Anexo Clase 4

8 de abril de 2024

CBC-MAC

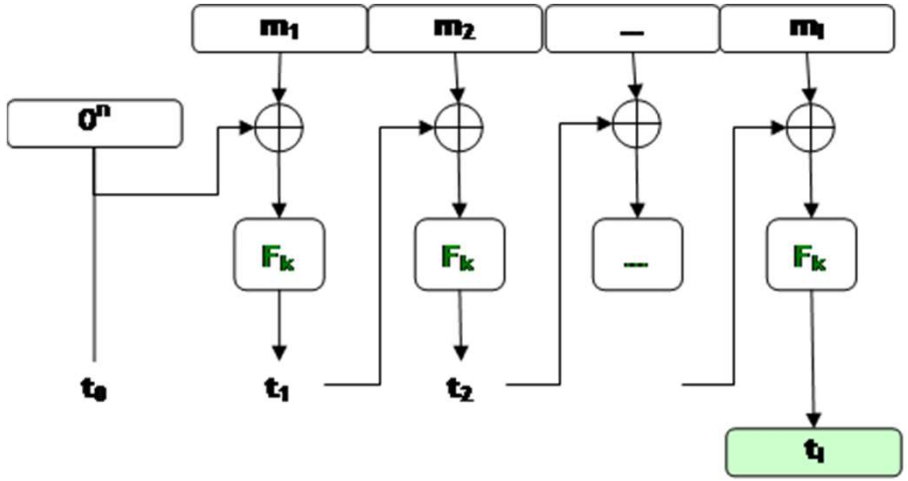
CBC-MAC (Gen, Mac, Vrfy)

- Gen: $k \leftarrow \{0,1\}^n$
- Mac:
  - $|k| = n$
  - $|m| = l(n).n$

El mensaje m se parte en l bloques de longitud n

$t_0 = 0^n$

$t_i = F_k(t_{i-1} \oplus m_i) \Rightarrow$ emite t_l



La construcción anterior es infalsificable SÓLO si se permiten mensajes de una misma longitud.

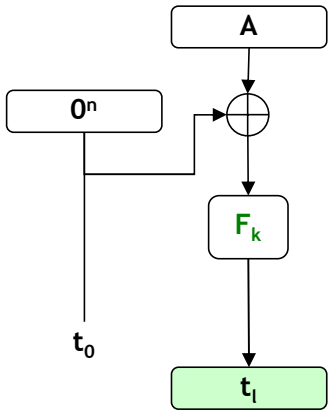
Ejemplo de falsificación si se permiten mensajes de distintas longitudes:

Experimento $MAC - Forge_{A,\pi}(n)$

1) k←Gen(n)
2) El adversario A que tiene acceso al oráculo, solicita el $Mac_k(m_1)$ donde m_1=A

   (A tiene el tamaño de un bloque)

Q = {<A, $F_k(A)$>}

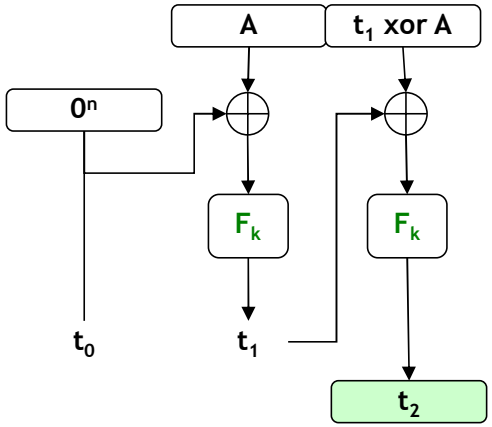


El adversario A puede obtener un par <m_2,t_2> válido haciendo

m_2 = A|| $t_1$xor A

El || es concatenacion

t_2 = t_1

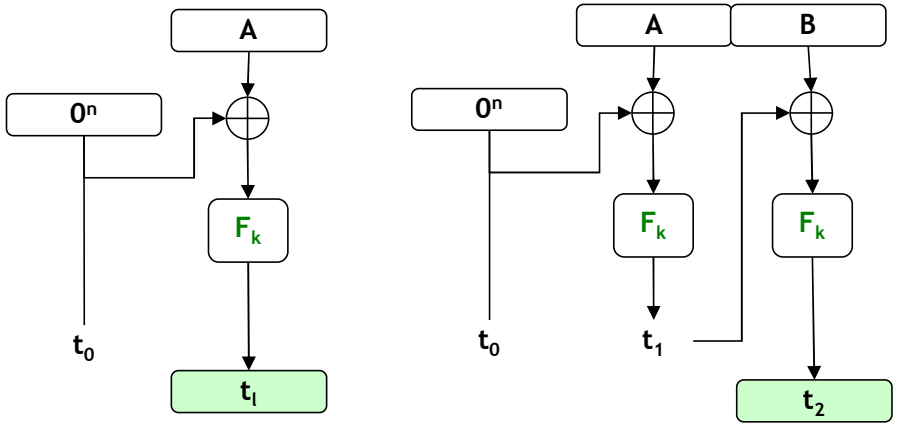


Otra forma de hacerlo:

Experimento $MAC - Forge_{A,\pi}(n)$

1) k←Gen(n)
2) El adversario A que tiene acceso al oráculo, solicita el $Mac_k(m_1)$ donde m_1=A
   y
   $Mac_k(m_2)$ donde m_2=A||B
   (A y B tienen el tamaño un bloque)

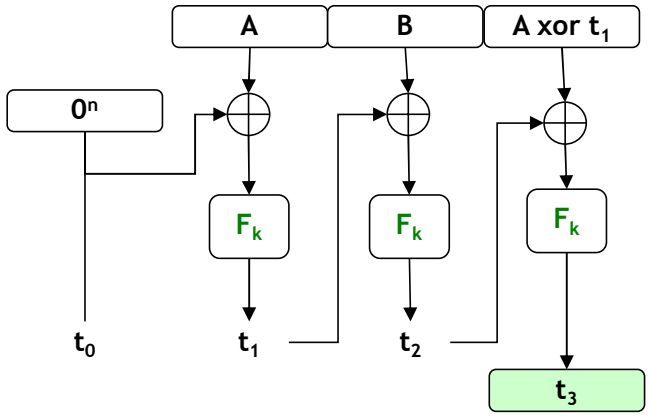
Q = {<A, F_k(A)>, <A||B, F_k(B xor F_k(A))>, }



El adversario A puede obtener un par <m_3,t_3> válido haciendo

m_3 = A||B||(A xor t_1)

t_3 = t_2



Opciones seguras para CBC-MAC para mensajes de longitud arbitraria:

Opción 1: $k_l := F_k(|m|)$ y $t \leftarrow CBC - MAC_{kl}(m)$

Opción 2: $m' := |m| \,\|\, m$ y $t \leftarrow CBC - MAC_k(m')$

Opción 3:

$$k_1 \leftarrow \{0,1\}^n$$
$$k_2 \leftarrow \{0,1\}^n$$
$$t \leftarrow CBC - MAC_{k1}(m)$$
$$\hat{t} \leftarrow F_{k2}(t)$$

¿por qué no es seguro poner la longitud del mensaje al final?

Experimento $MAC - Forge_{A,\pi}(n)$

El adversario A que tiene acceso al oráculo, que siempre efectúa:

$Mac_k(m)$=CBC-MACk(m') donde m'=m || |m|

Efectúa consultas para:

m_1=AAA

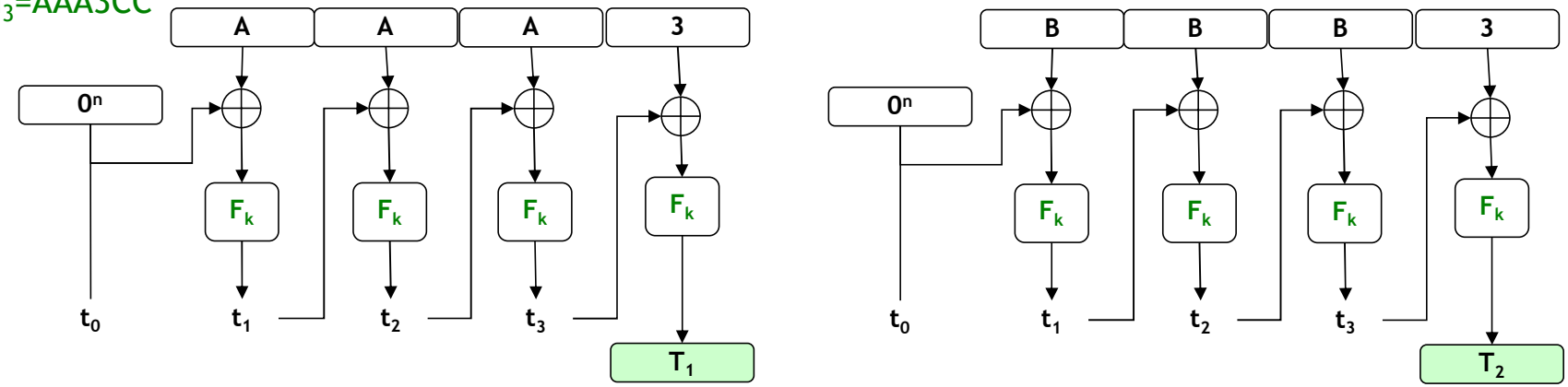
m_2=BBB

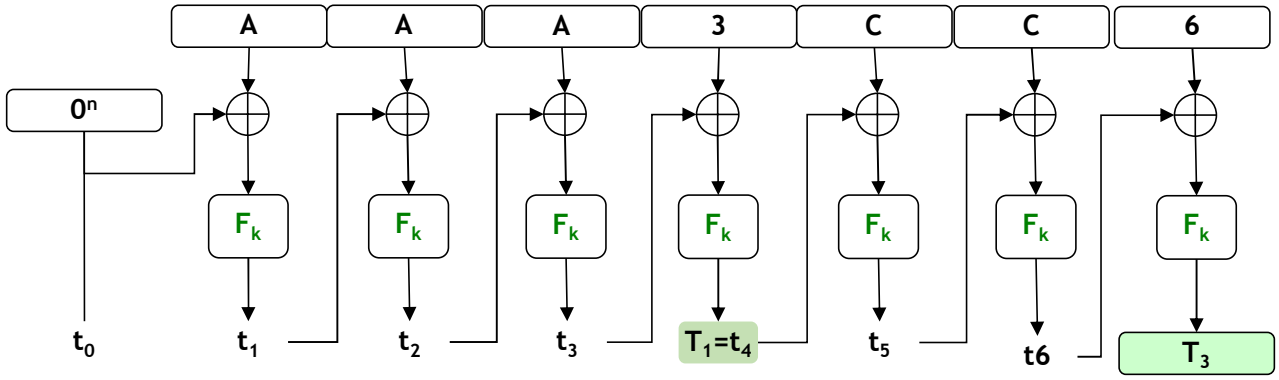
m_3=AAA3CC

Y obtiene una etiqueta válida para

m_4 = BBB3XC

Donde X = T_1 xor T_2 xor C





La etiqueta válida será

$T_4 = T_3$