1 Hard Problems

1.1 Factorization

\[ p, q \rightarrow n = p \times q \text{ (easy quadratic)} \]
\[ n = p \times q \rightarrow p, q \text{ difficult} \]

Example: RSA function, easy to encrypt but difficult to decrypt (without knowing the private key)

1.2 Discrete Logarithm (DL)

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

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

For any adversary \( A \),

\[
\forall x \in [1, q], \text{Adv}^{DL}(A) = Pr[A(g^x) \rightarrow x] \text{ is negligible.}
\]

1.3 Diffie-Hellman

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

\( A \rightarrow B : g^a \)
\( B \rightarrow A : g^b \)
\( A \rightarrow B : N_{g^{ab}} \)

1.3.1 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 \( A \),

\[
\forall x, y \in [1, q], \text{Adv}^{CDH}(A) = Pr[A(g^x, g^y) \rightarrow g^{xy}] \text{ is negligible.}
\]
1.3.2 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 \( A \),

\[
\forall x, y, r \in [1, q], \text{Adv}_{DDH}^D(A) = Pr[A(g^x, g^y, g^{xy}) \rightarrow 1] - Pr[A(g^x, g^y, g^r) \rightarrow 1] \text{ is negligible.}
\]

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

1.4 Summary

Most cryptographic constructions are based on hard problems. Their security is proved by reduction to the problems above (DL, CDH, DDH).

\[
\text{DDH} \leq \text{CDH} \leq \text{DL} \text{ or } \text{DL} \Rightarrow \text{CDH} \Rightarrow \text{DDH}
\]

Proof:

1. \( \text{DDH} \leq \text{CDH} \)
   \[
   \text{DDH}(g^x, g^y, z) \{
   \text{//OracleCDH returns } g^{xy}, \text{ then we just have to compare with the input value } z
   \text{//to know whether or not } z = g^{xy} \text{ or } z = g^r
   \text{if } (z == \text{OracleCDH}(g^x, g^y)) \text{ return true;}
   \text{else return false;}
   \}
   \]

2. \( \text{CDH} \leq \text{DL} \)
   \[
   \text{CDH}(g^x, g^y) \{
   x = \text{OracleDL}(g^x);
   y = \text{OracleDL}(g^y);
   \text{return } g^{xy};
   \}
   \]

Usage of DH assumption

The Diffie-Hellman problems are widely used in cryptography:

- Public key cryptosystems [ElGamal, Cramer & Shoup]
2 Adversaries

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...

The adversary is given access to oracles (encryption and decryption of all messages of his choice)

Three classical security levels:

- Chosen-Plain-text Attacks (CPA)
- Non adaptive Chosen-Cipher-text Attacks (CCA1)
- Adaptive Chosen-Cipher-text Attacks (CCA2)

2.1 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.*

2.2 Non adaptive Chosen-Cipher-text Attacks (CCA1)

Adversary knows the public key and 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]).
2.3 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]).

\[ CCA2 \Rightarrow CCA1 \Rightarrow CPA \]

3 Definition of Indistinguishability

Definition 1: A function \( \mu : \mathbb{N} \to \mathbb{R}^+ \) is negligible if for every positive polynomial \( p \) there is an \( N \) such that

\[ \forall n > N, \mu(n) < \frac{1}{p(n)} \]

Example: \( f(n) \to \left(\frac{1}{2}\right)^n \)

Definition 2: Objects are considered to be computationally equivalent if they cannot be differentiated by any efficient procedure.

Example: Considering an adversary A, we say that A does not distinguish the two distributions \( D_1 \) and \( D_2 \) if and only if

\[ \forall x \in U, |Pr(A(x_1) = 1) - Pr(A(x_0) = 1)| \]

is negligible.

where \( Pr(A(x_b) = 1) \) \((b \in 0, 1)\) is the probability for A to guess correctly that the element \( x \) is taken from \( D_b \).

4 Three security notions for the public key encryption

4.1 One-Wayness OW

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

\textit{It is equivalent to put the message in a translucent box: you can’t recover it completely but some parts may be accessible especially if the adversary has partial information about the message:}

\textit{FROM: XXXX}
4.2 Indistinguishability IND

Definition: The adversary is not able to guess in polynomial-time even a bit of the plaintext knowing the cipher-text, notion introduced by S. Goldwasser and S. Micali ([GM84]).

Message is now in a black bag but it is still possible to scramble the cipher to produce a new cipher-text which decrypts to a related plaintext.

4.3 Non-Malleability NM

Definition: 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]).

NM is the strongest level of security demands and it implies IND which implies OW.

\[
NM \Rightarrow IND \Rightarrow OW
\]

5 Computational security

Let \( XXX \in \{CPA, CCA1, CCA2\} \).

We recall that:
- for CPA, \( O_1 = O_2 = \emptyset \), the adversary has no access to the oracles.
- for CCA1, \( O_1 = \{D\}, O_2 = \emptyset \), the adversary can use the decryption oracle before the challenge.
- for CCA2, \( O_1 = O_2 = \{D\} \), in the case, the adversary can use the decryption oracle before and after the challenge.

5.1 The IND-XXX Games

Given an encryption scheme \( S = (K, E, D) \). An adversary is a pair \( A = (A_1, A_2) \) of polynomial-time probabilistic algorithms. Let \( b \in \{0, 1\} \).

First, we give to \( A_1 \) the public key. Then he chooses two messages \( m_0, m_1 \).

Then \( b \) is picked at random from the set \( \{0, 1\} \), and \( A_2 \) receive \( c \), the encryption of \( m_b \).

To conclude, \( A_2 \) answers \( b' \), and we check if the probability that \( b \) equals \( b' \) is 1/2 in order to see if the scheme is secure. Now we move on an algorithmic description.
Let $IND^b_{XXX}(A)$ be the following algorithm:

- Generate $(pk, sk) \overset{R}{\leftarrow} K(\eta)$
- $(s, m_0, m_1) \overset{R}{\leftarrow} A^1_1(\eta, pk)$ [s is just a variable which allow $A_1, A_2$ to communicate]
- $b' \overset{R}{\leftarrow} A^0_2(\eta, pk, s, E(pk, m_b))$ [$b' \in \{0, 1\}$]
- return $b'$

We define the function

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

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

**Example**: RSA is not IND-CPA

- The adversary $A_1$ is given the public key $p_k$
- He chooses two messages $m_0, m_1$
- $b = 0, 1$ is chosen at random and $c = E(m_b)$ is given to the adversary $A_2$
- $A_2$’s algorithm to choose between 0 and 1 ($b'$)
  - if $(RSA(m_0, p_k) == c)$ return 0;
  - else return 1;
- The probability $P(b = b') - \frac{1}{2}$ is not negligible so RSA is not IND-CPA.
- In fact, this proof shows that every encryption scheme which doesn’t imply a part of randomness can’t be more secure than OW.

**Example**: Proof that RSA is OW secure by reduction to factorization problem.

- We assume that we have an Oracle $O(n,e)$ which solves RSA and return the private key $d$.
- We let be $n$ the input of the factorization problem $n = p \times q$ we are looking for $p$ and $q$ two large prime numbers.
- We call our oracle with a small prime number $e = 2$ for example.
- We get $d$ such that $d \times e = \phi(n) \times k + 1$ whith $k \in \mathbb{N}$
We try all the possible values for $k$ until $\phi(n)$ is an integer: this is a loop of $d \times e$ operations but $e$ is small so it’s in $O(d)$.

$\phi(n) = (p - 1) \times (q - 1) = pq - (p + q) + 1$ so $p + q = p \times q + 1 - \phi(n) = n + 1 - \phi(n)$

We now have $p \times q$ and $p + q$ so finding $p$ and $q$ is possible by solving the equation $X^2 - (p + q) \times X + p \times q = 0$

5.2 The NM-XXX Games

Given an encryption scheme $PE = (K, E, D)$. An adversary is a pair $A = (A_1, A_2)$ of polynomial-time probabilistic algorithms. Let $b \in \{0, 1\}$.

The idea of this game is to chose a message space $M$ for $A_1$ (which has been given the public key), and to give to $A_2$ two messages $m$ and $m^*$ picked randomly in this message space.

Moreover, $c$, the encryption of $m$, is also given to $A_2$.

This one finally produces a cipher text $c'$, a binary relation $R$, then we calculate the difference between the probability that $m, D(c')$ are in relation and the probability that $m, m^*$ are in relation.

The result should be negligible if the scheme is secure. In other words:

Let $NM^b_{XXX}(A)$ be the following algorithm:

1. Generate $(pk, sk) \xleftarrow{R} K(\eta)$

2. $(s, M) \xleftarrow{R} A_1^b(\eta, pk)$; $m_0, m_1 \xleftarrow{R} M$ [is just a variable which allows $A_1, A_2$ to communicate]

3. $(R, C') \xleftarrow{R} A_2^b(\eta, pk, s, M, E(pk, m_b))$; $M' \xleftarrow{R} D(C')$ [$C'$ is a cipher-text space]

4. return $R(m_b, M')$

We define the function

$$ADV^b_{S,A}^{NM_{XXX}}(\eta) = Pr[R(m, M') \xleftarrow{R} NM^b_{XXX}(A) : R(m, M') = 1] - Pr[R(m, M'') \xleftarrow{R} NM^b_{XXX}(A) : R(m, M'') = 1]$$

**Definition**: An encryption scheme is NM-XXX secure, if for any adversary $A$ the function $ADV^b_{S,A}^{NM_{XXX}}$ is negligible.
5.3 Relations

We conclude this lecture with the different relations between the OW-CPA, IND-XXX and NM-XXX games.