I – Basic Notions

David Pointcheval
MPRI – Paris
Ecole normale supérieure/PSL, CNRS & INRIA

Outline

Cryptography
Provable Security
Basic Security Notions
Conclusion

Cryptography

Introduction
Kerckhoffs’ Principles
Formal Notations

Provable Security

Basic Security Notions

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

Scytale - Permutation

Alberti’s disk

Mono-alphabetical Substitution

Wheel – M 94 (CSP 488)

Poly-alphabetical Substitution

Outline

Cryptography

Introduction

Kerckhoffs’ Principles

Formal Notations

Provable Security

Basic Security Notions

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

Kerckhoffs’ Principles (3)

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:

\[ E^k \rightarrow G^k \rightarrow D \]

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 \( k \in \{0,1\}^n \):
  \[
  m = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad \text{plaintext}
  \]
  \[
  k = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{key} = \text{random mask}
  \]
  \[
  c = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{ciphertext}
  \]

- 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 \( c \)
\[\Rightarrow\] 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 \( k \)
  \[\Rightarrow\] 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
  \[\Rightarrow\] 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
Cryptography
Introduction
Kerckhoffs' Principles
Formal Notations
Provable Security
Basic Security Notions
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.

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

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:

The secrecy of the private key $sk$ guarantees the secrecy of communications

Provable Security

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

Outline

Cryptography

Provable Security

- Definition
- Computational Assumptions
- Some Reductions

Basic Security Notions

Conclusion
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

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

**Integer Factoring**
- 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>512 bits</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>768 bits</td>
</tr>
</tbody>
</table>

### Integer Factoring Variants

**RSA** [Rivest-Shamir-Adleman 1978]
- Given \( n = pq, e \) and \( y \in \mathbb{Z}_n^* \)
- Find \( x \) such that \( y = x^e \mod n \)

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 \( x = y^d \mod n \)

**Flexible RSA** [Baric-Pfitzmann and Fujisaki-Okamoto 1997]
- 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

**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 \( \mathcal{A} \) by its success probability in finding the solution:

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

We quantify the hardness of the problem by the success probability of the best adversary within time \( t \):

\[
\text{Succ}(t) = \max_{|\mathcal{A}| \leq t} \{ \text{Succ}(\mathcal{A}) \}.
\]

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 \( q \), and the generator \( g \), but \( x \) is randomly chosen:

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


**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 \( Z = g^{xy} \)

The adversary is called a distinguisher (outputs 1 bit).

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

\[
\text{Adv}^{\text{ddh}}_G(A) = \Pr[A(X, Y, Z) = 1 | Z = g^{xy}] - \Pr[A(X, Y, Z) = 1 | Z \leftarrow G]
\]

**Outline**

Cryptography

Provable Security

Definition

Computational Assumptions

Some Reductions

Basic Security Notions

Conclusion

\[
\text{DDH} \leq \text{CDH} \leq \text{DLP}
\]

**CDH ≤ DLP**

Let \( A \) be an adversary against the DLP within time \( t \), then we build an adversary \( B \) against the CDH: given \( X \) and \( Y \), \( B \) runs \( A \) on \( X \), that outputs \( x' \) (correct or not); then \( B \) outputs \( Y x' \).

The running time \( t' \) of \( B \) is the same as \( A \), plus one exponentiation:

\[
\text{Succ}^{\text{cdh}}_G(t') \geq \text{Succ}^{\text{cdh}}_G(B) = \Pr[B(X, Y) \rightarrow g^{xy} = Y^x] = \Pr[A(X) \rightarrow x] = \text{Succ}^{\text{dlp}}_G(A)
\]

Taking the maximum on the adversaries \( A \):

\[
\text{Succ}^{\text{cdh}}_G(t + \tau \exp) \geq \text{Succ}^{\text{dlp}}_G(t)
\]

\[
\text{DDH} \leq \text{CDH}
\]

Let \( A \) be an adversary against the CDH within time \( t \), we build an adversary \( B \) against the DDH: given \( X, Y \) and \( Z \), \( B \) runs \( A \) on \((X, Y)\), that outputs \( Z' \); then \( B \) outputs 1 if \( Z' = Z \) and 0 otherwise.

The running time of \( B \) is the same as \( A \):

\[
\text{Adv}^{\text{ddh}}_G(t) \geq \text{Succ}^{\text{cdh}}_G(t) - 1/q
\]

Taking the maximum on the adversaries \( A \):

\[
\text{Adv}^{\text{ddh}}_G(t) \geq \text{Succ}^{\text{cdh}}_G(t) - 1/q
\]
**Distribution Indistinguishability**

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

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

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

**Computational Indistinguishability**

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

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

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

**Relations between Indistinguishability Notions**

For any distinguisher $\mathcal{A}$, we have

\[ \text{Adv}^{D_0, D_1}(\mathcal{A}) = \Pr_{a \in D_0} [A(a) = 1] - \Pr_{a \in D_1} [A(a) = 1] \leq \sum_{x \in X} \left| \Pr_{a \in D_0} [a = x] - \Pr_{a \in D_1} [a = x] \right| \leq \text{Dist}(D_0, D_1) \]

**Theorem**

$\text{Dist}(D_0, D_1)$ is the best advantage any adversary could get, even within an unbounded time.

\[ \forall t, \quad \text{Adv}^{D_0, D_1}(t) \leq \text{Dist}(D_0, D_1). \]

With a better analysis, we can even get
Hybrid Technique

Let us consider the distributions $D_A$ and $D_B$:

$D_A = (g^x, g^y_1, g^{xy_1}, \ldots, g^y_n, g^{xy_n}) \subseteq \mathbb{G}^{2n+1}$

$D_B = (g^x, g^z_1, \ldots, g^y_n, g^z_n) \subseteq \mathbb{G}^{2n+1}$

Let $\mathcal{A}$ be an adversary within time $t$, against $D_A$ vs. $D_B$.

Given a DDH input $(X, Y, Z)$, we generate the hybrid instance:

$I_i = (X, g^y_1, X^{y_1}, \ldots, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^y_n, g^z_n)$

Note that

- if $Z = g^{xy}$, then $I \in D_i$
- if $Z \overset{R}{\leftarrow} \mathbb{G}$, then $I \in D_{i-1}$

$\text{Adv}^{D_A, D_B}(t) = \text{Adv}^{D_n, D_0}(t) \leq \sum_{i=1}^{n} \text{Adv}^{\text{ddh}}_{\mathcal{G}}(t')$

Theorem

$\forall t, \text{Adv}^{D_A, D_B}(t) \leq n \times \text{Adv}^{\text{ddh}}_{\mathcal{G}}(t + 2(n - 1)\tau_{\text{exp}})$

Outline

Cryptography

Provable Security

Basic Security Notions

Public-Key Encryption

Variants of Indistinguishability

Signatures

Conclusion
Public-Key Encryption

Goal: Privacy/Secrecy of the plaintext

OW − CPA

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

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

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 $EG$ is defined, in a group $G = \langle g \rangle$ of order $q$:

- $K(G, g, q)$: $x \overset{R}{\leftarrow} \mathbb{Z}_q$, and $sk \leftarrow x$ and $pk \leftarrow y = g^x$
- $E_{pk}(m)$: $r \overset{R}{\leftarrow} \mathbb{Z}_q$, $c_1 \leftarrow g^r$ and $c_2 \leftarrow y^r \times m = pk^r \times m$. Then, the ciphertext is $c = (c_1, c_2)$
- $D_{sk}(c)$ outputs $c_2/c_1^x = c_2/c_1^{sk}$

Theorem (ElGamal is OW − CPA)

$$\text{Succ}_{E_G}^\text{ow−cpa}(t) \leq \text{Succ}_{G}^\text{cdh}(t)$$
ElGamal is OW−CPA: Proof

\[ \text{Succ}_{\text{OW−CPA}}^G(t) \leq \text{Succ}_{\text{CDH}}^G(t) \]

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

- \( \mathcal{A} \) gets \( \text{pk} \leftarrow X \) from \( \mathcal{B} \)
- \( \mathcal{B} \) sets \( c_1 \leftarrow Y \)
- \( \mathcal{B} \) chooses \( c_2 \overset{\text{R}}{\leftarrow} \mathbb{G} \) (which virtually sets \( m^* \leftarrow C_2/\text{CDH}(X, Y) \)), and sends \( c = (c_1, c_2) \)
- \( \mathcal{B} \) receives \( m \) from \( \mathcal{A} \) and outputs \( c_2/m \)
- \( \text{Pr}[m = m^*] = \text{Succ}_{\text{OW−CPA}}^G(\mathcal{A}) = \text{Pr}[c_2/m = C_2/m] = \text{Pr}[c_2/m = \text{CDH}(X, Y)] \leq \text{Succ}_{\text{CDH}}^G(t) \)

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?


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 \( m' \)): in the game

\[(sk, pk) \leftarrow \mathcal{K}();(\mathcal{M}, f, \text{state}) \leftarrow \mathcal{A}(pk); m, m' \overset{\text{R}}{\leftarrow} \mathcal{M}; c = \mathcal{E}_{pk}(m); p \leftarrow \mathcal{A}(	ext{state}, c)\]

then,

\[\text{Adv}_{\text{sem}}^G(\mathcal{A}) = \left| \text{Pr}[p = f(m)] - \text{Pr}[p = f(m')] \right| \]

Indistinguishability

Another equivalent formulation (if efficiently computable predicate):

\[\text{IND−CPA}\]

After having chosen two plaintexts \( m_0 \) and \( m_1 \), upon receiving the encryption of \( m_b \) (for a random bit \( b \)), it should be hard to guess which message has been encrypted: in the game

\[(sk, pk) \leftarrow \mathcal{K}();(m_0, m_1, \text{state}) \leftarrow \mathcal{A}(pk); b \overset{\text{R}}{\leftarrow} \{0, 1\}; c = \mathcal{E}_{pk}(m_b); b' \leftarrow \mathcal{A}(	ext{state}, c)\]

then,

\[\text{Adv}_{\text{ind−CPA}}^G(\mathcal{A}) = \left| \text{Pr}[b' = 1|b = 1] - \text{Pr}[b' = 1|b = 0] \right| = \left| 2 \times \text{Pr}[b' = b] - 1 \right|\]
**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 $\mathcal{P}$; it gets $m_0$, $m_1 \overset{R}{\in} D$, and outputs them;
- the challenger chooses $m_b$ in $c$
- $B$ runs $A$, to get the guess $p$ of $A$ about the predicate $\mathcal{P}$ on the plaintext in $c$;
  - If $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $B$ outputs a random bit $b'$,
  - otherwise it outputs $b'$ such that $\mathcal{P}(m_{b'}) = p$.

Note that (if diff denotes the event that $\mathcal{P}(m) \neq \mathcal{P}(m')$)

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

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

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

$$\text{Adv}_{\text{ind}}(B) = \left| \Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0] \right| \leq \text{Adv}_{\text{ind}}'(t')$$

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

$$\text{Adv}_{\text{ind}}'(t) \leq \text{Adv}_{\text{ind}}(t + 2\tau_D + 2\tau_P)$$

$$\text{Adv}_{\text{sem}}(B) = \left| \Pr[p = \mathcal{P}(m)] - \Pr[p = \mathcal{P}(m')] \right|$$

$$\leq \left| \Pr[m = m_p] - \Pr[m' = m_p] \right|,$$

$$\text{Adv}_{\text{ind}}(A) = \left| \Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0] \right|$$

where $m = m_b$
Semantic Security implies Indistinguishability

\[ \text{Adv}^{\text{sem}}(B) = |\Pr[m = m'] - \Pr[m' = m']| \]
\[ = |\Pr[m_b = m_b'] - \Pr[m_d = m_d']| \]
\[ = |\Pr[b = b'] - \Pr[d = b']| \]
\[ = |\Pr[b = b'] - 1/2| \]
\[ = \text{Adv}^{\text{ind}}(A)/2 \leq \text{Adv}^{\text{sem}}(t') \]

The running time \( t' \) of \( B \) is one execution of \( A \) (time \( t \))

\[ \text{Adv}^{\text{ind}}(t) \leq 2 \times \text{Adv}^{\text{sem}}(t) \]

ElGamal Encryption

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

- \( \mathcal{K}(G, g, q): x \xleftarrow{\$} \mathbb{Z}_q \), and \( sk \leftarrow x \) and \( pk \leftarrow y = g^x \)
- \( \mathcal{E}_{pk}(m): r \xleftarrow{\$} \mathbb{Z}_q \), \( c_1 \leftarrow g^r \) and \( c_2 \leftarrow y^r \times m = pk^r \times m \).
  Then, the ciphertext is \( c = (c_1, c_2) \)
- \( D_{sk}(c) \) outputs \( c_2/c_1^k = c_2/c_1^{sk} \)

ElGamal is IND − CPA: Proof

Let \( A \) be an adversary against \( \mathcal{E}_G \), we build an adversary \( B \) against DDH: let us be given a 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 \( d = (b' = b) \)

\[ |2 \times \Pr[b' = b|Z = \text{CDH}(X, Y)] - 1| = \text{Adv}^{\text{ind−cpa}}(A) \]
\[ |2 \times \Pr[b' = b|Z \xleftarrow{\$} G] - 1| = 0 \]

As a consequence,

\[ \text{Adv}^{\text{ind−cpa}}(A) = 2 \times \left| \Pr[d = 1|Z = \text{CDH}(X, Y)] - \Pr[d = 1|Z \xleftarrow{\$} G] \right| \]
\[ = 2 \times \text{Adv}^{\text{ddh}}(B) \leq 2 \times \text{Adv}^{\text{ddh}}(t) \]
## RSA Encryption

The RSA encryption scheme $RSA$ is defined by

- $K(1^k)$: $p$ 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 $pk \leftarrow (n, e)$

- $E_{pk}(m)$: the ciphertext is $c = m^e \mod n$

- $D_{sk}(c)$: the plaintext is $m = c^d \mod n$

### Theorem ($RSA$ is OW – CPA, but...)

$Succ_{OW-CPA}^{RSA}(t) \leq Succ_{RSA}^{rsa}(t)$

A deterministic encryption scheme cannot be IND – CPA

---

## Outline

- Cryptography
- Provable Security
- Basic Security Notions
  - Public-Key Encryption
  - Variants of Indistinguishability
  - Signatures
- Conclusion

---

### Indistinguishability vs. Find-then-Guess

#### FtG – 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 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 $b$

$Adv_{S}^{f_{t-g}\text{-cpa}}(A) = \left| 2 \times \Pr[b' = b] - 1 \right|$

#### 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 $b$

$Adv_{S}^{l_{o-r}\text{-cpa}}(A) = \left| 2 \times \Pr[b' = b] - 1 \right|$

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

$LR_{b}(m_0, m_1)$: outputs the encryption of $m_b$ under $pk$
Find-then-Guess vs. Left-or-Right

**Theorem (FtG ∼ LoR)**

\[
\forall t, \quad \text{Adv}_S^{\text{ftg-cpa}}(t) \leq \text{Adv}_S^{\text{lor-cpa}}(t)
\]

\[
\forall t, \quad \text{Adv}_S^{\text{lor-cpa}}(t) \leq n \times \text{Adv}_S^{\text{ftg-cpa}}(t)
\]

where \( n \) is the number of LR queries

LoR ⇒ FtG is clear

FtG ⇒ 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: \( D_i = (0, \ldots, 0, 1, \ldots, 1) = 0^i 1^{n-i} \in \{0, 1\}^n \)

\[
\text{Dist}(D_0, D_n) = \text{Adv}_S^{\text{lor-cpa}}(A) \quad \text{Dist}(D_i, D_{i+1}) \leq \text{Adv}_S^{\text{ftg-cpa}}(t)
\]

Left-or-Right vs. Real-Random

**Theorem (LoR ∼ RoR)**

\[
\forall t, \quad \text{Adv}_S^{\text{lor-cpa}}(t) \leq \text{Adv}_S^{\text{roR-cpa}}(t)
\]

\[
\forall t, \quad \text{Adv}_S^{\text{roR-cpa}}(t) \leq 2 \times \text{Adv}_S^{\text{lor-cpa}}(t)
\]

LoR ⇒ RoR is clear (using \( m_0 = m \) and \( m_1 \sim_{\mathcal{M}} \))

RoR ⇒ LoR: \( B \) flips a bit \( d \), and uses \( m_d \) for the RR oracle, then forwards \( A \)'s answer

\[
\text{Pr}[d \leftarrow B|\text{Real}] = \text{Pr}[d \leftarrow A] \quad \text{Pr}[d \leftarrow B|\text{Random}] = 1/2
\]

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

\[
= |2 \times \text{Pr}[d \leftarrow B|\text{Real}] - 2 \times \text{Pr}[d \leftarrow B|\text{Random}]|
\]

\[
\leq \ 2 \times \text{Adv}_S^{\text{roR}}(B)
\]

Real-or-Random Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

RoR − 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 asks RR on any message \( m \) of its choice
- The challenger answers using RR

\[
\text{Adv}_S^{\text{ror-cpa}}(A) = |2 \times \text{Pr}[b' = b] - 1|
\]

Outline

Cryptography

Provable Security

Basic Security Notions

Public-Key Encryption

Variants of Indistinguishability

Signatures

Conclusion
**Signature**

- **Goal:** Authentication of the sender

**EUF – NMA**

**Existential Unforgeability**

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

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

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**

**EUF – NMA Security Game**

**RSA Signature**

The RSA signature scheme $RSA$ is defined by

- $K(1^k)$: $p$ and $q$ two random $k$-bit prime integers, and an exponent $v$ (possibly fixed, or not):
  $$sk \leftarrow s = v^{-1} \mod \varphi(n) \text{ and } pk \leftarrow (n, v)$$
- $S_{sk}(m)$: the signature is $\sigma = m^s \mod n$
- $V_{pk}(m, \sigma)$ checks whether $m = \sigma^v \mod n$

**Theorem (RSA is not EUF – NMA)**

The plain RSA signature is not secure at all!
Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available
- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques