Provable Security in the Computational Model

I - Basic Notions

David Pointcheval

Ecole normale supérieure, CNRS & INRIA

MPRI – Paris

Outline

1 Cryptography
   - Introduction
   - Kerckhoffs’ Principles
   - Formal Notations

2 Provable Security
   - Definition
   - Computational Assumptions
   - Some Reductions

3 Basic Security Notions
   - Public-Key Encryption
   - Variants of Indistinguishability
   - Signatures

4 Conclusion

Secrecy of Communications

One ever wanted to communicate secretly

With the all-digital world, security needs are even stronger
Old Methods

Substitutions and permutations

Security relies on the secrecy of the mechanism

Outline

1 Cryptography
   - Introduction
   - Kerckhoffs’ Principles
   - Formal Notations

2 Provable Security

3 Basic Security Notions

4 Conclusion

Kerckhoffs’ Principles (1)

La Cryptographie Militaire (1883)

Le système doit être matériellement, sinon mathématiquement, indéchiffrable

The system should be, if not theoretically unbreakable, unbreakable in practice

→ If the security cannot be formally proven, heuristics should provide some confidence.

Kerckhoffs’ Principles (2)

La Cryptographie Militaire (1883)

Il faut qu’il n’exige pas le secret, et qu’il puisse sans inconvénient tomber entre les mains de l’ennemi

Compromise of the system should not inconvenience the correspondents

→ The description of the mechanism should be public
La Cryptographie Militaire (1883)

La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants

The key should be rememberable without notes and should be easily changeable

→ The parameters specific to the users (the key) should be short

Use of (Secret) Key

A shared information (secret key) between the sender and the receiver parameterizes the mechanism:

- Vigenère: each key letter tells the shift
- Enigma: connectors and rotors

Security looks better: but broken (Alan Turing et al.)

Symmetric Encryption

Principles 2 and 3 define the concepts of symmetric cryptography:

\[
\begin{align*}
E_k(m) &\rightarrow c \\
D_k(c) &\rightarrow m
\end{align*}
\]

Secrecy

It is impossible/hard to recover \( m \) from \( c \) only (without \( k \))

Security

It is heuristic only: 1st principle

Perfect Secrecy?

Any security indeed vanished with statistical attacks!

Perfect secrecy? Is it possible?

Perfect Secrecy

The ciphertext does not reveal any (additional) information about the plaintext: no more than known before

- a priori information about the plaintext, defined by the distribution probability of the plaintext
- a posteriori information about the plaintext, defined by the distribution probability of the plaintext, given the ciphertext

Both distributions should be perfectly identical
One-Time Pad Encryption

Vernam’s Cipher (1929)

- Encryption of \( m \in \{0, 1\}^n \) under the key \( k \in \{0, 1\}^n \):
  \[
  m = \begin{array}{cccccccc}
  1 & 0 & 0 & 1 & 0 & 1 & 1 \\
  \end{array}, \quad \text{plaintext}
  \]
  \[
  k = \begin{array}{cccccccc}
  1 & 1 & 0 & 1 & 0 & 0 & 0 \\
  \end{array}, \quad \text{key = random mask}
  \]
  \[
  c = \begin{array}{cccccccc}
  0 & 1 & 0 & 0 & 0 & 1 & 1 \\
  \end{array}, \quad \text{ciphertext}
  \]

- Decryption of \( c \in \{0, 1\}^n \) under the key \( k \in \{0, 1\}^n \):
  \[
  c \oplus k = (m \oplus k) \oplus k = m \oplus (k \oplus k) = m
  \]

Which message is encrypted in the ciphertext \( c \in \{0, 1\}^n \)?
For any candidate \( m \in \{0, 1\}^n \), the key \( k = c \oplus m \) would lead to \( c \)
⇒ no information about \( m \) is leaked with \( c \)!

Information Theory

Drawbacks

- The key must be as long as the plaintext
- This key must be used once only (one-time pad)

Theorem (Shannon – 1949)

To achieve perfect secrecy, A and B have to share a common string truly random and as long as the whole communication.

Thus, the above one-time pad technique is optimal.

Practical Secrecy

Perfect Secrecy vs. Practical Secrecy

- No information about the plaintext \( m \) is in the ciphertext \( c \) without the knowledge of the key \( k \)
  ⇒ information theory
  No information about the plaintext \( m \) can be extracted from the ciphertext \( c \), even for a powerful adversary (unlimited time and/or unlimited power): perfect secrecy
- In practice: adversaries are limited in time/power
  ⇒ complexity theory

Shannon also showed that combining appropriately permutations and substitutions can hide information: extracting information from the ciphertext is time consuming

Modern Symmetric Encryption: DES and AES

Combination of substitutions and permutations

DES (1977)
Data Encryption Standard

AES (2001)
Advanced Encryption Standard
**Outline**

1. **Cryptography**
   - Introduction
   - Kerckhoffs' Principles
   - Formal Notations

2. Provable Security

3. Basic Security Notions

4. Conclusion

---

**Symmetric Encryption: Formalism**

**Symmetric Encryption - Secret Key Encryption**

One secret key only shared by Alice and Bob: this is a common parameter for the encryption and the decryption algorithms. This secret key has a symmetric capability.

![Diagram: Symmetric Encryption]

The secrecy of the key $k$ guarantees the secrecy of communications but requires such a common secret key!

How can we establish such a common secret key? Or, how to avoid it?

---

**Asymmetric Encryption: Intuition**

[Diffie-Hellman 1976]

Secrecy

- The recipient only should be able to open the message
- No requirement about the sender

Why would the sender need a secret key to encrypt a message?

---

**Asymmetric Encryption: Formalism**

**Public Key Cryptography - Diffie-Hellman (1976)**

- Bob's public key is used by Alice as a parameter to encrypt a message to Bob
- Bob's private key is used by Bob as a parameter to decrypt ciphertexts

Asymmetric cryptography extends the 2nd principle:

![Diagram: Asymmetric Encryption]

The secrecy of the private key $sk$ guarantees the secrecy of communications.
What is a Secure Cryptographic Scheme/Protocol?

Symmetric encryption:
The secrecy of the key $k$ guarantees the secrecy of communications

Asymmetric encryption:
The secrecy of the private key $sk$ guarantees the secrecy of communications

What does mean secrecy?
→ Security notions have to be formally defined

How to guarantee above security claims for concrete schemes?
→ Provable security

Provable Security

One can prove that:
- if an adversary is able to break the cryptographic scheme
- then one can break a well-known hard problem

General Method

Computational Security Proofs

In order to prove the security of a cryptographic scheme/protocol, one needs
- a formal security model (security notions)
- acceptable computational assumptions (hard problems)
- a reduction: if one can break the security notions, then one can break the hard problem
Integer Factoring

- Given \( n = pq \)
- Find \( p \) and \( q \)

<table>
<thead>
<tr>
<th>Year</th>
<th>Required Complexity</th>
<th>( n ) bitlength</th>
</tr>
</thead>
<tbody>
<tr>
<td>before 2000</td>
<td>64</td>
<td>768</td>
</tr>
<tr>
<td>before 2010</td>
<td>80</td>
<td>1024</td>
</tr>
<tr>
<td>before 2020</td>
<td>112</td>
<td>2048</td>
</tr>
<tr>
<td>before 2030</td>
<td>128</td>
<td>3072</td>
</tr>
<tr>
<td></td>
<td>192</td>
<td>7680</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>15360</td>
</tr>
</tbody>
</table>

Note that the reduction may be lossy: extra bits are then required

Integer Factoring Records

- Given \( n = pq \)
- Find \( p \) and \( q \)

<table>
<thead>
<tr>
<th>Digits</th>
<th>Date</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>129</td>
<td>April 1994</td>
<td>Quadratic Sieve</td>
</tr>
<tr>
<td>130</td>
<td>April 1996</td>
<td>Algebraic Sieve</td>
</tr>
<tr>
<td>140</td>
<td>February 1999</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>August 1999</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>April 2003</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>May 2005</td>
<td></td>
</tr>
<tr>
<td>232</td>
<td>December 2009</td>
<td></td>
</tr>
</tbody>
</table>

Note that this problem is hard without the prime factors \( p \) and \( q \), but becomes easy with them: if \( d = e^{-1} \mod \varphi(n) \), then \( x = y^d \mod n \)

RSA

- Given \( n = pq \), \( e \) and \( y \in \mathbb{Z}_n^* \)
- Find \( x \) such that \( y = x^e \mod n \)

Flexible RSA

- Given \( n = pq \) and \( y \in \mathbb{Z}_n^* \)
- Find \( x \) and \( e > 1 \) such that \( y = x^e \mod n \)

Both problems are assumed as hard as integer factoring: the prime factors are a trapdoor to find solutions
Discrete Logarithm Problem

- Given $G = \langle g \rangle$ a cyclic group of order $q$, and $y \in G$
- Find $x$ such that $y = g^x$

Possible groups: $G \in (\mathbb{Z}_p, \times)$, or an elliptic curve

(Computational) Diffie Hellman Problem

- Given $G = \langle g \rangle$ a cyclic group of order $q$, and $X = g^x$, $Y = g^y$
- Find $Z = g^{xy}$

The knowledge of $x$ or $y$ helps to solve this problem (trapdoor)

Success Probabilities

For any computational problem $P$, we quantify the quality of an adversary $A$ by its success probability in finding the solution:

$$\text{Succ}^P(A) = \Pr[A(\text{instance}) \rightarrow \text{solution}].$$

We quantify the hardness of the problem by the success probability of the best adversary within time $t$:

$$\text{Succ}(t) = \max_{|A| \leq t} \{\text{Succ}(A)\}.$$

Note that the probability space can be restricted:
- some inputs are fixed, and others only are randomly chosen.

Discrete Logarithm Problem

We usually fix the group $G = \langle g \rangle$ of order $q$, $X$ is randomly chosen:

$$\text{Succ}_{dlp}^G(A) = \Pr_{x \leftarrow \mathbb{Z}_q}[A(g^x) \rightarrow x].$$

Decisional Problem

(Decisional) Diffie Hellman Problem

- Given $G = \langle g \rangle$ a cyclic group of order $q$, and $X = g^x$, $Y = g^y$, as well as a candidate $Z \in G$
- Decide whether $Z = g^{xy}$

In such a case, the adversary is called a distinguisher (outputs 1 bit).

A good distinguisher should behave in significantly different manners according to the input distribution:

$$\text{Adv}_{\text{ddh}}^G(A) = \Pr[A(X, Y, Z) = 1|Z = g^{xy}] - \Pr[A(X, Y, Z) = 1|Z \not\in G].$$

Outline

1. Cryptography
2. Provable Security
   - Definition
   - Computational Assumptions
   - Some Reductions
3. Basic Security Notions
4. Conclusion
Distribution Indistinguishability

Indistinguishabilities

Let $D_0$ and $D_1$, two distributions on a finite set $X$:

- $D_0$ and $D_1$ are perfectly indistinguishable if
  \[
  \text{Dist}(D_0, D_1) = \sum_{x \in X} \Pr[a = x] - \Pr[a = x] = 0
  \]

- $D_0$ and $D_1$ are statistically indistinguishable if
  \[
  \text{Dist}(D_0, D_1) = \sum_{x \in X} \Pr[a = x] - \Pr[a = x] = \text{negl()}
  \]

Computational Indistinguishability

Let $D_0$ and $D_1$, two distributions on a finite set $X$,

- a distinguisher $A$ between $D_0$ and $D_1$ is characterized by its advantage
  \[
  \text{Adv}^{D_0, D_1}(A) = \Pr[A(a) = 1] - \Pr[A(a) = 1]
  \]

- the computational indistinguishability of $D_0$ and $D_1$ is measured by
  \[
  \text{Adv}^{D_0, D_1}(t) = \max_{|A| \leq t} \{\text{Adv}^{D_0, D_1}(A)\}.
  \]
Computational Indistinguishability

\[
\text{Adv}^{D_0:D_1}(A) = \Pr_{a \in D_1}[A(a) = 1] - \Pr_{a \in D_0}[A(a) = 1] \\
= \Pr[a \in D_1 : A(a) = 1] - \Pr[a \in D_0 : A(a) = 1] \\
= 2 \times \Pr[a \in D_b : A(a) = b \land b = 1] \\
+ 2 \times \Pr[a \in D_b : A(a) = b \land b = 0] - 1 \\
= 2 \times \Pr[a \in D_b : A(a) = b] - 1
\]

Equivalent Notation

Let \( D_0 \) and \( D_1 \), two distributions on a finite set \( X \),
\[
\text{Adv}^{D_0:D_1}(A) = 2 \times \Pr[a \in D_b : A(a) = b] - 1
\]

Hybrid Technique

Let us consider the distributions \( D_A \) and \( D_B \):
\[
D_A = (g^x, g^{y_1}, g^{xy_1, \ldots}, g^{y_n}, g^{xy_n}) \subseteq \mathbb{G}^{2n+1} \\
D_B = (g^x, g^{y_1}, g^{z_1, \ldots}, g^{y_n}, g^{z_n}) \subseteq \mathbb{G}^{2n+1}
\]
We define the hybrid distribution
\[
D_i = (g^x, g^{y_1}, g^{xy_1, \ldots}, g^{y_i}, g^{xy_i}, g^{z_{i+1}, \ldots}, g^{y_n}, g^{z_n})
\]
\[
D_0 = D_B \quad D_n = D_A.
\]

Relations between Indistinguishability Notions

For any distinguisher \( A \), we have
\[
\text{Adv}^{D_0:D_1}(A) = \Pr_{a \in D_0}[A(a) = 1] - \Pr_{a \in D_1}[A(a) = 1] \\
\leq \Pr_{x \in X}[a = x] - \Pr_{a \in D_0}[a = x] \\
\leq \text{Dist}(D_0, D_1)
\]

Theorem

\( \text{Dist}(D_0, D_1) \) is the best advantage any adversary could get, even within an unbounded time.
\[
\forall t, \quad \text{Adv}^{D_0:D_1}(t) \leq \text{Dist}(D_0, D_1).
\]

Hybrid Technique

Let \( A \) be an adversary within time \( t \), against \( D_A \) vs. \( D_B \).
Given a DDH input \( (X, Y, Z) \), we generate the hybrid instance:
\[
I_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})
\]
Note that
- if \( Z = g^{y_i} \), then \( I \in D_i \) \quad \text{Adv}^{D_i:D_{i-1}}(A) \leq \text{Adv}^{\text{ddh}}(t')
- if \( Z \not\supseteq \mathbb{G} \), then \( I \in D_{i-1} \) \quad \text{where} \ t' \leq t + 2(n-1)\tau_{\text{exp}}
\[
\text{Adv}^{D_A:D_B}(A) = \text{Adv}^{D_B:D_0}(A) \\
\leq \sum_{i=1}^{n} \text{Adv}^{\text{ddh}}(t')
\]

Theorem

\[
\forall t, \quad \text{Adv}^{D_A:D_B}(t) \leq n \times \text{Adv}^{\text{ddh}}(t + 2(n-1)\tau_{\text{exp}})
\]
**Public-Key Encryption**

**Outline**

1. Cryptography
2. Provable Security
3. Basic Security Notions
   - Public-Key Encryption
   - Variants of Indistinguishability
   - Signatures
4. Conclusion

---

**Public-Key Encryption**

**Goal: Privacy/Secrecy of the plaintext**

---

**OW − CPA Security Game**

**One-Wayness**

For a public-key encryption scheme $S = (K, E, D)$, without the secret key $sk$, it should be computationally impossible to recover the plaintext $m$ from the ciphertext $c$:

$$\text{Succ}^w_S(A) = \Pr[(sk, pk) \leftarrow K(); m \xleftarrow{R} M; c = E_{pk}(m) : A(pk, c) \rightarrow m]$$

should be negligible.

**Chosen-Plaintext Attacks**

In the public-key setting, the adversary has access to the encryption key (the public key), and thus can encrypt any plaintext of its choice: chosen-plaintext attack
ElGamal Encryption

The ElGamal encryption scheme \( \mathcal{E}G \) is defined, in a group \( G = \langle g \rangle \) of order \( q \):

\[
\begin{align*}
K(G, g, q): & \quad x \overset{\$}{\leftarrow} \mathbb{Z}_q, \text{ and } sk \leftarrow x \text{ and } pk \leftarrow y = g^x \\
E_{pk}(m): & \quad r \overset{\$}{\leftarrow} \mathbb{Z}_q, \quad c_1 \leftarrow g^r \text{ and } c_2 \leftarrow y^r \times m =.pk^r \times m. \\
D_{sk}(c): & \quad \text{outputs } c_2 / c_1^{sk}
\end{align*}
\]

**Theorem (ElGamal is OW−CPA)**

\[
\text{Succ}_{\mathcal{E}G}^{\text{OW−cpa}}(t) \leq \text{Succ}_{G}^{\text{cdh}}(t)
\]

ElGamal is OW−CPA: Proof

Let \( A \) be an adversary against \( \mathcal{E}G \), we build an adversary \( B \) against CDH: let us be given a CDH instance \((X, Y)\)

\[
\begin{align*}
A & \text{ gets } pk \leftarrow X \text{ from } B \\
B & \text{ sets } c_1 \leftarrow Y \\
B & \text{ chooses } \mathbf{m}^* \overset{\$}{\leftarrow} M \text{ and } c_2 \overset{\$}{\leftarrow} G, \text{ and sends } c = (c_1, c_2) \\
B & \text{ receives } \mathbf{m} \text{ from } A \text{ and outputs } c_2 / \mathbf{m} \\
\Pr[\mathbf{m} = \mathbf{m}^*] &= \text{Succ}_{\mathcal{E}G}^{\text{OW−cpa}}(A) \\
&= \Pr[\mathbf{c}_2 / \mathbf{m} = \mathbf{c}_2 / \mathbf{m}^*] = \Pr[\mathbf{c}_2 / \mathbf{m} = \text{CDH}(X, Y)] \leq \text{Succ}_{G}^{\text{cdh}}(t)
\end{align*}
\]

Is OW−CPA Enough?

For a yes/no answer or sell/buy order, one bit of information may be enough for the adversary!

How to model that no bit of information leaks?

**Semantic Security** [Goldwasser-Micali 1984]

For any predicate \( f \), \( \mathcal{E}(m) \) does not help to guess \( f(m) \), with better probability than \( f(m') \) (for a random but private \( m' \)):

\[
\begin{align*}
& \begin{align*}
& \begin{align*}
& (sk, pk) \leftarrow \mathcal{K}(); (\mathcal{M}, f, \text{state}) \leftarrow \mathcal{A}(pk); \\
& m, m' \overset{\$}{\leftarrow} \mathcal{M}; c = \mathcal{E}_{pk}(m); p \leftarrow \mathcal{A}(\text{state}, c)
\end{align*}
\end{align*}
\end{align*}
\]

then,

\[
\text{Adv}_{S}^{\text{sem}}(\mathcal{A}) = \Pr[p = f(m)] - \Pr[p = f(m')].
\]
Indistinguishability implies Semantic Security

Let \( A \) be an adversary within time \( t \) against semantic security, we build an adversary \( B \) against indistinguishability:

- \( B \) runs \( A \) to get \( D \) and a predicate \( P \);
- it gets \( m_0, m_1 \leftarrow D \), and outputs them;
- the challenger encrypts \( m_b \) in \( c \);
- \( B \) runs \( A \), to get the guess \( p \) of \( A \) about the predicate \( P \) on the plaintext in \( c \);
  - if \( P(m_b) = P(m_1) \), \( B \) outputs a random bit \( b^0 \);
  - otherwise it outputs \( b^0 \) such that \( P(m_b) = p \).

Note that (if diff denotes the event that \( P(m) \neq P(m') \))

\[
\text{Adv}_{\text{sem}}(A) = \Pr[p = P(m) | c = E(m)] - \Pr[p = P(m') | c = E(m)]
\]

\[
= \Pr[p = P(m) | c = E(m) \land \text{diff}] - \Pr[p = P(m') | c = E(m) \land \text{diff}] \times \Pr[\text{diff}]
\]

If \( \text{diff} \) denotes the event that \( P(m_0) \neq P(m_1) \)

\[
\text{Adv}_{\text{ind}}(B) = \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]
\]

\[
= \Pr[b' = 1 | b = 1 \land \text{diff}] - \Pr[b' = 1 | b = 0 \land \text{diff}] \times \Pr[\text{diff}]
\]

\[
= \Pr[P(m_1) = p | c = E(m_1) \land \text{diff}] - \Pr[P(m_1) = p | c = E(m_0) \land \text{diff}] \times \Pr[\text{diff}]
\]

\[
= \Pr[P(m_1) = p | c = E(m_1) \land \text{diff}] - \Pr[P(m_0) = p | c = E(m_1) \land \text{diff}] \times \Pr[\text{diff}]
\]

\[
= \text{Adv}_{\text{sem}}(A) \leq \text{Adv}_{\text{ind}}(t')
\]

The running time \( t' \) of \( B = \) one execution of \( A \) (time \( t \)), two sampling from \( D \) (time \( \tau_D \)), two evaluations of the predicate \( P \) (time \( \tau_P \))

\[
\text{Adv}_{\text{sem}}(t) \leq \text{Adv}_{\text{ind}}(t + 2\tau_D + 2\tau_P)
\]
Semantic Security implies Indistinguishability

Let $A$ be an adversary within time $t$ against indistinguishability, we build an adversary $B$ against semantic security:

- $B$ runs $A$ to get $(m_0, m_1)$;
  - it sets $D = \{m_0, m_1\}$, and $P(m) = (m \in D)$;
- the challenger chooses $m, m' \xleftarrow{\$} D$, and encrypts $m$ in $c$
- $B$ runs $A$, to get $b'$, that it forwards as its guess $p$

$$
\text{Adv}_{\text{sem}}(B) = \Pr[p = P(m)] - \Pr[p = P(m')]
= \Pr[m = m_b] - \Pr[m' = m_b]
= \Pr[m = m_b'] - \Pr[m' = m_b']
\text{Adv}_{\text{ind}}(A) = \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]
$$

where $m = m_b$

ElGamal Encryption

Let $A$ be an adversary against $\mathcal{E}_G$, we build an adversary $B$ against $\text{DDH}$: let us be given a $\text{DDH}$ instance $(X, Y, Z)$

- $A$ gets $pk \leftarrow X$ from $B$, and outputs $(m_0, m_1)$
- $B$ sets $c_1 \leftarrow Y$
- $B$ chooses $b \xleftarrow{\$} \{0, 1\}$, sets $c_2 \leftarrow Z \times m_b$, and sends $c = (c_1, c_2)$
- $B$ receives $b'$ from $A$ and outputs $d = (b' = b)$
- $2 \times \Pr[b' = b] - 1
\leq \text{Adv}_{\text{ind-cpa}}(A)$, if $Z = \text{CDH}(X, Y)$
\leq 0, otherwise
ElGamal is IND−CPA: Proof

As a consequence,

- $2 \times \Pr[b' = b | Z = \text{CDH}(X, Y)] - 1 = \text{Adv}_{\mathcal{EG}}^{\text{ind-CPA}}(A)$
- $2 \times \Pr[b' = b | Z \leftarrow \mathcal{G}] - 1 = 0$

$$\text{Adv}_{\mathcal{EG}}^{\text{ind-CPA}}(A) = 2 \times \Pr[d = 1 | Z = \text{CDH}(X, Y)] - \Pr[d = 1 | Z \leftarrow \mathcal{G}]$$

$$= 2 \times \text{Adv}_{\mathcal{G}}^{\text{ddh}}(B) \leq 2 \times \text{Adv}_{\mathcal{G}}^{\text{ddh}}(t)$$

### RSA Encryption

The RSA encryption scheme $\mathcal{RSA}$ is defined by

- $\mathcal{K}(1^k)$: $p$ and $q$ two random $k$-bit prime integers, and an exponent $e$ (possibly fixed, or not):
  - $sk \leftarrow d = e^{-1} \mod \varphi(n)$ and $pk \leftarrow (n, e)$
- $\mathcal{E}_{pk}(m)$: the ciphertext is $c = m^e \mod n$
- $\mathcal{D}_{sk}(c)$: the plaintext is $m = c^d \mod n$

#### Theorem ($\mathcal{RSA}$ is OW−CPA, but . . .)

$$\text{Succ}_{\mathcal{RSA}}^{\text{OW−CPA}}(t) \leq \text{Succ}_{\mathcal{RSA}}(t)$$

*A deterministic encryption scheme cannot be IND − CPA*

---

### Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG − CPA

- The challenger flips a bit $b$
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}(.)$
- The adversary receives the public key $pk$, and chooses 2 messages $m_0$ and $m_1$
- The challenger returns the encryption $c$ of $m_b$ under $pk$
- The adversary outputs its guess $b'$ on the bit $b$

$$\text{Adv}_{\mathcal{S}}^{\text{ftg−CPA}}(A) = \text{Adv}_{\mathcal{S}}^{\text{ind−CPA}}(A) = 2 \times \Pr[b' = b] - 1$$

Note: the adversary has access to the following oracle, only once:

$LR_b(m_0, m_1)$: outputs the encryption of $m_b$ under $pk$
### Left-or-Right Indistinguishability

**LoR − CPA**

- The challenger flips a bit $b$
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow K()$
- The adversary receives the public key $pk$, and asks LR on any pair $(m_0, m_1)$ of its choice
- The challenger answers using LR$_b$
- The adversary outputs its guess $b'$ on the bit $b$

$Adv^{lor-CPA}_S (A) = 2 \times Pr[b' = b] - 1$

### Find-then-Guess vs. Left-or-Right

**Theorem (FtG $\sim$ LoR)**

$\forall t$, $Adv^{ftg-CPA}_S (t) \leq Adv^{lor-CPA}_S (t)$

$\forall t$, $Adv^{lor-CPA}_S (t) \leq n \times Adv^{ftg-CPA}_S (t)$

*where $n$ is the number of LR queries*

**LoR $\Rightarrow$ FtG is clear**

**FtG $\Rightarrow$ LoR**:

Hybrid distribution of the sequence of bits $b$

- The Left distribution is $(0, 0, \ldots, 0) \in \{0, 1\}^n$, for the LR queries
- The Right distribution is $(1, 1, \ldots, 1) \in \{0, 1\}^n$, for the LR queries
- Hybrid distribution: $D_i = (0, \ldots, 0, 1, \ldots, 1) = 0^n1^{n-i} \in \{0, 1\}^n$

$Dist(D_0, D_n) = Adv^{lor-CPA}_S (A)$

$Dist(D_i, D_{i+1}) \leq Adv^{ftg-CPA}_S (t)$

### Real-or-Random Indistinguishability

**Left-or-Right vs. Real-Random**

**Theorem (LoR $\sim$ RoR)**

$\forall t$, $Adv^{ror-CPA}_S (t) \leq Adv^{lor-CPA}_S (t)$

$\forall t$, $Adv^{lor-CPA}_S (t) \leq 2 \times Adv^{ror-CPA}_S (t)$

**LoR $\Rightarrow$ RoR is clear** (using $m_0 = m$ and $m_1 \overset{R}{\leftarrow} M$)

**RoR $\Rightarrow$ LoR**:

Flip a bit $d$, and use $m_d$ for the RR oracle

$Pr[d \leftarrow A|Real] = Pr[d \leftarrow A] = Pr[d \leftarrow A|Random] = 1/2$

$Adv^{lor}_S (A) = 2 \times Pr[A \leftarrow d] - 1$

$= 2 \times Pr[d \leftarrow A|Real] - 2 \times Pr[d \leftarrow A|Random]$

$\leq 2 \times Adv^{ror}_S (t)$
### Signature

**Existential Unforgeability**

For a signature scheme $\mathcal{S}_G = (K, S, V)$, without the secret key $sk$, it should be computationally impossible to generate a valid pair $(m, \sigma)$:

$$\text{Succ}_{\mathcal{S}_G}(\mathcal{A}) = \Pr[(sk, pk) \leftarrow K(); (m, \sigma) \leftarrow \mathcal{A}(pk) : V_{pk}(m, \sigma) = 1]$$

should be negligible.

**No-Message Attacks**

In the public-key setting, the adversary has access to the verification key (the public key), but not necessarily to valid signatures:

no-message attack
The RSA signature scheme $\mathcal{RSA}$ is defined by

$K(1^k)$: $p$ and $q$ two random $k$-bit prime integers, and an exponent $\nu$ (possibly fixed, or not):

$sk \leftarrow s = \nu^{-1} \mod \varphi(n)$ and $pk \leftarrow (n, \nu)$

$S_{sk}(m)$: the signature is $\sigma = m^s \mod n$

$V_{pk}(m, \sigma)$ checks whether $m = \sigma^\nu \mod n$

**Theorem (RSA is not EUF − NMA)**

The plain RSA signature is not secure at all!