Lecture 15: Public Key Encryption: I

Instructor: Omkant Pandey

Spring 2017 (CSE 594)

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:**
  - $\mathsf{Gen}(1^n) \rightarrow (pk, sk)$
  - $\mathsf{Enc}(pk, m) \rightarrow c$
  - $\mathsf{Dec}(sk, c) \rightarrow m'$ 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?
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.:

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

1. <u>Think:</u> IND-CPA is stronger than weak IND-CPA
2. <u>Think</u> 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 Gen s.t. $\mathsf{Gen}(1^n)$ 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. $\mathsf{Gen}(1^n)$ 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. $\mathsf{Gen}(1^n)$ 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. $\mathsf{Gen}(1^n)$ 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.:

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

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

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 $(pk, sk) \leftarrow ((f_i, h_i), f_i^{-1})$

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 $\mathcal{F} = \{f_i : \mathcal{D}_i \rightarrow \mathcal{R}_i\}_{i \in \mathcal{I}}$ 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 $(f_i(r), h_i(r) \oplus m)$

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*

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

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 $e \cdot d = 1 \mod \Phi(N)$
- $f_{N,e}(x) = x^e \mod N$
- $f^{-1}_{N,d}(y) = y^d \mod N$

- 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.:

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

Theorem

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

Candidate Trapdoor Permutations (contd.)

Assumption (RSA Assumption)

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

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

- 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]