ICS - Information Security Systems
Lecture 2: Security Formalism

2025-2026

Outline

Red phone

The One-Time Pad

Inputs/Outputs :

- Message $m \in \{0,1\}^\ell$ (plaintext), message space : $\mathcal{M} = \{0,1\}^\ell$
- Key $k \in \{0,1\}^\ell$, key space : $\mathcal{K} = \{0,1\}^\ell$
- Ciphertext $c \in \{0,1\}^\ell$, ciphertext space : $\mathcal{C} = \{0,1\}^\ell$

Algorithms :

- Encryption : $Enc_k(m) = m \oplus k$
- Decryption : $Dec_k(c) = c \oplus k$
- Correctness : $Dec_k(Enc_k(m)) = (m \oplus k) \oplus k = m$

Advantages :

- Used during the Cold War
- Suitable for short messages/secrets
- Perfectly secure (information-theoretic security)

Drawbacks :

- Key must be as long as the message
- Can only be used once
- Key must be uniformly random

Perfect Indistinguishability (*i.e.* Information-Theoretic or Unconditional Security)

Key Idea

The ciphertext provides **no additional information** about the message to an adversary.

Indistinguishability Game for an Encryption Scheme Enc :

- Adversary : chooses two messages $m_0, m_1 \in \mathcal{M}$
- Challenger : picks $k \leftarrow_{\$} \mathcal{K}$ and $b \leftarrow_{\$} \{0, 1\}$, then computes $c \leftarrow Enc_k(m_b)$
- Adversary : receives c and outputs $\hat{b}$ (attempts to guess b)

Definition : Enc is **perfectly indistinguishable** if, for any adversarial strategy :

$$\Pr[\,\hat{b} = b\,] = \frac{1}{2}$$

(probability taken over the random choices of k, b, and all randomness used by the adversary)

Proof of the Theorem

An encryption system is said to be **perfect** if the knowledge of a ciphertext gives no information about the plaintext, even to an adversary with **unlimited computational resources**.

Theorem : For the One-Time Pad, for any adversary $\mathcal{A}$,

$$\Pr[\, \hat{b} = b \,] = \frac{1}{2}.$$

Proof of the Theorem

Theorem Reminder :

$$\text{For any strategy of } \mathcal{A}, \quad \Pr\left[\hat{b} = b\right] = \tfrac{1}{2}$$

Proof : Lemma : Let $m \in \mathcal{M}$ and $c \in \mathcal{C}$. If $k \leftarrow \mathcal{K}$, then :

$$\Pr\left[m \oplus k = c\right] = \Pr\left[k = m \oplus c\right] = \frac{1}{2^\ell}$$

Law of Total Probability :

$$\Pr[\hat{b} = b] = \Pr[\hat{b} = 0 \mid b = 0]\Pr[b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\Pr[b = 1]$$

where $\Pr[b = 0] = \Pr[b = 1] = \frac{1}{2}$.

Suppose $\mathcal{A}$ is deterministic : it partitions ciphertexts into two sets C_0 ($\mathcal{A} \to 0$) and C_1 ($\mathcal{A} \to 1$) depending on $\mathcal{A}$'s output.

$$\Pr[\hat{b} = 0 \mid b = 0] = \frac{\#C_0}{2^\ell}, \qquad \Pr[\hat{b} = 1 \mid b = 1] = \frac{\#C_1}{2^\ell}$$

Proof of the Theorem

Proof (Randomised Adversary) :

- Suppose now that $\mathcal{A}$ is randomised, i.e. it uses random bits $r \in \{0,1\}^*$.
- For each choice of r, we obtain a deterministic algorithm $\mathcal{A}_r$.

$$\Pr_{k,b,r}\left[\hat{b} = b\right] = \sum_{r \in \{0,1\}^*} \Pr\left[\hat{b} = b \mid \mathcal{A} \text{ uses } r\right] \Pr\left[\mathcal{A} \text{ uses } r\right]$$

- For each fixed r, $\mathcal{A}_r$ is deterministic :

$$\Pr\left[\hat{b} = b \mid \mathcal{A} \text{ uses } r\right] = \frac{1}{2}$$

hence :

$$\Pr_{k,b,r}\left[\hat{b} = b\right] = \frac{1}{2} \sum_{r \in \{0,1\}^*} \Pr\left[\mathcal{A} \text{ uses } r\right] = \frac{1}{2}$$

Another Example of Unconditional (or Perfect) Security

A key k is drawn uniformly at random from the key space $\mathcal{K}$, denoted $k \xleftarrow{\$} \mathcal{K}$.

Given an encryption function Enc, a decryption function Dec, and a message m.

An encryption scheme is secure if, for all $(m_0, m_1) \in \mathcal{M}^2$:

$$\Pr[\mathrm{Enc}_k(m_0) = c] = \Pr[\mathrm{Enc}_k(m_1) = c]$$

If the message space $\mathcal{M}$ is the same as $\mathcal{K}$, we can encrypt [1] with :

$$\mathrm{Enc}_k \colon m \mapsto m \cdot k$$

$$\mathrm{Dec}_k \colon c \mapsto c \cdot k^{-1}$$

1. usable once per key

Perfect Indistinguishability Implies No Information Leakage

Examples : Given $c \leftarrow \mathrm{Enc}_k(m)$, an adversary cannot learn :

- the least significant bit $m[0]$ of m
- the parity $\bigoplus_{i=0}^{\ell-1} m[i]$ of m
- whether m contains more 1s than 0s
- etc.

Equivalent Definition : The distribution of m does not depend on c.

$$\forall m^* \in \mathcal{M}, \forall c^* \in \mathcal{C} \text{ such that } \Pr[c = c^*] > 0, \qquad \Pr_{m,k}[m = m^* \mid c = c^*] = \Pr_{m,c,k}[m = m^*]$$

Limits of Perfect Indistinguishability : Shannon's Theorem

Theorem : A perfectly indistinguishable encryption scheme must satisfy :

1. $\#\mathcal{K} \geq \#\mathcal{M}$
2. If $\#\mathcal{K} = \#\mathcal{M}$, then k must be uniformly chosen in $\mathcal{K}$

Proof of (i) :

- Suppose $\#\mathcal{K} < \#\mathcal{M}$ and construct an adversary $\mathcal{A}$ such that $\Pr[\hat{b} = b] > \frac{1}{2}$
- For each $c \in \mathcal{C}$, define $\mathcal{M}_c = \{m \in \mathcal{M} \mid \exists k,\ \mathrm{Dec}_k(c) = m\}$. Since Dec is deterministic, $\#\mathcal{M}_c \leq \#\mathcal{K} < \#\mathcal{M}$
- Pick $c^* \in \mathcal{C}$ with $\mathcal{M}_{c^*} \neq \varnothing$ and $m_0 \notin \mathcal{M}_{c^*}$
- Define $\mathcal{A}(c)$ as :
    - 0 if $c = c^*$ (certain that $m \neq m_0$)
    - a random bit otherwise
- Then $\Pr[\hat{b} = b] > 1/2$, contradiction.

Conclusion on Vernam Encryption (OTP)

- One-Time Pad : perfectly indistinguishable but...
- ... perfect indistinguishability is impossible with a short key

Relaxing the Notion of Security :

- Allow some information leakage *statistical security*
- Limit the adversary's computational power *computational security*

Another Issue : malleability

$$c = m \oplus k \quad \Rightarrow \quad c \oplus m' = (m \oplus m') \oplus k$$

The adversary can modify any message.

Outline

What is Encryption ?

Definition (Asymmetric Encryption)

An *asymmetric encryption scheme* $\mathcal{E}$ is a set of probabilistic polynomial-time (PPT) algorithms consisting of :

$\mathsf{KeyGen}(1^\lambda)$ **:** a PPT algorithm that takes as input a security parameter λ, and outputs a key pair $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{Enc}(\mathsf{pk}, m)$ **:** a PPT algorithm that computes and returns a *ciphertext c* of the message m using the public key pk.

$\mathsf{Dec}(\mathsf{sk}, c)$ **:** a deterministic polynomial-time algorithm that takes as input a secret key sk and a ciphertext c, and returns the *plaintext message m*.

We require that $\mathcal{E}$ satisfies ...

How Do We Know if It's Secure ?

How can we define the confidentiality of an encryption scheme ?

How Do We Know if It’s Secure ?

How can we define the confidentiality of an encryption scheme ?

It must be difficult for an attacker (**what kind of attacker ?**) to gain information (**what kind of information ?**) from a ciphertext.

Adversary Model

Characteristics of the adversary :

- **Clever & capable** : Can perform any operation they wish.
- **Time-limited** :
  - Limited to fewer than 2^{128} computations, for instance.
  - Otherwise, a brute-force attack by enumeration is always possible.

Model used : **Any Turing Machine**.

- Represents all possible algorithms.
- Probabilistic : the adversary can generate keys, random numbers, etc.

Principles of Modern Cryptography

Formal Definitions

- What does it mean for an encryption scheme to be **secure**?
  (*good definition*) Whatever information an adversary already has about the message, the encryption gives them only *very little* additional information.
- What is an **adversary**?
  - ~~Ciphertext-only attack (COA)~~
  - ~~Known-plaintext attack (KPA)~~
  - Chosen-plaintext attack (CPA)
  - Chosen-ciphertext attack (CCA)

Specific Assumptions

- Adversary's computational power (complexity theory)
- Validity and comparison of assumptions, and which ones are *necessary*

Provable Security *Proving that a protocol satisfies a **security definition**, under given **assumptions**.*

Recap : Perfect Indistinguishability

Reminder : *perfect indistinguishability for Enc using a game*

Adversary : chooses two messages $m_0, m_1 \in \mathcal{M}$

Challenge : samples $k \xleftarrow{\$} \mathcal{K}$, $b \xleftarrow{\$} \{0,1\}$ and computes $c \leftarrow \mathrm{Enc}_k(m_b)$

Adversary : outputs a bit $\hat{b}$ (*tries to guess* b)

Enc is perfectly indistinguishable if $\Pr[\hat{b} = b] = \frac{1}{2}$

Characteristics

- Weak adversary model :
  - The adversary only sees the ciphertext c
  - Even with access to many ciphertexts, no difference !
  - But not KPA $\Rightarrow$ a single pair $(m, \mathrm{Enc}_k(m))$ reveals k *(one-time pad)*
- Strong guarantees : no matter how powerful the adversary, they learn nothing

$\Longrightarrow$ Need for a stronger adversary model.

IND-CPA : Indistinguishability under Chosen-Plaintext Attack

IND-CPA Game for Encryption

Setup : sample $k \xleftarrow{\$} \mathcal{K}$

Adversary : has oracle access : for each query x_i, it receives $c_i \leftarrow \mathrm{Enc}_k(x_i)$

Adversary : chooses two messages $m_0, m_1 \in \mathcal{M}$ of equal length

Challenge : samples $b \xleftarrow{\$} \{0, 1\}$ and computes $c \leftarrow \mathrm{Enc}_k(m_b)$

Adversary : outputs a bit $\hat{b}$ (*tries to guess* b)

Remarks

- The adversary can query $\mathrm{Enc}_k(m_0)$ and $\mathrm{Enc}_k(m_1)$
  - $\mathrm{Enc}_k(\cdot)$ must therefore be **randomised**
- The messages must have the same length
  - IND-CPA encryption may reveal message length
  - Generally necessary for efficiency
  - *Ad-hoc* solutions exist if length is sensitive information

Relaxing Guarantees : IND-CPA Advantage

What is the **advantage** of an adversary $\mathcal{A}$ compared to random guessing of $\hat{b}$?

Adversary's Advantage $\mathcal{A}$

$$\mathsf{Adv}^{\text{IND-CPA}}_{\text{Enc}}(\mathcal{A}) = 2\left|\Pr\left[\mathcal{A}^{\text{Enc}_k} \to b\right] - \frac{1}{2}\right|$$

Remarks

- The advantage is between 0 (perfectly indistinguishable) and 1
- $\mathsf{Adv}^{\text{IND-CPA}}_{\text{Enc}}(\mathcal{A}) = \left|\Pr\left[\mathcal{A}^{\text{Enc}_k} \to 1 \mid b = 1\right] - \Pr\left[\mathcal{A}^{\text{Enc}_k} \to 1 \mid b = 0\right]\right|$

IND-CPA Security

An adversary with **bounded resources** has a **negligible advantage.**

Asymptotic Security *definition from complexity theory*

- Fix a *security parameter* n
- Bounded resources : probabilistic polynomial-time adversaries $\mathcal{A}$ in n
- Negligible advantage : if $< \frac{1}{p(n)}$ for any polynomial p

Concrete Security *used in this course*

- **Advantage function** : $\mathrm{Adv}_{\mathrm{Enc}}^{\text{IND-CPA}}(q,t) = \max_{\mathcal{A}_{q,t}} \mathrm{Adv}_{\mathrm{Enc}}^{\text{IND-CPA}}(\mathcal{A}_{q,t})$ where $\mathcal{A}_{q,t}$ ranges over all probabilistic algorithms running in time $\leq t$ and making $\leq q$ queries
- No formal definition of *bounded resources* or *negligible* :

Orders of Magnitude (Time)

Computation Time

- $t \simeq 2^{40}$: ~ 1 day on my laptop
- $t \simeq 2^{60}$: feasible on a large CPU/GPU cluster *academic research*
- $t \simeq 2^{80}$: feasible with an ASIC cluster *Bitcoin mining*
- $t \simeq 2^{128}$: seems sufficiently hard

Example : performing 2^{128} operations in 34 years ($\simeq 2^{30}$ seconds)

- **Assumptions :**
  - Hardware at 2^{50} ops/s *quite fast*
  - Highly parallelisable *not always true*
  - 1000 W per device *reasonable*
- **Results :**
  - Requires $\simeq \frac{2^{128}}{2^{50} \cdot 2^{30}} = 2^{48}$ machines $> 280 \cdot 10^{12}$
  - Requires ~ 280,000 TW $> 1.7 \cdot 10^{9}$ nuclear power plants

Orders of Magnitude (Probabilities)

Probabilities

- $p = \frac{1}{2}$: getting heads with a fair coin
- $p = \frac{1}{6}$: rolling a 6 with a fair die
- $p \simeq 2^{-24}$: probability of winning the French national lottery
- $p \simeq 2^{-72}$: probability of winning the French lottery three times in a row

Example :

- An attack taking 1 second and succeeding with probability 2^{-60} would have been expected to succeed fewer than once since the Big Bang
- CPU errors due to cosmic rays occur with far higher probability !

Combining Orders of Magnitude. If $\mathrm{Adv}_{\mathrm{Enc}}^{\text{IND-CPA}}(2^{128}) < 2^{-60}$, then the encryption can be considered (IND-CPA) secure.

Outline

Fonction négligeable

A fonction $\epsilon\colon \mathbb{N} \to \mathbb{R}^+$ is *negligible*, if

$$\forall c > 0, \exists k_0 \in \mathbb{N}, \forall k > k_0, |\epsilon(k)| < \frac{1}{|k^c|}.$$

Properties

Let f and g be two negligible functions, then

1. $f.g$ is negligible.
2. For any $k > 0$, f^k is negligible.
3. For any λ, μ in $\mathbb{R}$, $\lambda.f + \mu.g$ is negligible.

Exercise : Proofs

Noticeable Functions

Instead of "there exists an N such that for all $n > N$ " we will in the following often say "for all sufficiently large n".

We call a function $\nu : \mathbb{N} \to \mathbb{R}$ noticeable if there exists a positive polynomial p such that for all sufficiently large n, we have :

$$\nu(n) > \frac{1}{p(n)}$$

Note : A function can be neither noticeable nor negligible.

Exercises

Prove or disprove the following statements :

1. If both $f, g \geq 0$ are noticeable, then $f - g$ and $f + g$ are noticeable.
2. If both $f, g \geq 0$ are not noticeable, then $f - g$ is not noticeable.
3. If both $f, g \geq 0$ are not noticeable, then $f + g$ is not noticeable.
4. If $f \geq 0$ is noticeable, and $g \geq 0$ is negligible, then $f.g$ is negligible.
5. If both $f, g > 0$ are negligible, then f/g is noticeable.

Exercise : Prove or disprove :

- The function $f(n) := (\frac{1}{2})^n$ is negligible.
- The function $f(n) := 2^{-\sqrt{n}}$ is negligible.
- The function $f(n) := n^{-log(n)}$ is negligible.

Outline

How to prove the security ?

Theorem

A cryptosystem C has a security property P under a hypothesis H

$$H \Rightarrow C \text{ has } P$$

$$(A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow \neg A)$$

$$[H \Rightarrow C \text{ has } P] \Leftrightarrow [\neg(C \text{ has } P) \Rightarrow \neg H]$$

Proof by Reduction

1. Assume that there exists an adversary A that breaks the security property of C.
2. Construct an adversary B that uses A to breaks the hypothesis H in a polynomial time.

Modelling an Adversary

Let $\mathcal{A}$ be an algorithm that attempts to decrypt encrypted messages using randomly generated keys, *i.e.*,

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda).$$

An encryption algorithm $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc})$ is said to be practically/computationally secure if, for any $\mathcal{A}$ capable of decrypting a message, its success probability is negligible in the key size λ.

Formally : Let $\mathsf{Adv}_{\mathcal{A}}(\mathcal{E}, \lambda)$ denote the probability of obtaining information about a message. We have :

$$\mathsf{Adv}_{\mathcal{E},\mathcal{A}}(\lambda) \leq \epsilon(\lambda).$$

Outline

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)
  Decryption oracle only before the challenge

- Adaptive Chosen-Cipher-text Attacks (CCA2)
  unlimited access to the decryption oracle (except for the challenge)

Other Attack Scenarios : Attacker's Goal

- **Non-Malleability (NM) :** It is impossible to transform a ciphertext of a message m into a ciphertext of a related message $f(m)$ for some known function f.

Other Attack Scenarios : Attacker's Goal

- **Non-Malleability (NM) :** It is impossible to transform a ciphertext of a message m into a ciphertext of a related message $f(m)$ for some known function f.
- **Indistinguishability (IND) :** It is impossible to distinguish a ciphertext of a message m from a ciphertext of another message m'.

Other Attack Scenarios : Attacker's Goal

- **Non-Malleability (NM) :** It is impossible to transform a ciphertext of a message m into a ciphertext of a related message $f(m)$ for some known function f.
- **Indistinguishability (IND) :** It is impossible to distinguish a ciphertext of a message m from a ciphertext of another message m'.
- **One-Way (OW) :** It is impossible to recover the encrypted message.

Is OW Insufficient ?

"One-wayness" means that the attacker cannot recover the entire message. But they might still recover half of it !

Is OW Insufficient ?

"One-wayness" means that the attacker cannot recover the entire message. But they might still recover half of it !

Let's take a concrete example :

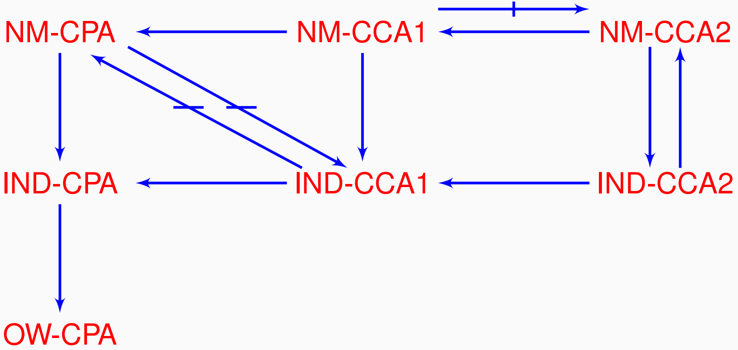
The message can no longer be read (one-wayness).

However, one can still tell whether the paper is white, red, green, etc.

We gain one bit of information about the paper — and maybe that information is crucial !

Relations

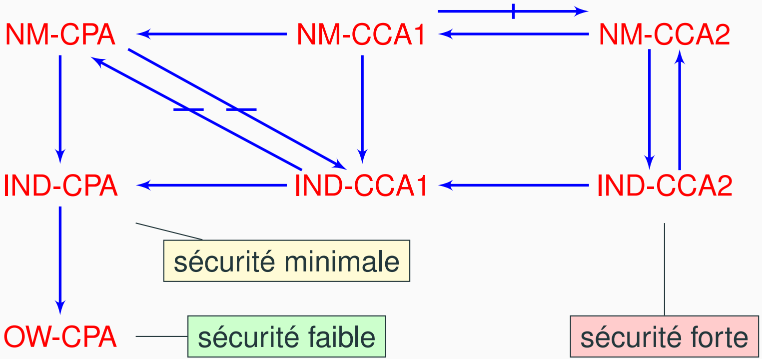
CCA2 $\Rightarrow$ CCA resistance $\Rightarrow$ CPA resistance $\Rightarrow$ KPA.



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

Relations

$\boxed{\text{CCA2} \Rightarrow \text{CCA resistance} \Rightarrow \text{CPA resistance} \Rightarrow \text{KPA.}}$



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

Outline

Security

IND-CPA security

Negligible Functions

Reduction Proof

Different Adversaries

Security Assumptions

The Diffie-Hellman protocol

g, p are public parameters.

- Diffie chooses x and computes $g^x \mod p$
- Diffie sends $g^x \mod p$
- Hellman chooses y and computes $g^y \mod p$
- Hellman sends $g^x \mod p$

Shared key : $(g^x)^y = g^{xy} = (g^y)^x$

Basic Diffie-Hellman key-exchange : initiator I and responder R exchange public "half-keys" to arrive at mutual session key $\mathrm{k} = \mathrm{g}^{\mathrm{xy}} \mod \mathrm{p}$.

Hard Problems

Most cryptographic constructions are based on ***hard problems***. Their security is proved by reduction to these problems :

- **RSA-OAEP.** Given $N = pq$ and $e \in \mathbb{Z}^*_{\varphi(N)}$, compute the inverse of e modulo $\varphi(N) = (p-1)(q-1)$. **Factorization**
- **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).

The Discrete 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 A,

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

is negligible.

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 A,

$$\mathbf{Adv}^{CDH}(A) = Pr\left[A(g^x, g^y) \rightarrow g^{xy} \middle| x, y \xleftarrow{R} [1, q]\right]$$

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 A, the advantage of A

$$\mathbf{Adv}^{DDH}(A) = Pr\left[A(g^x, g^y, g^{xy}) \to 1 \middle| x, y \xleftarrow{R} [1, q]\right]$$
$$-Pr\left[A(g^x, g^y, g^r) \to 1 \middle| x, y, r \xleftarrow{R} [1, q]\right]$$

is negligible.

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

Relation between the problems

Proposition

Solve DL $\Longrightarrow$ Solve CDH $\Longrightarrow$ Solve DDH. (Exercise)

Moaurer & Wolf

For many groups, $DL \Longleftrightarrow CDH$

Joux & Wolf

There are groups for which DDH is easier than CDH.

Conclusion

One-Time Pad

- First example of an encryption scheme
- Very strong security… but in a very weak model !
- Practically unusable

Computational Security

- Game + advantage $\rightarrow$ notion of security
- Various security models depending on the experiment
    - Define the goals and the capabilities

What's Next ?

- Symmetric and public-key encryption
- Authentication and integrity