Definitions and concepts

Ciphers and security definitions

Cipher -- Definition

Let $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ be key, message and ciphertext spaces. A **cipher** is defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and is a pair of efficient encryption-decryption algorithms (E, D) suc that:

- E may be randomized, D is deterministic.

$$E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}; \quad c = E(k, m)$$

$$D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}; \quad m = D(k, c)$$

- has the **Corectness propriety**:

$$\forall m \in \mathcal{M}, k \in \mathcal{K} \Rightarrow D(k, (E(k, m))) = m$$

- Is efficient
  - Polynomial time
  - In good time

Example: One time pad
$E(k, m) = k \oplus m$
$D(k, c) = k \oplus c$
$D(k, E(k, c)) = k \oplus k \oplus m = m$

Kerchoff's principle

The method must not be required to be secret, and it must be able to fall into the enemy's hands without causing inconvenience.

Intuition: all of the security of the system should be concentrated in the **secrecy of the key**, not the secrecy of the algorithms.

Perfect security

A cipher (E, D) has perfect secrecy if $\forall m_0, m_1 \in \mathcal{M}, c \in \mathcal{C}$ and the uniform key distribution K we have:

$$Pr[E(K, m_0) = c] = Pr[E(K, m_1) = c]$$

Intuition:

- All messages are **equally** likely to be the ciphertext
- No adversary can learn something about m given c. This means that if we have the distribution of messages M and the distribution of ciphertexts C the following holds for all ciphertexts c and messages m:

$$Pr[M = m | C = c] = Pr[M = m]$$

  In short C and M are independent.
- No chosen ciphertext attack.

We can also develop another characterization of perfect security. Suppose we have an adversary that can tinker with the ciphertexts. We call this tinkering a predicate ϕ. Now we can rephrase the definition to include all possible predicates ϕ that can be applied on $\mathcal{C}$:

$$Pr[\phi(E(K, m_0))] = Pr[\phi(E(K, m_1))]$$

Example:
OTP is perfectly secure.

However, the bad news:

- If (E, D) has perfect secrecy $\Rightarrow |\mathcal{K}| > |\mathcal{M}| \Rightarrow$ So for each message we will need to send a key that is the same or bigger size => impractical
- If we already have a secure channel to communicate the key there is no use for the OTP.
- Don't reuse the key or you'll be open to vulnerabilities.

Semantic security -- Definition

Instead of insisting that the probabilities from perfect security are equal we insist they are close:

$$Pr[\phi(E(K,m_0))] - Pr[\phi(E(K,m_1))] < \epsilon$$

where ϵ is negligible quantity.

Remark:

- In order to achieve computationally efficient results we need to relax the definition of security.
- Practical Relaxation: For all practical purposes instead of considering all possible ϕ we consider only the efficient ones instead of taking into account all possible generated messages m_0, m_1 we consider only those that can be generated by efficient algorithms.

Security as attack games.

There are other ways to define security concepts. One of the most common ones is to formulate a game with a **challenger** and an **adversary**. The challenger sets up a game and offers some information and the adversary must win the game or gain a significant **advantage**.
This way the challenger can control the difficulty of the game. If the game is too hard we can choose to give the adversary more power or set a lower advantage threshold for the adversary to achieve. Using these tools we can adjust how strong the security definitions are.

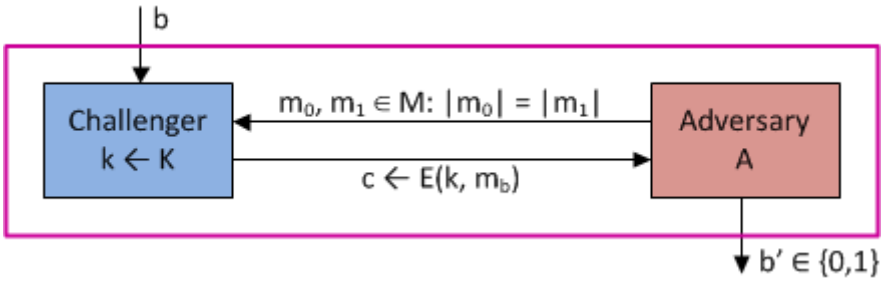
Semantic security -- Attack game

- Let $\mathcal{E} = (E, D)$ be a cipher
- Let $EXP(0)$, and $EXP(1)$ be two experiments
- An adversary A sends $m_0, m_1 \in M$ to the challenger
- The challenger sends an encryption of **one** of them ($EXP(0)$ or $EXP(1)$)
- The adversary must guess which one of the experiments was received.
- $W_b =$ event that $EXP(b) = 1$ = event that in $EXP(b)$ the Adversary outputs 1
- $\mathrm{Adv}_{SS}(A, \mathcal{E}) = |Pr[W_0] - Pr[W_1]| \in [0, 1]$
  - *intuition*: We look if the adversary behaves diferently if he is given one ciphertext or the other
  - If Adv is close to 1 => The adversary can distinguish between the encryptions

$\mathcal{E}$ is semantically secure if for all efficient adversaries A

- $\mathrm{Adv}_{SS}(A, \mathcal{E}) < \epsilon$

Intuition: This is similar to a bit guessing game. In this game the challenger rolls a bit b and based on the outcome decides which ciphertext (c_0 or c_1) to send to the adversary. The adversary must guess the bit's value.



Pseudorandomness

Computtional indistinguishability -- attack game

- The challenger computes a string s and samples another one r.
- Then it flips a bit b and gives one of them to the adversary.
- The adversary is tasked to say if he received s or r.

The definition can be expanded to probability distributions:

Computantional indistinguishability 2 -- attack game

Let P_1, P_2 be two distributions over $\{0,1\}^n$. We say that P_1, P_2 are **computationally indisinguishable** if $\forall$ efficient statistical tests A

$$|Pr_{x \leftarrow P_1}[A(x) = 1] - Pr_{x \leftarrow P_2}[A(x) = 1] < \epsilon$$

where ϵ is **negligible**.

Pseudorandom generators -- definition

Let $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^L$. An **efficient deterministic algorithm** $G : \{0,1\}^l \to \{0,1\}^L$ is called a **pseudorandom generator**. The output $G(s)$ must be **computationally indistinguishable** from a random string $r \in \{0,1\}^L$

Intuition: Given a seed $s \in \{0,1\}^l$ where $l \ll L$ we want to stretch the seed into a a longer key.

Remark Since $l \ll L$ the PRG cannot achieve perfect security.

Stream cipher

If we have a PRG G and we use the strings generated by it as keys like in OTP we are building a stream cipher. See image below.

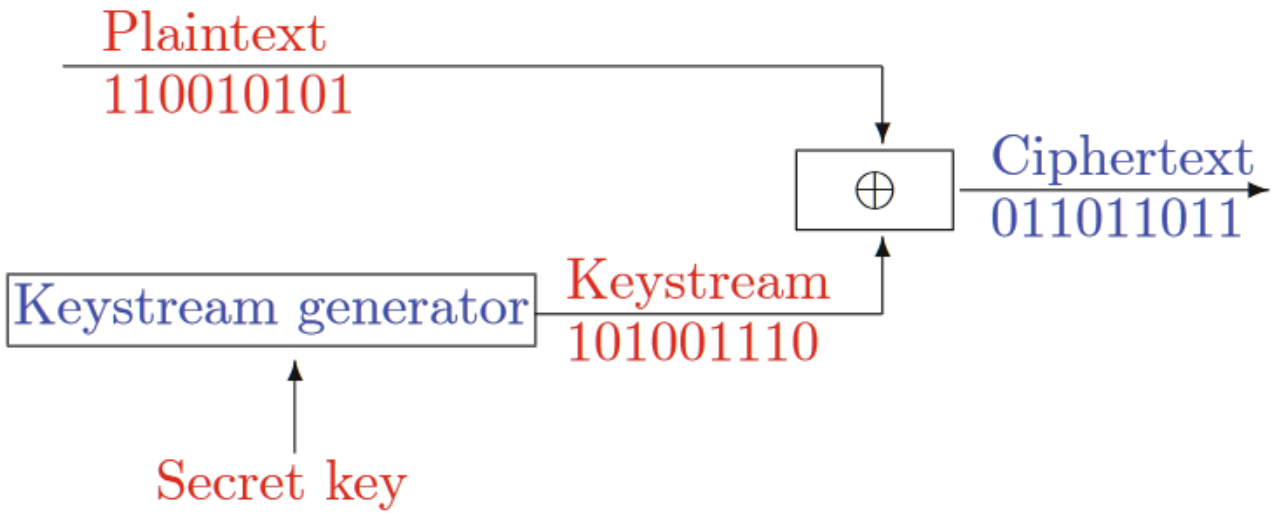


FIGURE 10.2. Stream ciphers

Pseudorandom function -- Definition

A pseudo-random function (PRF) $F : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}$ is a deterministic algorithm that has two inputs: key $k \in \mathcal{K}$ and an input data block $x \in \mathcal{X}$

Its output $y := F(k, x)$

Idea: for a randomly chosen key k the PRF F must look like a random function from $\mathcal{X}$ to $\mathcal{Y}$

PRF Security

A PRF F is secure if it's indistinguishable from a random function (The advantage for all efficient adversaries is negligible)

Weak security

A PRF F is secure if it's indistinguishable from a random function when the queries are limited(The advantage for all efficient adversaries is negligible)