

Models and analysis of security protocols

1st Semester 2009-2010

Indistinguishability

Lecture 2

Pascal Lafourcade

Université Joseph Fourier, Verimag

Master: 24th September 2009

Last Time (I)

Introduction

- ▶ Presentation
- ▶ Organization
- ▶ Motivation
- ▶ Mathematics Recalls
- ▶ Birthday Paradox
- ▶ Negligible functions

Remarks, questions, comments ?

Last Time (II)

Exercises to do IMPORTANT

- ▶ Give the security properties for an international airport
- ▶ Drug Test
- ▶ Independant variables
- ▶ Negligible and Noticeable Functions.

Others Exercises

- ▶ Expectation properties
- ▶ Proofs of different probabilistic theorems.
- ▶ Generalization of Birthday Paradox.

Outline of Today:

Hard Problems

- Factorization

- Discret Logarithm

- Diffie-Hellman

- Summary

Outline of Today:

Hard Problems

- Factorization

- Discret Logarithm

- Diffie-Hellman

- Summary

Intuition of Indistinguishability

Outline of Today:

Hard Problems

- Factorization

- Discret Logarithm

- Diffie-Hellman

- Summary

Intuition of Indistinguishability

Different Adversaries

Outline of Today:

Hard Problems

- Factorization

- Discret Logarithm

- Diffie-Hellman

- Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Outline of Today:

Hard Problems

- Factorization

- Discret Logarithm

- Diffie-Hellman

- Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Outline of Today:

Hard Problems

Factorization

Discret Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

Outline

Hard Problems

Factorization

Discret Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

Integer Factoring and RSA

→ Use of algorithmically hard problems.

Factorization

- ▶ $p, q \mapsto n = p \cdot q$ easy (quadratic)
- ▶ $n = p \cdot q \mapsto p, q$ difficult

RSA function $n = pq$, p and q primes.

e : public exponent

- ▶ $x \mapsto x^e \pmod n$ easy (cubic)
- ▶ $y = x^e \pmod n \mapsto x \pmod n$ difficult
 $x = y^d$ where $d = e^{-1} \pmod{\phi(n)}$

Complexity Estimates

Estimates for integer factoring Lenstra-Verheul 2000

Modulus (bits)	Operations (\log_2)
512	58
1024	80
2048	111
4096	149
8192	156

$\approx 2^{60}$ years

→ Can be used for RSA too.

Outline

Hard Problems

Factorization

Discret Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

The Discret Logarithm (DL)

Let $G = (\langle g \rangle, *)$ be any finite cyclic group of prime order.

Idea: it is hard for any adversary to produce x if he only knows g^x .

For any adversary \mathcal{A} ,

$$\mathbf{Adv}^{DL}(\mathcal{A}) = Pr \left[\mathcal{A}(g^x) \rightarrow x \mid x, y \stackrel{R}{\leftarrow} [1, q] \right]$$

is negligible.

Outline

Hard Problems

Factorization

Discret Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

The Diffie-Hellman Key Exchange Protocol

Let g be a generator of a cyclic group of prime order q .

$$A \rightarrow B : g^a$$

$$B \rightarrow A : g^b$$

$$A \rightarrow B : \{N\}_{g^{ab}}$$

Computational Diffie-Hellman (CDH)

Idea: it is hard for any adversary to produce g^{xy} if he only knows g^x and g^y .

For any adversary \mathcal{A} ,

$$\mathbf{Adv}^{CDH}(\mathcal{A}) = Pr\left[\mathcal{A}(g^x, g^y) \rightarrow g^{xy} \mid x, y \stackrel{R}{\leftarrow} [1, q]\right]$$

is negligible.

Decisional Diffie-Hellman (DDH)

Idea: Knowing g^x and g^y , it should be hard for any adversary to distinguish between g^{xy} and g^r for some random value r .

For any adversary \mathcal{A} , the advantage of \mathcal{A}

$$\begin{aligned} \mathbf{Adv}^{DDH}(\mathcal{A}) = & Pr\left[\mathcal{A}(g^x, g^y, g^{xy}) \rightarrow 1 \mid x, y \stackrel{R}{\leftarrow} [1, q]\right] \\ & - Pr\left[\mathcal{A}(g^x, g^y, g^r) \rightarrow 1 \mid x, y, r \stackrel{R}{\leftarrow} [1, q]\right] \end{aligned}$$

is negligible.

This means that an adversary cannot extract a single bit of information on g^{xy} from g^x and g^y .

Outline

Hard Problems

Factorization

Discret Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

Hard Problems

Most cryptographic constructions are based on *hard problems*.

Their security is proved by reduction to these problems:

- ▶ **Discrete Logarithm** problem, DL. Given a group $\langle g \rangle$ and g^x , compute x .
- ▶ **Computational Diffie-Hellman**, CDH Given a group $\langle g \rangle$, g^x and g^y , compute g^{xy} .
- ▶ **Decisional Diffie-Hellman**, DDH Given a group $\langle g \rangle$, distinguish between the distributions (g^x, g^y, g^{xy}) and (g^x, g^y, g^r) .
- ▶ **RSA**. Given $N = pq$ and $e \in \mathbb{Z}_{\varphi(N)}^*$, compute the inverse of e modulo $\varphi(N) = (p-1)(q-1)$. **Factorization**

$$DDH \leq CDH \leq DL$$

Relation between the problems

Exercise

$DL \Rightarrow CDH \Rightarrow DDH$.

Prop (Moaurer & Wolf)

For many groups, $DL \Leftrightarrow CDH$

Prop (Joux & Wolf)

There are groups for which DDH is easier than CDH .

Usage of DH assumption

The Diffie-Hellman problems are widely used in cryptography:

- ▶ Public key cryptosystems [ElGamal, Cramer& Shoup]
- ▶ Pseudo-random functions [Noar& Reingold, Canetti]
- ▶ Pseudo-random generators [Blum& Micali]
- ▶ (Group) key exchange protocols [many]

Outline

Hard Problems

Factorization

Discrete Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

Notion of Indistinguishability

Objects are considered to be computationally equivalent if they cannot be differentiated by any efficient procedure.

Hence, two distributions are said to be computationally indistinguishable if no efficient procedure can tell them apart.

Example with Distributions

Given an efficient algorithm D , we consider the probability that D accepts a string taken from the first distribution, and the probability for the second distribution. If these two probabilities are close, we say that D does not distinguish the two distributions.

Concrete Example (I)

Consider that in Box 1 there are 9 blue numerated balls and in Box 2 there are 9 red numerated balls, with uniform distributions.

Alice picks one ball into one of the two boxes and says the number of the ball.

Where did Alice pick the ball?

$$|Pr[A(Box1|Number) = 1] - Pr[A(Box2|Number) = 1]|$$

is negligible.

Concrete Example (II)

Consider now that Alice has $1/2$ probability to pick ball number 1 between the red balls and $1/16$ for the others $(2, \dots, 9)$.

Hence an adversary has a non negligible advantage to know which Box the ball comes from.

Medical Issue

Consider two sets of patients following two indistinguishable distributions of probability. We give in similar conditions to the first set a new medicine and only water to the second set.

If the results are significant then the treatment is efficient, i.e., the probability of distribution for the results with medicine is distinguishable from the fictive one.

Cryptographic Issue

For a perfect encryption scheme we wish:

$$|Pr[Enc(1) = 1] - Pr[Enc(0) = 1]|$$

is negligible.

Outline

Hard Problems

Factorization

Discret Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

Which adversary?



Adversary Model

Qualities of the adversary:

- ▶ **Clever:** Can perform all operations he wants
- ▶ **Limited time:**
 - ▶ Do not consider attack in 2^{60} .
 - ▶ Otherwise a Brute force by enumeration is always possible.

Model used: **Any Turing Machine.**

- ▶ Represents all possible algorithms.
- ▶ Probabilistic: adversary can generate keys, random number...

Adversary Models

The adversary is given access to oracles :

- encryption of all messages of his choice
- decryption of all messages of his choice

Three classical security levels:

- ▶ Chosen-Plain-text Attacks (CPA)
- ▶ Non adaptive Chosen-Cipher-text Attacks (CCA1)
only before the challenge
- ▶ Adaptive Chosen-Cipher-text Attacks (CCA2)
unlimited access to the oracle (except for the challenge)



Chosen-Plain-text Attacks (CPA)



Adversary can obtain all cipher-texts from any plain-texts.
It is always the case with a Public Encryption scheme.

Non adaptive Chosen-Cipher-text Attacks (CCA1)



Adversary knows the public key, has access to a **decryption oracle multiple times before to get the challenge** (cipher-text), also called “Lunchtime Attack” introduced by M. Naor and M. Yung ([NY90]).

Adaptive Chosen-Cipher-text Attacks (CCA2)



Adversary knows the public key, has access to a **decryption oracle multiple times before and AFTER to get the challenge**, but of course cannot decrypt the challenge (cipher-text) introduced by C. Rackoff and D. Simon ([RS92]).

Summary of Adversaries

CCA2: $\mathcal{O}_1 = \mathcal{O}_2 = \{\mathcal{D}\}$ Adaptive Chosen Cipher text Attack



CCA1: $\mathcal{O}_1 = \{\mathcal{D}\}$, $\mathcal{O}_2 = \emptyset$ Non-adaptive Chosen Ciphertext Attack



CPA: $\mathcal{O}_1 = \mathcal{O}_2 = \emptyset$ Chosen Plain text Attack



Outline

Hard Problems

Factorization

Discret Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

Symmetric key and public key encryption

- Symmetric key encryption



- Public key encryption



One-Wayness (OW)

Put your message in a translucent bag, but you cannot read the text.



One-Wayness (OW)

Put your message in a translucent bag, but you cannot read the text.



Without the private key, it is computationally **impossible to recover the plain-text.**

Is it secure ?



Is it secure ?



Is it secure ?



- ▶ you cannot read the text but you can distinguish which one has been encrypted.

Is it secure ?



- ▶ you cannot read the text but you can distinguish which one has been encrypted.
- ▶ Does not exclude to recover half of the plain-text
- ▶ Even worse if one has already partial information of the message:
 - ▶ Subject: XXXX
 - ▶ From: XXXX

Indistinguishability (IND)

Put your message in a black bag, you can not read anything.



Now a black bag is of course IND and it implies OW.

Indistinguishability (IND)

Put your message in a black bag, you can not read anything.



Now a black bag is of course IND and it implies OW.
The adversary is not able to **guess in polynomial-time even a bit of the plain-text knowing the cipher-text**, notion introduced by S. Goldwasser and S.Micali ([GM84]).

Is it secure?



Is it secure?



Is it secure?



- ▶ It is possible to scramble it in order to produce a new cipher. In more you know the relation between the two plain text because you know the moves you have done.

Non Malleability (NM)

Put your message in a black box.



But in a black box you cannot touch the cube (message), hence NM implies IND.

Non Malleability (NM)

Put your message in a black box.



But in a black box you cannot touch the cube (message), hence NM implies IND.

The adversary should **not be able to produce a new cipher-text** such that the plain-texts are meaningfully related, notion introduced by D. Dolev, C. Dwork and M. Naor in 1991 ([DDN91,BDPR98,BS99]).

Summary of Security Notions

Non Malleability



Indistinguishability



One-Wayness



Outline

Hard Problems

Factorization

Discret Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

Asymmetric Encryption

An asymmetric encryption scheme $\mathcal{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

- ▶ \mathcal{K} : **key generation**
- ▶ \mathcal{E} : **encryption**
- ▶ \mathcal{D} : **decryption**

$$\mathcal{K}(\eta) = (k_e, k_d)$$

$$\mathcal{E}_{k_e}(m, r) = c$$

$$\mathcal{D}(c, k_d) = m$$

One-Wayness (OW)

Adversary \mathcal{A} : any polynomial time Turing Machine (PPTM)

Basic security notion: One-Wayness (OW)



Without the private key, it is computationally impossible to recover the plain text:

$$\Pr_{m,r}[\mathcal{A}(c) = m \mid c = E(m, r)]$$

is negligible.

Indistinguishability (IND)



Game Adversary: $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

1. The adversary \mathcal{A}_1 is given the public key pk .
2. The adversary \mathcal{A}_1 chooses two messages m_0, m_1 .
3. $b = 0, 1$ is chosen at random and $c = E(m_b)$ is given to the adversary.
4. The adversary \mathcal{A}_2 answers b' .

The probability $Pr[b = b'] - \frac{1}{2}$ should be negligible.

The IND-CPA Games



Given an encryption scheme $\mathcal{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. An adversary is a pair $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ of polynomial-time probabilistic algorithms, $b \in \{0, 1\}$.

Let $\text{IND}_{CPA}^b(\mathcal{A})$ be the following algorithm:

- ▶ Generate $(pk, sk) \xleftarrow{R} \mathcal{K}(\eta)$.
- ▶ $(s, m_0, m_1) \xleftarrow{R} \mathcal{A}_1(\eta, pk)$
- ▶ $b' \xleftarrow{R} \mathcal{A}_2(\eta, pk, s, \mathcal{E}(pk, m_b))$
- ▶ return b' .

Then, we define the advantage against the IND-CPA game by:

$$\text{ADV}_{\mathcal{S}, \mathcal{A}}^{\text{IND}_{CPA}}(\eta) = \\ \Pr[b' \xleftarrow{R} \text{IND}_{CPA}^1(\mathcal{A}) : b' = 1] - \Pr[b' \xleftarrow{R} \text{IND}_{CPA}^0(\mathcal{A}) : b' = 1]$$

The IND-CCA1 Games



Given an encryption scheme $\mathcal{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. An adversary is a pair $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ of polynomial-time probabilistic algorithms, $b \in \{0, 1\}$.

Let $\text{IND}_{\text{CCA1}}^b(\mathcal{A})$ be the following algorithm:

- ▶ Generate $(pk, sk) \xleftarrow{R} \mathcal{K}(\eta)$.
- ▶ $(s, m_0, m_1) \xleftarrow{R} \mathcal{A}_1^{\mathcal{O}_1}(\eta, pk)$ **where** $\mathcal{O}_1 = \mathcal{D}$
- ▶ $b' \xleftarrow{R} \mathcal{A}_2(\eta, pk, s, \mathcal{E}(pk, m_b))$
- ▶ return b' .

Then, we define the advantage against the IND-CCA1 game by:

$$\text{ADV}_{\mathcal{S}, \mathcal{A}}^{\text{IND}_{\text{CCA1}}}(\eta) = \\ \Pr[b' \xleftarrow{R} \text{IND}_{\text{CCA1}}^1(\mathcal{A}) : b' = 1] - \Pr[b' \xleftarrow{R} \text{IND}_{\text{CCA1}}^0(\mathcal{A}) : b' = 1]$$

The IND-CCA2 Games



Given an encryption scheme $\mathcal{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. An adversary is a pair $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ of polynomial-time probabilistic algorithms, $b \in \{0, 1\}$.

Let $\text{IND}_{\text{CCA2}}^b(\mathcal{A})$ be the following algorithm:

- ▶ Generate $(pk, sk) \xleftarrow{R} \mathcal{K}(\eta)$.
- ▶ $(s, m_0, m_1) \xleftarrow{R} \mathcal{A}_1^{\mathcal{O}_1}(\eta, pk)$ where $\mathcal{O}_1 = \mathcal{D}$
- ▶ $b' \xleftarrow{R} \mathcal{A}_2^{\mathcal{O}_2}(\eta, pk, s, \mathcal{E}(pk, m_b))$ **where** $\mathcal{O}_2 = \mathcal{D}$
- ▶ return b' .

Then, we define the advantage against the IND-CCA2 game by:

$$\text{ADV}_{\mathcal{S}, \mathcal{A}}^{\text{IND}_{\text{CCA2}}(\eta)} = \Pr[b' \xleftarrow{R} \text{IND}_{\text{CCA2}}^1(\mathcal{A}) : b' = 1] - \Pr[b' \xleftarrow{R} \text{IND}_{\text{CCA2}}^0(\mathcal{A}) : b' = 1]$$

IND-XXX Security



Definition

An encryption scheme is *IND-XXX secure*, if for any adversary \mathcal{A} the function $\text{ADV}_{S,\mathcal{A}}^{\text{IND-XXX}}$ is negligible.

Exercise

Prove that

$$\begin{aligned}\text{ADV}_{S,\mathcal{A}}^{\text{IND}^{\text{XXX}}}(\eta) &= \Pr[b' \stackrel{R}{\leftarrow} \text{IND}^1(\mathcal{A}) : b' = 1] \\ &\quad - \Pr[b' \stackrel{R}{\leftarrow} \text{IND}^0(\mathcal{A}) : b' = 1] \\ &= 2\Pr[b' \stackrel{R}{\leftarrow} \text{IND}^b(\mathcal{A}) : b' = b] - 1\end{aligned}$$

Summary



Given $\mathcal{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\text{IND}_{\text{XXX}}^b(\mathcal{A})$ follows:

- ▶ Generate $(pk, sk) \xleftarrow{R} \mathcal{K}(\eta)$.
- ▶ $(s, m_0, m_1) \xleftarrow{R} \mathcal{A}_1^{\mathcal{O}_1}(\eta, pk)$
- ▶ $b' \xleftarrow{R} \mathcal{A}_2^{\mathcal{O}_2}(\eta, pk, s, \mathcal{E}(pk, m_b))$
- ▶ return b' .

$$\text{ADV}_{\mathcal{S}, \mathcal{A}}^{\text{IND}_{\text{XXX}}}(\eta) = \\ \Pr[b' \xleftarrow{R} \text{IND}_{\text{XXX}}^1(\mathcal{A}) : b' = 1] - \Pr[b' \xleftarrow{R} \text{IND}_{\text{XXX}}^0(\mathcal{A}) : b' = 1]$$



IND-CPA: $\mathcal{O}_1 = \mathcal{O}_2 = \emptyset$ Chosen Plain text Attack

IND-CCA1: $\mathcal{O}_1 = \{\mathcal{D}\}$, $\mathcal{O}_2 = \emptyset$ Non-adaptive Chosen Cipher text Attack

IND-CCA2: $\mathcal{O}_1 = \mathcal{O}_2 = \{\mathcal{D}\}$ Adaptive Chosen Cipher text Attack.

Definition of Non Malleability



Game Adversary: $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

1. The adversary \mathcal{A}_1 is given the public key pk .
2. The adversary \mathcal{A}_1 chooses a message space M .
3. Two messages m and m^* are chosen at random in M and $c = E(m; r)$ is given to the adversary.
4. The adversary \mathcal{A}_2 outputs a binary relation R and a cipher-text c' .

Probability $Pr[R(m, m')]$ – $Pr[R(m, m^*)]$ is negligible,
where $m' = \mathcal{D}(c')$

Non-Malleability - XXX

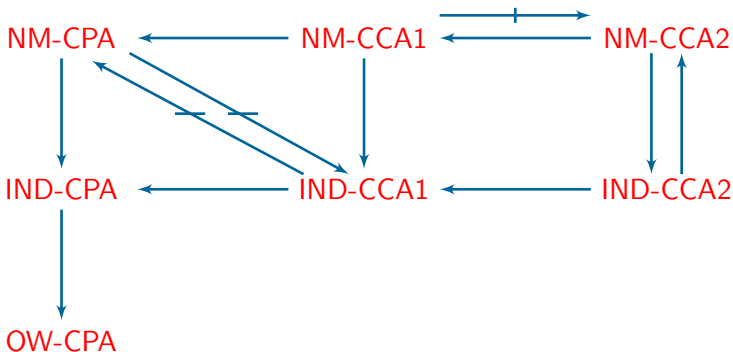


- ▶ Let $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and $A = (A_1, A_2)$.
- ▶ For $b \in \{0, 1\}$ we define the experiment $\mathbf{Exp}_{\mathcal{PE}, A}^{atk-b}(k)$:
 - $(pk, sk) \leftarrow \mathcal{K}(k)$; $(M, s) \leftarrow A_1^{O_1(\cdot)}(pk)$; $x_0, x_1 \leftarrow M$
 - $y \leftarrow \mathcal{E}_{pk}(x_b)$; $(\mathcal{R}, \vec{y}) \leftarrow A_2^{O_2(\cdot)}(M, s, y)$; $\vec{x} \leftarrow \mathcal{D}_{pk}(\vec{y})$;
 - If $y \notin \vec{y} \wedge \perp \notin \vec{x} \wedge \mathcal{R}(x_b, \vec{x})$ then $d \leftarrow 1$ else $d \leftarrow 0$
 - Return d
- ▶ For $atk \in \{cpa, cca1, cca2\}$ and $k \in \mathbb{N}$, the advantage

$$\mathbf{Adv}_{\mathcal{PE}, A}^{atk}(k) = Pr \left[\mathbf{Exp}_{\mathcal{PE}, A}^{atk-1}(k) = 1 \right] - Pr \left[\mathbf{Exp}_{\mathcal{PE}, A}^{atk-0}(k) = 1 \right]$$

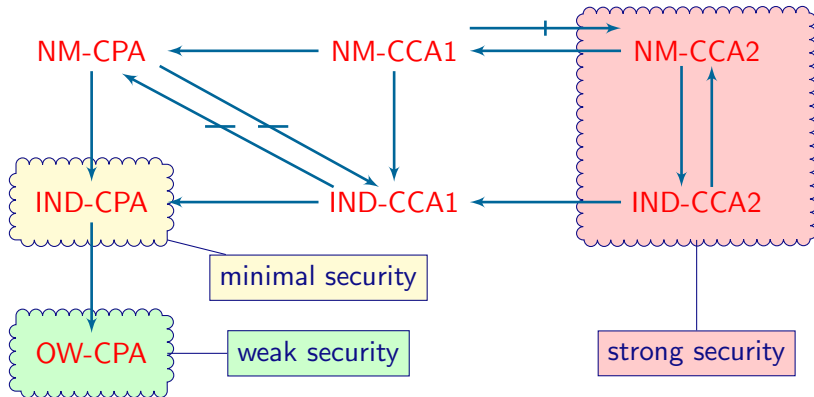
has to be negligible for \mathcal{PE} to be considered secure, assuming A , M and \mathcal{R} can be computed in time $p(k)$.

Relations



“Relations Among Notions of Security for Public-Key Encryption Schemes”, **Crypto’98**, by Mihir Bellare, Anand Desai, David Pointcheval and Phillip Rogaway [BDPR’98]

Relations



"Relations Among Notions of Security for Public-Key Encryption Schemes", **Crypto'98**, by Mihir Bellare, Anand Desai, David Pointcheval and Phillip Rogaway [BDPR'98]

Outline

Hard Problems

Factorization

Discret Logarithm

Diffie-Hellman

Summary

Intuition of Indistinguishability

Different Adversaries

Intuition of Computational Security

Definitions of Computational Security

Conclusion

Summary of Today

Today

- ▶ DL, DDH, CDH
- ▶ Indistinguishability
- ▶ Adversary: CPA, CCA1, CCA2
- ▶ Security Notions: OW, IND, NM

Exercices TODO

Today

- ▶ DL, DDH, CDH
- ▶ Elgamal OW and IND-CPA

Next Time

- ▶ Reduction Proof
- ▶ Hybrid Argument

Thank you for your attention.

Questions ?