Secret-Key & Public-Key Encryption

601.642/442: Modern Cryptography

Fall 2022

Secret-Key Encryption

The Setting

- Alice and Bob share a secret key $s \in \{0,1\}^n$
- Alice wants to send a private message m to Bob
- Goals:
  - **Correctness:** Alice can compute an encoding c of m using s. Bob can decode m from c correctly using s
  - **Security:** No eavesdropper can distinguish between encodings of m and m'

Definition

- **Syntax:**
  - $\mathsf{Gen}(1^n) \to s$
  - $\mathsf{Enc}(s, m) \to c$
  - $\mathsf{Dec}(s, c) \to m'$ or $\perp$

  All algorithms are polynomial time

- **Correctness:** For every m, $\mathsf{Dec}(s, \mathsf{Enc}(s, m)) = m$, where $s \xleftarrow{\$} \mathsf{Gen}(1^n)$

- **Security:** We have already seen *one-time* security. Today, we will consider **multi-message** security.

Multi-message Secure Encryption

Definition (Multi-message Secure Encryption)

A secret-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is multi-message secure if for all n.u. PPT adversaries $\mathcal{A}$, for all polynomials $q(\cdot)$, there exists a negligible function $\mu(\cdot)$ s.t.:

$$\Pr\left[\begin{array}{r} s \xleftarrow{\$} \mathsf{Gen}(1^n), \\ \left\{\left(m_0^i, m_1^i\right)\right\}_{i=1}^{q(n)} \leftarrow \mathcal{A}(1^n), \\ b \xleftarrow{\$} \{0,1\} \end{array} : \mathcal{A}\left(\left\{\mathsf{Enc}\left(m_b^i\right)\right\}_{i=1}^{q(n)}\right) = b\right] \leqslant \frac{1}{2} + \mu(n)$$

1. Think: Security against *adaptive* adversaries (who may choose message pairs in an adaptive manner based on previously seen ciphertexts)?

Necessity of Randomized Encryption

Necessity of Randomized Encryption

Theorem (Randomized Encryption)

A multi-message secure encryption scheme cannot be deterministic and stateless.

Necessity of Randomized Encryption

Theorem (Randomized Encryption)

A multi-message secure encryption scheme cannot be deterministic and stateless.

Think: Proof?

Encryption using PRFs

Let $\{f_s : \{0,1\}^n \rightarrow \{0,1\}^n\}$ be a family of PRFs

Encryption using PRFs

Let $\{f_s : \{0,1\}^n \rightarrow \{0,1\}^n\}$ be a family of PRFs

- $\mathsf{Gen}(1^n)$: $s \xleftarrow{\$} \{0,1\}^n$

Encryption using PRFs

Let $\{f_s : \{0,1\}^n \to \{0,1\}^n\}$ be a family of PRFs

- $\mathsf{Gen}(1^n)$: $s \xleftarrow{\$} \{0,1\}^n$
- $\mathsf{Enc}(s, m)$: Pick $r \xleftarrow{\$} \{0,1\}^n$. Output $(r, m \oplus f_s(r))$

Encryption using PRFs

Let $\{f_s : \{0,1\}^n \to \{0,1\}^n\}$ be a family of PRFs

- $\mathsf{Gen}(1^n)$: $s \xleftarrow{\$} \{0,1\}^n$
- $\mathsf{Enc}(s, m)$: Pick $r \xleftarrow{\$} \{0,1\}^n$. Output $(r, m \oplus f_s(r))$
- $\mathsf{Dec}(s, (r, c))$: Output $c \oplus f_s(r)$

Encryption using PRFs

Let $\{f_s : \{0,1\}^n \to \{0,1\}^n\}$ be a family of PRFs

- $\mathsf{Gen}(1^n)$: $s \xleftarrow{\$} \{0,1\}^n$
- $\mathsf{Enc}(s, m)$: Pick $r \xleftarrow{\$} \{0,1\}^n$. Output $(r, m \oplus f_s(r))$
- $\mathsf{Dec}(s, (r, c))$: Output $c \oplus f_s(r)$

Theorem (Encryption from PRF)

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a multi-message secure encryption scheme*

Encryption using PRFs

Let $\{f_s : \{0,1\}^n \rightarrow \{0,1\}^n\}$ be a family of PRFs

- $\mathsf{Gen}(1^n)$: $s \xleftarrow{\$} \{0,1\}^n$
- $\mathsf{Enc}(s, m)$: Pick $r \xleftarrow{\$} \{0,1\}^n$. Output $(r, m \oplus f_s(r))$
- $\mathsf{Dec}(s, (r, c))$: Output $c \oplus f_s(r)$

Theorem (Encryption from PRF)

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a multi-message secure encryption scheme*

- <u>Think:</u> Proof?

Proof of Security

Proof via hybrids:

- H_1: Real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 0$)

Proof of Security

Proof via hybrids:

- H_1: Real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 0$)
- H_2: Replace f_s with random function $f \xleftarrow{\$} \mathcal{F}_n$

Proof of Security

Proof via hybrids:

- H_1: Real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 0$)
- H_2: Replace f_s with random function $f \xleftarrow{\$} \mathcal{F}_n$
- H_3: Switch to one-time pad encryption

Proof of Security

Proof via hybrids:

- H_1: Real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 0$)
- H_2: Replace f_s with random function $f \xleftarrow{\$} \mathcal{F}_n$
- H_3: Switch to one-time pad encryption
- H_4: Switch to encryption of $m_1^1, \ldots, m_1^{q(n)}$

Proof of Security

Proof via hybrids:

- H_1: Real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 0$)
- H_2: Replace f_s with random function $f \xleftarrow{\$} \mathcal{F}_n$
- H_3: Switch to one-time pad encryption
- H_4: Switch to encryption of $m_1^1, \ldots, m_1^{q(n)}$
- H_5: Use random function $f \xleftarrow{\$} \mathcal{F}_n$ to encrypt

Proof of Security

Proof via hybrids:

- H_1: Real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 0$)
- H_2: Replace f_s with random function $f \xleftarrow{\$} \mathcal{F}_n$
- H_3: Switch to one-time pad encryption
- H_4: Switch to encryption of $m_1^1, \ldots, m_1^{q(n)}$
- H_5: Use random function $f \xleftarrow{\$} \mathcal{F}_n$ to encrypt
- H_6: Encrypt using f_s. Same as real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 1$)

Proof of Security

Proof via hybrids:

- H_1: Real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 0$)
- H_2: Replace f_s with random function $f \xleftarrow{\$} \mathcal{F}_n$
- H_3: Switch to one-time pad encryption
- H_4: Switch to encryption of $m_1^1, \ldots, m_1^{q(n)}$
- H_5: Use random function $f \xleftarrow{\$} \mathcal{F}_n$ to encrypt
- H_6: Encrypt using f_s. Same as real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 1$)

Proof of Security

Proof via hybrids:

- H_1: Real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 0$)
- H_2: Replace f_s with random function $f \xleftarrow{\$} \mathcal{F}_n$
- H_3: Switch to one-time pad encryption
- H_4: Switch to encryption of $m_1^1, \ldots, m_1^{q(n)}$
- H_5: Use random function $f \xleftarrow{\$} \mathcal{F}_n$ to encrypt
- H_6: Encrypt using f_s. Same as real experiment with $m_0^1, \ldots, m_0^{q(n)}$ (i.e., $b = 1$)

<u>Think</u>: Non-adaptive vs adaptive queries

Semantic Security

Definition (Semantic Security)

A secret-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is semantically secure if there exists a PPT simulator algorithm $\mathcal{S}$ s.t. the following two experiments generate computationally indistinguishable outputs:

$$\left\{ \begin{array}{r} (m, z) \leftarrow M(1^n), \\ s \leftarrow \mathsf{Gen}(1^n), \\ \text{Output } (\mathsf{Enc}(s, m), z) \end{array} \right\} \approx \left\{ \begin{array}{l} (m, z) \leftarrow M(1^n), \\ \text{Output } S(1^n, z) \end{array} \right\}$$

where M is a machine that randomly samples a message from the message space and arbitrary auxiliary information.

Semantic Security

Definition (Semantic Security)

A secret-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is semantically secure if there exists a PPT simulator algorithm $\mathcal{S}$ s.t. the following two experiments generate computationally indistinguishable outputs:

$$\left\{ \begin{array}{r} (m, z) \leftarrow M(1^n), \\ s \leftarrow \mathsf{Gen}(1^n), \\ \text{Output } (\mathsf{Enc}(s, m), z) \end{array} \right\} \approx \left\{ \begin{array}{l} (m, z) \leftarrow M(1^n), \\ \text{Output } \mathcal{S}(1^n, z) \end{array} \right\}$$

where M is a machine that randomly samples a message from the message space and arbitrary auxiliary information.

- Indistinguishability security $\Leftrightarrow$ Semantic security

Semantic Security

Definition (Semantic Security)

A secret-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is semantically secure if there exists a PPT simulator algorithm $\mathcal{S}$ s.t. the following two experiments generate computationally indistinguishable outputs:

$$\left\{ \begin{array}{r} (m, z) \leftarrow M(1^n), \\ s \leftarrow \mathsf{Gen}(1^n), \\ \text{Output } (\mathsf{Enc}(s, m), z) \end{array} \right\} \approx \left\{ \begin{array}{l} (m, z) \leftarrow M(1^n), \\ \text{Output } \mathcal{S}(1^n, z) \end{array} \right\}$$

where M is a machine that randomly samples a message from the message space and arbitrary auxiliary information.

- Indistinguishability security $\Leftrightarrow$ Semantic security
- Think: Proof?

Food for Thought

Secret-key Encryption in practice:

- Block ciphers with fixed input length (e.g., AES)
- Encryption modes to encrypt arbitrarily long messages (e.g., CBC)
- Stream ciphers for stateful encryption
- Cryptanalysis (e.g., Differential Cryptanalysis)

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 <u>don't</u> 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 <u>don't</u> 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:**
  - $\mathsf{Gen}(1^n) \rightarrow (pk, sk)$
  - $\mathsf{Enc}(pk, m) \rightarrow c$
  - $\mathsf{Dec}(sk, c) \rightarrow m'$ or $\perp$

  All algorithms are polynomial time

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

- **Security:** ?

Security

Definition ((Weak) Indistinguishability Security)

A public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 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.:

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

1. Think: Semantic security style definition?

Security

Definition ((Weak) Indistinguishability Security)

A public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 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.:

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

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 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 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.:

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

1. <u>Think:</u> IND-CPA is stronger than weak IND-CPA

Security (contd.)

A stronger definition:

Definition (Indistinguishability Security)

A public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 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.:

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

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

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 $\mathcal{F} = \{f_i : \mathcal{D}_i \to \mathcal{R}_i\}_{i \in \mathcal{I}}$ satisfying the following properties:

- **Sampling function:** $\exists$ a PPT Gen s.t. $\mathsf{Gen}(1^n)$ 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 Gen s.t. $\mathsf{Gen}(1^n)$ 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 $\mathcal{F} = \{f_i : \mathcal{D}_i \to \mathcal{R}_i\}_{i \in \mathcal{I}}$ satisfying the following properties:

- **Sampling function:** $\exists$ a PPT Gen s.t. $\mathsf{Gen}(1^n)$ 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 $\mathcal{F} = \{f_i : \mathcal{D}_i \to \mathcal{R}_i\}_{i \in \mathcal{I}}$ satisfying the following properties:

- **Sampling function:** $\exists$ a PPT Gen s.t. $\mathsf{Gen}(1^n)$ 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.:

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

Trapdoor Permutations

Definition (Trapdoor OWPs)

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

- **Sampling function:** $\exists$ a PPT Gen s.t. $\mathsf{Gen}(1^n)$ 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.:

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

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

Public-key Encryption from Trapdoor Permutations

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

Public-key Encryption from Trapdoor Permutations

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

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

Public-key Encryption from Trapdoor Permutations

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

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

Public-key Encryption from Trapdoor Permutations

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

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

Public-key Encryption from Trapdoor Permutations

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

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

Theorem (PKE from Trapdoor Permutations)

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

Public-key Encryption from Trapdoor Permutations

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

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

Theorem (PKE from Trapdoor Permutations)

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

- Think: Proof?

Public-key Encryption from Trapdoor Permutations

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

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

Theorem (PKE from Trapdoor Permutations)

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

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

Candidate Trapdoor Permutations

Definition (RSA Collection)

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

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

Candidate Trapdoor Permutations (contd.)

Definition (RSA Assumption)

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

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

Candidate Trapdoor Permutations (contd.)

Definition (RSA Assumption)

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

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

- Think: RSA assumption implies the factoring assumption

Candidate Trapdoor Permutations (contd.)

Definition (RSA Assumption)

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

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

- 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]