Public-Key Encryption Theory

Yu Zhang

Harbin Institute of Technology

Cryptography, Autumn, 2020

Outline

1. Definitions and Securities of Public-Key Encryption

2. Trapdoor Permutations

3. Security Against Chosen-Ciphertext Attacks

4. Public-Key Encryption from TDP in ROM

Content

1. Definitions and Securities of Public-Key Encryption

2. Trapdoor Permutations

3. Security Against Chosen-Ciphertext Attacks

4. Public-Key Encryption from TDP in ROM

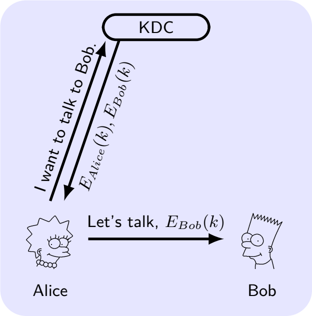
Limitations of Private-Key Cryptography

- The key-distribution need physically meeting.
- The number of keys for U users is $\Theta(U^2)$.
- Secure communication in open system:

Solutions that are based on private-key cryptography are not sufficient to deal with the problem of secure communication in open systems where parties cannot physically meet, or where parties have transient interactions.

Needham-Schroeder Protocol for Symmetric Key

- Key Distribution Center (KDC) as Trusted Third Party (TTP), which has the shared key with Alice, and with Bob, respectively.
- $E_{Bob}(k)$ is a **ticket** to access Bob, k is **session key**.
- Used in MIT's Kerberos protocol (in Windows).



Strength:

- each one stores one key
- no updates

Weakness:

- single-point-of-failure

Merkle Puzzles (Key Exchange W/O TTP)

Alice prepares 2^{32} puzzles Puzzle_i, and sends to Bob.

$$\text{Puzzle}_i \leftarrow \text{Enc}_{(0^{96}\|p_i)}(\text{“Puzzle \#”}x_i\|k_i),$$

where Enc is 128-bit, $p_i \leftarrow \{0,1\}^{32}$ and $x_i, k_i \leftarrow \{0,1\}^{128}$.

Bob chooses Puzzle_j randomly, guesses p_j in 2^{32} time, obtains x_j, k_j and sends x_j to Alice.

Alice lookups puzzle with x_j, and uses k_j as secret key.

- **Adversary** needs 2^{32+32} time.

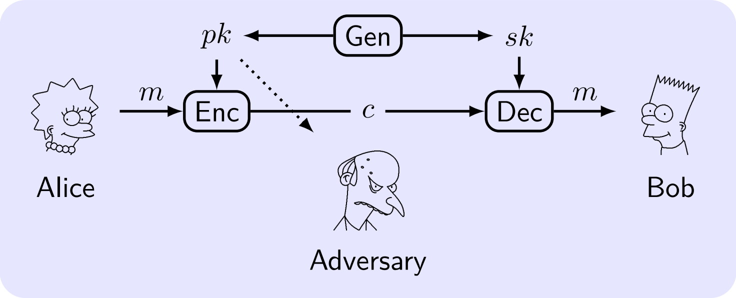
Better Gap?

Quadratic gap is best possible if we treat cipher as a black box oracle.

Public-Key Revolution

- In 1976, Whitfield Diffie and Martin Hellman published "*New Directions in Cryptography*".
- **Asymmetric** or **public-key** encryption schemes:
  - **Public key** as the encryption key.
  - **Private key** as the decryption key.
- **Public-key primitives**:
  - Public-key encryption.
  - Digital signatures. (non-repudiation)
  - Interactive key exchange.
- **Strength**:
  - Key distribution over public channels.
  - Reduce the need to store many keys.
  - Enable security in open system.
- **Weakness**: 2 or 3 orders of magnitude slower than private-key encryptions, active attack on public key distribution.

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

Symmetric ciphers are possible to encrypt a 32-bit message and obtain a 32-bit ciphertext (e.g. with the one time pad). Can the same be done with a public-key system?

Theorem 2

Q: Would a deterministic public-key encryption scheme be secure in the presence of an eavesdropper?

Proposition 3

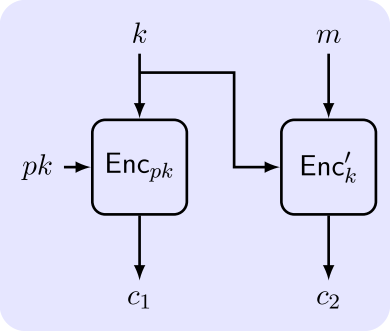
Q: If Π is secure in the presence of an eavesdropper, is Π also CPA-secure? and is it secure for multiple encryptions?

Proposition 4

Q: is perfectly-secret public-key encryption possible?

Construction of Hybrid Encryption

To speed up the encryption, use private-key encryption Π' (data-encapsulation mechanism, DEM) in tandem with public-key encryption Π (key-encapsulation mechanism, KEM).



Construction 5

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

- $\mathsf{Gen}^{\mathsf{hy}}$:
  $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.
- $\mathsf{Enc}^{\mathsf{hy}}$: 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. $m := \mathsf{Dec}'_k(c_2)$.

Q: is hybrid encryption a public-key enc. or private-key enc. ?

Security of Hybrid Encryption

Theorem 6

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

1 Definitions and Securities of Public-Key Encryption

2 **Trapdoor Permutations**

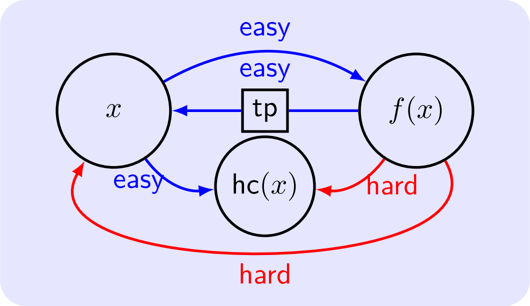
3 Security Against Chosen-Ciphertext Attacks

4 Public-Key Encryption from TDP in ROM

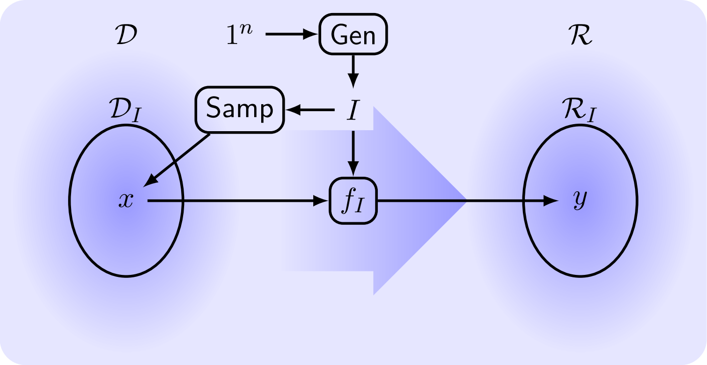
Overview

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])



Families of Functions



Definition 7

$\Pi = (\mathsf{Gen}, \mathsf{Samp}, f)$ is a **family of functions** if:

1. **Parameter-generation** algorithm: $I \leftarrow \mathsf{Gen}(1^n)$.
2. **sampling** algorithm: $x \leftarrow \mathsf{Samp}(I)$.
3. The deterministic **evaluation** algorithm: $y := f_I(x)$.

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

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

Examples

Let f with $< I, \mathrm{td} >$ be a TDP. Which of the following f' is also a TDP?

- $f'(x) = f(x) \| 000$
- $f'(x) = f(x) \| \mathsf{td}$
- $f'(x \| x') = f(x) \| \mathsf{Inv}_{\mathsf{td}}(f(x'))$
- $f'(x \| x') = f(x) \| f(x')$
- $f'(x) = f(x) \oplus I$
- $f'(x) = \begin{cases} f(x) & \text{if } x[0,1,2,3] \neq 1010 \\ x & \text{otherwise} \end{cases}$

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

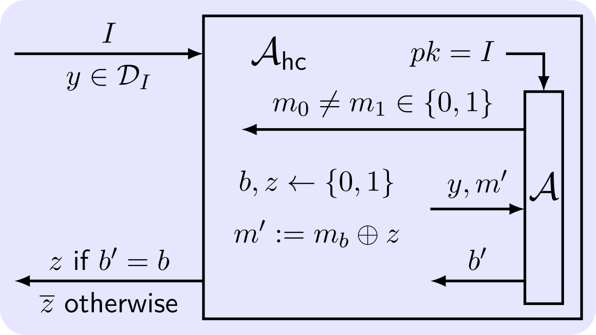
If $\widehat{\Pi} = (\widehat{Gen}, f)$ *is TDP, and* hc *is HCP for* $\widehat{\Pi}$*, then Construction* Π *is CPA-secure.*

Is the following scheme is secure?

$\mathsf{Enc}_I(m) = f_I(m)$, $\mathsf{Dec}_{\mathsf{td}}(c) = f_I^{-1}(c)$.

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 $z \neq \mathsf{hc}_I(x)$, $m' = m_b \oplus \overline{\mathsf{hc}}_I(x) = m_{\overline{b}} \oplus \mathsf{hc}_I(x)$,
which means $m_{\overline{b}}$ is encrypted.

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

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

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

Content

1 Definitions and Securities of Public-Key Encryption

2 Trapdoor Permutations

3 **Security Against Chosen-Ciphertext Attacks**

4 Public-Key Encryption from TDP in ROM

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 10

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.

Examples

Let (Gen, E, D) be CCA-secure on message space $\{0,1\}^{128}$. Which of the following is also CCA-secure?

- $E'(pk, m) = (E(pk, m), 0^{128})$
  $$D'(sk, (c_1, c_2)) = \begin{cases} D(sk, c_1) & \text{if } c_2 = 0^{128} \\ \perp & \text{otherwise} \end{cases}$$
- $E'(pk, m) = (E(pk, m), E(pk, 0^{128}))$
  $D'(sk, (c_1, c_2)) = D(sk, c_1)$

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

1 Definitions and Securities of Public-Key Encryption

2 Trapdoor Permutations

3 Security Against Chosen-Ciphertext Attacks

4 **Public-Key Encryption from TDP in ROM**

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

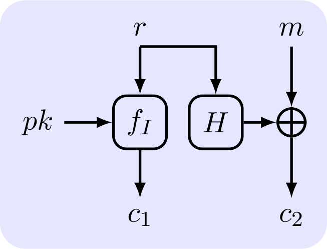
An RO maps n_1-bit inputs to n_2-bit outputs.

- An RO as an 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': $H(x') = y$
- An 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 an RO: $n_1 = 2n$, $n_2 = n$.

  $F_k(x) \overset{\text{def}}{=} H(k \| x),\ \ |k| = |x| = n.$

Security Against CPA

Idea: PubK CPA = PrivK + (Secret Key = TDP + RO)



Construction 11

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

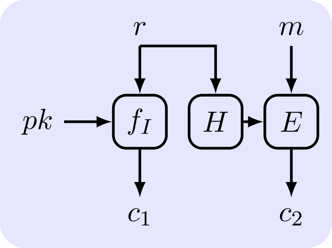
Theorem 12

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 13

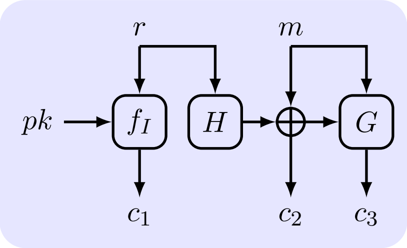
- Π' *is PrivK*
- Gen*:* $pk = I$, $sk =$ td*.*
- Enc*:* $k := H(r), r \leftarrow D_I$*, output* $\langle c_1 = f_I(r), c_2 = \text{Enc}'_k(m)\rangle$*.*
- Dec*:* $r := f_{\text{td}}^{-1}(c_1)$*,* $k := H(r)$*, output* $\text{Dec}'_k(c_2)$*.*

Theorem 14

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 15

- Gen*:* $pk = I$*,* $sk =$ 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*:* $r := f_{\text{td}}^{-1}(c_1)$*,* $m := H(r) \oplus c_2$*. If* $G(c_2 \| m) = c_3$ *output* m*, otherwise* $\perp$

Theorem 16

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

Key Size Comparison

NIST recommends the **key lengths** (in bits) with comparable security. NIST deems a 112-bit effective key length acceptable for security until the year 2030, but recommends 128-bit or higher key lengths for applications where security is required beyond then.

AES	RSA (N)/DH (p)	ECC (order q)
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512