VII. Public-key encryption

Private-key encryption

- very efficient,
- but needs shared secret key.
- key distribution, key agreement

Public-key encryption

- no shared keys,
- but less efficient than private-key encryption.
- used in combination with private-key encryption
- hybrid encryption

Public-key encryption schemes

Definition 7.1 A public-key encryption scheme is a triple $(\text{Gen}, \text{Enc}, \text{Dec})$ of ppts such that:

1. Gen on input 1^n outputs pair of keys (pk, sk). pk called public key, sk called secret key, $|pk|, |sk| \geq n$.
2. Enc on input a public key pk and a message m (from set depending on pk) outputs a ciphertext c, $c \leftarrow \text{Enc}_{pk}(m)$.
3. Dec on input a private key and a ciphertext c outputs a message m or a special failure symbol $\perp$. We assume Dec is deterministic and write $m := \text{Dec}_{sk}(c)$.

There must be a negligible function μ such that for all $(pk, sk) \leftarrow \text{Gen}(1^n)$ and all possible messages

$$\Pr\left[\text{Dec}_{sk}\left(\text{Enc}_{pk}(m)\right) \neq m\right] \leq \mu(n).$$

Public-key encryption

Alice

- encrypts message m with pk_B
- sends encrypted message/ciphertext c

Bob

- generates pair of public key pk_B and secret key sk_B
- makes pk_B public
- decrypts with sk_B

The eavesdropping game

Eavesdropping indistinguishability game $\mathrm{PubK}^{eav}_{A,\Pi}$

1. $(pk, sk) \leftarrow Gen(1^n)$.
2. A is given pk and outputs pair of message m_0, m_1 with $|m_0| = |m_1|$.
3. $b \leftarrow \{0,1\}, c \leftarrow Enc_{pk}(m_b)$ and c is given to A.
4. A outputs bit b'.
5. Output of experiment is 1, if $b = b'$, otherwise output is 0.

Write $\mathrm{PubK}^{eav}_{A,\Pi} = 1$, if output is 1. Say A has succeded or A has won.

The CPA game

CPA indistinguishability game $PubK^{cpa}_{A,\Pi}(n)$

1. $(pk, sk) \leftarrow Gen(1^n)$.
2. A is given pk and oracle access to $Enc_{pk}(\cdot)$.

   Outputs two plaintexts m_0, m_1 with $|m_0| = |m_1|$.
3. $b \leftarrow \{0,1\}, c \leftarrow Enc_k(m_b)$. c given to A.
4. A continues to have oracle access to $Enc_{pk}(\cdot)$.

   It outputs b'.
5. Output of experiment is 1, if $b = b'$, otherwise output is 0.

Write $PubK^{cpa}_{A,\Pi}(n) = 1$, if output is 1. Say A has succeded or A has won.

The indistinguishability game

Definition 7.2 $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ has indistinguishable encryptions under an eavesdropping attack if for every probabilistic polynomial time algorithm A there is a negligible function $\mu : \mathbb{N} \to \mathbb{R}^+$ such that

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

Definition 7.3 $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ has indistinguishable encryptions under a chosen plaintext attack if for every probabilistic polynomial time algorithm A there is a negligible function $\mu : \mathbb{N} \to \mathbb{R}^+$ such that

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

Eavesdropping, CPAs, multiple encryptions

Theorem 7.4 A public-key encryption scheme has indistinguishable encryptions under an eavesdropping attack if and only if it has indistinguishable encryptions under a chosen plaintext attack.

Theorem 7.5 A public-key encryption scheme has indistinguishable encryptions under an eavesdropping attack if and only if it has multiple indistinguishable encryptions under an eavesdropping attack.

Multiple messages

Multiple messages eavesdropping game $PubK^{mult}_{A,\Pi}(n)$

1. $(pk, sk) \leftarrow Gen(1^n)$
2. A is given pk and on input 1^n generates two vectors of messages $M_0 = (m_0^1, \ldots, m_0^t), M_1 = (m_1^1, \ldots, m_1^t)$ with $|m_0^i| = |m_1^i|$ for all i.
3. $b \leftarrow \{0,1\}, c_i \leftarrow Enc_{pk}(m_b^i)$.
   $C = (c_1, \ldots, c_t)$ is given to A.
4. $b' \leftarrow A(1^n, C)$.
5. Output of experiment is 1, if $b = b'$, otherwise output is 0.

From multiple messages to single message

A adversary against $\mathrm{PubK}^{\mathrm{mult}}_{A,\Pi}(\cdot)$

A' on input 1^n

1. A', given pk, runs A(pk) to obtain $M_0 = (m_0^1, \ldots, m_0^t)$ and $M_1 = (m_1^1, \ldots, m_1^t)$
2. A' chooses $i \leftarrow \{1, \ldots, t\}$ and outputs m_0^i, m_1^i. A' is given ciphertext c^i.
3. For $j < i$, A' computes $c^j := \mathrm{Enc}_{pk}(m_0^j)$.
   For $j > i$, A' computes $c^j := \mathrm{Enc}_{pk}(m_1^j)$.
4. A' runs $A(c^1, \ldots, c^t)$ and outputs the bit b' that A outputs.

Trapdoor permutations

Definition 7.6 A quadruple $\Pi = (\text{Gen}, \text{Samp}, f, \text{Inv})$ of ppts is called a family of trapdoor permutations, if

1. $\text{Gen}(1^n)$ outputs parameters (I, td) with $|I| \geq n$, where each pair (I, td) defines a finite set $D_I = D_{td}$.
2. By Gen_1 denote the algorithm obtained from Gen by restricting the output to I. Then $(\text{Gen}_1, \text{Samp}, f)$ is a family of one-way permutations.
3. Inv is deterministic and on input td, $y \in D_I$ outputs $x \in D_I$. We require that for all $(I, td) \leftarrow \text{Gen}(1^n)$ and all $x \in D_I$ $\text{Inv}_{td}(f_I(x)) = x$.

Function families

Definition 6.3 (restated) A triple $\Pi = (\mathrm{Gen}, \mathrm{Samp}, f)$ of ppts is called a family of functions, if

1. $\mathrm{Gen}(1^n)$ outputs parameters I with $|I| \geq n$, where each I defines finite sets D_I and R_I for a function $f_I : D_I \to R_I$ defined below.
2. $\mathrm{Samp}(I)$ outputs $x \leftarrow D_I$.
3. f is deterministic and on input I, $x \in D_I$ outputs $y \in R_I$, $y := f_I(x)$.

Π is a family of permutations, if in addition for all I $D_I = R_I$ and f_I is a bijection.

The inverting game

Inverting game $\mathrm{Invert}_{A,\Pi}(n)$

1. $I \leftarrow \mathrm{Gen}(1^n), x \leftarrow \mathrm{Samp}(I), y := f_I(x)$.
2. A given input 1^n, I and y, outputs x'.
3. Output of game is 1, if $f_I(x') = y$, otherwise output is 0.

Definition 6.4 (restated) A family of functions $\Pi = (\mathrm{gen}, \mathrm{Samp}, f)$ is called one-way, if for every probabilistic polynomial time algorithm A there is a negligible function $\mu : \mathbb{N} \rightarrow \mathbb{R}^+$ such that $\Pr[\mathrm{Invert}_{A,\Pi}(n) = 1] \leq \mu(n)$.

The RSA trapdoor permutation

$\mathrm{Gen}(1^n)$ computes 2 n-bit primes $p, q, p \neq q$, sets $N := p \cdot q$, $\varphi(N) := (p-1)(q-1)$. It computes $e, d \in \mathbb{Z}^*_{\varphi(N)}$ such that $e \cdot d = 1 \bmod \varphi(N)$. It outputs $I := (N, e)$, $td := (N, d)$. D_I is defined as $\mathbb{Z}_N$.

$\mathrm{Samp}(N, e)$ outputs $x \leftarrow \mathbb{Z}_N$.

$f_{(N,e)}(x)$ outputs $c := x^e \bmod N$.

$\mathrm{Inv}_{(N,d)}(c)$ outputs $x := c^d \bmod N$.

Hardcore predicates

Definition 7.7 Let $\Pi = (\text{Gen}, \text{Samp}, f, \text{Inv})$ be a family of trapdoor permutations. Let hc be a deterministic algorithm that, on input I and $x \in D_I$, outputs a single bit $hc_I(x)$.

Algorithm hc is a hardcore predicate for Π, if for every ppt A there is a negligible function μ such that

$$\Pr\left[A\left(I, f_I(x)\right) = hc_I(x)\right] \leq \frac{1}{2} + \mu(n),$$

where $(I, td) \leftarrow \text{Gen}(1^n), x \leftarrow D_I$.

The RSA trapdoor permutation

$\text{Gen}(1^n)$ computes 2 n-bit primes $p,q,p \neq q$, sets $N := p \cdot q$, $\varphi(N) := (p-1)(q-1)$. It computes $e,d \in \mathbb{Z}^*_{\varphi(N)}$ such that $e \cdot d = 1 \bmod \varphi(N)$. It outputs $I := (N,e)$, $td := (N,d)$. D_I is defined as $\mathbb{Z}_N$.

$\text{Samp}(N,e)$ outputs $x \leftarrow \mathbb{Z}_N$.

$f_{(N,e)}(x)$ outputs $c := x^e \bmod N$.

$\text{Inv}_{(N,d)}(c)$ outputs $x := c^d \bmod N$.

Fact The least significant bit is a hardcore predicate for the RSA trapdoor permutation.

From trapdoor permutations to encryption

Construction 7.8 Let $\mathrm{T} = \left(\mathrm{Gen}_{\mathrm{T}}, \mathrm{Samp}, f, \mathrm{Inv}\right)$ be a family of trapdoor permutations, and let hc be a hardcore predicate for T. Define the public-key encryption scheme $\Pi = \left(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec}\right)$ with message space $\{0,1\}$ as follows:

- **Gen:** on input 1^n, run $\mathrm{Gen}_{\mathrm{T}}$ to obtain (I, td). Output the public key I and the private key td.
- **Enc:** on input a public key I and message $m \in \{0,1\}$, choose $x \leftarrow D_I$ and output ciphertext $\left(f_I(x), hc_I(x) \oplus m\right)$.
- **Dec:** on input a private key td and a ciphertext $(y, s), y \in D_I$, compute $x := \mathrm{Inv}_{td}(y)$ and output $m := hc_I(x) \oplus s$.

From trapdoor permutations to encryption

Construction 7.8 Let $\mathrm{T} = (\mathrm{Gen}_{\mathrm{T}}, \mathrm{Samp}, f, \mathrm{Inv})$ be a family of trapdoor permutations, and let hc be a hardcore predicate for T. Define the public-key encryption scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ with message space $\{0,1\}$ as follows:

- **Gen:** on input 1^n, run $\mathrm{Gen}_{\mathrm{T}}$ to obtain (I, td). Output the public key I and the private key td.
- **Enc:** on input a public key I and message $m \in \{0,1\}$, choose $x \leftarrow D_I$ and output ciphertext $(f_I(x), hc_I(x) \oplus m)$.
- **Dec:** on input a private key td and a ciphertext $(y, s), y \in D_I$, compute $x := f_I^{-1}(y)$ and output $m := hc_I(x) \oplus s$.

Theorem 7.9 An encryption scheme as in Construction 6.8 has indistinguishable encryptions under a chosen plaintext attack.

From adversaries to predictors

A ppt adversary against Π from Construction 7.8.

A_{hc} on input $I, y \in D_I$

1. Set $pk = I$ and run $A(pk)$ to obtain $m_0, m_1 \in \{0,1\}$
2. Choose independent random bit z and b. Set $m' := m_b \oplus z$.
3. Give the ciphertext (y, m') to A and obtain an output bit b'.
4. If $b = b'$, output z; otherwise output $\bar{z}$.

Encrypting longer messages

$m = m_1 m_2 \dots m_k,\ m_i \in \{0,1\}$

First solution :

1. $x_i \leftarrow D_I, i = 1, \dots, k$
2. Output $\langle f_I(x_1), m_1 \oplus hc_I(x_1) \rangle, \dots, \langle f_I(x_k), m_k \oplus hc_I(x_k) \rangle$

Second solution :

1. $x_1 \leftarrow D_I, x_{i+1} = f(x_i), i = 1, \dots, k$
2. Output $\langle x_{k+1}, m_1 \oplus hc_I(x_1), \dots, m_k \oplus hc_I(x_k) \rangle$

Trapdoor permutations & hardcore predicates

Theorem 7.10 If a family of trapdoor permutations Π exists, then a family of trapdoor permutations $\hat{\Pi}$ together with a hardcore predicate hc exists.

Hybrid encryption – have your cake and eat it!

Private-key encryption

- very efficient,
- but needs shared secret key.
- key distribution, key agreement

Public-key encryption

- no shared keys,
- but less efficient than private-key encryption.
- used in combination with private-key encryption
- hybrid encryption

Hybrid encryption – have your cake and eat it!

$\Pi = (\text{Gen},\text{Enc},\text{Dec})$ public-key encryption scheme
$\Pi' = (\text{Gen}',\text{Enc}',\text{Dec}')$ private-key encryption scheme

$\Pi^{hy} = (\text{Gen}^{hy},\text{Enc}^{hy},\text{Dec}^{hy})$ defined by

Gen^{hy} on input 1^n run $\text{Gen}(1^n)$ to obtain (pk,sk)

Enc^{hy} on input a public key pk and a message $m \in \{0,1\}^*$ do

1. choose $k \leftarrow \text{Gen}'(1^n)$
2. compute $c_1 \leftarrow \text{Enc}_{pk}(k)$ and $c_2 \leftarrow \text{Enc}'_k(m)$.
3. output ciphertext $c = (c_1, c_2)$

Dec^{hy} on input private key sk and ciphertext $c = (c_1, c_2)$ do

1. compute $k := \text{Dec}_{sk}(c_1)$
2. output message $m := \text{Dec}'_k(c_2)$

Hybrid encryption – have your cake and eat it!

Theorem 7.11 If Π is a cpa-secure public-key encryption scheme and if Π' is a private key encryption scheme that has indistinguishable encryptions against eavesdropping adversaries, then Π^{hy} is a cpa-secure public-key encryption scheme.

Three adversaries – A_1

A^{hy} ppt adversary against public-key encryption scheme Π^{hy}.

A_1 on input $1^n, pk$

1. A_1 chooses $k \leftarrow \{0,1\}^n$ and obtains c_1, where $b \leftarrow \{0,1\}$ and $c_1 = Enc_{pk}(k)$ if $b = 0$, and $c_1 = Enc_{pk}(0^n)$ if $b = 1$
2. A_1 runs $A^{hy}(pk)$ to obtain two messages m_0, m_1
3. A_1 computes $c_2 \leftarrow Enc'_k(m_0)$, then runs $A^{hy}(c_1, c_2)$ and outputs the bit b' that A^{hy} outputs.

Three adversaries – A_2

A^{hy} ppt adversary against public-key encryption scheme Π^{hy}.

A_2 on input $1^n, pk$

1. A_2 chooses $k \leftarrow \{0,1\}^n$ and obtains c_1, where $b \leftarrow \{0,1\}$ and $c_1 = Enc_{pk}(0^n)$ if $b = 0$, and $c_1 = Enc_{pk}(k)$ if $b = 1$
2. A_2 runs $A^{hy}(pk)$ to obtain two messages m_0, m_1
3. A_2 computes $c_2 \leftarrow Enc'_k(m_1)$, then runs $A^{hy}(c_1, c_2)$ and outputs the bit b' that A^{hy} outputs.

Three adversaries – A’

A^{hy} ppt adversary against public-key encryption scheme Π^{hy}.

A' on input 1^n :

1. A' runs $Gen(1^n)$ to obtain a key pair (pk,sk).
2. A' runs $A^{hy}(pk)$ to obtain two messages m_0, m_1 and obtains $c_2 = Enc'_k(m_b)$, where $b \leftarrow \{0,1\}$.
3. A' computes $c_1 \leftarrow Enc_{pk}(0^n)$. Then A' runs $A^{hy}(c_1, c_2)$ and outputs the bit b' that A^{hy} outputs.