I – Basics

David Pointcheval
Ecole normale supérieure, CNRS & INRIA

IACR-SEAMS School
Cryptographie: Foundations and New Directions
November 2016 – Hanoi – Vietnam

Outline

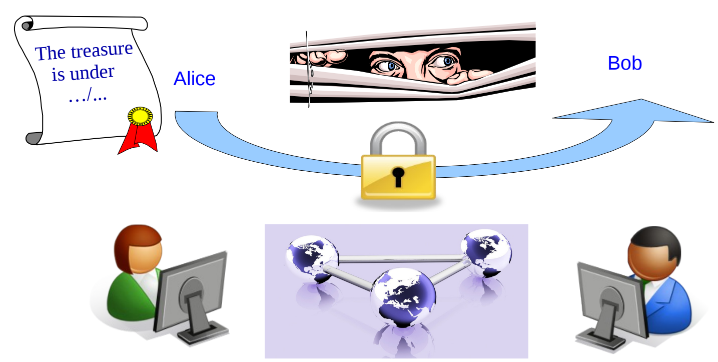
1. Cryptography
   - Introduction
   - Formal Notations
2. Provable Security
   - Definition
   - Computational Assumptions
   - Some Reductions
3. Public-Key Encryption
   - One-Wayness
   - Indistinguishability
4. Conclusion

Outline

1. Cryptography
   - Introduction
   - Formal Notations
2. Provable Security
3. Public-Key Encryption
4. Conclusion

Secrecy of Communications

One ever wanted to communicate secretly



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

What Does Secrecy Mean?

Shannon provides a definition of secrecy:

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

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

Outline

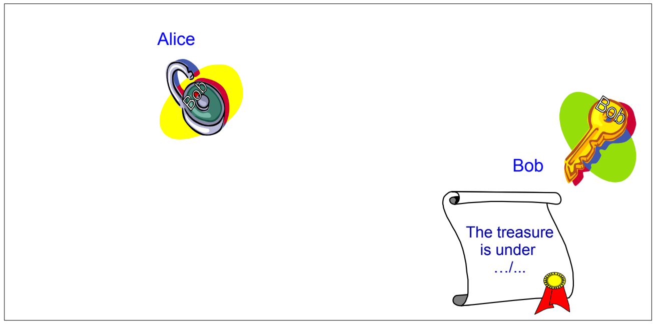
1. Cryptography
   - Introduction
   - Formal Notations
2. Provable Security
3. Public-Key Encryption
4. Conclusion

Asymmetric Encryption: Intuition [Diffie-Hellman 1976]

Secrecy

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

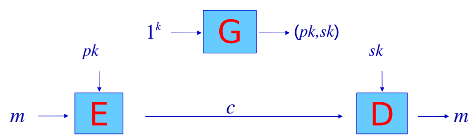
Why would the sender need a secret key to encrypt a message?



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



Secrecy of the private key $sk \Rightarrow$ secrecy of communications
Because of pk, perfect secrecy is definitely impossible!

Outline

1 Cryptography

2 Provable Security
- Definition
- Computational Assumptions
- Some Reductions

3 Public-Key Encryption

4 Conclusion

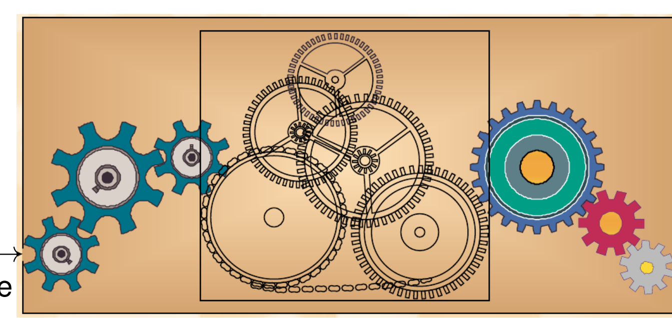
What is a Secure Cryptographic Scheme/Protocol?

- Public-key encryption:
  Secrecy of the private key $sk \Rightarrow$ secrecy of communications
- What does mean secrecy?
  $\rightarrow$ Security notions have to be formally defined
- How to guarantee above security claims for concrete schemes?
  $\rightarrow$ 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

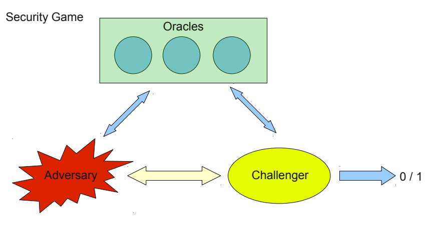


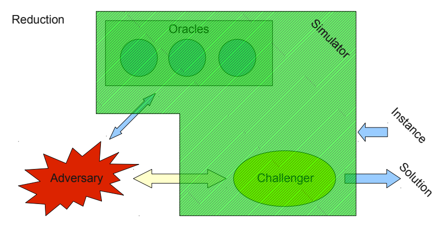
General Method

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





Outline

1 Cryptography

2 Provable Security
- Definition
- Computational Assumptions
- Some Reductions

3 Public-Key Encryption

4 Conclusion

Integer Factoring

[Lenstra-Verheul 2000]

Integer Factoring

- Given $n = pq$
- Find p and q

Year	Required Complexity	n bitlength
before 2000	64	768
before 2010	80	1024
before 2020	112	2048
before 2030	128	3072
	192	7680
	256	15360

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

Integer Factoring Records

Integer Factoring

- Given $n = pq$
- Find p and q

Digits	Date	Details
129	April 1994	Quadratic Sieve
130	April 1996	Algebraic Sieve
140	February 1999	
155	August 1999	512 bits
160	April 2003	
200	May 2005	
232	December 2009	768 bits

Integer Factoring Variants

RSA [Rivest-Shamir-Adleman 1978]

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

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

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

- Given $n = pq$ and $y \in \mathbb{Z}_n^\star$
- Find x and $e > 1$ such that $y = x^e \bmod n$

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

Discrete Logarithm

Discrete Logarithm Problem

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

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

(Computational) Diffie Hellman Problem

- Given $\mathbb{G} = \langle g \rangle$ 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:

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

We quantify the hardness of the problem by the success probability of the best adversary within time t: $\mathbf{Succ}(t) = \max_{|\mathcal{A}| \leq t}\{\mathbf{Succ}(\mathcal{A})\}$

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 $\mathbb{G} = \langle g \rangle$ of order q, X is randomly chosen:

$$\mathbf{Succ}_{\mathbb{G}}^{\mathbf{dlp}}(\mathcal{A}) = \Pr_{x \stackrel{R}{\leftarrow} \mathbb{Z}_q}[\mathcal{A}(g^x) \rightarrow x]$$

Decisional Problem

(Decisional) Diffie Hellman Problem

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

$$\mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(\mathcal{A}) = \Pr[\mathcal{A}(X, Y, Z) = 1 | Z = g^{xy}] - \Pr[\mathcal{A}(X, Y, Z) = 1 | Z \stackrel{R}{\leftarrow} \mathbb{G}]$$

$$\mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(t) = \max_{|\mathcal{A}| \leq t}\{\mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(\mathcal{A})\}$$

Distribution Indistinguishability

Indistinguishabilities

Let $\mathcal{D}_0$ and $\mathcal{D}_1$, two distributions on a finite set X:

- $\mathcal{D}_0$ and $\mathcal{D}_1$ are perfectly indistinguishable if

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1) = \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_1}[a = x] - \Pr_{a \in \mathcal{D}_0}[a = x] \right| = 0$$

- $\mathcal{D}_0$ and $\mathcal{D}_1$ are statistically indistinguishable if

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1) = \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_1}[a = x] - \Pr_{a \in \mathcal{D}_0}[a = x] \right| = \mathrm{negl}()$$

Distribution Indistinguishability

Computational Indistinguishability

Let $\mathcal{D}_0$ and $\mathcal{D}_1$, two distributions on a finite set X,

- a distinguisher $\mathcal{A}$ between $\mathcal{D}_0$ and $\mathcal{D}_1$

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) = \Pr_{a \in \mathcal{D}_1}[\mathcal{A}(a) = 1] - \Pr_{a \in \mathcal{D}_0}[\mathcal{A}(a) = 1]$$

- the computational indistinguishability of $\mathcal{D}_0$ and $\mathcal{D}_1$ is

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(t) = \max_{|\mathcal{A}| \leq t}\{\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A})\}$$

Theorem

$$\forall t, \quad \mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(t) \leq \mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1)$$

Outline

1 Cryptography

2 Provable Security
- Definition
- Computational Assumptions
- Some Reductions

3 Public-Key Encryption

4 Conclusion

DDH $\leq$ CDH $\leq$ DLP

CDH $\leq$ DLP

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

The running time t' of $\mathcal{B}$ is the same as $\mathcal{A}$, plus one exponentiation:

$$\begin{aligned}\mathbf{Succ}^{\mathsf{cdh}}_{\mathbb{G}}(t') \geq \mathbf{Succ}^{\mathsf{cdh}}_{\mathbb{G}}(\mathcal{B}) &= \Pr[\mathcal{B}(X, Y) \rightarrow g^{xy} = Y^x] \\ &= \Pr[\mathcal{A}(X) \rightarrow x] = \mathbf{Succ}^{\mathsf{dlp}}_{\mathbb{G}}(\mathcal{A})\end{aligned}$$

Taking the maximum on the adversaries $\mathcal{A}$:

$$\mathbf{Succ}^{\mathsf{cdh}}_{\mathbb{G}}(t + \tau_{\mathrm{exp}}) \geq \mathbf{Succ}^{\mathsf{dlp}}_{\mathbb{G}}(t)$$

DDH $\leq$ CDH $\leq$ DLP

DDH $\leq$ CDH

Let $\mathcal{A}$ be an adversary against the **CDH** within time t, we build an adversary $\mathcal{B}$ against the **DDH**: given X, Y and Z, $\mathcal{B}$ runs $\mathcal{A}$ on (X, Y), that outputs Z'; then $\mathcal{B}$ outputs 1 if $Z' = Z$ and 0 otherwise
The running time of $\mathcal{B}$ is the same as $\mathcal{A}$: and $\mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t)$ is greater than

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(\mathcal{B}) &= \Pr[\mathcal{B} \rightarrow 1 | Z = g^{xy}] - \Pr[\mathcal{B} \rightarrow 1 | Z \stackrel{R}{\leftarrow} \mathbb{G}] \\
&= \Pr[\mathcal{A}(X, Y) \rightarrow Z | Z = g^{xy}] - \Pr[\mathcal{A}(X, Y) \rightarrow Z | Z \stackrel{R}{\leftarrow} \mathbb{G}] \\
&= \Pr[\mathcal{A}(X, Y) \rightarrow g^{xy}] - \Pr[\mathcal{A}(X, Y) \rightarrow Z | Z \stackrel{R}{\leftarrow} \mathbb{G}] \\
&= \mathbf{Succ}^{\mathsf{cdh}}_{\mathbb{G}}(\mathcal{A}) - 1/q
\end{aligned}$$

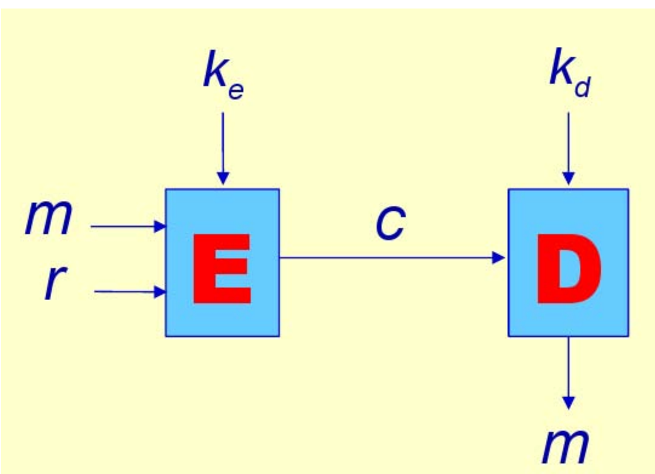
Taking the maximum on the adversaries $\mathcal{A}$:

$$\mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t) \geq \mathbf{Succ}^{\mathsf{cdh}}_{\mathbb{G}}(t) - 1/q$$

Outline

1 Cryptography
2 Provable Security
3 Public-Key Encryption
- One-Wayness
- Indistinguishability
4 Conclusion

Public-Key Encryption



Goal: Privacy/Secrecy of the plaintext

OW − CPA

One-Wayness

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

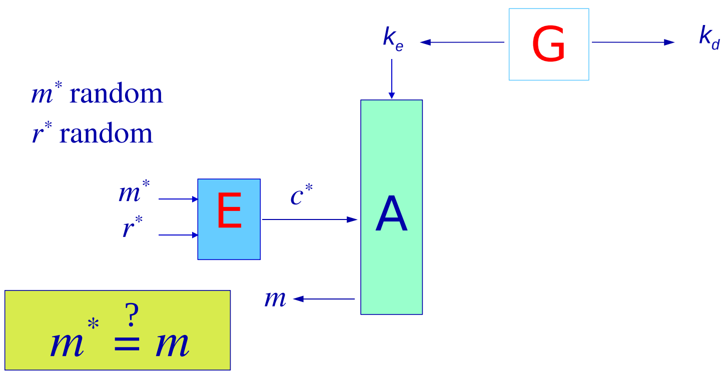
$$\mathbf{Succ}^{\mathsf{ow}}_{\mathcal{S}}(\mathcal{A}) = \Pr[(sk, pk) \leftarrow \mathcal{K}(); m \stackrel{R}{\leftarrow} \mathcal{M}; c = \mathcal{E}_{pk}(m) : \mathcal{A}(pk, c) \rightarrow m]$$

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

OW − CPA Security Game



ElGamal Encryption

[ElGamal 1985]

ElGamal Encryption

The ElGamal encryption scheme $\mathcal{EG}$ is defined, in a group $\mathbb{G} = \langle g \rangle$ of order q, for $m \in \mathbb{G}$

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

Theorem (ElGamal is OW − CPA)

$$\mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(t) \leq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t)$$

ElGamal is OW − CPA: Proof

$$\mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(t) \leq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t)$$

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

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $c_2 \stackrel{R}{\leftarrow} \mathbb{G}$ (this implicitly defines $m^* = c_2 / \mathbf{CDH}(X, Y)$), and sends $c = (c_1, c_2)$
- $\mathcal{B}$ receives m from $\mathcal{A}$ and outputs c_2 / m
- $\Pr[m = m^*] = \mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(\mathcal{A})$
  $= \Pr[c_2 / m = c_2 / m^*] = \Pr[c_2 / m = \mathbf{CDH}(X, Y)] \leq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t)$

Outline

1 Cryptography

2 Provable Security

3 Public-Key Encryption
- One-Wayness
- Indistinguishability

4 Conclusion

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?

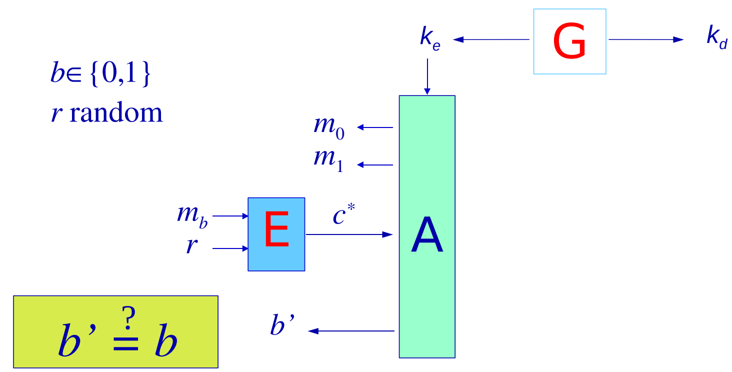
Semantic Security / Indistinguishability [Goldwasser-Micali 1984]

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:

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

$$\mathbf{Adv}^{\text{ind}-\text{cpa}}_{\mathcal{S}}(\mathcal{A}) = \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]$$

IND − CPA Security Game



ElGamal Encryption

ElGamal Encryption

The ElGamal encryption scheme $\mathcal{EG}$ is defined,
in a group $\mathbb{G} = \langle g \rangle$ of order q, for $m \in \mathbb{G}$

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

Theorem (ElGamal is IND − CPA)

$$\mathbf{Adv}^{\text{ind}-\text{cpa}}_{\mathcal{EG}}(t) \leq 2 \times \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t)$$

ElGamal is IND − CPA: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, 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 $d = (b' = b)$
- $2 \times \Pr[b' = b] - 1$
  $= \mathbf{Adv}^{\text{ind}-\text{cpa}}_{\mathcal{EG}}(\mathcal{A})$, if $Z = \mathbf{CDH}(X, Y)$
  $= 0$, otherwise

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

As a consequence,

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

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ind-cpa}}_{\mathcal{EG}}(\mathcal{A}) &= 2 \times \left(\begin{array}{l} \Pr[d = 1 | Z = \mathbf{CDH}(X,Y)] \\ -\Pr[d = 1 | Z \stackrel{R}{\leftarrow} \mathbb{G}] \end{array} \right) \\ &= 2 \times \mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(\mathcal{B}) \leq 2 \times \mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t)\end{aligned}$$

$\mathcal{RSA}$ Encryption

[Rivest-Shamir-Adleman 1978]

$\mathcal{RSA}$ 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): $sk \leftarrow d = e^{-1} \bmod \varphi(n)$ and $pk \leftarrow (n, e)$
- $\mathcal{E}_{pk}(m)$: the ciphertext is $c = m^e \bmod n$
- $\mathcal{D}_{sk}(c)$: the plaintext is $m = c^d \bmod n$

Theorem ($\mathcal{RSA}$ is $\mathbf{OW-CPA}$, but...)

$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{RSA}}(t) \leq \mathbf{Succ}^{\mathsf{rsa}}(t)$$

A deterministic encryption scheme cannot be $\mathbf{IND-CPA}$

Outline

1 Cryptography
- Introduction
- Formal Notations

2 Provable Security
- Definition
- Computational Assumptions
- Some Reductions

3 Public-Key Encryption
- One-Wayness
- Indistinguishability

4 Conclusion

Conclusion

Global methodology for provable security:

- 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

We will apply this methodology

- on advanced security notions for encryption
- to signature schemes