheme  $(\text{Gen}, \text{Enc}, \text{Dec})$  is indistinguishably secure under chosen plaintext attack (IND-CPA) if for all n.u. PPT adversaries  $\mathcal{A}$ , there exists a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr \left[ \begin{array}{l} (pk, sk) \xleftarrow{\$} \text{Gen}(1^n), \\ (m_0, m_1) \leftarrow \mathcal{A}(1^n, pk), \\ b \xleftarrow{\$} \{0, 1\} \end{array} : \mathcal{A}(pk, \text{Enc}(m_b)) = b \right] \leq \frac{1}{2} + \mu(n)$$

- 1 Think: IND-CPA is stronger than weak IND-CPA

## Security (contd.)

A stronger definition:

### Definition (Indistinguishability Security)

A public-key encryption scheme  $(\text{Gen}, \text{Enc}, \text{Dec})$  is indistinguishably secure under chosen plaintext attack (IND-CPA) if for all n.u. PPT adversaries  $\mathcal{A}$ , there exists a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr \left[ \begin{array}{l} (pk, sk) \xleftarrow{\$} \text{Gen}(1^n), \\ (m_0, m_1) \leftarrow \mathcal{A}(1^n, pk), \\ b \xleftarrow{\$} \{0, 1\} \end{array} : \mathcal{A}(pk, \text{Enc}(m_b)) = b \right] \leq \frac{1}{2} + \mu(n)$$

- 1 Think: IND-CPA is stronger than weak IND-CPA
- 2 Think Multi-message security?

# Multi-message security

## Lemma (Multi-message security)

*One-message security implies multi-message security for public-key encryption*

# Multi-message security

## Lemma (Multi-message security)

*One-message security implies multi-message security for public-key encryption*

- 1 Think: Proof?

# Multi-message security

## Lemma (Multi-message security)

*One-message security implies multi-message security for public-key encryption*

- 1 Think: Proof?
- 2 Corollary: Suffices to consider single-bit message

# One-way Functions, Revisited

## Definition (Collection of OWFs)

A collection of one-way functions is a family  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following conditions:



# One-way Functions, Revisited

## Definition (Collection of OWFs)

A collection of one-way functions is a family  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following conditions:

- **Sampling function:** There exists a PPT  $\text{Gen}$  s.t.  $\text{Gen}(1^n)$  outputs  $i \in \mathcal{I}$

# One-way Functions, Revisited

## Definition (Collection of OWFs)

A collection of one-way functions is a family  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following conditions:

- **Sampling function:** There exists a PPT Gen s.t.  $\text{Gen}(1^n)$  outputs  $i \in \mathcal{I}$
- **Sampling from domain:** There exists a PPT algorithm that on input  $i$  outputs a uniformly random element of  $\mathcal{D}_i$

# One-way Functions, Revisited

## Definition (Collection of OWFs)

A collection of one-way functions is a family  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following conditions:

- **Sampling function:** There exists a PPT Gen s.t.  $\text{Gen}(1^n)$  outputs  $i \in \mathcal{I}$
- **Sampling from domain:** There exists a PPT algorithm that on input  $i$  outputs a uniformly random element of  $\mathcal{D}_i$
- **Evaluation:** There exists a PPT algorithm that on input  $i, x \in \mathcal{D}_i$  outputs  $f_i(x)$

# One-way Functions, Revisited

## Definition (Collection of OWFs)

A collection of one-way functions is a family  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following conditions:

- **Sampling function:** There exists a PPT Gen s.t. Gen( $1^n$ ) outputs  $i \in \mathcal{I}$
- **Sampling from domain:** There exists a PPT algorithm that on input  $i$  outputs a uniformly random element of  $\mathcal{D}_i$
- **Evaluation:** There exists a PPT algorithm that on input  $i, x \in \mathcal{D}_i$  outputs  $f_i(x)$
- **Hard to invert::** For every n.u. PPT adversary  $\mathcal{A}$ , there exists a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr [i \leftarrow \text{Gen}(1^n), x \leftarrow \mathcal{D}_i, y \leftarrow f_i(x) : f_i(\mathcal{A}(1^n, i, y)) = y] \leq \mu(n)$$

# One-way Functions, Revisited (contd.)

## Theorem

*There exists a collection of one-way functions iff there exists a strong one-way function*

# One-way Functions, Revisited (contd.)

## Theorem

*There exists a collection of one-way functions iff there exists a strong one-way function*

Think: Proof?

# Collection of One-way Permutations

## Definition

A collection  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  is a collection of one-way permutations if  $\mathcal{F}$  is a collection of OWFs and for every  $i \in \mathcal{I}$ ,  $f_i$  is a permutation.

# Trapdoor Permutations

## Definition (Trapdoor OWPs)

A collection of trapdoor permutations is a family of permutations  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following properties:



# Trapdoor Permutations

## Definition (Trapdoor OWPs)

A collection of trapdoor permutations is a family of permutations  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following properties:

- **Sampling function:**  $\exists$  a PPT  $\text{Gen}$  s.t.  $\text{Gen}(1^n)$  outputs  $(i, t) \in \mathcal{I}$

# Trapdoor Permutations

## Definition (Trapdoor OWPs)

A collection of trapdoor permutations is a family of permutations  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following properties:

- **Sampling function:**  $\exists$  a PPT  $\text{Gen}$  s.t.  $\text{Gen}(1^n)$  outputs  $(i, t) \in \mathcal{I}$
- **Sampling from domain:**  $\exists$  a PPT algorithm that on input  $i$  outputs a uniformly random element of  $\mathcal{D}_i$

# Trapdoor Permutations

## Definition (Trapdoor OWPs)

A collection of trapdoor permutations is a family of permutations  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following properties:

- **Sampling function:**  $\exists$  a PPT  $\text{Gen}$  s.t.  $\text{Gen}(1^n)$  outputs  $(i, t) \in \mathcal{I}$
- **Sampling from domain:**  $\exists$  a PPT algorithm that on input  $i$  outputs a uniformly random element of  $\mathcal{D}_i$
- **Evaluation:**  $\exists$  PPT that on input  $i, x \in \mathcal{D}_i$  outputs  $f_i(x)$

# Trapdoor Permutations

## Definition (Trapdoor OWPs)

A collection of trapdoor permutations is a family of permutations  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following properties:

- **Sampling function:**  $\exists$  a PPT  $\text{Gen}$  s.t.  $\text{Gen}(1^n)$  outputs  $(i, t) \in \mathcal{I}$
- **Sampling from domain:**  $\exists$  a PPT algorithm that on input  $i$  outputs a uniformly random element of  $\mathcal{D}_i$
- **Evaluation:**  $\exists$  PPT that on input  $i, x \in \mathcal{D}_i$  outputs  $f_i(x)$
- **Hard to invert:**  $\forall$  n.u. PPT adversary  $\mathcal{A}$ ,  $\exists$  a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr [i \leftarrow \text{Gen}(1^n), x \leftarrow \mathcal{D}_i, y \leftarrow f_i(x) : f_i(\mathcal{A}(1^n, i, y)) = y] \leq \mu(n)$$

# Trapdoor Permutations

## Definition (Trapdoor OWPs)

A collection of trapdoor permutations is a family of permutations  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  satisfying the following properties:

- **Sampling function:**  $\exists$  a PPT  $\text{Gen}$  s.t.  $\text{Gen}(1^n)$  outputs  $(i, t) \in \mathcal{I}$
- **Sampling from domain:**  $\exists$  a PPT algorithm that on input  $i$  outputs a uniformly random element of  $\mathcal{D}_i$
- **Evaluation:**  $\exists$  PPT that on input  $i, x \in \mathcal{D}_i$  outputs  $f_i(x)$
- **Hard to invert:**  $\forall$  n.u. PPT adversary  $\mathcal{A}$ ,  $\exists$  a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr [i \leftarrow \text{Gen}(1^n), x \leftarrow \mathcal{D}_i, y \leftarrow f_i(x) : f_i(\mathcal{A}(1^n, i, y)) = y] \leq \mu(n)$$

- **Inversion with trapdoor:**  $\exists$  a PPT algorithm that given  $(i, t, y)$  outputs  $f_i^{-1}(y)$

# Public-key Encryption from Trapdoor Permutations

Let  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  be a family of trapdoor permutations

# Public-key Encryption from Trapdoor Permutations

Let  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  be a family of trapdoor permutations

- $\text{Gen}(1^n)$ :  $(f_i, f_i^{-1}) \leftarrow \text{Gen}_T(1^n)$ . Output  $(pk, sk) \leftarrow ((f_i, h_i), f_i^{-1})$

# Public-key Encryption from Trapdoor Permutations

Let  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  be a family of trapdoor permutations

- $\text{Gen}(1^n)$ :  $(f_i, f_i^{-1}) \leftarrow \text{Gen}_T(1^n)$ . Output  $(pk, sk) \leftarrow ((f_i, h_i), f_i^{-1})$
- $\text{Enc}(pk, m)$ : Pick  $r \xleftarrow{\$} \{0, 1\}^n$ . Output  $(f_i(r), h_i(r) \oplus m)$



# Public-key Encryption from Trapdoor Permutations

Let  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  be a family of trapdoor permutations

- $\text{Gen}(1^n)$ :  $(f_i, f_i^{-1}) \leftarrow \text{Gen}_T(1^n)$ . Output  $(pk, sk) \leftarrow ((f_i, h_i), f_i^{-1})$
- $\text{Enc}(pk, m)$ : Pick  $r \xleftarrow{\$} \{0, 1\}^n$ . Output  $(f_i(r), h_i(r) \oplus m)$
- $\text{Dec}(sk, (c_1, c_2))$ :  $r \leftarrow f_i^{-1}(c_1)$ . Output  $c_2 \oplus h_i(r)$

# Public-key Encryption from Trapdoor Permutations

Let  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  be a family of trapdoor permutations

- $\text{Gen}(1^n)$ :  $(f_i, f_i^{-1}) \leftarrow \text{Gen}_T(1^n)$ . Output  $(pk, sk) \leftarrow ((f_i, h_i), f_i^{-1})$
- $\text{Enc}(pk, m)$ : Pick  $r \xleftarrow{\$} \{0, 1\}^n$ . Output  $(f_i(r), h_i(r) \oplus m)$
- $\text{Dec}(sk, (c_1, c_2))$ :  $r \leftarrow f_i^{-1}(c_1)$ . Output  $c_2 \oplus h_i(r)$

## Theorem (PKE from Trapdoor Permutations)

$(\text{Gen}, \text{Enc}, \text{Dec})$  is IND-CPA secure public-key encryption scheme

# Public-key Encryption from Trapdoor Permutations

Let  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  be a family of trapdoor permutations

- $\text{Gen}(1^n)$ :  $(f_i, f_i^{-1}) \leftarrow \text{Gen}_T(1^n)$ . Output  $(pk, sk) \leftarrow ((f_i, h_i), f_i^{-1})$
- $\text{Enc}(pk, m)$ : Pick  $r \xleftarrow{\$} \{0, 1\}^n$ . Output  $(f_i(r), h_i(r) \oplus m)$
- $\text{Dec}(sk, (c_1, c_2))$ :  $r \leftarrow f_i^{-1}(c_1)$ . Output  $c_2 \oplus h_i(r)$

## Theorem (PKE from Trapdoor Permutations)

$(\text{Gen}, \text{Enc}, \text{Dec})$  is IND-CPA secure public-key encryption scheme

- Think: Proof?

# Public-key Encryption from Trapdoor Permutations

Let  $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  be a family of trapdoor permutations

- $\text{Gen}(1^n)$ :  $(f_i, f_i^{-1}) \leftarrow \text{Gen}_T(1^n)$ . Output  $(pk, sk) \leftarrow ((f_i, h_i), f_i^{-1})$
- $\text{Enc}(pk, m)$ : Pick  $r \xleftarrow{\$} \{0, 1\}^n$ . Output  $(f_i(r), h_i(r) \oplus m)$
- $\text{Dec}(sk, (c_1, c_2))$ :  $r \leftarrow f_i^{-1}(c_1)$ . Output  $c_2 \oplus h_i(r)$

## Theorem (PKE from Trapdoor Permutations)

$(\text{Gen}, \text{Enc}, \text{Dec})$  is IND-CPA secure public-key encryption scheme

- Think: Proof?
- How to build trapdoor permutations?

# Candidate Trapdoor Permutations

## Definition (RSA Collection)

**RSA** =  $\{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  where:

- $\mathcal{I} = \{(N, e) \mid N = p \cdot q \text{ s.t. } p, q \in \Pi_n, e \in \mathbb{Z}_{\Phi(N)}^*\}$
- $\mathcal{D}_i = \{x \mid x \in \mathbb{Z}_N^*\}$
- $\mathcal{R}_i = \mathbb{Z}_N^*$
- $\text{Gen}(1^n) \rightarrow ((N, e), d)$  where  $(N, e) \in \mathcal{I}$  and  $e \cdot d = 1 \pmod{\Phi(N)}$
- $f_{N,e}(x) = x^e \pmod{N}$
- $f_{N,d}^{-1}(y) = y^d \pmod{N}$

# Candidate Trapdoor Permutations

## Definition (RSA Collection)

**RSA** =  $\{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$  where:

- $\mathcal{I} = \{(N, e) \mid N = p \cdot q \text{ s.t. } p, q \in \Pi_n, e \in \mathbb{Z}_{\Phi(N)}^*\}$
- $\mathcal{D}_i = \{x \mid x \in \mathbb{Z}_N^*\}$
- $\mathcal{R}_i = \mathbb{Z}_N^*$
- $\text{Gen}(1^n) \rightarrow ((N, e), d)$  where  $(N, e) \in \mathcal{I}$  and  $e \cdot d = 1 \pmod{\Phi(N)}$
- $f_{N,e}(x) = x^e \pmod{N}$
- $f_{N,d}^{-1}(y) = y^d \pmod{N}$

- Think: Why is  $f_{N,e}$  a permutation?

## Candidate Trapdoor Permutations (contd.)

### Assumption (RSA Assumption)

For any n.u. PPT adversary  $\mathcal{A}$ , there exists a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr \left[ \begin{array}{l} p, q \xleftarrow{\$} \Pi_n, N = p \cdot q, e \xleftarrow{\$} \mathbb{Z}_{\Phi(N)}^*, \\ y \xleftarrow{\$} \mathbb{Z}_N^*; x \leftarrow \mathcal{A}(N, e, y) \end{array} : x^e = y \pmod N \right] \leq \mu(n)$$

## Candidate Trapdoor Permutations (contd.)

### Assumption (RSA Assumption)

For any n.u. PPT adversary  $\mathcal{A}$ , there exists a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr \left[ \begin{array}{l} p, q \xleftarrow{\$} \Pi_n, N = p \cdot q, e \xleftarrow{\$} \mathbb{Z}_{\Phi(N)}^*, \\ y \xleftarrow{\$} \mathbb{Z}_N^*; x \leftarrow \mathcal{A}(N, e, y) \end{array} : x^e = y \pmod N \right] \leq \mu(n)$$

- Think: RSA assumption implies the factoring assumption



## Candidate Trapdoor Permutations (contd.)

### Assumption (RSA Assumption)

For any n.u. PPT adversary  $\mathcal{A}$ , there exists a negligible function  $\mu(\cdot)$  s.t.:

$$\Pr \left[ \begin{array}{l} p, q \xleftarrow{\$} \Pi_n, N = p \cdot q, e \xleftarrow{\$} \mathbb{Z}_{\Phi(N)}^*, \\ y \xleftarrow{\$} \mathbb{Z}_N^*; x \leftarrow \mathcal{A}(N, e, y) \end{array} : x^e = y \pmod N \right] \leq \mu(n)$$

- Think: RSA assumption implies the factoring assumption

### Theorem

Assuming the RSA assumption, the RSA collection is a family of trapdoor permutations

# Food for Thought

- Direct (more efficient) constructions of PKE (e.g., El-Gamal)
- Stronger security notions:
  - Indistinguishability under chosen-ciphertext attacks (IND-CCA) [Naor-Segev],[Dolev-Dwork-Naor],[Sahai]
  - Circular security/key-dependent message security [Boneh-Halevi-Hamburg-Ostrovsky]
  - Leakage-resilient encryption [Dziembowski-Pietrzak],[Akavia-Goldwasser-Vaikuntanathan]
- Weaker security notions:
  - Deterministic encryption [Bellare-Boldyreva-O'Neill]