UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Integridad de Mensajes

Alan Reyes-Figueroa

Criptografía y Cifrado de Información (Aula 10) 31.agosto.2021

Integridad de Mensajes

En las próximas aulas discutimos el tema de autenticación de mensajes. Ahora el objetivo no es cifrar un mensaje, sino más bien autenticar y validar que dicho mensaje no ha sido alterado.

Más adelante veremos cómo incorporar de forma conjunta cifrado y autenticación.

Objetivo: Integridad, no confidencialidad.

Ejemplos:

- Archivos binarios en disco:
  - Archivos de sistema operativo (Windows, Linux, ...)
  - Descargas de instaladores o binarios.

  (importante asegurarse que ho hayan sido modificados por algún virus o *malware*).
- *Banners* o *adds* en sitios web.
  (el creador del add quiere asegurarse que el contenido de su publicidad no sea alterado).

MACs

La forma usual de hacer esto es a través de una Código Autenticador de Mensajes (*Message Authentication Code*, *MAC*).

Definición

Un ***código autenticador de mensajes*** *(MAC) es un par* $I = (S, V)$ *definido sobre* $(\mathcal{K}, \mathcal{M}, \mathcal{T})$*, con funciones* $S : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ *y* $V : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0, 1\}$*, donde*

- *la función de signatura (signing algorithm)* $tag = S(\mathbf{k}, \mathbf{m})$ *produce* $tag \in \mathcal{T}$*,*
- *la función verificadora* $V(\mathbf{k}, \mathbf{m}, tag)$ *responde Falso o Verdadero.*

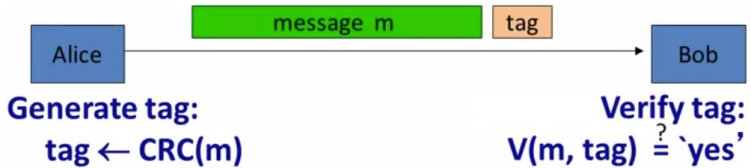
MACs

De nuevo, requerimos que se cumpla una condición de consistencia

$$V(\mathbf{k}, \mathbf{m}, S(\mathbf{k}, \mathbf{m})) = 1, \quad \forall \mathbf{k} \in \mathcal{K}, \forall \mathbf{m} \in \mathcal{M}.$$

Importante! Verificar integridad requiere compartir una clave secreta **k**.

Ejemplo:



- CRC = *Cyclic Redundancy Check*.
- Un adversario puede fácilmente modificar un mensaje y recalcular el CRC.
- CRC está diseñado para detectar errores aleatorios, no malintencionados.

MACs Seguros

Supongamos que un adversario puede elegir mensajes $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_q$ de manera que éste puede conocer los respectivos tags de cada mensaje

$$tag_i = S(\mathbf{k}, \mathbf{m}_i), \quad \text{para } i = 1, 2, \ldots, q.$$

Y supongamos que el objetivo del atacante es lograr producir un nuevo par (mensaje, tag) que sea válido. Esto se conoce como **falsificación existencial** (*existential forgery*).

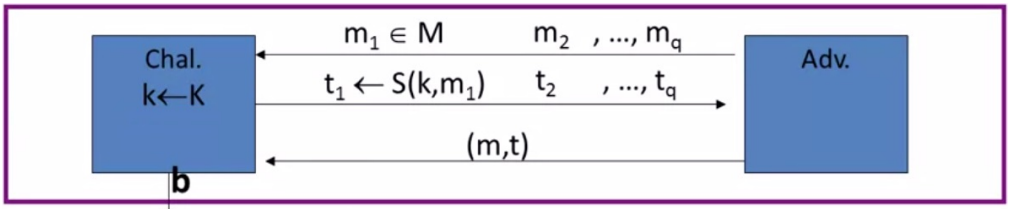
- producir $(\mathbf{m}, t)$ nuevo

$$(\mathbf{m}, t) \notin \{(\mathbf{m}_1, tag_1), (\mathbf{m}_2, tag_2), \ldots, (\mathbf{m}_q, tag_q)\}.$$

Un MAC es **seguro** si no se puede hacer falsificación existencial, esto es, un atacante no puede producir pares nuevos $(\mathbf{m}, tag)$ que sean válidos a partir de una colección $\{((\mathbf{m}_i, i)\}_{i=1}^{q}$ de pares válidos.

MACs Seguros

Para una MAC $I = (S, V)$ y un adversario A definimos el siguiente mecanismo:



Definición

*Una MAC $I = (S, V)$ es un **MAC seguro** si para todo algoritmos eficiente A, la ventaja*

$$\mathrm{Adv}_{MAC}(A, I) = \mathbb{P}(\textit{Chal. responde } 1) \textit{ es negligible.}$$

MACs Seguros

Consecuencia, la longitud del tag no debe ser muy corta.

En la práctica se una tags de longitud 90, 96 ó 128 bits.

$$\mathbb{P}(\text{adversario adivinar un tag}) = \frac{1}{96} \approx 1.26 \times 10^{-29}.$$

Aplicación

Instalación de sistema operativo:
Supongamos que se intala un sistema operativo en una máquina (*e.g.* Windows, Ubuntu, ...) Al momento de la instalación, la máquina calcula

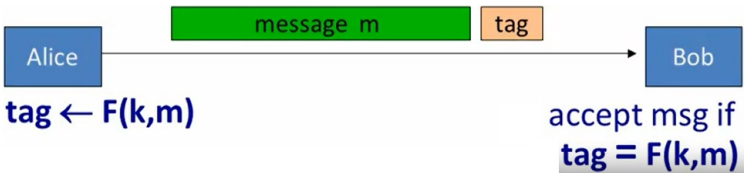
- Si luego un virus infecta y corrompe algunos archivos del sistema,
- el usuario reinicia el sistema desde un OS limpio (por ejemplo, un USB, o SSD),
- una vez el sistema reinicia, el usuario da su password. El nuevo sistema verifica el MAC de cada archivo de sistema $\Rightarrow$ los archivos corruptos son detectados.

Construcción de MACs Seguros

Ejemplo: Una PRF segura $\Rightarrow$ MAC seguro.

Para una función pseudo-aleatoria (PRF) segura $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, definimos un MAC de la siguiente forma. Hacemos $I_F = (S, V)$, donde

$$
\begin{aligned}
S(\mathbf{k}, \mathbf{m}) &= F(\mathbf{k}, \mathbf{m}), \\
V(\mathbf{k}, \mathbf{m}, t) &= \begin{cases} 1, & \text{si } t = F(\mathbf{k}, \mathbf{m}); \\ 0, & \text{caso contrario.} \end{cases}
\end{aligned}
$$



Construcción de MACs Seguros

Teorema

Si $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ es una PRF segura, y $\frac{1}{|\mathcal{Y}|}$ es negligible (esto es, $|\mathcal{Y}|$ es grande), entonces el MAC I_F es seguro.

En particular, para todo adversario MAC eficiente A, atacanto al MAC I_F, existe un adversario PRF eficiente B, atacando a la función F, tal que

$$\mathrm{Adv}_{MAC}(A, I_F) = \mathrm{Adv}_{PRF}(B, F) + \tfrac{1}{|\mathcal{Y}|}.$$

Así, I_F es seguro, desde que $|\mathcal{Y}|$ sea largo, digamos $|\mathcal{Y}| \geq 2^{80}$.
(Por eso conviene usar tags de tamaño ≥ 80 (*e.g.* 96 bits o 128 bits).

Construcción de MACs Seguros

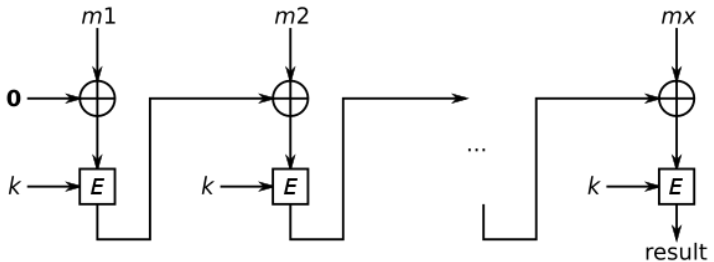
Ejemplos:

- AES: Vimos que AES es una PRF segura. Se puede usar para producir un MAC para mensajes de 128-bits.
- **Pregunta:** ¿Cómo convertir una small-MAC en un Big-MAC?
  Este es el llamado *problema de MacDonalds*.

  Esto es cómo podemos, a partir de una MAC que actúa en cadenas pequeñas (128 bits), en un MAC seguro que actúe sobre cadenas extensas (gigabytes)?
- Dos construcciones populares son:
  - **CBC-MAC**: (banca electrónica ACH, ANSI X9.9, X9.19, FIPS 186-3).
  - **HMAC**: (protocolos web: SSL, IPsec, SSH, ...)
- Estas convierten una PRF pequeña y la transforman en una PRF grande.

Construcción de MACs Seguros

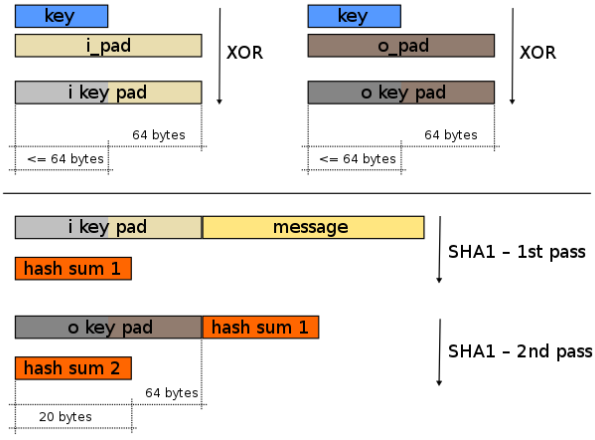
CBC-MAC: Usa un esquema de cifrado en bloques, en modo CBC (*Cipher Block Chain*).



Esquema de un CBC-MAC.

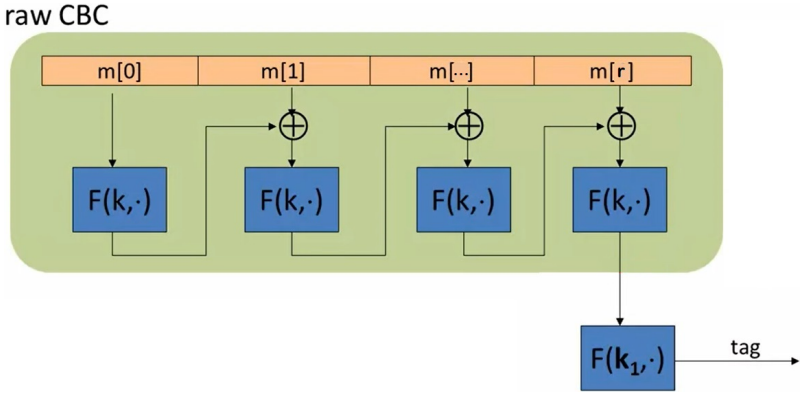
Construcción de MACs Seguros

HMAC: Usa un esquema basado en funciones hash criptográficas.



Encrypted CBC-MAC

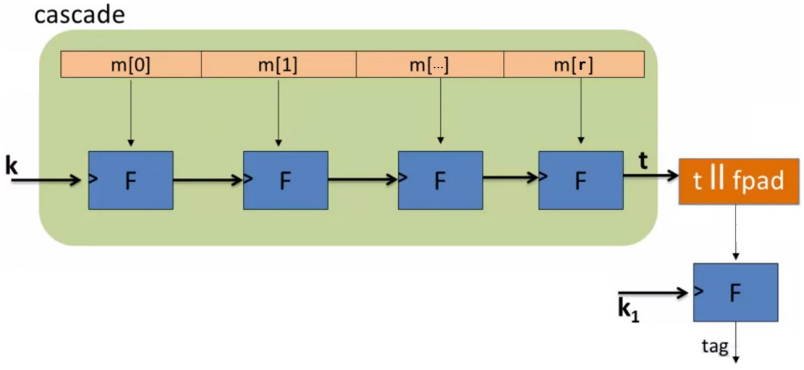
Dada una PRP segura $F : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ (por ejemplo un AES), definimos una nueva PRF como $F_{eCBC} : \mathcal{K}^2 \times \mathcal{X}^{\leq L} \to \mathcal{X}$.



Esquema eCBC-MAC.

Nested MAC

Dada una PRF segura $F : \mathcal{K} \times \mathcal{X} \to \mathcal{K}$ (por ejemplo un AES), definimos una nueva PRF como $F_{NMAC} : \mathcal{K}^2 \times \mathcal{X}^{\leq L} \to \mathcal{X}$.



Esquema NMAC.

Aquí $\mathcal{X}^{\leq L} = \bigcup_{i=1}^{L} \mathcal{X}$.

Seguridad de MACs

Teorema

Para cualquier $L > 0$, y para todo algoritmo adversario PRF eficiente A que ataca a F_{eCBC} ó F_{NMAC} con q consultas (esto es el adversario tiene a disposición q pares $(\mathbf{m}, tag)$), existen adversiarios eficientes B tales que

$$\begin{aligned} \mathrm{Adv}_{PRF}(A, F_{eCBC}) &\leq \mathrm{Adv}_{PRP}(B, F) + 2\frac{q^2}{|\mathcal{X}|}, \\ \mathrm{Adv}_{PRF}(A, F_{NMAC}) &\leq qL\,\mathrm{Adv}_{PRP}(B, F) + \frac{q^2}{2|\mathcal{X}|}. \end{aligned}$$

Consecuencias:

- El esquema CBC-MAC es seguro desde que $q << |\mathcal{X}|^{1/2}$.
- El esquema NMAC es seguro desde que $q << |\mathcal{K}|^{1/2}$ (2^{64} para el AES-128).

Seguridad de MACs

Un ejemplo:

$$\text{Adv}_{PRF}(A, F_{eCBC}) \leq \text{Adv}_{PRP}(B, F) + 2\frac{q^2}{|\mathcal{X}|}.$$

Supongamos, por ejemplo, que queremos $\text{Adv}_{PRF}(A, F_{eCBC}) \leq \frac{1}{2^{32}}$, entonces requerimos

$$\frac{q^2}{|\mathcal{X}|} \leq \frac{1}{2^{32}}.$$

- AES: $|\mathcal{X}| = 2^{128}$ (128 bits). $\Rightarrow\ q^2 \leq 2^{96} \Rightarrow\ q \leq 2^{48}$.
  Así, después de 2^{48} mensajes debemos cambiar de clave **k**.
- 3DES: $|\mathcal{X}| = 2^{64}$ (64 bits). $\Rightarrow\ q^2 \leq 2^{32} \Rightarrow\ q \leq 2^{16}$.
  Así, después de $2^{16} = 65536$ mensajes debemos cambiar de clave. Por eso 3DES hoy no es seguro.

Seguridad de MACs

Comparación:

- **eCBC-MAC** es comúnente usado con el cifrado AES, produciendo un MAC basado en AES-128.
  - suele utilizar el modo de encripción CCM (*counter with CBC-MAC mode*).
  - Es un NIST estándar, llamado **CMAC**.
- **NMAC** no suele usarse con AES o 3DES.
  - La razón principal es que se necesitaría cambiar la clave del AES en cada bloque, y requiere estar calculando la expansión de clave AES.
  - Sin embargo, NMAC es la base del otro esquema de códigos de verificación HMAC (basados en bunciones hash).