

Criptografie și Securitate

- Prelegerea 14 - Confidențialitate și autentificare a mesajelor

Adela Georgescu, Ruxandra F. Olimid

Facultatea de Matematică și Informatică
Universitatea din București

Cuprins

1. Transmitere sigură a mesajelor
2. Abordări diferite pentru a combina criptarea și autentificarea

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;
- ▶ În 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;
- ▶ În 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 abordări uzuale pentru a combina criptarea și autentificarea mesajelor:

Confidențialitate și integritate

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

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

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

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

Confidențialitate și integritate

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

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

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

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

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

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

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

Confidențialitate și integritate

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

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

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

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

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

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

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 $\Pi_E = (\text{Enc}, \text{Dec})$ o schemă de criptare arbitrară și $\Pi_M = (\text{Mac}, \text{Vrfy})$ un cod de autentificare a mesajelor. O *schemă de transmitere a mesajelor* $\Pi' = (\text{EncMac}', \text{Dec}')$ constă din următorii algoritmi:

Securitate

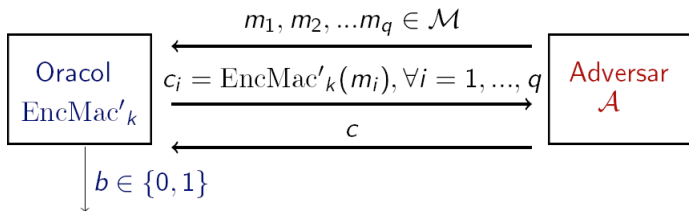
- ▶ Pentru a analiza care combinație de confidențialitate și integritate este sigură, definim ce înseamnă "combinație sigură";
- ▶ Introducem noțiunea de *schemă de transmitere a mesajelor*
- ▶ Fie $\Pi_E = (\text{Enc}, \text{Dec})$ o schemă de criptare arbitrară și $\Pi_M = (\text{Mac}, \text{Vrfy})$ un cod de autentificare a mesajelor. O *schemă de transmitere a mesajelor* $\Pi' = (\text{EncMac}', \text{Dec}')$ 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} , întorcând un text clar m sau simbolul \perp de eroare.

Securitate

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

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

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

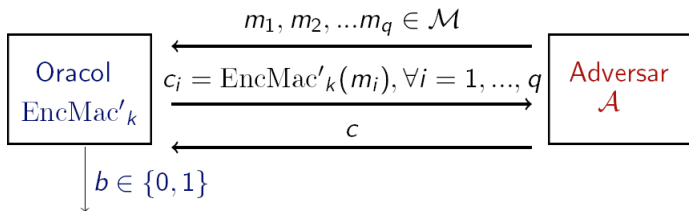


Securitate

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

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

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



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

Definiție

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

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

Definiție

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

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

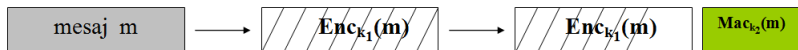
Definiție

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

Securitate Criptare-și-autentificare

- Revenim la cele trei combinații de criptare și autentificare:

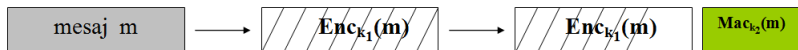
1. *Criptare-și-autentificare (Encrypt AND Authenticate)*



Securitate Criptare-și-autentificare

- Revenim la cele trei combinații de criptare și autentificare:

1. *Criptare-și-autentificare (Encrypt AND Authenticate)*

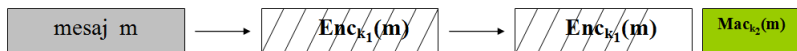


- Combinația aceasta **nu** este neaparat sigură; un MAC sigur nu implică nici un fel de confidențialitate;

Securitate Criptare-și-autentificare

- Revenim la cele trei combinații de criptare și autentificare:

1. *Criptare-și-autentificare (Encrypt AND Authenticate)*



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

Securitate Autentificare-apoi-criptare

2. Autentificare-apoi-criptare (*Authenticate THEN Encrypt*)



Securitate Autentificare-apoi-criptare

2. Autentificare-apoi-criptare (*Authenticate THEN Encrypt*)



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

Securitate Autentificare-apoi-criptare

2. Autentificare-apoi-criptare (*Authenticate THEN Encrypt*)



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

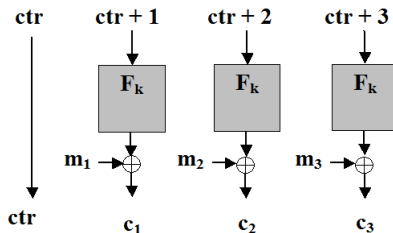
Securitate Autentificare-apoi-criptare

2. Autentificare-apoi-criptare (*Authenticate THEN Encrypt*)



- ▶ Combinația aceasta **nu** este neapărat sigură;
- ▶ Se poate construi o schemă de criptare CPA-sigură care împreună cu orice MAC sigur nu poate fi CCA-sigură;
- ▶ Definim $\text{Transform}(m)$ 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 $\text{Enc}_k(m) = \text{Enc}'_k(\text{Transform}(m))$ 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 $c = \text{Enc}'_{k_1}(\text{Transform}(m || \text{Mac}_{k_2}(m)))$, 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 $c = \text{Enc}'_{k_1}(\text{Transform}(m || \text{Mac}_{k_2}(m)))$, 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 $c = \text{Enc}'_{k_1}(\text{Transform}(m || \text{Mac}_{k_2}(m)))$, 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 $\text{Transform}(m)$ 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 $\text{Transform}(m)$ 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 $\text{Transform}(m)$ 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 $\text{Transform}(m)$ 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 $\text{Transform}(m)$ sunt 00 și prin complementare devin 11;

Securitate Autentificare-apoi-criptare

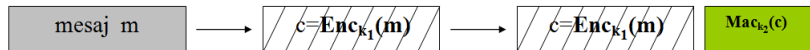
- ▶ Pentru că în acest caz, primii doi biți ai lui $\text{Transform}(m)$ sunt 01 sau 10 și o modificare a lor oferă o codificare validă a lui m .
- ▶ Tag-ul rămâne valid pentru că este aplicat pe m ;
- ▶ Dacă însă primul bit al lui m este 0, c modificat nu este valid...
- ▶ ... pentru că primii doi biți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

Securitate Autentificare-apoi-criptare

- ▶ Totuși, anumite instanțieri ale acestei combinații pot fi sigure;
- ▶ O astfel de combinație este folosită și în SSL și CCMP (WPA2).

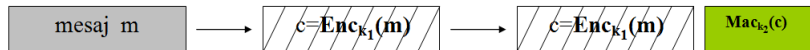
Securitate Criptare-apoi-autentificare

3. Criptare-apoi-autentificare (*Encrypt THEN Authenticate*)



Securitate Criptare-apoi-autentificare

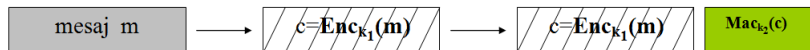
3. Criptare-apoi-autentificare (*Encrypt THEN Authenticate*)



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

Securitate Criptare-apoi-autentificare

3. Criptare-apoi-autentificare (*Encrypt THEN Authenticate*)



- ▶ 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 $\text{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0, 1\}^{n/2}$, $r \xleftarrow{R} \{0, 1\}^{n/2}$, 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 $\text{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0, 1\}^{n/2}$, $r \xleftarrow{R} \{0, 1\}^{n/2}$, iar $F_k(\cdot)$ o permutare pseudoaleatoare;
- ▶ Definim $\text{Mac}_k(c) = F_k^{-1}(c)$;

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 $\text{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0, 1\}^{n/2}$, $r \xleftarrow{R} \{0, 1\}^{n/2}$, iar $F_k(\cdot)$ o permutare pseudoaleatoare;
- ▶ Definim $\text{Mac}_k(c) = F_k^{-1}(c)$;
- ▶ 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 $\text{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0, 1\}^{n/2}$, $r \xleftarrow{R} \{0, 1\}^{n/2}$, iar $F_k(\cdot)$ o permutare pseudoaleatoare;
- ▶ Definim $\text{Mac}_k(c) = F_k^{-1}(c)$;
- ▶ Schema de criptare și MAC-ul sunt sigure dar...
- ▶ $\langle \text{Enc}_k(m), \text{Mac}_k(\text{Enc}_k(m)) \rangle = \langle F_k(m||r), F_k^{-1}(F_k(m||r)) \rangle = \langle F_k(m||r), m||r \rangle$.

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).