# Quelles garanties avec la cryptographie ?

David Pointcheval

Ecole normale supérieure, CNRS & INRIA

Collège de France
27 avril 2011

# Outline


1. Cryptography
2. Provable Security
3. Security of Signatures
4. Security of Encryption

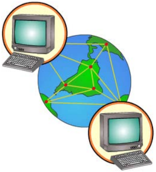
# Security of Communications

One ever wanted to exchange information securely

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

In your pocket


But also at home



# Cryptography

## 3 Historical Goals

- Confidentiality: The content of a message is concealed
- Authenticity: The author of a message is well identified
- Integrity: Messages have not been altered

between a sender and a recipient, against an adversary.

Also within groups, with insider adversaries

Cannot address availability, but should not affect it!

# First Encryption Mechanisms

**The goal of encryption is to hide a message**

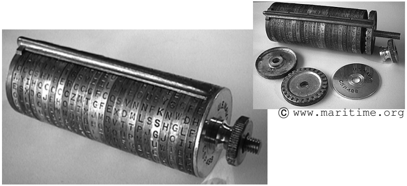
Scytale
Permutation

Substitutions and permutations

**Security** relies on
the secrecy of the mechanism

⇒ **How to widely use them?**

Alberti's disk
Mono-alphabetical Substitution





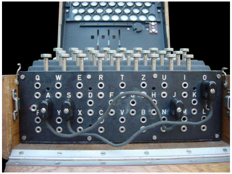
Wheel – M 94 (CSP 488)
Poly-alphabetical Substitution

# Use of a (Secret) Key

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

**Enigma**:

choice of the connectors
and the rotors



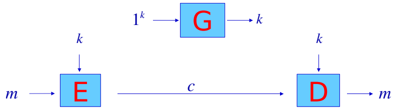
Security **looks** better: but broken (Alan Turing *et al.*)

$\Rightarrow$ **Security analysis is required**

# Modern Cryptography

## Secret Key Encryption

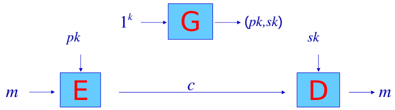
One secret key only shared by Alice and Bob:
this is a common parameter for both E and D




## Public Key Cryptography

[Diffie-Hellman – 1976]

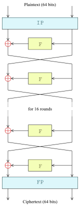
- Bob's public key is used by Alice as a parameter to E
- Bob's private key is used by Bob as a parameter to D

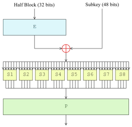


# DES and AES

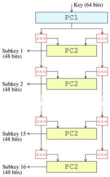
Still substitutions and permutations,
but considering various classes of attacks (statistic)

DES: Data Encryption Standard



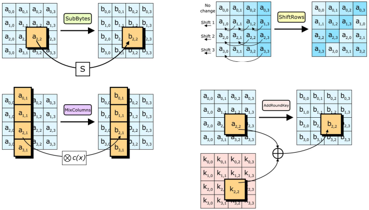

*Round Function F*




“Broken” in 1998 by brute force:
too short keys (56 bits)!

**⇒ No better attack granted a safe design!**

New standard since 2001: Advanced Encryption Standard




Longer keys: from 128 to 256 bits

**Criteria: Security arguments against many attacks**

**What does security mean?**

# Practical Secrecy

## Perfect Secrecy vs. Practical Secrecy

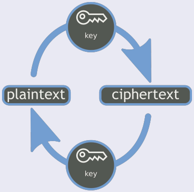
- No information about the plaintext *m* can be extracted from the ciphertext *c*, even for a powerful adversary (unlimited time and/or unlimited power): perfect secrecy
  $\Rightarrow$ information theory
- In practice: adversaries are limited in time/power
  $\Rightarrow$ complexity theory

We thus model all the players (the legitimate ones and the adversary) as Probabilistic Polynomial Time Turing Machines:

**computers that run programs**

# Provable Security

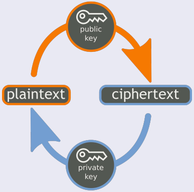
## Symmetric Cryptography




**The secrecy of the key guarantees the secrecy of communications**

**To be proven**

## Asymmetric Cryptography




**The secrecy of the private key guarantees the secrecy of communications**

**To be proven**

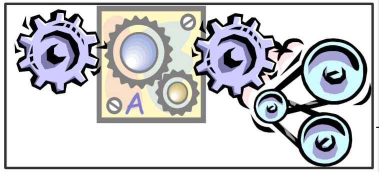
# What is a Secure Cryptographic Scheme?

- What does security mean?
  → Security notions have to be formally defined
- How to guarantee above security claims for concrete schemes?
  → Provable security

## Provable Security

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

hard instance →



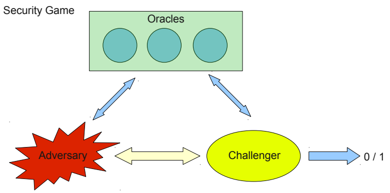
→ solution

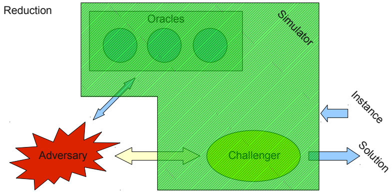
# General Method

## Computational Security Proofs

To prove the security of a cryptographic scheme, one needs

- a formal security model (security notions)
- a reduction: if one (Adversary) can break the security notions, then one (Simulator + Adversary) can break a hard problem
- acceptable computational assumptions (hard problems)




Proof by contradiction

# Integer Factoring

## Records

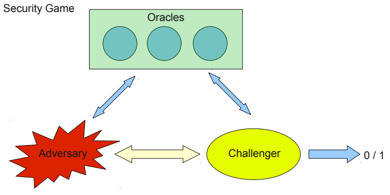
Given $n = pq \longrightarrow$ Find $p$ and $q$

| Digits | Date | Bit-Length |
|---|---|---|
| 130 | April 1996 | 431 bits |
| 140 | February 1999 | 465 bits |
| 155 | August 1999 | 512 bits |
| 160 | April 2003 | 531 bits |
| 200 | May 2005 | 664 bits |
| 232 | December 2009 | 768 bits |

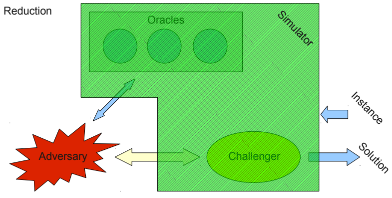
## Complexity

| | |
|---|---|
| 768 bits $\rightarrow 2^{64}$ op. | 3072 bits $\rightarrow 2^{128}$ op. |
| 1024 bits $\rightarrow 2^{80}$ op. | 7680 bits $\rightarrow 2^{192}$ op. |
| 2048 bits $\rightarrow 2^{112}$ op. | 15360 bits $\rightarrow 2^{256}$ op. |

# Reduction




Adversary running time $t$




Algorithm running time $T = f(t)$

- Lossy reduction: $T = k^3 \times t$

| Modulus Bit-length | Adversary Complexity | Algorithm Complexity | Best Known Complexity | |
|---|---|---|---|---|
| $k = 1024$ | $t < 2^{80}$ | $T < 2^{110}$ | $2^{80}$ | ✗ |
| $k = 2048$ | $t < 2^{80}$ | $T < 2^{113}$ | $2^{112}$ | ✗ |
| $k = 3072$ | $t < 2^{80}$ | $T < 2^{115}$ | $2^{128}$ | ✓ |

- Tight reduction: $T \approx t$

  With $k = 1024$ and $t < 2^{80}$, one gets $T < 2^{80}$ ✓

# One-Way Functions

## One-Way Functions

- $\mathcal{F}(1^k)$ generates a function $f : X \to Y$
- From $x \in X$, it is easy to compute $y = f(x)$
- Given $y \in Y$, it is hard to find $x \in X$ such that $y = f(x)$

## RSA Problem

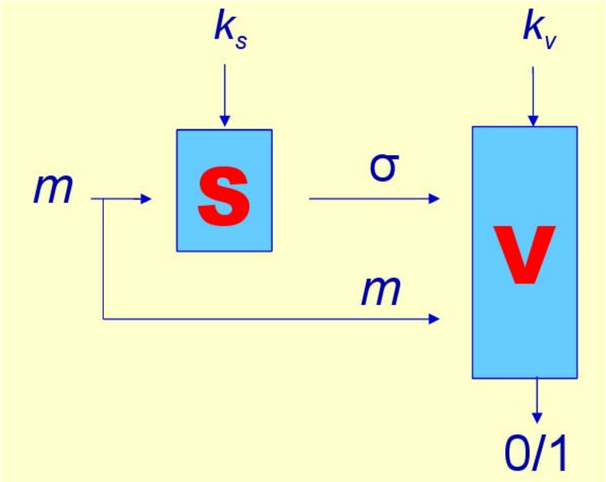
[Rivest-Shamir-Adleman 1978]

- Given $n = pq$, $e$ and $y \in \mathbb{Z}_n^\star$
- Find $x$ such that $y = x^e \bmod n$

This problem is hard without the prime factors $p$ and $q$
It becomes easy with them: if $d = e^{-1} \bmod \varphi(n)$, then $x = y^d \bmod n$

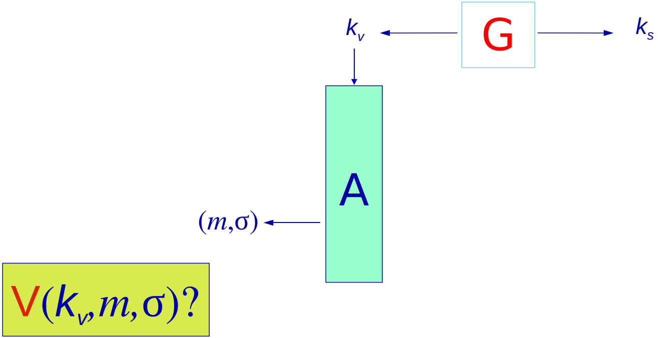
This problem is assumed as hard as integer factoring:
the prime factors are a trapdoor to find solutions
$\Rightarrow$ trapdoor one-way permutation

# Signature




Goal: Authentication of the sender

# EUF – NMA: Security Game




$$\mathbf{Succ}^{\mathsf{euf}}_{\mathcal{SG}}(\mathcal{A}) = \Pr[(k_s, k_v) \leftarrow \mathcal{G}(); (m, \sigma) \leftarrow \mathcal{A}(k_v) : \mathcal{V}(k_v, m, \sigma) = 1]$$

should be negligible.

$\mathcal{A}$ knows the public key only $\Rightarrow$ **No-Message Attack (NMA)**

# $\mathbf{EUF} - \mathbf{NMA}$

## One-Way Function

- $\mathcal{G}(1^k)$: $f \stackrel{R}{\leftarrow} \mathcal{F}(1^k)$ and $x \stackrel{R}{\leftarrow} X$, set $y = f(x)$,
  $k_s = x$ and $k_v = (f, y)$
- $\mathcal{S}(x, m) = k_s = x$
- $\mathcal{V}((f, y), m, x')$ checks whether $f(x') = y$

Under the one-wayness of $\mathcal{F}$, $\mathbf{Succ}^{\mathsf{euf-nma}}(\mathcal{A})$ is small.

But given one signature, one can “sign” any other message!
Signatures are public! $\Rightarrow$ **Known-Message Attacks ($\mathbf{KMA}$)**

The adversary has access to a list of messages-signatures

# EUF − KMA

## One-Way Functions

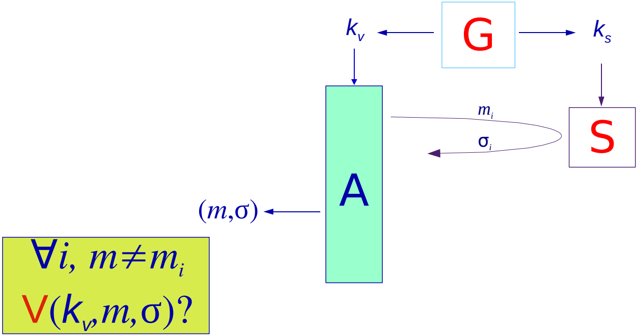
- $\mathcal{G}(1^k)$: $f \stackrel{R}{\leftarrow} \mathcal{F}(1^k)$, and $\vec{x} = (x_{1,0}, x_{1,1}, \ldots, x_{k,0}, x_{k,1}) \stackrel{R}{\leftarrow} X^{2k}$,
  $y_{i,j} = f(x_{i,j})$ for $i = 1, \ldots, k$ and $j = 0, 1$,
  $k_s = \vec{x}$ and $k_v = (f, \vec{y})$
- $\mathcal{S}(\vec{x}, m) = (x_{i,m_i})_{i=1,\ldots,k}$
- $\mathcal{V}((f, \vec{y}), m, (x'_i))$ checks whether $f(x'_i) = y_{i,m_i}$ for $i = 1, \ldots, k$

Under the one-wayness of $\mathcal{F}$, $\mathbf{Succ}^{\text{euf-nma}}(\mathcal{A})$ is small.
With the signature of $m = 0^k$, I cannot forge any other signature.

With the signatures of $m = 0^k$ and $m' = 1^k$, I learn $\vec{x}$: the secret key
Messages can be under the control of the adversary!

$\Rightarrow$ **Chosen-Message Attacks** (**CMA**)

# EUF − CMA




The adversary has access to any signature of its choice: Chosen-Message Attacks (oracle access):

$$\mathbf{Succ}^{\mathsf{euf-cma}}_{\mathcal{SG}}(\mathcal{A}) = \Pr\left[\begin{array}{l}(k_s, k_v) \leftarrow \mathcal{G}(); (m, \sigma) \leftarrow \mathcal{A}^{\mathcal{S}(k_s, \cdot)}(k_v) : \\ \forall i, m \neq m_i \wedge \mathcal{V}(k_v, m, \sigma) = 1\end{array}\right]$$

# The $\mathcal{RSA}$ Signature

[Rivest-Shamir-Adleman 1978]

## The $\mathcal{RSA}$ Signature

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

- $\mathcal{G}(1^k)$: $p$ and $q$, two random primes, and an exponent $v$

  $$n = pq,\ k_s \leftarrow s = v^{-1} \bmod \varphi(n) \text{ and } k_v \leftarrow (n, v)$$
- $\mathcal{S}(k_s, m)$: the signature is $\sigma = m^s \bmod n$
- $\mathcal{V}(k_v, m, \sigma)$ checks whether $m = \sigma^v \bmod n$

## Theorem (The Plain $\mathcal{RSA}$ is not $\mathrm{EUF} - \mathrm{NMA}$)

*The plain RSA signature is not secure at all!*

## Proof.

Choose a random $\sigma \in \mathbb{Z}_n^\star$, and set $m = \sigma^v \bmod n$.

By construction, $\sigma$ is a valid signature of $m$ □

# Full-Domain Hash Signature

[Bellare-Rogaway – Eurocrypt '96]

## Full-Domain Hash $\mathcal{RSA}$ Signature

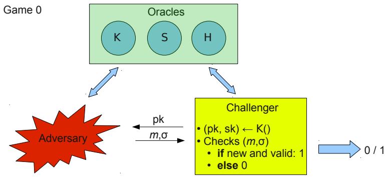
The FDH-RSA signature scheme is defined by

- $\mathcal{G}(1^k)$: $p$ and $q$, two random primes, and an exponent $v$
  $$n = pq,\ k_s \leftarrow s = v^{-1} \bmod \varphi(n) \text{ and } k_v \leftarrow (n, v)$$
- $\mathcal{H}$ is a hash function onto $\mathbb{Z}_n^\star$
- $\mathcal{S}(k_s, m)$: the signature is $\sigma = \mathcal{H}(m)^s \bmod n$
- $\mathcal{V}(k_v, m, \sigma)$ checks whether $\mathcal{H}(m) = \sigma^v \bmod n$

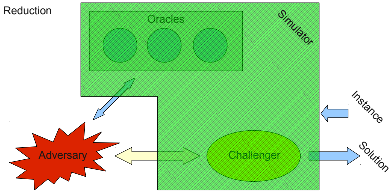
## Theorem (Security of the FDH-RSA)

*The FDH-RSA is* $\mathbf{EUF-CMA}$ *under appropriate assumptions on* $\mathcal{H}$*, and assuming the RSA problem is hard*

# FDH-RSA Security




Adversary running time $t$




Algorithm running time $T = f(t)$

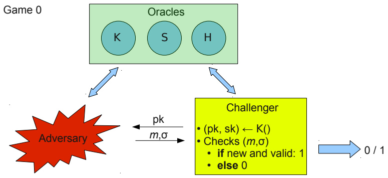
Initial reduction: $T \approx q_H \times t$ [Bellare-Rogaway – Eurocrypt '96]

(where $q_H$ is number of Hashing queries $\approx 2^{60}$)

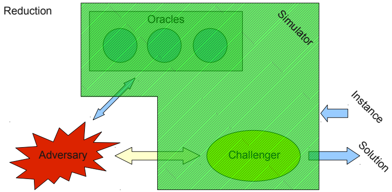
| | | | | |
|---|---|---|---|---|
| $k = 1024$ | $(2^{80})$ | $t < 2^{80}$ | $T < 2^{140}$ | ✗ |
| $k = 2048$ | $(2^{112})$ | $t < 2^{80}$ | $T < 2^{140}$ | ✗ |
| $k = 3072$ | $(2^{128})$ | $t < 2^{80}$ | $T < 2^{140}$ | ✗ |

$\Longrightarrow$ large modulus required!

# Improved Security




Adversary running time $t$




Algorithm running time $T = f(t)$

By exploiting the random self-reducibility of RSA: $(xr)^e = x^e r^e \bmod n$

$\Longrightarrow$ Improved reduction: $T \approx q_S \times t$ [Coron – Crypto '00]

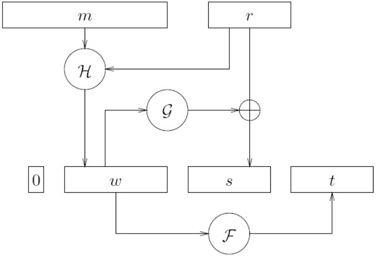
(where $q_S$ is the number is Signing queries $\leq 2^{30}$)

With $k = 2048$ and $t < 2^{80}$, one gets $T < 2^{110}$ ✓

(Best algorithm in $2^{112}$)

# RSA-PSS (PKCS #1 v2.1)

[Bellare-Rogaway – Eurocrypt '96]




- $m$ is the message to encrypt
- $r$ is the additional randomness to make encryption probabilistic

After the transformation,
$w\|s\|t$ goes in the plain RSA

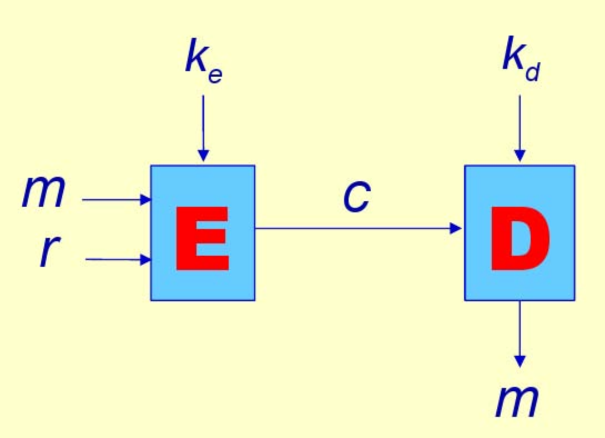
**Theorem (EUF-CMA Security** [Bellare-Rogaway – Eurocrypt '96]**)**

*RSA-PSS is* **EUF-CMA** *secure under the RSA assumption*

Security reduction between $\mathbf{EUF-CMA}$ and the RSA assumption:
$T \approx t$

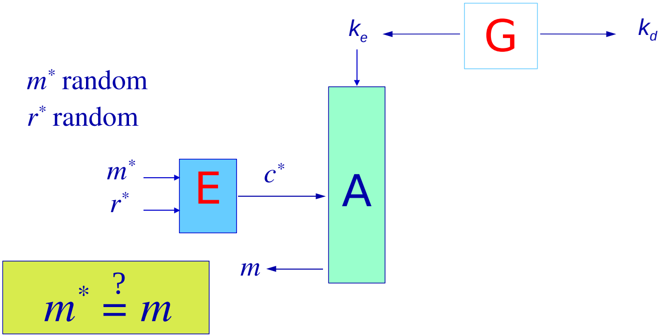
$\Longrightarrow$ 1024-bit RSA moduli provide $2^{80}$ security

# Public-Key Encryption




Goal: Privacy/Secrecy of the plaintext

# OW – CPA: Security Game




$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{S}}(\mathcal{A}) = \Pr\left[\begin{array}{c}(k_d, k_e) \leftarrow \mathcal{G}(); m^* \stackrel{R}{\leftarrow} \mathcal{M}; c = \mathcal{E}(k_e, m^*, r^*) : \\ \mathcal{A}(k_e, c^*) \rightarrow m^*\end{array}\right]$$

should be negligible.

# $\mathbf{OW}-\mathbf{CPA}$: Is it Enough?

## The $\mathcal{RSA}$ Encryption [Rivest-Shamir-Adleman 1978]

- $\mathcal{G}(1^k)$: $p$ and $q$, two random primes, and an exponent $e$: $n = pq$, $sk \leftarrow d = e^{-1} \bmod \varphi(n)$ and $pk \leftarrow (n, e)$
- $\mathcal{E}(pk, m) = c = m^e \bmod n$ ; $\mathcal{D}(sk, c) = m = c^d \bmod n$

$\mathcal{RSA}$ encryption is $\mathbf{OW}-\mathbf{CPA}$, under the RSA assumption

## $\mathbf{OW}-\mathbf{CPA}$ Too Weak

- $\mathcal{G}' = \mathcal{G}$; $\mathcal{E}'(pk, m = m_1 \| m_2) = \mathcal{E}(pk, m_1) \| m_2 = c_1 \| c_2$
- $\mathcal{D}'(sk, c_1 \| c_2)$: $m_1 = \mathcal{D}(sk, c_1)$, $m_2 = c_2$, output $m = m_1 \| m_2$

If $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is $\mathbf{OW}-\mathbf{CPA}$: then $(\mathcal{G}', \mathcal{E}', \mathcal{D}')$ is $\mathbf{OW}-\mathbf{CPA}$ too

But this is clearly not enough: **half or more of the message leaks!**

# OW − CPA: Is it 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?

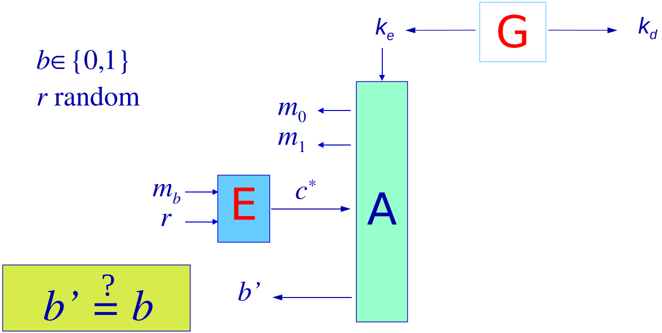
**Perfect Secrecy vs. Computational Secrecy**

- **Perfect secrecy**: the distribution of the ciphertext is perfectly independent of the plaintext
- **Computational secrecy**: the distribution of the ciphertext is computationally independent of the plaintext

Idea: No adversary can distinguish
a ciphertext of $m_0$ from a ciphertext of $m_1$.

Probabilistic encryption is required!

# IND – CPA: Security Game




$$(k_d, k_e) \leftarrow \mathcal{G}(); (m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}(k_e);$$

$$b \stackrel{R}{\leftarrow} \{0,1\}; c^* = \mathcal{E}(k_e, m_b, r); b' \leftarrow \mathcal{A}(\mathsf{state}, c^*)$$

$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = 2 \times \Pr[b' = b] - 1$ should be negligible.

# ElGamal Encryption

[ElGamal 1985]

## The ElGamal Encryption ($\mathcal{EG}$)

- $\mathcal{G}(1^k)$: $\mathbb{G} = \langle g \rangle$ of order $q$, $sk = x \stackrel{R}{\leftarrow} \mathbb{Z}_q$ and $pk \leftarrow y = g^x$
- $\mathcal{E}(pk, m, r) = (c_1 = g^r, c_2 = y^r m)$
- $\mathcal{D}(sk, (c_1, c_2)) = c_2 / c_1^x$

The ElGamal encryption is $\mathbf{IND} - \mathbf{CPA}$, under the **DDH** assumption

## Decisional Diffie-Hellman Problem

For $\mathbb{G} = \langle g \rangle$ of order $q$, and $x, y \stackrel{R}{\leftarrow} \mathbb{Z}_q$,

- Given $X = g^x$, $Y = g^y$ and $Z = g^z$, for either $z \stackrel{R}{\leftarrow} \mathbb{Z}_q$ or $z = xy$
- Decide whether $z = xy$

This problem is assumed hard to decide in appropriate groups $\mathbb{G}$!

# ElGamal is $\mathrm{IND}-\mathrm{CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$: $\mathcal{B}$ is an adversary against **DDH**: let us be given a **DDH** instance $(X = g^x, Y = g^y, Z = g^z)$

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$, and outputs $(m_0, m_1)$
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $b \stackrel{R}{\leftarrow} \{0,1\}$, sets $c_2 \leftarrow Z \times m_b$, and sends $c = (c_1, c_2)$
- $\mathcal{B}$ receives $b'$ from $\mathcal{A}$ and outputs $d = (b' = b)$
- $2 \times \Pr[b' = b] - 1$
  $= \mathbf{Adv}^{\mathsf{ind-cpa}}_{\mathcal{EG}}(\mathcal{A})$, if $z = xy$
  $= 0$, if $z \stackrel{R}{\leftarrow} \mathbb{Z}_q$

# ElGamal is $\mathbf{IND-CPA}$: Proof

As a consequence,

- $2 \times \Pr[b' = b | z = xy] - 1 = \mathbf{Adv}_{\mathcal{EG}}^{\mathsf{ind-cpa}}(\mathcal{A})$
- $2 \times \Pr[b' = b | z \overset{R}{\leftarrow} \mathbb{Z}_q] - 1 = 0$

If one subtracts the two lines:

$$\begin{aligned}\mathbf{Adv}_{\mathcal{EG}}^{\mathsf{ind-cpa}}(\mathcal{A}) &= 2 \times \begin{pmatrix} \Pr[d = 1 | z = xy] \\ - \Pr[d = 1 | z \overset{R}{\leftarrow} \mathbb{Z}_q] \end{pmatrix} \\ &= 2 \times \mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(\mathcal{B}) \leq 2 \times \mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(t)\end{aligned}$$

# IND – CPA: Is it Enough?

## The ElGamal Encryption [ElGamal 1985]

- $\mathcal{G}(1^k)$: $G = \langle g \rangle$ of order $q$, $sk = x \stackrel{R}{\leftarrow} \mathbb{Z}_q$ and $pk \leftarrow y = g^x$
- $\mathcal{E}(pk, m, r) = (c_1 = g^r, c_2 = y^r m)$ ; $\mathcal{D}(sk, (c_1, c_2)) = c_2 / c_1^x$

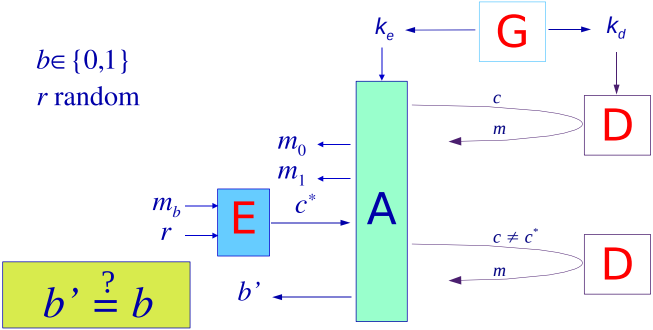
## Private Auctions

All the players $P_i$ encrypt their bids $c_i = \mathcal{E}(pk, b_i)$ for the authority; the authority opens all the $c_i$; the highest bid $b_I$ wins

- **IND – CPA** guarantees privacy of the bids
- Malleability: from $c_i = \mathcal{E}(pk, b_i)$, without knowing $b_i$, one can generate $c' = \mathcal{E}(pk, 2b_i)$: **an unknown higher bid!**

**IND – CPA** does not imply Non-Malleability

# IND − CCA: Security Game




The adversary can ask any decryption of its choice:

$\Rightarrow$ **Chosen-Ciphertext Attacks** (CCA)

**Theorem (NM vs. CCA** [Bellare-Desai-Pointcheval-Rogaway – Crypto '98]**)**

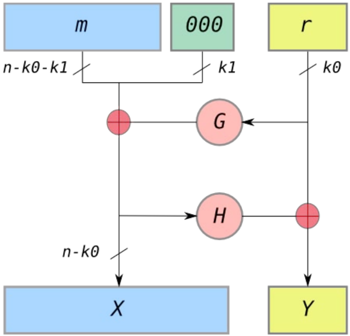
*The chosen-ciphertext security implies non-malleability*

$\Longrightarrow$ *the highest security level*

# RSA-OAEP (PKCS #1 v2.1)

[Bellare-Rogaway – Eurocrypt '94]

The $\mathcal{RSA}$ encryption is $\mathbf{OW-CPA}$, under the RSA assumption, but even not $\mathbf{IND-CPA}$: **need of randomness and redundancy**



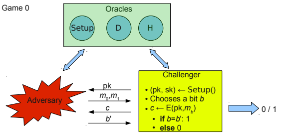

- $m$ is the message to encrypt
- $r$ is the additional randomness to make encryption probabilistic
- $00\ldots00$ is redundancy to be checked at decryption time

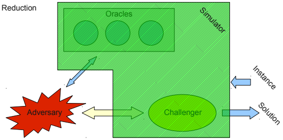
After the transformation, $X\|Y$ goes in the plain RSA

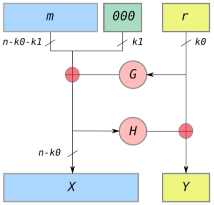
**Theorem (IND-CCA Security** [Fujisaki-Okamoto-Pointcheval-Stern – Crypto '01]**)**

*RSA-OAEP is IND-CCA secure under the RSA assumption*

# RSA-OAEP Security Proof [Fujisaki-Okamoto-Pointcheval-Stern – Crypto '01]



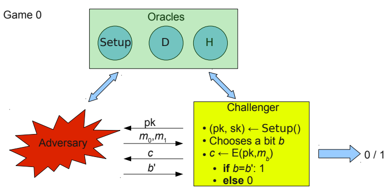

$$c = f(X \| Y)$$

More precisely, to get information on $m$, encrypted in $c = f(X\|Y)$, one must have asked $\mathcal{H}(X) \Longrightarrow$ partial inversion of $f$

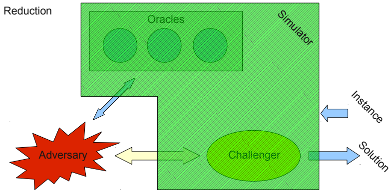
For RSA: partial inversion and full inversion are equivalent
(but at a computational loss)

# RSA-OAEP Security

[Fujisaki-Okamoto-Pointcheval-Stern – Crypto ’01]




Adversary running time $t$




Algorithm running time $T = f(t)$

If there is an adversary that distinguishes, within time $t$,
the two ciphertexts with overwhelming advantage (close to 1),
one can break RSA within time $T \approx 2t + 3{q_H}^2 k^3$
(where $q_H$ is number of Hashing queries $\approx 2^{60}$)

| | | | | |
|---|---|---|---|---|
| $k = 1024$ | $(2^{80})$ | $t < 2^{80}$ | $T < 2^{152}$ | ✗ |
| $k = 2048$ | $(2^{112})$ | $t < 2^{80}$ | $T < 2^{155}$ | ✗ |
| $k = 3072$ | $(2^{128})$ | $t < 2^{80}$ | $T < 2^{158}$ | ✗ |

$\Longrightarrow$ large modulus: $>$ 4096 bits!

# REACT-RSA Security

[Okamoto-Pointcheval – CT-RSA '01]

## REACT-RSA

- $\mathcal{G}(1^k)$: $p$ and $q$, two random primes, and an exponent $e$: $n = pq$, $sk \leftarrow d = e^{-1} \bmod \varphi(n)$ and $pk \leftarrow (n, e)$
- $\mathcal{E}(pk, m, r) =$

$$(c_1 = r^e \bmod n, c_2 = G(r) \oplus m, c_3 = H(r, m, c_1, c_2))$$

- $\mathcal{D}(sk, (c_1, c_2, c_3))$: $r = c_1^d \bmod n$, $m = c_2 \oplus G(r)$, if $c_3 = H(r, m, c_1, c_2)$ then output $m$, else output $\perp$

Security reduction between $\mathbf{IND} - \mathbf{CCA}$ and the RSA assumption:

$T \approx t$

$\Longrightarrow$ 1024-bit RSA moduli provide $2^{80}$ security

# Conclusion

With provable security, one can precisely get:

- the security games one wants to resist against any adversary
- the security level, according to the resources of the adversary

But, it is under some assumptions:

- the best attacks against famous problems (integer factoring, etc)
- no leakage of information excepted from the given oracles

Cryptographers' goals are thus

- to analyze the intractability of the underlying problems
- to define realistic and strong security notions (games)
- to correctly model the leakage of information (oracle access)
- to design schemes with tight security reductions

**Implementations and uses must satisfy the constraints!**