Encryption

A Graduate Course in Applied Cryptography: Capítulo 2

Sebastián J. Giraudo

FaMAF - UNC

sebastian.giraudo@mi.unc.edu.ar

September 16, 2025

Presentation Overview

Desde donde partimos

Suposiciones: Alice y Bob comparten una clave secreta k. Alice quiere transmitir un mensaje m a Bob por una red, manteniendo el mensaje secreto ante la presencia de un eavesdropper.

Desde donde partimos

Suposiciones: Alice y Bob comparten una clave secreta k. Alice quiere transmitir un mensaje m a Bob por una red, manteniendo el mensaje secreto ante la presencia de un eavesdropper.

Vamos a resolver el problema principal, pero no resolvemos todo lo referido a una computación segura. En particular:

- Se transmite sólo un mensaje por key.
- No se asegura *message integrity*.
- No se resuelve el mecanismo para obtener una llave secreta compartida.

Cifrado de Shannon

Sea $\mathcal{K}$ un conjunto de llaves, $\mathcal{M}$ un conjunto de mensajes y $\mathcal{C}$ un conjunto de textos cifrados. Por simplicidad, asumimos que son conjuntos finitos.

Un cifrado de Shannon es un par de funciones $\mathcal{E} = (E, D)$ tal que:

- E es una **función de encriptación** $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ tal que

$$c = E(k, m)$$

- D es una **función de descifrado** $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$ tal que

$$M = D(k, c)$$

Cifrado de Shannon

Sea $\mathcal{K}$ un conjunto de llaves, $\mathcal{M}$ un conjunto de mensajes y $\mathcal{C}$ un conjunto de textos cifrados. Por simplicidad, asumimos que son conjuntos finitos.

Un cifrado de Shannon es un par de funciones $\mathcal{E} = (E, D)$ tal que:

- E es una **función de encriptación** $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ tal que

$$c = E(k, m)$$

- D es una **función de descifrado** $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$ tal que

$$M = D(k, c)$$

- El descifrado "deshace" la encriptación, o sea se cumple la propiedad de **correctitud**: para todas las keys k y mensajes m tenemos que

$$D(k, E(k, m)) = m$$

Decimos que $\mathcal{E}$ está definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

Ejemplo: One-time pad

Sean

$$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^L$$

para un parámetro fijo L

Las funciones E, D se definen:

$$E(k,m) = k \oplus m$$

$$D(k,c) = k \oplus c$$

con $\oplus$ función *exclusive or* computada bit a bit.

$$D(k, E(k,m)) = D(k, k \oplus m) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = 0^L \oplus m = m$$

Ejemplo: One-time pad con largo variable

Sean

$$\mathcal{K} = \{0,1\}^L$$

y

$$\mathcal{M} = \mathcal{C} = \{0,1\}^{\leq L}$$

para un parámetro fijo L

Las funciones E, D se definen:

$$E(k,m) = k[0..\ell - 1] \oplus m$$

$$D(k,c) = k[0..\ell - 1] \oplus c$$

con $k[0..\ell - 1]$ función que trunca los primeros ℓ bits de k.

Ejemplo: Cifrado por sustitución

Sean

$$\mathcal{M} = \mathcal{C} = \Sigma^L$$

para un alfabeto Σ y un parámetro fijo L.

$k : \Sigma \to \Sigma$ es una permutación en Σ.

Las funciones E, D se definen:

$$E(k, m) = (k(m[0]), k(m[1]), \ldots, k(m[L-1]))$$

$$D(k, c) = (k^{-1}(c[0]), k^{-1}(c[1]), \ldots, k^{-1}(c[L-1])$$

con $m[i]$ denotando la i-ésima entrada de m y $k(m[i])$ denotando la aplicación de la permutación k al símbolo $m[i]$

Ejemplo: One-time pad aditivo

Sean

$$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0, \ldots, n-1\}$$

para entero positivo n.

Las funciones E, D se definen:

$$E(k, m) = m + k \mod n$$

$$D(k, c) = c - k \mod n$$

Seguridad perfecta

¿Que es un cifrado “seguro”?

Seguridad perfecta

¿Que es un cifrado “seguro”?

Definamos la noción de **seguridad perfecta**:

Seguridad perfecta

¿Que es un cifrado “seguro”?

Definamos la noción de **seguridad perfecta**:

Definition

Sea $\mathcal{E} = (E, D)$ un cifrado de Shannon definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Consideramos un experimento probabilístico en donde la variable aleatoria $\boldsymbol{k}$ está uniformemente distribuida sobre $\mathcal{K}$. Si para todo $m_0, m_1 \in \mathcal{M}$ y todo $\boldsymbol{c} \in \mathcal{C}$ tenemos que

$$\Pr[E(\boldsymbol{k}, m_0) = c] = \Pr[E(\boldsymbol{k}, m_1) = c]$$

entonces decimos que $\mathcal{E}$ es un cifrado de Shannon con **seguridad perfecta**.

Otras formas de pensar a la seguridad perfecta

Theorem

Sea $\mathcal{E} = (E, D)$ un cifrado de Shannon definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Las siguientes afirmaciones son equivalentes:

i. *$\mathcal{E}$ es perfectamente seguro.*
ii. *Para cada $c \in \mathcal{C}$ existe un entero N_c, tal que para todo $m \in \mathcal{M}$, tenemos*

$$|\{k \in \mathcal{K} : E(k, m) = c\}| = N_c$$

iii. *Si la variable aleatoria $\boldsymbol{k}$ está uniformemente distribuida en $\mathcal{K}$, entonces cada una de las variables aleatorias $E(\boldsymbol{k}, m)$ para $m \in \mathcal{M}$, tiene la misma distribución.*

One-time pad y seguridad perfecta

Theorem

One-time pad es un cifrado de Shannon con seguridad perfecta.

One-time pad y seguridad perfecta

Theorem

One-time pad es un cifrado de Shannon con seguridad perfecta.

Proof.

Para cada mensaje $m \in \mathcal{M}$ y cifrado $c \in \mathcal{C}$ hay sólo una $k \in \mathcal{K}$ que satisface

$$k \oplus m = c,$$

$k = m \oplus c$. Entonces $\mathcal{E}$ satisface (ii) del teorema anterior ($N_c = 1$ para cada c).

□

Análisis de one-time pad con largo variable

Encriptemos ‘yes’ y ‘no’ usando one-time pad de longitud variable

Análisis de one-time pad con largo variable

Encriptemos ‘yes’ y ‘no’ usando one-time pad de longitud variable

‘yes’ $= m_0$, ‘no’ $= m_1$. $|m_0| = 3$ y $|m_1| = 2$.

Análisis de one-time pad con largo variable

Encriptemos ‘yes’ y ‘no’ usando one-time pad de longitud variable

‘yes’ $= m_0$, ‘no’ $= m_1$. $|m_0| = 3$ y $|m_1| = 2$.

Luego $|c_0| = 3$ y $|c_1| = 2$.

Análisis del cifrado por sustitución

- Análisis de frecuencia: letras más y menos frecuentes.
- Se preservan patrones: longitudes de palabra, letras dobles y repeticiones.
- Ataques prácticos: texto conocido e *ingeniería social* (adivinar frases o palabras frecuentes, headers, etc).

Caracterización de la seguridad perfecta

Theorem

Sea $\mathcal{E} = (E, D)$ un cifrado de Shannon definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Consideremos un experimento probabilístico en donde $\boldsymbol{k}$ es una variable aleatoria uniformemente distribuida en $\mathcal{K}$. Entonces $\mathcal{E}$ es perfectamente segura si y solo si para cada predicado ϕ en $\mathcal{C}$, para cada $m_0, m_1 \in \mathcal{M}$, tenemos

$$\Pr[\phi(E(\boldsymbol{k}, m_0))] = \Pr[\phi(E(\boldsymbol{k}, m_1))]$$

Caracterización de la seguridad perfecta

Theorem

Sea $\mathcal{E} = (E, D)$ un cifrado de Shannon definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Consideremos un experimento probabilístico en donde $\boldsymbol{k}$ es una variable aleatoria uniformemente distribuida en $\mathcal{K}$. Entonces $\mathcal{E}$ es perfectamente segura si y solo si para cada predicado ϕ en $\mathcal{C}$, para cada $m_0, m_1 \in \mathcal{M}$, tenemos

$$\Pr[\phi(E(\boldsymbol{k}, m_0))] = \Pr[\phi(E(\boldsymbol{k}, m_1))]$$

El cifrado no revela nada sobre el mensaje.

Caracterización de la seguridad perfecta

Theorem

Sea $\mathcal{E} = (E, D)$ un cifrado de Shannon definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Consideremos un experimento probabilístico en donde $\boldsymbol{k}$ es una variable aleatoria uniformemente distribuida en $\mathcal{K}$. Entonces $\mathcal{E}$ es perfectamente segura si y solo si para cada predicado ϕ en $\mathcal{C}$, para cada $m_0, m_1 \in \mathcal{M}$, tenemos

$$\Pr[\phi(E(\boldsymbol{k}, m_0))] = \Pr[\phi(E(\boldsymbol{k}, m_1))]$$

El cifrado no revela nada sobre el mensaje.

$$\Pr[\boldsymbol{m} = m | \boldsymbol{c} = c] = \Pr[\boldsymbol{m} = m]$$

Caracterización de la seguridad perfecta

Theorem

Sea $\mathcal{E} = (E, D)$ un cifrado de Shannon definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Consideremos un experimento probabilístico en donde $\boldsymbol{k}$ es una variable aleatoria uniformemente distribuida en $\mathcal{K}$. Entonces $\mathcal{E}$ es perfectamente segura si y solo si para cada predicado ϕ en $\mathcal{C}$, para cada $m_0, m_1 \in \mathcal{M}$, tenemos

$$\Pr[\phi(E(\boldsymbol{k}, m_0))] = \Pr[\phi(E(\boldsymbol{k}, m_1))]$$

El cifrado no revela nada sobre el mensaje.

$$\Pr[\boldsymbol{m} = m | \boldsymbol{c} = c] = \Pr[\boldsymbol{m} = m]$$

$$\Pr[\boldsymbol{c} = c | \boldsymbol{m} = m] = \Pr[\boldsymbol{c} = c]$$

Malas noticias

... perfect security is such a powerful notion that one can really do no better than the one-time pad.

Malas noticias

... perfect security is such a powerful notion that one can really do no better than the one-time pad.

Theorem

Sea $\mathcal{E} = (E, D)$ un cifrado de Shannon definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Si $\mathcal{E}$ tiene seguridad perfecta, entonces $|\mathcal{K}| \geq |\mathcal{M}|$.

- La única forma de tener un cifrado con seguridad perfecta es tener keys tan largas como los mensajes.

- La única forma de tener un cifrado con seguridad perfecta es tener keys tan largas como los mensajes.

Relajemos los requerimientos de seguridad.

- La única forma de tener un cifrado con seguridad perfecta es tener keys tan largas como los mensajes.

Relajemos los requerimientos de seguridad.

Nos mantenemos en el mundo real:

- Adversarios computacionalmente viables:
  - Computadoras reales.
  - Cantidad razonable de tiempo y memoria.

Cifrado computacional

Definition

Un cifrado computacional $\mathcal{E} = (E, D)$ es un par de *algoritmos eficientes* (es decir, que se ejecutan en tiempo polinomial).

Definition

Un cifrado computacional $\mathcal{E} = (E, D)$ es un par de *algoritmos eficientes* (es decir, que se ejecutan en tiempo polinomial).

Permitimos que E sea un algoritmo probabilístico

$$c \xleftarrow{\$} E(k, m)$$

Definition

Un cifrado computacional $\mathcal{E} = (E, D)$ es un par de *algoritmos eficientes* (es decir, que se ejecutan en tiempo polinomial).

Permitimos que E sea un algoritmo probabilístico

$$c \xleftarrow{\$} E(k, m)$$

Luego la propiedad de correctitud adaptada es:
Para todas las keys $k \in \mathcal{K}$ y mensajes $m \in \mathcal{M}$, si ejecutamos

$$c \xleftarrow{\$} E(k, m), m' \leftarrow D(k, c),$$

entonces $m = m'$ con probabilidad 1.

Seguridad semántica

En seguridad perfecta habíamos visto que para todos los predicados ϕ y todos los mensajes m_0, m_1 tenemos

$$\Pr[\phi(E(\boldsymbol{k}, m_0))] = \Pr[\phi(E(\boldsymbol{k}, m_1))]$$

Seguridad semántica

En seguridad perfecta habíamos visto que para todos los predicados ϕ y todos los mensajes m_0, m_1 tenemos

$$\Pr[\phi(E(\boldsymbol{k}, m_0))] = \Pr[\phi(E(\boldsymbol{k}, m_1))]$$

Ahora en lugar de pedir que sean iguales, buscaremos que estén cerca:

$$|\Pr[\phi(E(\boldsymbol{k}, m_0))] - \Pr[\phi(E(\boldsymbol{k}, m_1))]| \leq \epsilon$$

para un ϵ muy pequeño o despreciable (negligible).

Seguridad semántica

En seguridad perfecta habíamos visto que para todos los predicados ϕ y todos los mensajes m_0, m_1 tenemos

$$\Pr[\phi(E(\boldsymbol{k}, m_0))] = \Pr[\phi(E(\boldsymbol{k}, m_1))]$$

Ahora en lugar de pedir que sean iguales, buscaremos que estén cerca:

$$|\Pr[\phi(E(\boldsymbol{k}, m_0))] - \Pr[\phi(E(\boldsymbol{k}, m_1))]| \leq \epsilon$$

para un ϵ muy pequeño o despreciable (negligible).

Seguro para todos los propósitos prácticos.

Detalles previos a la definición formal

Describimos un *attack game* o juego de ataque entre dos partes:

- *challenger* o retador
- *adversary* o adversario

Mediante el juego definimos un espacio de probabilidades, lo que define la *ventaja del adversario*, que se determina por la probabilidad de uno o mas eventos.

Juego de ataque a la seguridad semántica

Definition

Definimos el juego de ataque a la seguridad semántica: para un esquema $\mathcal{E} = (E, D)$ definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, para un adversario $\mathcal{A}$, y para $b = 0, 1$ definimos:

Experimento b:

- El adversario computa $m_0, m_1 \in \mathcal{M}$ del mismo tamaño y lo envía al challenger.
- El challenger computa $k \xleftarrow{\$} \mathcal{K}, c \xleftarrow{\$} E(k, m_b)$ y envía c al adversario.
- El adversario devuelve un bit $\hat{b} \in \{0, 1\}$

Para $b = 0, 1$, sea W_b el evento que $\mathcal{A}$ devuelve 1 en el experimento b. Definimos la ventaja ante la seguridad semántica con respecto a $\mathcal{E}$ como

$$SSadv[\mathcal{A}, \mathcal{E}] = |\Pr[W_0] - \Pr[W_1]|$$

Definición de seguridad semántica

Definition

Un esquema $\mathcal{E}$ es semánticamente seguro si para todo adversario eficiente $\mathcal{A}$, el valor $SSadv[\mathcal{A}, \mathcal{E}]$ es negligible.

“Eficiente”, “negligible”, “poly-bounded”

Veamos ideas intuitivas de estos conceptos

- Despreciable (negligible) significa tan pequeño que se puede considerar igual a cero para todos los propósitos prácticos.
- Un adversario eficiente es uno que corre en tiempo razonable.
- Un valor N es llamado super-poly si $1/N$ es negligible.
- Un valor poly-bounded es un número con tamaño “razonable”. En particular, el tiempo de ejecución de un adversario eficiente es poly-bounded.

Seguridad semántica con experimento bit guessing

Definition

Para un esquema $\mathcal{E} = (E, D)$ definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, para un adversario $\mathcal{A}$, y para $b = 0, 1$ definimos:

Experimento b:

- El adversario computa $m_0, m_1 \in \mathcal{M}$ del mismo tamaño y lo envía al challenger.
- El challenger computa $k \xleftarrow{\$} \mathcal{K}$, $c \xleftarrow{\$} E(k, m_b)$ y envía c al adversario.
- El aversario devuelve un bit $\hat{b} \in \{0, 1\}$

Para $b = 0, 1$, sea W_b el evento que $\mathcal{A}$ devuelve 1 en el experimento b.

Seguridad semántica con experimento bit guessing

Definition

Para un esquema $\mathcal{E} = (E, D)$ definido sobre $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ y para un adversario $\mathcal{A}$ el juego consiste en:

- El adversario computa $m_0, m_1 \in \mathcal{M}$ del mismo tamaño y lo envía al challenger.
- El challenger computa $b \xleftarrow{\$} \{0,1\}, k \xleftarrow{\$} \mathcal{K}, c \xleftarrow{\$} E(k, m_b)$ y envía c al adversario.
- El aversario devuelve un bit $\hat{b} \in \{0,1\}$

El adversario gana si $\hat{b} = b$.

Transformar a bit guessing

Cuánta ventaja tiene un adversario en la forma bit guessing comparado al juego anterior?

Transformar a bit guessing

Cuánta ventaja tiene un adversario en la forma bit guessing comparado al juego anterior?

Theorem

Para cada esquema de cifrado $\mathcal{E}$ y adversario $\mathcal{A}$, tenemos

$$SSadv[\mathcal{A}, \mathcal{E}] = 2 \cdot SSadv^*[\mathcal{A}, \mathcal{E}]$$

Generalización

Dado una propiedad de seguridad X para un esquema criptográfico $\mathcal{S}$. Para $b = 0, 1$ definimos W_b el evento que $\mathcal{A}$ devuelva 1 en el experimento b, y definimos

$$Xadv[\mathcal{A}, \mathcal{S}] = |\Pr[W_0] - \Pr[W_1]|$$

la ventaja de $\mathcal{A}$ en relación a X.

Podemos definir la versión bit-guessing de este juego de ataques, en el que el challenger elige aleatoriamente $b \in \{0, 1\}$ y corre el experimento b. Si W es el evento en que el adversario devuelve un output igual a b, definimos

$$Xadv^*[\mathcal{A}, \mathcal{S}] = |\Pr[W] - \frac{1}{2}|$$

la ventaja de $\mathcal{A}$ en relación a X bit-guessing.

Tenemos además que

$$Xadv[\mathcal{A}, \mathcal{S}] = 2 \cdot Xadv^*[\mathcal{A}, \mathcal{S}]$$

Negligible

Definition

Una función $f : \mathbb{Z}_{\geq 1} \to \mathbb{R}$ se llama *negligible* si para todo $c \in \mathbb{R}_{>0}$ existe $n_0 \in \mathbb{Z}_{\geq 1}$ tal que, para todo entero $n \geq n_0$, se cumple que

$$|f(n)| < \frac{1}{n^c}.$$

Negligible

Definition

Una función $f : \mathbb{Z}_{\geq 1} \to \mathbb{R}$ se llama *negligible* si para todo $c \in \mathbb{R}_{>0}$ existe $n_0 \in \mathbb{Z}_{\geq 1}$ tal que, para todo entero $n \geq n_0$, se cumple que

$$|f(n)| < \frac{1}{n^c}.$$

Theorem

Una función $f : \mathbb{Z}_{\geq 1} \to \mathbb{R}$ es negligible si y sólo si para todo $c > 0$ se cumple que

$$\lim_{n \to \infty} f(n) n^c = 0.$$

Super-poly

Definition

Una función $f : \mathbb{Z}_{\geq 1} \to \mathbb{R}$ se llama *super-poly* si $\frac{1}{f}$ es despreciable.

Poly-bounded

Definition

Una función $f : \mathbb{Z}_{\geq 1} \to \mathbb{R}$ se llama *acotada por polinomios* (*poly-bounded*) si existen $c, d \in \mathbb{R}_{>0}$ tales que, para todo entero $n \geq 0$, se cumple que

$$|f(n)| \leq n^c + d.$$

Poly-bounded

Definition

Una función $f : \mathbb{Z}_{\geq 1} \to \mathbb{R}$ se llama *acotada por polinomios* (*poly-bounded*) si existen $c, d \in \mathbb{R}_{>0}$ tales que, para todo entero $n \geq 0$, se cumple que

$$|f(n)| \leq n^c + d.$$

Nota: si f es una función acotada por polinomios, entonces $1/f$ definitivamente no es una función despreciable.

Formalidades de cifrados computacionales

En el modelado de cifrados computacionales $\mathcal{E}$, usamos familias de keys, mensajes y cifrados indexadas por

- Un **parámetro de seguridad** denotado por λ (número entero).
- Un **parámetro de sistema** denotado por Λ (string).

Entonces en lugar de conjuntos finitos $\mathcal{K}$, $\mathcal{M}$ y $\mathcal{C}$, tenemos familias de conjuntos

$$\{\mathcal{K}_{\lambda,\Lambda}\}_{\lambda,\Lambda}, \qquad \{\mathcal{M}_{\lambda,\Lambda}\}_{\lambda,\Lambda}, \qquad \{\mathcal{C}_{\lambda,\Lambda}\}_{\lambda,\Lambda}$$

Formalidades de cifrados computacionales

En el modelado de cifrados computacionales $\mathcal{E}$, usamos familias de keys, mensajes y cifrados indexadas por

- Un **parámetro de seguridad** denotado por λ (número entero).
- Un **parámetro de sistema** denotado por Λ (string).

Entonces en lugar de conjuntos finitos $\mathcal{K}$, $\mathcal{M}$ y $\mathcal{C}$, tenemos familias de conjuntos

$$\{\mathcal{K}_{\lambda,\Lambda}\}_{\lambda,\Lambda}, \qquad \{\mathcal{M}_{\lambda,\Lambda}\}_{\lambda,\Lambda}, \qquad \{\mathcal{C}_{\lambda,\Lambda}\}_{\lambda,\Lambda}$$

Al variarlos el sistema es mas seguro o más eficiente.

Definition (máquina interactiva eficiente)

Decimos que M es una máquina interactiva eficiente si existe una función acotada por polinomios (poly-bounded) t y una función negligible ϵ tal que para todo entorno (incluso los no acotados polinomialmente) la probabilidad de que el tiempo de ejecución de M exceda $t(\lambda)$ sea a lo sumo $\epsilon(\lambda)$.

Definition (máquina interactiva eficiente)

Decimos que M es una máquina interactiva eficiente si existe una función acotada por polinomios (poly-bounded) t y una función negligible ϵ tal que para todo entorno (incluso los no acotados polinomialmente) la probabilidad de que el tiempo de ejecución de M exceda $t(\lambda)$ sea a lo sumo $\epsilon(\lambda)$.

Un adversario eficiente es simplemente una máquina interactiva eficiente.

¡Gracias!