HAI709I : Fondements cryptographiques de la sécurité, Université de Montpellier, 2023

07/10/2024. Lecture 5.

1 Computational security: encryption schemes secure against and adversary computable in polynomial time

In this section we discuss encryption schemes suitable for "realistic" settings. On a one hand, this makes some requirements for the scheme even more restrictive: all operations performed by the sender and the receiver (Alice and Bob) must be done efficiently. This means that the algorithm for encoding and decoding must run in polynomial time. On the other hand, this makes the security restrictions imposed on the scheme much weaker: the scheme need to remain secure only against adversaries computable in polynomial time.

Definition 1. A triple of algorithms $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ is called an encryption scheme *computable in polynomial time* if for every n there are families of sets $\mathcal{M}_n, \mathcal{E}_n, \mathcal{K}_n$ (which are the spaces of *clear texts*, *cyphertexts*, and *secret keys* for the value of security parameter n), and

- $\mathrm{Gen}(\underbrace{11\ldots1}_{n})$ samples a key $k \in \mathcal{K}_n$ (a secret key used for messages of length n)
- $\mathrm{Enc}(m, k, \underbrace{11\ldots1}_{n})$ applied to a clear message $m \in \mathcal{M}_n$ and a secret key $k \in \mathcal{K}_n$ returns an encrypted message $e \in \mathcal{E}_n$
- $\mathrm{Dec}(m, k, \underbrace{11\ldots1}_{n})$ applied to an encrypted message $e \in \mathcal{E}_n$ and a secret key $k \in \mathcal{K}_n$ returns either $m \in \mathcal{M}_n$ such that $\mathrm{Enc}(m, k) = e$ or the symbol $\perp$ (failure)
- each of the algorithms $\langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ terminates in polynomial time
- for every $m \in \mathcal{M}_n$, for a randomly chosen $k \leftarrow Gen(\underbrace{11\ldots1}_{n})$, the probability

$$\mathrm{Prob}[\mathrm{Dec}(\mathrm{Enc}(m, k), k) = \perp]$$

  is a negligible function.

It is not so simple to say what it means that a scheme is secure against a restricted adversary. Unlike the situation with an unlimitedly powerful adversary, we cannot require that the information revealed to the adversary does not affect the conditional distribution on the set of potentially possible clear messages. Instead we will use the definition based on *the attack with a chosen pair of clear messages*. We will see later that this definition implies all properties of security that we may need in practice (see the discussion of *semantic security* a few lectures later).

The basic attack (attack with a chosen pair of clear messages): Let $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ be an encryption scheme, where $\mathcal{M}, \mathcal{E}, \mathcal{K}$ are the spaces of *clear messages*, *encrypted messages*, and *secret key* respectively. Let us consider the following game between an adversary and Alice.

- Adversary uses an algorithm $Adv_1(\underbrace{11\ldots1}_{n})$ that chooses two clear messages $m_a, m_b \in \mathcal{M}_n$;

- Alice chooses at random $i \in \{a, b\}$ (with equal probabilities), samples a secret key $k \leftarrow \mathrm{Gen}(\underbrace{11\ldots1}_{n})$, and computes the encrypted message $e = \mathrm{Enc}(m_i, k, \underbrace{11\ldots1}_{n})$;
- Adversary computes $j \in \{a, b\}$ using another algorithm $j \leftarrow Adv_2(e, m_a, m_b, \underbrace{11\ldots1}_{n})$.

The success of the adversary is defined as follows:

$$\mathbf{success} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

In words: the adversary prepares a pair of messages m_a, m_b; Alice decides which message to encrypt; then the adversary tries to understand which of the two messages was encrypted.

Definition 2. An encryption scheme $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ is called *secure against an adversary computable in polynomial time*, if for the game between Adversary and Alice following the protocol of the *attack with a pair of chosen clear messages* (explained above) where the adversary's algorithms Adv_1 and Adv_2 are computable in polynomial time, the gap

$$\left| \mathrm{Prob}[\mathbf{succes}] - \frac{1}{2} \right|$$

is a negligible function.

Remark 1. In practice we want to use encryption schemes that are at once *computable in polynomial time* and *secure against an adversary computable in polynomial time*.

The *attack with a pair of chosen clear messages* might look a bit artificial. However, security against this attack is an absolutely fundamental property. It implies security against many other natural types of attacks. In the class we discussed several examples of such facts, see below.

Example 1: Attack on the first bit of the clear messages. Let $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ be an encryption scheme, where $\mathcal{M}, \mathcal{E}, \mathcal{K}$ are the spaces of *clear messages*, *encrypted messages*, and *secret key* respectively. Let us assume that $\mathcal{M} = \{0, 1\}^n$, and consider the following game between an adversary and Alice.

- Alice chooses at random a clear message $(m_1 \ldots m_n) \in \{0, 1\}^n$ (with the uniform distribution of probabilities, i.e., each message is chosen with probability $1/2^n$), samples a secret key

  $$k \leftarrow \mathrm{Gen}(\underbrace{11\ldots1}_{n}),$$

  and computes the encrypted message $e = \mathrm{Enc}((m_1 \ldots m_n), k, \underbrace{11\ldots1}_{n})$
- Adversary computes $j \leftarrow Adv_{example1}(e, \underbrace{11\ldots1}_{n})$.

The success of the adversary is defined as follows:

$$\mathbf{success}_{\text{example1}} = \begin{cases} 1, & \text{if } j = m_1, \\ 0, & \text{otherwise.} \end{cases}$$

In words: Alice encrypts a randomly chosen message; Adversary intercepts the cyphertext and tries to guess the value of the very first bit of the clear message.

Proposition 1. *If a scheme* $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ *is secure against an adversary computable in polynomial time (in the sense of Definition 2) then* Π *is secure against an attack on the first bits of the clear message, i.e., for every adversary* Adv *computable in polynomial time*

$$\left| \mathrm{Prob}[\mathbf{success}_{\text{example1}} = 1] - \frac{1}{2} \right|$$

is negligible.

Informal meaning of Proposition 1: if a scheme is secure against polynomial-time computable adversaries, then one cannot extract from an encrypted message any useful information on the 1st bit of the clear text.

Proof. We want to prove

$$\Pi \text{ is secure in the sense of Definition 2} \Longrightarrow \left| \mathrm{Prob}[\mathbf{success}_{\text{example1}} = 1] - \frac{1}{2} \right| \text{ is negligible}$$

It is convenient to reformulate this implication in the counter-positive form:

$$\left| \mathrm{Prob}[\mathbf{success}_{\text{example1}} = 1] - \frac{1}{2} \right| \text{ is } \textit{non}\text{-negligible} \Longrightarrow \Pi \text{ is } \textit{not} \text{ secure in the sense of Definition 2}$$

A bit more precisely, this means that if there exist an adversary that uses a polynomial time algorithm $Adv_{example1}$, and

$$\mathrm{Prob}[\mathbf{success}_{\text{example1}} = 1] = \frac{1}{2} + \delta_n$$

for a *non-negligible* δ_n, then there exists a pair of polynomial time algorithms Adv_1, Adv_2 such that in the game from Definition 2 the differnece

$$\left| \mathrm{Prob}[\mathbf{success} = 1] - \frac{1}{2} \right|$$

is also non-negligible. In fact, we will use $Adv_{example1}$ to construct Adv_1, Adv_2 such that

$$\mathrm{Prob}[\mathbf{success} = 1] = \frac{1}{2} + \delta_n$$

(with exactly the same non-negligible δ_n). We do it as follows.

- $Adv_1(\underbrace{11\ldots1}_{n})$ samples two clear messages $m_a, m_b \in \mathcal{M}_n = \{0,1\}^n$

$$\begin{aligned} m_a &= 0\underbrace{**\ldots*}_{n-1} \\ m_b &= 1\underbrace{**\ldots*}_{n-1} \end{aligned}$$

  such that the first bit of m_a is 0 and the first bit of m_b is 1 (the other bits of m_a and m_b denoted above "$*$" are chosen at random, uniformly and independently);

- Alice chooses at random $i \in \{a, b\}$ (with equal probabilities), samples a secret key $k \leftarrow \mathrm{Gen}(\underbrace{11\ldots1}_{n})$, and computes the encrypted message $e = \mathrm{Enc}(m_i, k, \underbrace{11\ldots1}_{n})$;

- $Adv_2(e, m_a, m_b, \underbrace{11\ldots1}_{n})$ computes $j \leftarrow Adv_{example1}(e)$

  and then returns a in case $j = 0$ and returns b in case $j = 1$

Idea behind the construction: The machine $Adv_{example1}$ can distinguish between encrypted messages obtain from clear messages $0 * *\ldots*$ and from clear messages $1 * *\ldots*$; we can apply this machine (without even understanding how it works) to distinguish between $\mathrm{Enc}(m_a, k, 1^n)$ and $\mathrm{Enc}(m_b, k, 1^n)$, if the clear texts m_a and m_b begin with 0 and 1 respectively.

Observe that we include $Adv_{example1}$ in the construction of Adv_2 as a "black box". If $Adv_{example1}$ runs in polynomial time, then we can claim that Adv_2 also runs in polynomial time. The algorithm Adv_1 always run in polynomial time, its construction is pretty simple.

We assumed that all digits in $m_a = 0\underbrace{**\ldots*}_{n-1}$ and in $m_b = 1\underbrace{**\ldots*}_{n-1}$ (except for the very first position) are chosen at random, uniformly and independently. Hence, when Adv_1 samples these m_a and m_b, and then Alice chooses $i \in \{a, b\}$ at random, then the resulting string

$$m_i = \underbrace{**\ldots*}_{n}$$

is uniformly distributed on the whole set $\{0,1\}^n$. So, when we feed $e = \mathrm{Enc}(m_i, k, \underbrace{11\ldots1}_{n})$ into $Adv_{example1}$, we are in the setting of Example 1 (i.e., $Adv_{example1}$ is applied to an encrypted messages obtained from a randomly chosen clear message). Therefore, we may use the assumption that $Adv_{example1}(e)$ with probability $\frac{1}{2} + \delta_n$ correctly guesses the first bits of the clear message m_i.

Further, the strings m_a and m_b are chosen so that they must differ at the first bit. Thus, the answer returned by $Adv_{example1}(e)$ allows to understand which of the two messages (m_a or m_b) was encoded. And this is exactly what Adv_2 is doing!

It is not hard to see that the attack of (Adv_1, Adv_2) succeeds if and only if $Adv_{example1}(e)$ correctly determines the first bit of the used clear message. Therefore, this attack succeeds with probability $\frac{1}{2} + \delta_n$. If δ_n is non-negligible, we conclude that Π is not secure, and the proposition is proven. □

Example 2: Attack on the XOR of the bits of the clear message. Let $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ be an encryption scheme, where $\mathcal{M}, \mathcal{E}, \mathcal{K}$ are the spaces of *clear messages*, *encrypted messages*, and *secret key* respectively. We assume again that $\mathcal{M} = \{0,1\}^n$ and consider the following game between an adversary and Alice.

- Alice chooses at random a clear message $(m_1 \dots m_n) \in \{0,1\}^n$ (with the uniform distribution of probabilities, i.e., each message is chosen with probability $1/2^n$), samples a secret key

$$k \leftarrow \mathrm{Gen}(\underbrace{11\dots1}_{n}),$$

  and computes the encrypted message $e = \mathrm{Enc}((m_1 \dots m_n), k, \underbrace{11\dots1}_{n})$

- Adversary computes $j \leftarrow Adv_{example2}(e, \underbrace{11\dots1}_{n})$.

The success of the adversary is defined as follows:

$$\mathbf{success}_{\mathrm{example2}} = \begin{cases} 1, & \text{if } j = m_1 \oplus m_2 \oplus \dots \oplus m_n, \\ 0, & \text{otherwise.} \end{cases}$$

In words: Alice encrypts a randomly chosen message; Adversary intercepts the cyphertext and tries to guess the XOR of all bits of the clear message.

Proposition 2. *If a scheme* $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ *is secure against an adversary computable in polynomial time (in the sense of Definition 2) then* Π *is secure against an attack on the XOR of the bits of the clear message, i.e., for every adversary* Adv *computable in polynomial time*

$$\left| \mathrm{Prob}[\mathrm{success}_{\mathrm{example2}} = 1] - \frac{1}{2} \right|$$

is negligible.

Informal meaning of Proposition 2: if a scheme is secure in the sense of Definition 2, then one cannot extract from an encrypted message any useful information on the XOR of all bits of the clear text.

Proof of Proposition 2 is similar to the proof of Proposition 1. The key idea: the adversary should sample two random clear messages m_a *and* m_b *with different parity of their bits.*

Example 3: Attack on the value of the 50**th bit of the clear message given as a side information the values of the first** 49 **bits of the clear message.** Let $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ be an encryption scheme, where $\mathcal{M}, \mathcal{E}, \mathcal{K}$ are the spaces of *clear messages*, *encrypted messages*, and *secret key* respectively and let $\mathcal{M} = \{0,1\}^n$. We consider the following game between an adversary and Alice.

- Alice chooses at random a clear message $(m_1 \dots m_n) \in \{0,1\}^n$ (with the uniform distribution of probabilities, i.e., each message is chosen with probability $1/2^n$), samples a secret key

$$k \leftarrow \mathrm{Gen}(\underbrace{11\dots1}_{n}),$$

  and computes the encrypted message $e = \mathrm{Enc}((m_1 \dots m_n), k, \underbrace{11\dots1}_{n})$

- Adversary computes $j \leftarrow Adv_{example3}(e, m_1, m_2, \ldots, m_{49}, \underbrace{11\ldots1}_{n})$.

The success of the adversary is defined as follows:

$$\mathbf{success}_{\text{example3}} = \begin{cases} 1, & \text{if } j = m_{50}, \\ 0, & \text{otherwise.} \end{cases}$$

In words: Alice encrypts a randomly chosen message; Adversary intercepts the cyphertext and gets in addition the values of the first 49 bits of the clear text; given all this information, Adversary tries to guess the 50th bits of the clear message.

Proposition 3. *If a scheme $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ is secure against an adversary computable in polynomial time (in the sense of Definition 2) then Π is secure against an attack on the value of the 50th bit of the clear message given as a side information the values of the first 49 bits of the clear message, i.e., for every adversary Adv computable in polynomial time*

$$\left| \mathrm{Prob}[\mathrm{success}_{\text{example3}} = 1] - \frac{1}{2} \right|$$

is negligible.

Informal meaning of Proposition 3: if a scheme is secure in the sense of Definition 2, then one cannot extract from an encrypted message any useful information on the $50th$ bit of the clear text even if we get an access to the first 49 bits of the clear text.

Proof of Proposition 3 is similar to the proof of Proposition 1 and Proposition 2. The key idea: the adversary samples to clear messages m_a and m_b that agree at the first 49 positions and differ in the 50th position.

2 Pseudo-random generators and computationally secure schemes

Reminder:

Definition 3. A function

$$G \,:\, \{0,1\}^{\ell(n)} \to \{0,1\}^n$$

is called a *pseudo-random generator* if

- $\ell(n) < n$
- $G(x)$ is computed (by a deterministic algorithm) in polynomial time
- for every poly-time algorithm TEST (deterministic or randomised) the difference

$$\left| \mathrm{Prob}_{x \in_R \{0,1\}^{\ell(n)}}[\mathrm{TEST}(G(x)) = 1] - \mathrm{Prob}_{y \in_R \{0,1\}^n}[\mathrm{TEST}(y) = 1] \right|$$

  is negligibly small.

Theorem 1. *We fix $G_n \,:\, \{0,1\}^{\ell(n)} \to \{0,1\}^n$ and define an encryption scheme $\Pi = \langle Gen, Enc, Dec \rangle$ as follows. We let $\mathcal{M}_n = \mathcal{E}_n = \{0,1\}^n$ (the spaces of clear texts and cyphertexts), and $\mathcal{K}_n = \{0,1\}^{\ell(n)}$ (the space of secret keys), and assume that*

- *the algorithm* $Gen(\underbrace{11\ldots1}_{n})$ *takes a random* $k \in \{0,1\}^{\ell(n)}$
- *the algorithm* $Enc(m,k)$ *computes a bitwise XOR of the open message* m *and* $k' = G_n(k)$
- *the algorithm* $Dec(e,k)$ *computes a bitwise XOR of the encrypted message* e *and the key* $k' = G_n(k)$

If G_n *is a pseudo-random generator then this scheme* Π *is secure against any adversary computable in polynomial time.*

Remark 2. This scheme allows to use secret keys that are much shorter than the clear messages. Such a scheme is interesting if $\ell(n) \ll n$, e.g., $\ell(n) = \sqrt[10]{n}$. We know that if the secret keys are shorter than the clear messages, then a scheme cannot be absolutely secure (against an adversary with unbounded computational resources). However, Theorem 1 guarantees that we can make secret keys much shorter without loosing security against the adversaries *computable in polynomial time*.

We will finish the proof of this theorem in the next lecture.

References

[1] J. Katz, Y. Lindell. Introduction to modern cryptography, CRC Press, 2021

[2] B. Martin. Codage, cryptologie et applications. PPUR presses polytechniques, 2004