Security Models
Lecture 1
Security Notions

Pascal Lafourcade

2020-2021

Outline

1. Negligible Functions
2. Diffie-Hellman
3. Reduction Proof
4. Different Adversaries
5. Intuition of Computational Security
6. Definitions of Computational Security
7. Conclusion

Negligible functions

We call a function $\mu : \mathbb{N} \to \mathbb{R}^+$ negligible if for every positive polynomial p there exists an N such that for all $n > N$

$$\mu(n) < \frac{1}{p(n)}$$

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

Negligible Functions

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 $f(n) := n^{-logn}$ is negligible.

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.

Outline

1. Negligible Functions
2. Diffie-Hellman
3. Reduction Proof
4. Different Adversaries
5. Intuition of Computational Security
6. Definitions of Computational Security
7. Conclusion

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. 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 $\mathcal{A}$,

$$\mathrm{Adv}^{DL}(\mathcal{A}) = Pr\left[\mathcal{A}(g^x) \to x \middle| x, y \overset{R}{\leftarrow} [1, q]\right]$$

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 $\mathcal{A}$,

$$\mathsf{Adv}^{CDH}(\mathcal{A}) = Pr\left[\mathcal{A}(g^x, g^y) \to g^{xy} \middle| x, y \stackrel{R}{\leftarrow} [1, q]\right]$$

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 $\mathcal{A}$

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

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

Prop

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

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

Outline

1. Negligible Functions
2. Diffie-Hellman
3. Reduction Proof
4. Different Adversaries
5. Intuition of Computational Security
6. Definitions of Computational Security
7. Conclusion

How to prove the security ?

Theorem

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

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

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

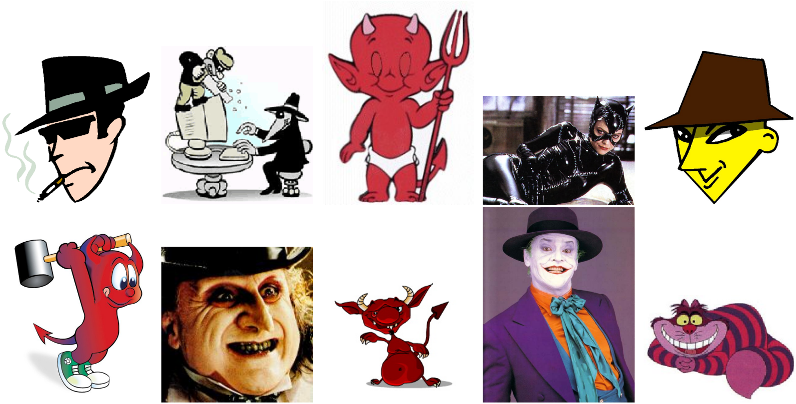
Proof by Reduction

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.

Outline

1. Negligible Functions
2. Diffie-Hellman
3. Reduction Proof
4. Different Adversaries
5. Intuition of Computational Security
6. Definitions of Computational Security
7. 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 generates keys, random number...

Adversary Models

The adversary is given access to oracles :

$\rightarrow$ encryption of all messages of his choice
$\rightarrow$ 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

$\Downarrow$

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

$\Downarrow$

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

Outline

1. Negligible Functions
2. Diffie-Hellman
3. Reduction Proof
4. Different Adversaries
5. Intuition of Computational Security
6. Definitions of Computational Security
7. Conclusion

One-Wayness (OW)

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

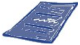
One-Wayness (OW)

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

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

Is it secure ?

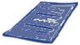
Is it secure ?



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

$\Downarrow$

Indistinguishability

$\Downarrow$

One-Wayness

Outline

1. Negligible Functions
2. Diffie-Hellman
3. Reduction Proof
4. Different Adversaries
5. Intuition of Computational Security
6. Definitions of Computational Security
7. 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**

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

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

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

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 $\textsc{Ind}^{b}_{CPA}(\mathcal{A})$ be the following algorithm:

- Generate $(pk, sk) \stackrel{R}{\leftarrow} \mathcal{K}(\eta)$.
- $(s, m_0, m_1) \stackrel{R}{\leftarrow} \mathcal{A}_1(\eta, pk)$
- Sample $b \stackrel{R}{\leftarrow} \{0, 1\}$.
- $b' \stackrel{R}{\leftarrow} \mathcal{A}_2(\eta, pk, s, \mathcal{E}(pk, m_b))$
- return b'.

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

$$\textsc{Adv}^{\textsc{Ind}_{CPA}}_{\mathcal{S},\mathcal{A}}(\eta) =$$

$$Pr[b' \stackrel{R}{\leftarrow} \textsc{Ind}^{1}_{CPA}(\mathcal{A}) : b' = 1] - Pr[b' \stackrel{R}{\leftarrow} \textsc{Ind}^{0}_{CPA}(\mathcal{A}) : b' = 1]$$

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 $\textsc{Ind}^b_{CCA1}(\mathcal{A})$ be the following algorithm:

- Generate $(pk, sk) \stackrel{R}{\leftarrow} \mathcal{K}(\eta)$.
- $(s, m_0, m_1) \stackrel{R}{\leftarrow} \mathcal{A}_1^{\mathcal{O}_1}(\eta, pk)$ **where** $\mathcal{O}_1 = \mathcal{D}$
- Sample $b \stackrel{R}{\leftarrow} \{0, 1\}$.
- $b' \stackrel{R}{\leftarrow} \mathcal{A}_2(\eta, pk, s, \mathcal{E}(pk, m_b))$
- return b'.

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

$$\textsc{Adv}^{\textsc{Ind}_{CCA1}}_{\mathcal{S},\mathcal{A}}(\eta) =$$

$$Pr[b' \stackrel{R}{\leftarrow} \textsc{Ind}^1_{CCA1}(\mathcal{A}) : b' = 1] - Pr[b' \stackrel{R}{\leftarrow} \textsc{Ind}^0_{CCA1}(\mathcal{A}) : b' = 1]$$

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

- Generate $(pk, sk) \stackrel{R}{\leftarrow} \mathcal{K}(\eta)$.
- $(s, m_0, m_1) \stackrel{R}{\leftarrow} \mathcal{A}_1^{\mathcal{O}_1}(\eta, pk)$ where $\mathcal{O}_1 = \mathcal{D}$
- Sample $b \stackrel{R}{\leftarrow} \{0, 1\}$.
- $b' \stackrel{R}{\leftarrow} \mathcal{A}_2^{\mathcal{O}_2}(\eta, pk, s, \mathcal{E}(pk, m_b))$ **where** $\mathcal{O}_2 = \mathcal{D}$
- return b'.

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

$$\text{ADV}^{\text{IND}_{CCA2}}_{\mathcal{S},\mathcal{A}}(\eta) =$$

$$Pr[b' \stackrel{R}{\leftarrow} \text{IND}^1_{CCA2}(\mathcal{A}) : b' = 1] - Pr[b' \stackrel{R}{\leftarrow} \text{IND}^0_{CCA2}(\mathcal{A}) : b' = 1]$$

Summary

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

- Generate $(pk, sk) \overset{R}{\leftarrow} \mathcal{K}(\eta)$.
- $(s, m_0, m_1) \overset{R}{\leftarrow} \mathcal{A}_1^{\mathcal{O}_1}(\eta, pk)$
- Sample $b \overset{R}{\leftarrow} \{0, 1\}$.
- $b' \overset{R}{\leftarrow} \mathcal{A}_2^{\mathcal{O}_2}(\eta, pk, s, \mathcal{E}(pk, m_b))$
- return b'.

$$\text{ADV}^{\text{IND}_{XXX}}_{\mathcal{S},\mathcal{A}}(\eta) =$$

$$Pr[b' \overset{R}{\leftarrow} \text{IND}^{1}_{XXX}(\mathcal{A}) : b' = 1] - Pr[b' \overset{R}{\leftarrow} \text{IND}^{0}_{XXX}(\mathcal{A}) : b' = 1]$$

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.

IND-XXX Security




Definition

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

Exercise

Prove that

$$\begin{array}{rcl}\textsc{Adv}^{\textsc{Ind}XXX}_{\mathcal{S},\mathcal{A}}(\eta) & = & Pr[b' \stackrel{R}{\leftarrow} \textsc{Ind}^1(\mathcal{A}) : b' = 1] \\ & - & Pr[b' \stackrel{R}{\leftarrow} \textsc{Ind}^0(\mathcal{A}) : b' = 1] \\ & = & 2Pr[b' \stackrel{R}{\leftarrow} \textsc{Ind}^b(\mathcal{A}) : b' = b] - 1\end{array}$$

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

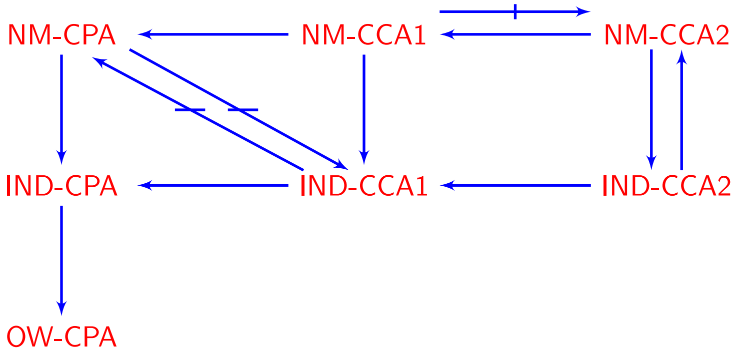



- Let $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and $A = (A_1, A_2)$.
- For $b \in \{0,1\}$ we define the experiment $\mathbf{Exp}^{atk-b}_{\mathcal{PE},A}(k)$:
  $(pk, sk) \leftarrow \mathcal{K}(k)$; $(M, s) \leftarrow A_1^{O_1(.)}(pk)$; $x_0, x_1 \leftarrow M$
  $y \leftarrow \mathcal{E}_{pk}(x_b)$; $(\mathcal{R}, \vec{y}) \leftarrow A_2^{O_2(.)}(M, s, y)$; $\vec{x} \leftarrow \mathcal{D}_{pk}(\vec{y})$;
  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

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

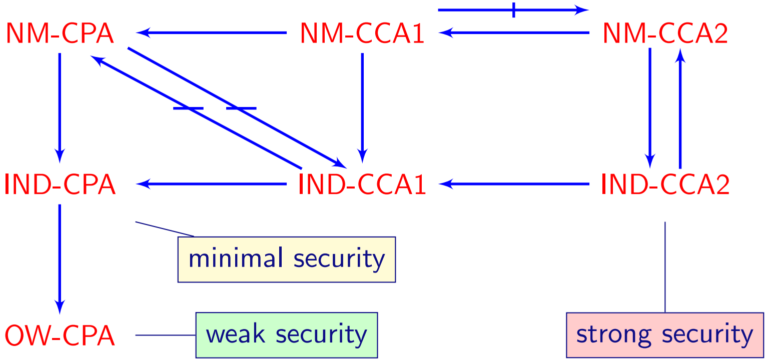
  has to be negligible for $\mathcal{PE}$ to be considered secure, assuming A, M and $\mathcal{R}$ can be computed in time $p(k)$.

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]

Example: RSA

public	private
$n = pq$	$d = e^{-1} \mod \phi(n)$
e (public key)	(private key)

RSA Encryption

- $E(m) = m^e \mod n$
- $D(c) = c^d \mod n$

OW-CPA = RSA problem by definition!

But not semantically secure because it is deterministic.

Outline

1. Negligible Functions
2. Diffie-Hellman
3. Reduction Proof
4. Different Adversaries
5. Intuition of Computational Security
6. Definitions of Computational Security
7. Conclusion

Today

1. DH
2. OW & IND & NM
3. CPA & CCA1 & CCA2
4. Reduction technique

Thank you for your attention.

Questions ?