Criptografie și Securitate

- Prelegerea 13 -
Confidențialitate și autentificarea mesajelor

Adela Georgescu, Ruxandra F. Olimid

Facultatea de Matematică și Informatică
Universitatea din București

Cuprins

Confidențialitate și integritate

- Am văzut cum putem obține confidențialitate folosind scheme de criptare;

Confidențialitate și integritate

- Am văzut cum putem obține confidențialitate folosind scheme de criptare;
- Am văzut cum putem garanta integritatea datelor folosind MAC-uri;

Confidențialitate și integritate

- Am văzut cum putem obține confidențialitate folosind scheme de criptare;
- Am văzut cum putem garanta integritatea datelor folosind MAC-uri;
- In practică avem nevoie de ambele proprietăți de securitate: confidențialitate și integritatea datelor;

Confidențialitate și integritate

- Am văzut cum putem obține confidențialitate folosind scheme de criptare;
- Am văzut cum putem garanta integritatea datelor folosind MAC-uri;
- In practică avem nevoie de ambele proprietăți de securitate: confidențialitate și integritatea datelor;
- Nu orice combinație de schemă de criptare sigură și MAC sigur oferă cele două proprietăți de securitate!

Confidențialitate și integritate

- Iată trei abordari uzuale pentru a combina criptarea și autentificarea mesajelor:

Confidențialitate și integritate

- Iată trei abordari uzuale pentru a combina criptarea și autentificarea mesajelor:
  1. *Criptare-și-autentificare*: criptarea și autentifcarea se fac independent. Pentru un mesaj clar m, se transmite mesajul criptat $\langle c, t \rangle$ unde

     $$c \leftarrow \mathrm{Enc}_{k_1}(m) \text{ și } t \leftarrow \mathrm{Mac}_{k_2}(m)$$

     La recepție, $m = \mathrm{Dec}_{k_1}(c)$ și dacă $\mathrm{Vrfy}_{k_2}(m, t) = 1$, atunci întoarce m; altfel întoarce $\perp$.

Confidențialitate și integritate

- Iată trei abordari uzuale pentru a combina criptarea și autentificarea mesajelor:

  1. *Criptare-și-autentificare*: criptarea și autentifcarea se fac independent. Pentru un mesaj clar m, se transmite mesajul criptat $\langle c, t \rangle$ unde

     $$c \leftarrow \mathrm{Enc}_{k_1}(m) \text{ și } t \leftarrow \mathrm{Mac}_{k_2}(m)$$

     La recepție, $m = \mathrm{Dec}_{k_1}(c)$ și dacă $\mathrm{Vrfy}_{k_2}(m, t) = 1$, atunci întoarce m; altfel întoarce $\perp$.

  2. *Autentificare-apoi-criptare*: întâi se calculează tag-ul t apoi mesajul și tag-ul sunt criptate împreună

     $$t \leftarrow \mathrm{Mac}_{k_2}(m) \text{ si } c \leftarrow \mathrm{Enc}_{k_1}(m||t)$$

     La recepție, $m||t = \mathrm{Dec}_{k_1}(c)$ și dacă $\mathrm{Vrfy}_{k_2}(m, t) = 1$, atunci întoarce m; altfel întoarce $\perp$.

Confidențialitate și integritate

3. *Criptare-apoi-autentificare*: întâi se criptează mesajul și apoi se calculează tag-ul

$$c \leftarrow \mathrm{Enc}_{k_1}(m) \text{ si } t \leftarrow \mathrm{Mac}_{k_2}(c)$$

La recepție, se verifică întâi t înainte de a decripta c; aceasta este chiar construcția pentru schema CCA-sigură.

Confidențialitate și integritate

3. *Criptare-apoi-autentificare*: întâi se criptează mesajul și apoi se calculează tag-ul

$$c \leftarrow \mathrm{Enc}_{k_1}(m) \text{ si } t \leftarrow \mathrm{Mac}_{k_2}(c)$$

La recepție, se verifică întâi t înainte de a decripta c; aceasta este chiar construcția pentru schema CCA-sigură.

- Vom analiza fiecare abordare la instanțierea cu o schemă de criptare CPA-sigură și un MAC sigur (cu tag-uri unice).

Confidențialitate și integritate

3. *Criptare-apoi-autentificare*: întâi se criptează mesajul și apoi se calculează tag-ul

$$c \leftarrow \mathrm{Enc}_{k_1}(m) \text{ si } t \leftarrow \mathrm{Mac}_{k_2}(c)$$

La recepție, se verifică întâi t înainte de a decripta c; aceasta este chiar construcția pentru schema CCA-sigură.

- Vom analiza fiecare abordare la instanțierea cu o schemă de criptare CPA-sigură și un MAC sigur (cu tag-uri unice).
- Ne vor interesa doar acele abordări care oferă confidențialitate și integritate pentru *orice* schemă de criptare sigură și *orice* MAC sigur.

Securitate

- Pentru a analiza care combinație de confidențialitate și integritate este sigură, definim ce înseamnă "combinație sigură";

Securitate

- Pentru a analiza care combinație de confidențialitate și integritate este sigură, definim ce înseamnă "combinație sigură";
- Introducem noțiunea de *schemă de transmitere a mesajelor*

Securitate

- Pentru a analiza care combinație de confidențialitate și integritate este sigură, definim ce înseamnă "combinație sigură";
- Introducem noțiunea de *schemă de transmitere a mesajelor*
- Fie $\Pi_E = (\mathrm{Enc}, \mathrm{Dec})$ o schemă de criptare arbitrară și $\Pi_M = (\mathrm{Mac}, \mathrm{Vrfy})$ un cod de autentificare a mesajelor. *O schemă de transmitere a mesajelor* $\Pi' = (\mathrm{EncMac}', \mathrm{Dec}')$ constă din următorii algoritmi:

Securitate

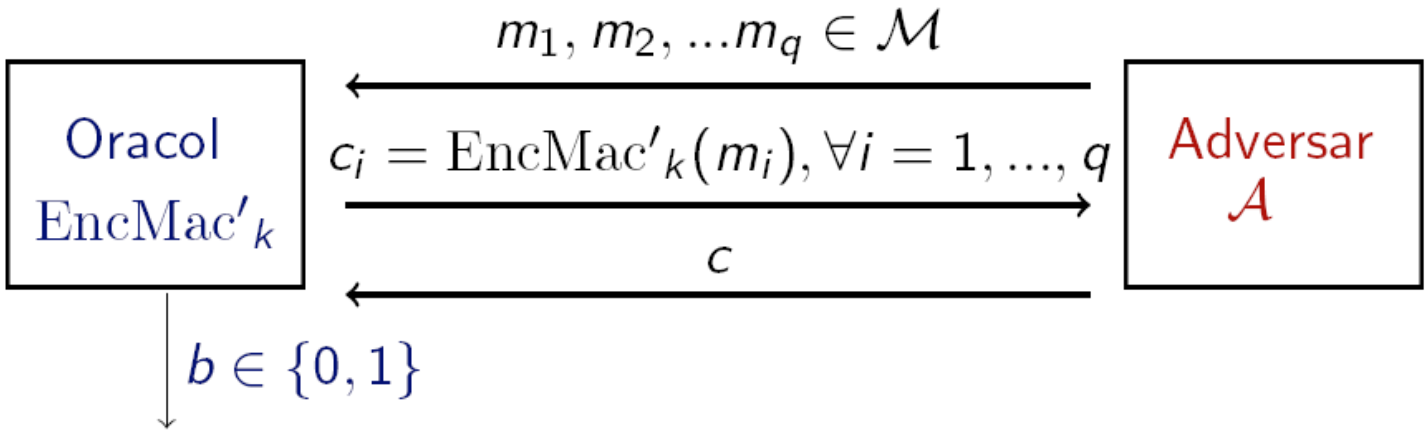
- Pentru a analiza care combinație de confidențialitate și integritate este sigură, definim ce înseamnă "combinație sigură";
- Introducem noțiunea de *schemă de transmitere a mesajelor*
- Fie $\Pi_E = (\mathrm{Enc}, \mathrm{Dec})$ o schemă de criptare arbitrară și $\Pi_M = (\mathrm{Mac}, \mathrm{Vrfy})$ un cod de autentificare a mesajelor. *O schemă de transmitere a mesajelor* $\Pi' = (\mathrm{EncMac}', \mathrm{Dec}')$ constă din următorii algoritmi:
  - EncMac - algoritm de transmitere a mesajelor care pentru o cheie (k_1, k_2) și un mesaj m, întoarce o valoare c derivată prin aplicarea unei combinații a algoritmilor Enc_{k_1} și Mac_{k_2};
  - Dec - algoritm de decriptare care pentru o cheie (k_1, k_2) și un mesaj transmis c, aplică o combinație a algoritmilor Dec_{k_1} și Vrfy_{k_2}, întorcand un text clar m sau simbolul $\perp$ de eroare.

Securitate

- Corectitudinea schemei cere ca $\forall n\, \forall (k_1, k_2)\, \forall m \in \{0,1\}^*$

$$\mathrm{Dec}'_{k_1,k_2}(\mathit{EncMac}'_{k_1,k_2}(m)) = m$$

- Pentru a defini securitatea unei astfel de scheme, folosim un experiment $\mathrm{Auth}_{\mathcal{A},\Pi}(n)$:

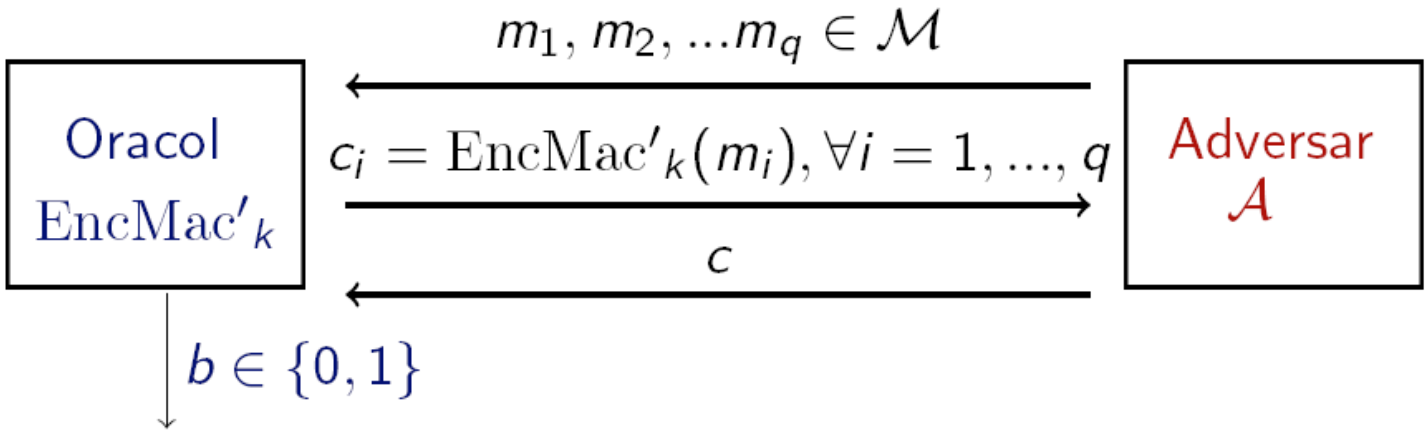


Securitate

- Corectitudinea schemei cere ca $\forall n\, \forall (k_1, k_2)\, \forall m \in \{0,1\}^*$

$$\mathrm{Dec}'_{k_1,k_2}(\mathit{EncMac}'_{k_1,k_2}(m)) = m$$

- Pentru a defini securitatea unei astfel de scheme, folosim un experiment $\mathrm{Auth}_{\mathcal{A},\Pi}(n)$:



- Output-ul experimentului este 1 dacă și numai dacă: (1) $m \neq \perp$ și (2) $m \notin \{m_1, \ldots, m_q\}$ unde $m = \mathrm{Dec}'_k(c)$;

Securitate

Definiție

O schemă de transmitere a mesajelor Π' oferă comunicare autentificată dacă pentru orice adversar polinomial $\mathcal{A}$ există o funcție neglijabilă negl așa încât

$$Pr[\mathrm{Auth}_{\mathcal{A},\Pi}(n) = 1] \leq negl(n).$$

Securitate

Definiție

O schemă de transmitere a mesajelor Π' oferă comunicare autentificată dacă pentru orice adversar polinomial $\mathcal{A}$ există o funcție neglijabilă negl așa încât

$$Pr[\mathrm{Auth}_{\mathcal{A},\Pi}(n) = 1] \leq negl(n).$$

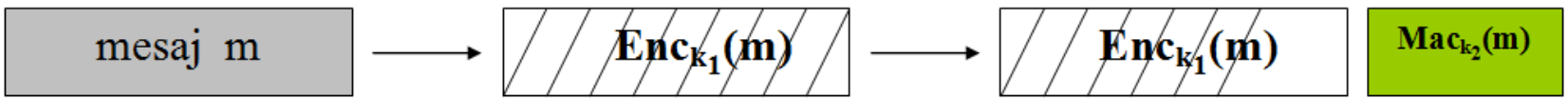
Definiție

O schemă de transmitere a mesajelor Π' este sigură dacă este o schemă de criptare CCA-sigură și oferă comunicare autentificată.

Securitate Criptare-și-autentificare

- Revenim la cele trei combinații de criptare și autentificare:

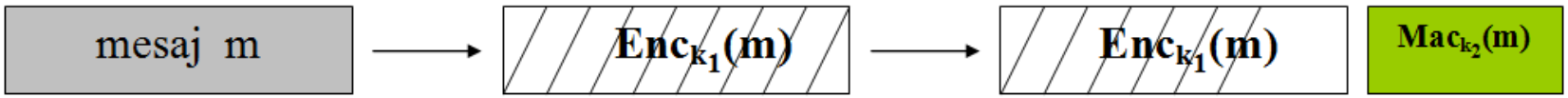
1. *Criptare-și-autentificare*



Securitate Criptare-și-autentificare

- Revenim la cele trei combinații de criptare și autentificare:

1. *Criptare-și-autentificare*

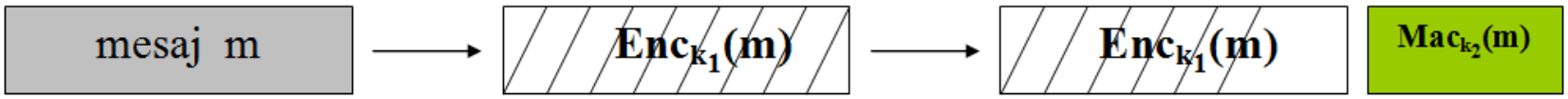


- Combinația aceasta **nu** este neaparat sigură; un MAC sigur nu implică nici un fel de confidențialitate;

Securitate Criptare-și-autentificare

- Revenim la cele trei combinații de criptare și autentificare:

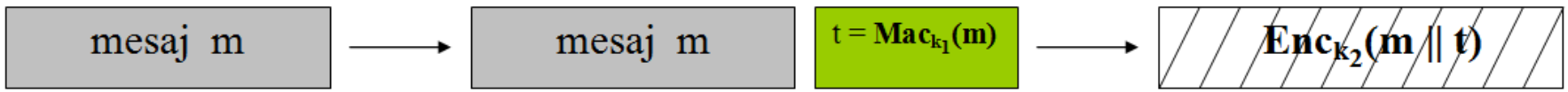
1. *Criptare-și-autentificare*



- Combinația aceasta **nu** este neaparat sigură; un MAC sigur nu implică nici un fel de confidențialitate;
- Dacă $(\mathrm{Mac}, \mathrm{Vrfy})$ este un MAC sigur atunci și schema definită de $\mathrm{Mac}'_k(m) = (m, Mac_k(m))$ este un MAC sigur dar dezvăluie mesajul m

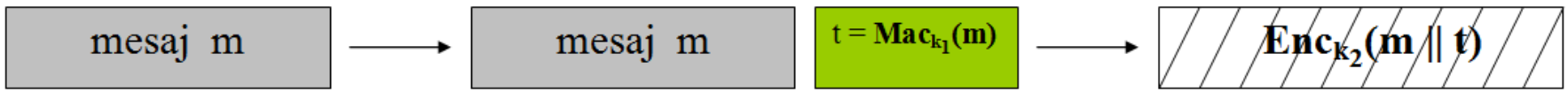
Securitate Autentificare-apoi-criptare

2. *Autentificare-apoi-criptare*



Securitate Autentificare-apoi-criptare

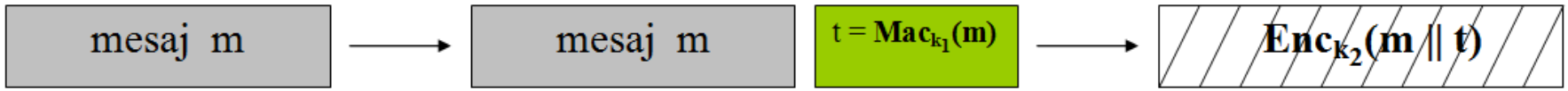
2. *Autentificare-apoi-criptare*



- Combinația aceasta **nu** este neapărat sigură;

Securitate Autentificare-apoi-criptare

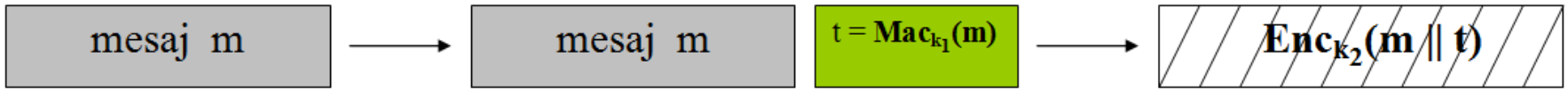
2. *Autentificare-apoi-criptare*



- Combinația aceasta **nu** este neapărat sigură;
- Se poate construi o schemă de criptare CPA-sigură care împreuna cu orice MAC sigur nu poate fi CCA-sigură;

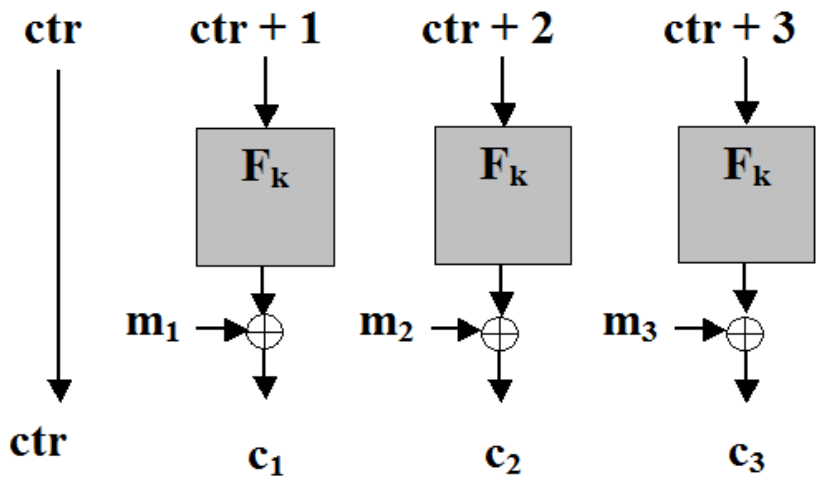
Securitate Autentificare-apoi-criptare

2. *Autentificare-apoi-criptare*



- Combinația aceasta **nu** este neapărat sigură;
- Se poate construi o schemă de criptare CPA-sigură care împreuna cu orice MAC sigur nu poate fi CCA-sigură;
- Definim $\mathrm{Transform}(m)$ astfel:
  - orice 0 din m se transformă în 00;
  - orice 1 din m se transformă arbitrar în 01 sau 10;

Securitate Autentificare-apoi-criptare



- Definim $\mathrm{Enc}_k(m) = \mathrm{Enc}'_k(\mathit{Transform}(m))$ unde Enc' reprezintă criptare în modul CTR folosind o funcție pseudoaleatoare;

Securitate Autentificare-apoi-criptare

- Arătăm că o combinație de tipul autentificare-apoi-criptare a schemei de criptare de mai sus cu *orice* MAC nu este sigură la un atac de tip CCA.

Securitate Autentificare-apoi-criptare

- Arătăm că o combinație de tipul autentificare-apoi-criptare a schemei de criptare de mai sus cu *orice* MAC nu este sigură la un atac de tip CCA.
- Atacul funcționează atâta timp cât un adversar poate verifica dacă un text criptat dat este valid;

Securitate Autentificare-apoi-criptare

- Arătăm că o combinație de tipul autentificare-apoi-criptare a schemei de criptare de mai sus cu *orice* MAC nu este sigură la un atac de tip CCA.
- Atacul funcționează atâta timp cât un adversar poate verifica dacă un text criptat dat este valid;
- Fiind dată o provocare $c = \mathrm{Enc}'_{k_1}(\mathrm{Transform}(m||\mathrm{Mac}_{k_2}(m)))$, atacatorul modifică primii doi biți din al 2-lea bloc al lui c și verifică dacă rezultatul este valid;

Securitate Autentificare-apoi-criptare

- Arătăm că o combinație de tipul autentificare-apoi-criptare a schemei de criptare de mai sus cu *orice* MAC nu este sigură la un atac de tip CCA.
- Atacul funcționează atâta timp cât un adversar poate verifica dacă un text criptat dat este valid;
- Fiind dată o provocare $c = \mathrm{Enc}'_{k_1}(\mathrm{Transform}(m||\mathrm{Mac}_{k_2}(m)))$, atacatorul modifică primii doi biți din al 2-lea bloc al lui c și verifică dacă rezultatul este valid;
- Dacă primul bit al mesajului clar m este 1, atunci c modificat este valid;

Securitate Autentificare-apoi-criptare

- Arătăm că o combinație de tipul autentificare-apoi-criptare a schemei de criptare de mai sus cu *orice* MAC nu este sigură la un atac de tip CCA.
- Atacul funcționează atâta timp cât un adversar poate verifica dacă un text criptat dat este valid;
- Fiind dată o provocare $c = \mathrm{Enc}'_{k_1}(\mathrm{Transform}(m||\mathrm{Mac}_{k_2}(m)))$, atacatorul modifică primii doi biți din al 2-lea bloc al lui c și verifică dacă rezultatul este valid;
- Dacă primul bit al mesajului clar m este 1, atunci c modificat este valid;
- Intrebare: De ce?

Securitate Autentificare-apoi-criptare

- Pentru că în acest caz, primii doi biți ai lui $\mathrm{Transform}(m)$ sunt 01 sau 10 și o modificare a lor oferă o codificare validă a lui m.

Securitate Autentificare-apoi-criptare

- Pentru că în acest caz, primii doi biți ai lui $\mathrm{Transform}(m)$ sunt 01 sau 10 și o modificare a lor oferă o codificare validă a lui m.
- Tag-ul rămâne valid pentru că este aplicat pe m;

Securitate Autentificare-apoi-criptare

- Pentru că în acest caz, primii doi biți ai lui $\mathrm{Transform}(m)$ sunt 01 sau 10 și o modificare a lor oferă o codificare validă a lui m.
- Tag-ul rămâne valid pentru că este aplicat pe m;
- Dacă însă primul bit al lui m este 0, c modificat nu este valid...

Securitate Autentificare-apoi-criptare

- Pentru că în acest caz, primii doi biți ai lui $\mathrm{Transform}(m)$ sunt 01 sau 10 și o modificare a lor oferă o codificare validă a lui m.
- Tag-ul rămâne valid pentru că este aplicat pe m;
- Dacă însă primul bit al lui m este 0, c modificat nu este valid...
- ... pentru că primii doi biți din $\mathrm{Transform}(m)$ sunt 00 și prin complementare devin 11;

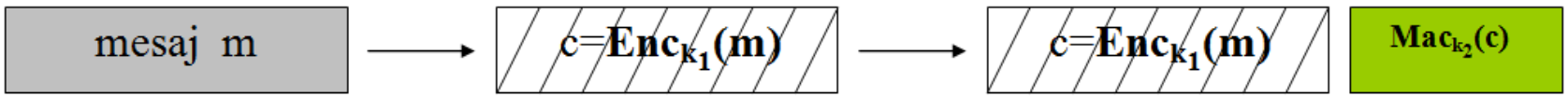
Securitate Autentificare-apoi-criptare

- Pentru că în acest caz, primii doi biți ai lui $\mathrm{Transform}(m)$ sunt 01 sau 10 și o modificare a lor oferă o codificare validă a lui m.
- Tag-ul rămâne valid pentru că este aplicat pe m;
- Dacă însă primul bit al lui m este 0, c modificat nu este valid...
- ... pentru că primii doi biți din $\mathrm{Transform}(m)$ sunt 00 și prin complementare devin 11;
- Atacul poate fi aplicat pe fiecare bit din m, recuperând astfel întreg mesajul m.

- Totusi, anumite instanțieri ale acestei combinații pot fi sigure;
- O astfel de combinație este folosita și în SSL.

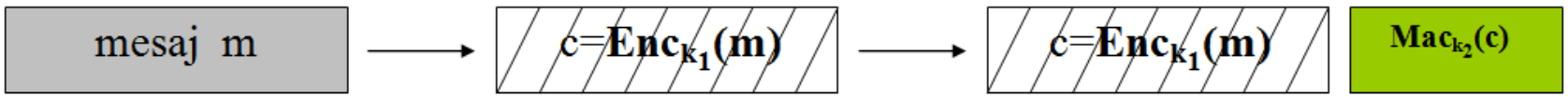
Securitate Criptare-apoi-autentificare

3. *Criptare-apoi-autentificare*



Securitate Criptare-apoi-autentificare

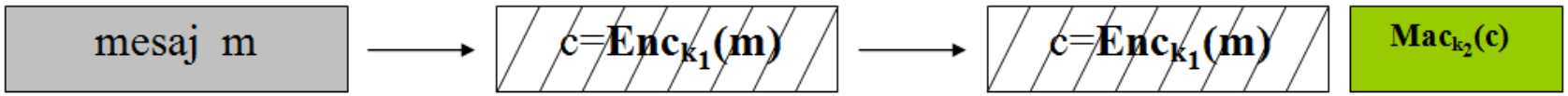
3. *Criptare-apoi-autentificare*



- Combinația aceasta este întotdeauna sigură; se folosește în IPsec;

Securitate Criptare-apoi-autentificare

3. *Criptare-apoi-autentificare*



- Combinația aceasta este întotdeauna sigură; se folosește în IPsec;
- Deși folosim aceeași construcție pentru a obține securitate CCA și transmitere sigură a mesajelor, scopurile urmărite sunt diferite în fiecare caz;

Necesitatea de a folosi chei diferite

- Pentru scopuri diferite de securitate trebuie să folosim întotdeauna chei diferite;

Necesitatea de a folosi chei diferite

- Pentru scopuri diferite de securitate trebuie să folosim întotdeauna chei diferite;
- Să urmărim ce se întâmplă dacă folosim metoda criptare-apoi-autentificare atunci când folosim aceeași cheie k atât pentru criptare cât și pentru autentificare;

Necesitatea de a folosi chei diferite

- Pentru scopuri diferite de securitate trebuie să folosim întotdeauna chei diferite;
- Să urmărim ce se întâmplă dacă folosim metoda criptare-apoi-autentificare atunci când folosim aceeași cheie k atât pentru criptare cât și pentru autentificare;
- Definim $\mathrm{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0,1\}^{n/2}$, $r \leftarrow^R \{0,1\}^{n/2}$, iar $F_k(\cdot)$ o permutare pseudoaleatoare;

Necesitatea de a folosi chei diferite

- Pentru scopuri diferite de securitate trebuie să folosim întotdeauna chei diferite;
- Să urmărim ce se întâmplă dacă folosim metoda criptare-apoi-autentificare atunci când folosim aceeași cheie k atât pentru criptare cât și pentru autentificare;
- Definim $\mathrm{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0,1\}^{n/2}$, $r \leftarrow^R \{0,1\}^{n/2}$, iar $F_k(\cdot)$ o permutare pseudoaleatoare;
- Definim $\mathrm{Mac}_k(c) = F_k^{-1}(c)$;

Necesitatea de a folosi chei diferite

- Pentru scopuri diferite de securitate trebuie să folosim întotdeauna chei diferite;
- Să urmărim ce se întâmplă dacă folosim metoda criptare-apoi-autentificare atunci când folosim aceeași cheie k atât pentru criptare cât și pentru autentificare;
- Definim $\mathrm{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0,1\}^{n/2}$, $r \leftarrow^R \{0,1\}^{n/2}$, iar $F_k(\cdot)$ o permutare pseudoaleatoare;
- Definim $\mathrm{Mac}_k(c) = F_k^{-1}(c)$;
- Schema de criptare și MAC-ul sunt sigure dar...

Necesitatea de a folosi chei diferite

- Pentru scopuri diferite de securitate trebuie să folosim întotdeauna chei diferite;
- Să urmărim ce se întâmplă dacă folosim metoda criptare-apoi-autentificare atunci când folosim aceeași cheie k atât pentru criptare cât și pentru autentificare;
- Definim $\mathrm{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0,1\}^{n/2}$, $r \leftarrow^R \{0,1\}^{n/2}$, iar $F_k(\cdot)$ o permutare pseudoaleatoare;
- Definim $\mathrm{Mac}_k(c) = F_k^{-1}(c)$;
- Schema de criptare și MAC-ul sunt sigure dar...
- $\langle \mathrm{Enc}_k(m), \mathrm{Mac}_k(\mathrm{Enc}_k(m)) \rangle = \langle F_k(m||r), F_k^{-1}(F_k(m||r)) \rangle = \langle F_k(m||r), m||r \rangle$.

Important de reținut!

- Metoda sigură de a combina criptarea și autentificarea este *criptare-apoi-autentificare*;
- Este important să se folosească chei simetrice diferite pentru a atinge scopuri diferite (criptare și autentificare).