Public-Key Encryption Theory

Yu Zhang

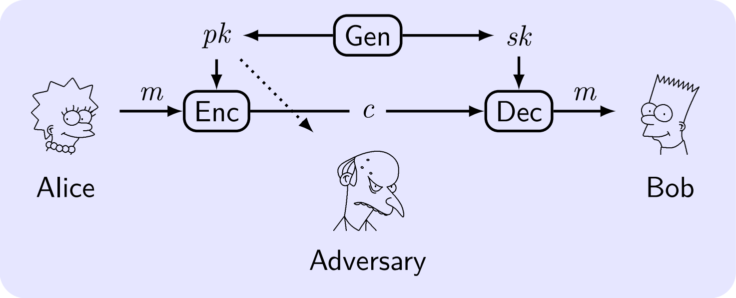
HIT/CST/NIS

Cryptography, Autumn, 2014

Outline

Content

Definitions



- **Key-generation** algorithm: $(pk, sk) \leftarrow$ Gen, key length $\geq n$.
- **Plaintext space** $\mathcal{M}$ is associated with pk.
- **Encryption** algorithm: $c \leftarrow \mathsf{Enc}_{pk}(m)$.
- **Decryption** algorithm: $m := \mathsf{Dec}_{sk}(c)$, or outputs $\perp$.
- **Requirement**: $\Pr[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m] \geq 1 - \mathsf{negl}(n)$.

Security against Eavesdroppers = CPA

The eavesdropping indistinguishability experiment $\mathsf{PubK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$:

1. $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ **is given input** $\mathbf{pk}$ **and so oracle access to** $\mathsf{Enc}_{\mathbf{pk}}(\cdot)$, outputs m_0, m_1 of the same length.
3. $b \leftarrow \{0,1\}$. $c \leftarrow \mathsf{Enc}_{pk}(m_b)$ (challenge) is given to $\mathcal{A}$.
4. $\mathcal{A}$ **continues to have access to** $\mathsf{Enc}_{\mathbf{pk}}(\cdot)$ and outputs b'.
5. If $b' = b$, $\mathcal{A}$ succeeded $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi} = 1$, otherwise 0.

Definition 1

Π is **CPA-secure** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that

$$\Pr\left[\mathsf{PubK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n).$$

Security Properties of Public-Key Encryption

Theorem 2

No deterministic public-key encryption scheme is secure in the presence of an eavesdropper.

Proposition 3

If Π is secure in the presence of an eavesdropper, then Π also is CPA-secure.

Theorem 4

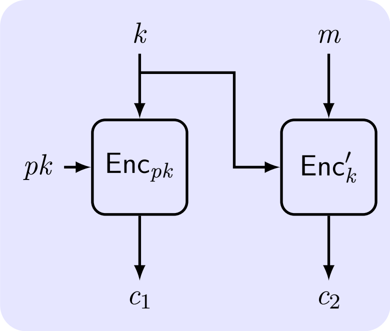
If Π is secure in the presence of an eavesdropper, then Π is secure for multiple encryptions.

Proposition 5

Perfectly-secret public-key encryption is impossible.

Construction of Hybrid Encryption

To speed up the encryption of long message, use private-key encryption Π' in tandem with public-key encryption Π.



Construction 6

$\Pi^{\mathsf{hy}} = (\mathsf{Gen}^{\mathsf{hy}}, \mathsf{Enc}^{\mathsf{hy}}, \mathsf{Dec}^{\mathsf{hy}})$:

- $\mathsf{Gen}^{\mathsf{hy}}$:
  $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.
- $\mathsf{Enc}^{\mathsf{hy}}$: pk and m.
  1. $k \leftarrow \{0,1\}^n$.
  2. $c_1 \leftarrow \mathsf{Enc}_{pk}(k)$,
     $c_2 \leftarrow \mathsf{Enc}'_k(m)$.
- $\mathsf{Dec}^{\mathsf{hy}}$: sk and $\langle c_1, c_2 \rangle$.
  1. $k := \mathsf{Dec}_{sk}(c_1)$.
  2. $m := \mathsf{Dec}'_k(c_2)$.

Hybrid encryption is a public-key encryption without any secret key in advance.

Theorem 7

If Π is a CPA-secure public-key encryption scheme and Π' is a private-key encryption scheme that has indistinguishable encryptions in the presence of an eavesdropper, then Π^{hy} is a CPA-secure public-key encryption scheme.

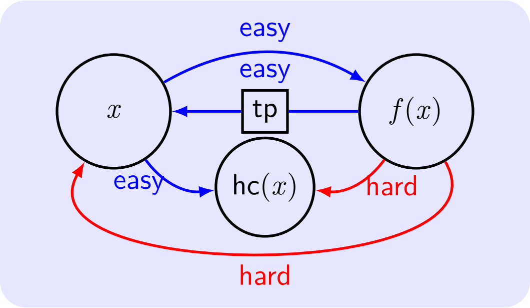
$$
\begin{tikzcd}
\langle pk, \mathsf{Enc}_{pk}(k), \mathsf{Enc}'_k(m_0)\rangle \arrow[r, leftrightarrow, dotted, "\text{(by transitivity)}"] \arrow[d, leftrightarrow, "\text{(by security of } \Pi\text{)}"] & \langle pk, \mathsf{Enc}_{pk}(k), \mathsf{Enc}'_k(m_1)\rangle \arrow[d, leftrightarrow, "\text{(by security of } \Pi\text{)}"'] \\
\langle pk, \mathsf{Enc}_{pk}(0^n), \mathsf{Enc}'_k(m_0)\rangle \arrow[r, leftrightarrow, "\text{(by security of } \Pi'\text{)}"'] & \langle pk, \mathsf{Enc}_{pk}(0^n), \mathsf{Enc}'_k(m_1)\rangle
\end{tikzcd}
$$

Content

Overview

Trapdoor function: is easy to compute, yet difficult to find its inverse without special info., the "trapdoor". (One Way Function with the "trapdoor")

A public-key encryption scheme can be constructed from any trapdoor permutation. ("*Theory and Applications of Trapdoor Functions*", [Yao, 1982])



Definition of Families of Trapdoor Permutations

A tuple of polynomial-time algorithms $\Pi = (\mathsf{Gen}, \mathsf{Samp}, f, \mathsf{Inv})$ is a **family of trapdoor permutations (TDP)** if:

- **parameter generation** algorithm Gen, on input 1^n, outputs (I, td) with $|I| \geq n$. (I, td) defines a set $\mathcal{D}_I = \mathcal{D}_{\mathsf{td}}$.
- Gen_I outputs only I. $(\mathsf{Gen}_I, \mathsf{Samp}, f)$ is OWP.
- deterministic **inverting algorithm** Inv. $\forall (I, \mathsf{td})$ and $\forall x \in \mathcal{D}_I$,

$$\mathsf{Inv}_{\mathsf{td}}(f_I(x)) = x.$$

Deterministic polynomial-time algorithm hc is a **hard-core predicate** of Π if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that

$$\Pr[\mathcal{A}(I, f_I(x)) = \mathsf{hc}_I(x)] \leq \frac{1}{2} + \mathsf{negl}(n).$$

Public-key Encryption Schemes from TDPs

Construction 8

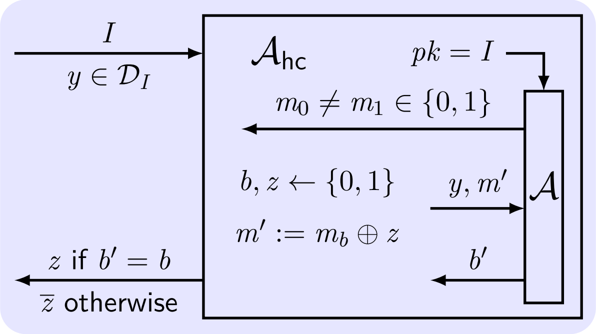
- Gen: $(I, \mathsf{td}) \leftarrow \widehat{Gen}$ *output* ***public key*** I *and* ***private key*** td.
- Enc: *on input* I *and* $m \in \{0,1\}$, *choose a random* $x \leftarrow \mathcal{D}_I$ *and output* $\langle f_I(x), \mathsf{hc}_I(x) \oplus m \rangle$.
- Dec: *on input* td *and* $\langle y, m' \rangle$, *compute* $x := f_I^{-1}(y)$ *and output* $\mathsf{hc}_I(x) \oplus m'$.

Theorem 9

If $\widehat{\Pi} = (\widehat{Gen}, f)$ *is TDP, and* hc *is HCP for* $\widehat{\Pi}$, *then Construction* Π *is CPA-secure.*

Proof

Idea: $\mathsf{hc}_I(x)$ is pseudorandom. Reduce $\mathcal{A}_{\mathsf{hc}}$ for hc to $\mathcal{A}$ for Π.



$$\Pr[\mathcal{A}_{\mathsf{hc}}(I, f_I(x)) = \mathsf{hc}_I(x)] =$$

$$\frac{1}{2} \cdot (\Pr[b' = b | z = \mathsf{hc}_I(x)] + \Pr[b' \neq b | z \neq \mathsf{hc}_I(x)]).$$

Proof (Cont.)

$$\Pr[b' = b | z = \mathsf{hc}_I(x)] = \Pr[\mathsf{PubK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1] = \varepsilon(n).$$

If $z \neq \mathsf{hc}_I(x)$, $m' = m_b \oplus \overline{\mathsf{hc}}_I(x) = m_{\overline{b}} \oplus \mathsf{hc}_I(x)$,

$$\Pr[b' = b | z \neq \mathsf{hc}_I(x)] = \Pr[\mathsf{PubK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 0] = 1 - \varepsilon(n).$$

$$\Pr[b' \neq b | z \neq \mathsf{hc}_I(x)] = \varepsilon(n).$$

$$\Pr[\mathcal{A}_{\mathsf{hc}}(I, f_I(x)) = \mathsf{hc}_I(x)] = \frac{1}{2} \cdot (\varepsilon(n) + \varepsilon(n)) = \varepsilon(n).$$

Encrypting Longer Messages

Theorem 10

If $\exists$ TDP Π, then $\exists$ TDP $\widehat{\Pi}$ with a HCP hc *for $\widehat{\Pi}$.*

Example: If RSA assumption holds then the least-significant bit is hard-core for the RSA family of TDP.

an ℓ-it message $m = m_1 \cdots m_\ell$, the public key I, the ciphertext is

$$\langle f_I(x_1), \mathsf{hc}_I(x_I) \oplus m_1 \rangle, \ldots, \langle f_I(x_\ell), \mathsf{hc}_I(x_\ell) \oplus m_\ell \rangle,$$

with $x_1, \ldots, x_\ell$ chosen independently and *u.r.a* from $\mathcal{D}_I$.

An alternative way: $x_1 \leftarrow \mathcal{D}_I$ and compute $x_{i+1} := f_I(x_i)$ for $i = 1$ to ℓ. the ciphertext is

$$\langle x_{\ell+1}, \mathsf{hc}_I(x_1) \oplus m_1, \ldots, \mathsf{hc}_I(x_\ell) \oplus m_\ell \rangle.$$

Content

Scenarios of CCA in Public-Key Setting

1. An adversary $\mathcal{A}$ observes the ciphertext c sent by $\mathcal{S}$ to $\mathcal{R}$.
2. $\mathcal{A}$ send c' to $\mathcal{R}$ in the name of $\mathcal{S}$ or its own.
3. $\mathcal{A}$ infer m from the decryption of c' to m'.

Scenarios

- **login to on-line bank with the password**: trial-and-error, learn info from the feedback of bank.
- **reply an e-mail with the quotation of decrypted text**.
- **malleability of ciphertexts**: e.g. doubling others' bids at an auction.

Definition of Security Against CCA/CCA2

The CCA/CCA2 indistinguishability experiment $\mathsf{PubK}^{\mathsf{cca}}_{\mathcal{A},\Pi}(n)$:

1. $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ **is given input** pk **and oracle access to** $\mathsf{Dec}_{sk}(\cdot)$, outputs m_0, m_1 of the same length.
3. $b \leftarrow \{0, 1\}$. $c \leftarrow \mathsf{Enc}_{pk}(m_b)$ is given to $\mathcal{A}$.
4. $\mathcal{A}$ **have access to** $\mathsf{Dec}_{sk}(\cdot)$ **except for** c **in CCA2**[1] and outputs b'.
5. If $b' = b$, $\mathcal{A}$ succeeded $\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi} = 1$, otherwise 0.

Definition 11

Π has **CCA/CCA2-secure** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that

$$\Pr\left[\mathsf{PubK}^{\mathsf{cca}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n).$$

[1] CCA is also called Lunchtime attacks; CCA2 is also called Adaptive CCA.

State of the Art on CCA2-secure Encryption

- **Zero-Knowledge Proof**: complex, and impractical. (e.g., Dolev-Dwork-Naor)
- **Random Oracle** model: efficient, but not realistic (to consider CRHF as RO). (e.g., RSA-OAEP and Fujisaki-Okamoto)
- **DDH(Decisional Diffie-Hellman assumption) and UOWHF(Universal One-Way Hashs Function)**: x2 expansion in size, but security proved w/o RO or ZKP (e.g., Cramer-Shoup system).

CCA2-secure implies Plaintext-aware: an adversary cannot produce a valid ciphertext without "knowing" the plaintext.

Open problem

Constructing a CCA2-secure scheme based on RSA problem as efficient as "Textbook RSA".

Content

Random Oracle Model (ROM) – Overview

- **Random oracle (RO)**: a truly random function H answers every possible query with a random response.
  - **Consistent**: If H ever outputs y for an input x "on-the-fly", then it always outputs the same answer given the same input.
  - No one "knows" the entire function H.
- **Random oracle model (ROM)**: the existence of a public RO.
- **Methodology**: for constructing proven security in ROM.
  1. a scheme is designed and proven secure in ROM.
  2. Instantiate H with a hash function $\hat{H}$, such as SHA-1.
- No one seriously claims that a random oracle exists.[2]

With ROM, it is easy to achieve proven security, while keeping the efficiency by appropriate instantiation.

[2] There exists schemes that are proven secure in ROM but are insecure no matter how the random oracle is instantiated.

Simple Illustrations of ROM

A RO maps n_1-bit inputs to n_2-bit outputs.

- A RO as a OWF, experiment:
  1. A random function H is chosen.
  2. A random $x \in \{0,1\}^{n_1}$ is chosen, and $y := H(x)$ is evaluated.
  3. $\mathcal{A}$ is given y, and succeeds if it outputs x': $H(x') = y$.
- A RO as a CRHF, experiment:
  1. A random function H is chosen.
  2. $\mathcal{A}$ succeeds if it outputs x, x' with $H(x) = H(x')$ but $x \neq x'$.
- Constructing a PRF from a RO: $n_1 = 2n$, $n_2 = n$.
  $F_k(x) \overset{\text{def}}{=} H(k \| x), \;\; |k| = |x| = n.$

Security Against CPA

Construction 12

- Gen: $pk = I$, $sk = \mathsf{td}$.
- Enc: $r \leftarrow \{0,1\}^*$, *output* $\langle f_I(r), H(r) \oplus m \rangle$.
- Dec: *input* (c_1, c_2); *compute* $r := f_{\mathsf{td}}^{-1}(c_1)$, *output* $H(r) \oplus c_2$.

Theorem 13

If f is TPD and H is RO, Construction is CPA-secure.

H can not be replaced by PRG, since the partial info on r may be leaked by c_1.

CCA-secure based on Private Key Encryption

Idea: PubK CCA = PrivK CCA + (Secret Key = TPD + RO).

Construction 14

$\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *is a private-key encryption scheme.*

- Gen: $pk = I$, $sk = \mathsf{td}$.
- Enc: $r \leftarrow D_I$ *and compute* $k := H(r)$, *output* $\langle f_I(r), \mathsf{Enc}'_k(m) \rangle$.
- Dec: *input* $\langle c_1, c_2 \rangle$, *compute* $r := f^{-1}_{\mathsf{td}}(c_1)$, $k := H(r)$, *output* $\mathsf{Dec}'_k(c_2)$.

Theorem 15

If f is TDP, Π' is CCA-secure, and H is RO, Construction is CCA-secure.

CCA-secure based on TPD in ROM

Idea: PubK CCA $=$ TDP $+$ 2 RO (one for enc, one for mac).

Construction 16

- Gen*:* $pk = I$*,* $sk = \mathsf{td}$*.*
- Enc*:* $r \leftarrow D_I$*, output* $\langle c_1 = f_I(r), c_2 = H(r) \oplus m, c_3 = G(c_2 \| m) \rangle$*.*
- Dec*:* $r := f_{\mathsf{td}}^{-1}(c_1)$*,* $m := H(r) \oplus c_2$*. If* $G(c_2 \| m) = c_3$ *output* m*, otherwise* $\perp$*.*

Theorem 17

If f is TDP, G, H are ROs, Construction is CCA-secure.

Private Key Encryption vs. Public Key Encryption

	Private Key	**Public Key**
Secret Key	both parties	receiver
Weakest Attack	Eav	CPA
Probabilistic	CPA/CCA	always
Assumption against CPA	OWF	TDP
Assumption against CCA	OWF	TDP+RO
Efficiency	fast	slow