Securitatea Sistemelor Informatice

- Curs 7.1 -

Coduri de autentificare a mesajelor - MAC

Adela Georgescu

Facultatea de Matematică și Informatică
Universitatea din București
Anul universitar 2022-2023, semestrul I

Comunicare sigură și integritatea mesajelor

- Un scop de bază al criptografiei este să asigure comunicarea sigură de-a lungul unui canal public de comunicare;

Comunicare sigură și integritatea mesajelor

- Un scop de bază al criptografiei este să asigure comunicarea sigură de-a lungul unui canal public de comunicare;
- Am vazut cum putem obține confidențialitatea cu ajutorul schemelor de criptare;

Comunicare sigură și integritatea mesajelor

- Un scop de bază al criptografiei este să asigure comunicarea sigură de-a lungul unui canal public de comunicare;
- Am vazut cum putem obține confidențialitatea cu ajutorul schemelor de criptare;
- Insă, nu ne interesează doar ca adversarul să nu aibă acces la mesajele trimise, ci...

Comunicare sigură și integritatea mesajelor

- Un scop de bază al criptografiei este să asigure comunicarea sigură de-a lungul unui canal public de comunicare;
- Am vazut cum putem obține confidențialitatea cu ajutorul schemelor de criptare;
- Insă, nu ne interesează doar ca adversarul să nu aibă acces la mesajele trimise, ci...
- Vrem să garantăm integritatea mesajelor (sau autentificarea mesajelor)

Comunicare sigură și integritatea mesajelor

- Un scop de bază al criptografiei este să asigure comunicarea sigură de-a lungul unui canal public de comunicare;
- Am vazut cum putem obține confidențialitatea cu ajutorul schemelor de criptare;
- Insă, nu ne interesează doar ca adversarul să nu aibă acces la mesajele trimise, ci...
- Vrem să garantăm integritatea mesajelor (sau autentificarea mesajelor)
- Aceasta înseamnă ca mesajul primit de Bob este exact mesajul trimis de Alice.

Comunicare sigură și integritatea mesajelor

- Iată un exemplu:

Comunicare sigură și integritatea mesajelor

- Iată un exemplu:
- Să consideram cazul în care un mare lanț de supermarket-uri trimite o comandă pe email către un furnizor pentru a achiziționa 10.000 bax-uri de apă minerală;

Comunicare sigură și integritatea mesajelor

- Iată un exemplu:
- Să consideram cazul în care un mare lanț de supermarket-uri trimite o comandă pe email către un furnizor pentru a achiziționa 10.000 bax-uri de apă minerală;
- Odată primită comanda, furnizorul trebuie să verifice următoarele:

Comunicare sigură și integritatea mesajelor

- Iată un exemplu:
- Să consideram cazul în care un mare lanț de supermarket-uri trimite o comandă pe email către un furnizor pentru a achiziționa 10.000 bax-uri de apă minerală;
- Odată primită comanda, furnizorul trebuie să verifice următoarele:
  1. Comanda este autentică? A fost trimisă cu adevărat de un supermarket sau de către un adversar care a furat contul de email al clientului respectiv ?

Comunicare sigură și integritatea mesajelor

- Iată un exemplu:
- Să consideram cazul în care un mare lanț de supermarket-uri trimite o comandă pe email către un furnizor pentru a achiziționa 10.000 bax-uri de apă minerală;
- Odată primită comanda, furnizorul trebuie să verifice următoarele:
  1. Comanda este autentică? A fost trimisă cu adevărat de un supermarket sau de către un adversar care a furat contul de email al clientului respectiv ?
  2. Dacă s-a convins de autenticitatea comenzii, trebuie verificat dacă detaliile ei sunt cele originale sau au fost modificate pe parcurs de un adversar.

Comunicare sigură și integritatea mesajelor

- In exemplul precedent, problema este doar de integritate a mesajelor, și nu de confidențialitate (comanda nu e secretă);

Comunicare sigură și integritatea mesajelor

- In exemplul precedent, problema este doar de integritate a mesajelor, și nu de confidențialitate (comanda nu e secretă);
- In general nu ne putem baza pe încredere în ceea ce privește integritatea mesajelor transmise, indiferent că ele sunt:

Comunicare sigură și integritatea mesajelor

- In exemplul precedent, problema este doar de integritate a mesajelor, și nu de confidențialitate (comanda nu e secretă);
- In general nu ne putem baza pe încredere în ceea ce privește integritatea mesajelor transmise, indiferent că ele sunt:
  - comenzi efectuate online
  - operațiuni bancare online
  - email, SMS

Comunicare sigură și integritatea mesajelor

- In exemplul precedent, problema este doar de integritate a mesajelor, și nu de confidențialitate (comanda nu e secretă);
- In general nu ne putem baza pe încredere în ceea ce privește integritatea mesajelor transmise, indiferent că ele sunt:
  - comenzi efectuate online
  - operațiuni bancare online
  - email, SMS
- Vom vedea cum putem folosi tehnici criptografice pentru a preveni modificarea nedectată a mesajelor transmise.

Criptare vs. autentificarea mesajelor

- Criptarea, în general, **NU** oferă integritatea mesajelor!
- Nu folositi criptarea cu scopul de a obtine autentificarea mesajelor

Criptare vs. autentificarea mesajelor

- Criptarea, în general, **NU** oferă integritatea mesajelor!
- Nu folositi criptarea cu scopul de a obtine autentificarea mesajelor
- Dacă un mesaj este transmis criptat de-a lungul unui canal de comunicare, nu înseamnă că un adversar nu poate modifica/altera mesajul așa încât modificarea să aiba sens în textul clar;

Criptare vs. autentificarea mesajelor

- Criptarea, în general, **NU** oferă integritatea mesajelor!
- Nu folositi criptarea cu scopul de a obtine autentificarea mesajelor
- Dacă un mesaj este transmis criptat de-a lungul unui canal de comunicare, nu înseamnă că un adversar nu poate modifica/altera mesajul așa încât modificarea să aiba sens în textul clar;
- Verificăm, în continuare, că nici o schemă de criptare studiată nu oferă integritatea mesajelor;

Criptare vs. autentificarea mesajelor

- Criptarea folosind sisteme fluide
  - $Enc_k(m) = G(k) \oplus m$, unde G este un PRG;

Criptare vs. autentificarea mesajelor

- Criptarea folosind sisteme fluide
  - $Enc_k(m) = G(k) \oplus m$, unde G este un PRG;
  - Dacă modificăm un singur bit din textul criptat c, modificarea se va reflecta imediat în același bit din textul clar;

Criptare vs. autentificarea mesajelor

- Criptarea folosind sisteme fluide
  - $Enc_k(m) = G(k) \oplus m$, unde G este un PRG;
  - Dacă modificăm un singur bit din textul criptat c, modificarea se va reflecta imediat în același bit din textul clar;
  - Consecințele pot fi grave: de pildă, să considerăm transferul unei sume de bani în dolari criptate, reprezentată în binar;

Criptare vs. autentificarea mesajelor

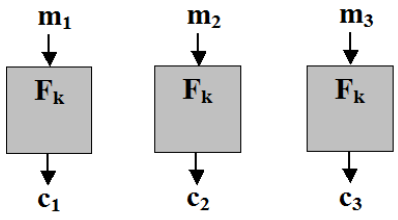
- Criptarea folosind sisteme fluide
  - $Enc_k(m) = G(k) \oplus m$, unde G este un PRG;
  - Dacă modificăm un singur bit din textul criptat c, modificarea se va reflecta imediat în același bit din textul clar;
  - Consecințele pot fi grave: de pildă, să considerăm transferul unei sume de bani în dolari criptate, reprezentată în binar;
  - Modificarea unui bit poate schimba suma foarte mult (al 11 lsb schimbă suma cu mai mult de 1000$);

Criptare vs. autentificarea mesajelor

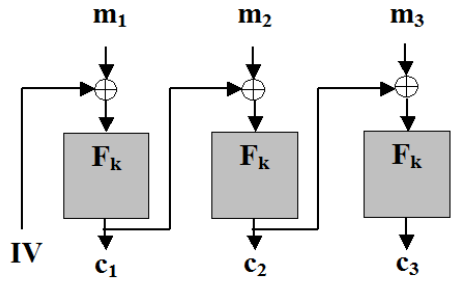
- Criptarea folosind sisteme fluide
  - $Enc_k(m) = G(k) \oplus m$, unde G este un PRG;
  - Dacă modificăm un singur bit din textul criptat c, modificarea se va reflecta imediat în același bit din textul clar;
  - Consecințele pot fi grave: de pildă, să considerăm transferul unei sume de bani în dolari criptate, reprezentată în binar;
  - Modificarea unui bit poate schimba suma foarte mult (al 11 lsb schimbă suma cu mai mult de 1000$);
  - Același atac se poate aplica și la OTP.

Criptare vs. autentificarea mesajelor

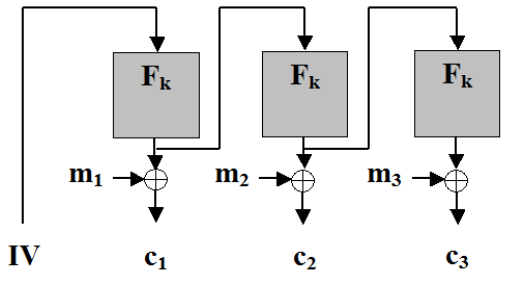
- Criptarea folosind sisteme bloc
  - Intrebare: Atacul de mai sus se poate aplica și pentru sistemele bloc cu modurile de operare studiate?



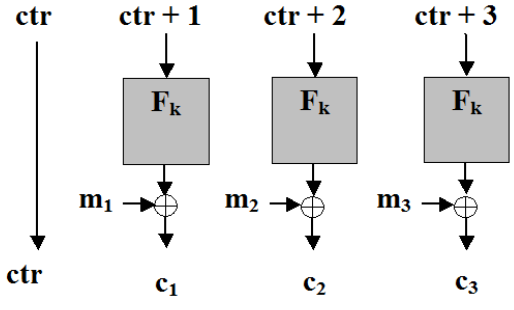
(a) ECB



(b) CBC



(c) OFB



(d) CTR

Criptare vs. autentificarea mesajelor

- Răspuns: Atacul se aplică identic pentru modurile OFB și CTR;

Criptare vs. autentificarea mesajelor

- Răspuns: Atacul se aplică identic pentru modurile OFB și CTR;
- Pentru modul ECB, modificarea unui bit din al i-lea bloc criptat afectează numai al i-lea bloc clar, dar este foarte greu de prezis efectul exact;

Criptare vs. autentificarea mesajelor

- Răspuns: Atacul se aplică identic pentru modurile OFB și CTR;
- Pentru modul ECB, modificarea unui bit din al i-lea bloc criptat afectează numai al i-lea bloc clar, dar este foarte greu de prezis efectul exact;
- Mai mult, ordinea blocurilor la ECB poate fi schimbată;

Criptare vs. autentificarea mesajelor

- Răspuns: Atacul se aplică identic pentru modurile OFB și CTR;
- Pentru modul ECB, modificarea unui bit din al i-lea bloc criptat afectează numai al i-lea bloc clar, dar este foarte greu de prezis efectul exact;
- Mai mult, ordinea blocurilor la ECB poate fi schimbată;
- Pentru modul CBC, schimbarea bitului j din IV va schimba bitul j din primul bloc;

Criptare vs. autentificarea mesajelor

- Răspuns: Atacul se aplică identic pentru modurile OFB și CTR;
- Pentru modul ECB, modificarea unui bit din al i-lea bloc criptat afectează numai al i-lea bloc clar, dar este foarte greu de prezis efectul exact;
- Mai mult, ordinea blocurilor la ECB poate fi schimbată;
- Pentru modul CBC, schimbarea bitului j din IV va schimba bitul j din primul bloc;
- Toate celelalte blocuri de text clar rămân neschimbate ($m_i = F_k^{-1}(c_i) \oplus c_{i-1}$ iar blocurile c_i si c_{i-1} nu au fost modificate).

Coduri de autentificare a mesajelor - MAC

- Așa cum am vazut, criptarea nu rezolvă problema autentificarii mesajelor;

Coduri de autentificare a mesajelor - MAC

- Așa cum am vazut, criptarea nu rezolvă problema autentificarii mesajelor;
- Vom folosi un mecanism diferit, numit cod de autentificare a mesajelor - MAC (Message Authentication Code);

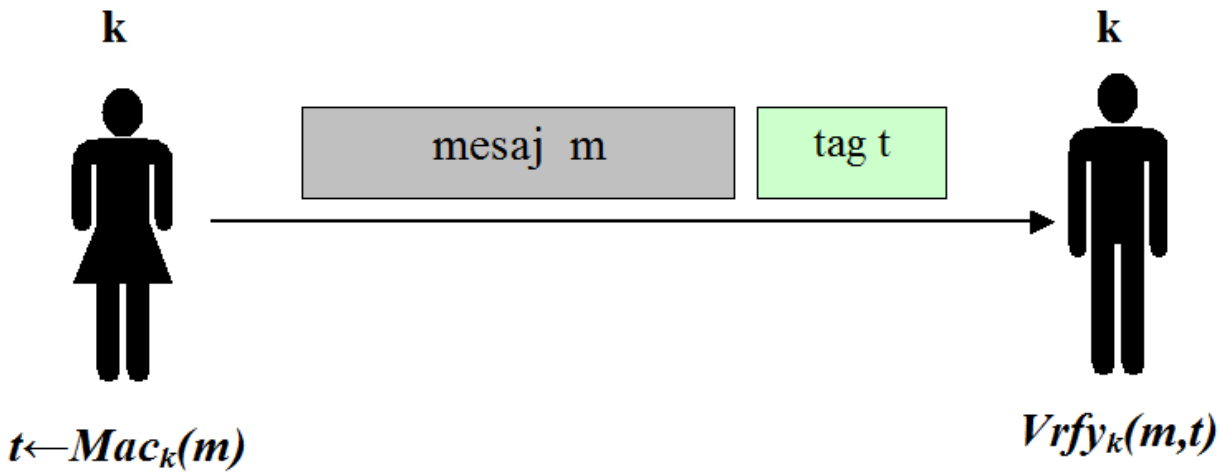
Coduri de autentificare a mesajelor - MAC

- Așa cum am vazut, criptarea nu rezolvă problema autentificarii mesajelor;
- Vom folosi un mecanism diferit, numit cod de autentificare a mesajelor - MAC (Message Authentication Code);
- Scopul lor este de a împiedica un adversar să modifice un mesaj trimis fără ca părțile care comunică să nu detecteze modificarea;

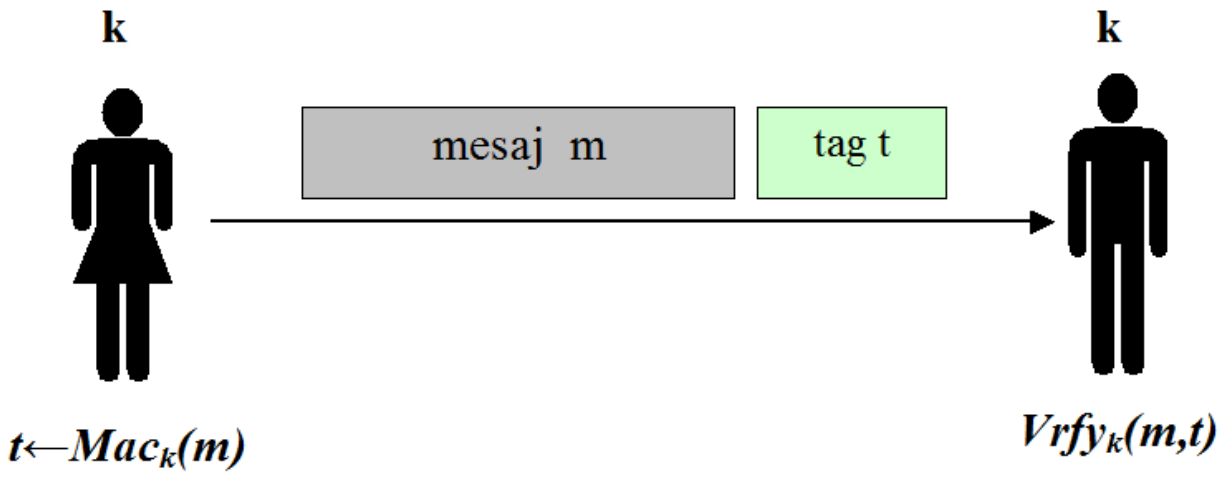
Coduri de autentificare a mesajelor - MAC

- Așa cum am vazut, criptarea nu rezolvă problema autentificarii mesajelor;
- Vom folosi un mecanism diferit, numit cod de autentificare a mesajelor - MAC (Message Authentication Code);
- Scopul lor este de a împiedica un adversar să modifice un mesaj trimis fără ca părțile care comunică să nu detecteze modificarea;
- Vom lucra în continuare în contextul criptografiei cu cheie secretă unde părțile trebuie să prestabilească de comun acord o cheie secretă.

MAC - Definiție

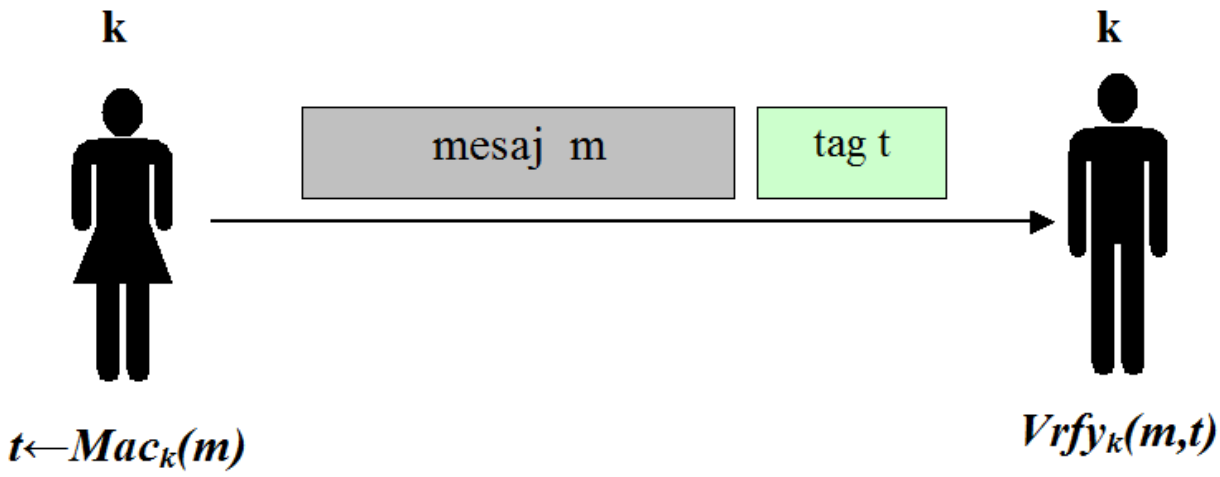


MAC - Definiție



- Alice și Bob stabliesc o cheie secretă k pe care o partajează;

MAC - Definiție



- Alice și Bob stabliesc o cheie secretă k pe care o partajează;
- Când Alice vrea să îi trimită un mesaj m lui Bob, calculează mai întâi un tag t (o etichetă) pe baza mesajului m și a cheii k și trimite perechea (m, t);

MAC - Definiție

- Tag-ul este calculat folosind un algoritm de generare a tag-urilor numit `Mac`;

MAC - Definiție

- Tag-ul este calculat folosind un algoritm de generare a tag-urilor numit `Mac`;
- La primirea perechii (m, t) Bob verifică dacă tag-ul este valid (în raport cu cheia k) folosind un algoritm de verificare `Vrfy`;

MAC - Definiție

- Tag-ul este calculat folosind un algoritm de generare a tag-urilor numit `Mac`;
- La primirea perechii (m, t) Bob verifică dacă tag-ul este valid (în raport cu cheia k) folosind un algoritm de verificare `Vrfy`;
- In continuare prezentăm definiția formală a unui cod de autentificare a mesajelor.

MAC - Definiție

Definiție

Un cod de autentificare a mesajelor (MAC) definit peste $(\mathcal{K}, \mathcal{M}, \mathcal{T})$ *este format dintr-un triplet de algoritmi polinomiali* $(\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$ *unde:*

1. $\mathrm{Gen}(1^n)$*: este algoritmul de generare a cheii* k *(aleasă uniform pe* n *biți)*
2. $\mathrm{Mac} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ *este algoritmul de generare a tag-urilor* $t \leftarrow \mathrm{Mac}_k(m)$*;*
3. $\mathrm{Vrfy} : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0, 1\}$
   este algoritmul de verificare ce întoarce un bit $b = \mathrm{Vrfy}_k(m, t)$ *cu semnificația că:*
   - $b = 1$ *înseamnă* valid
   - $b = 0$ *înseamnă* invalid

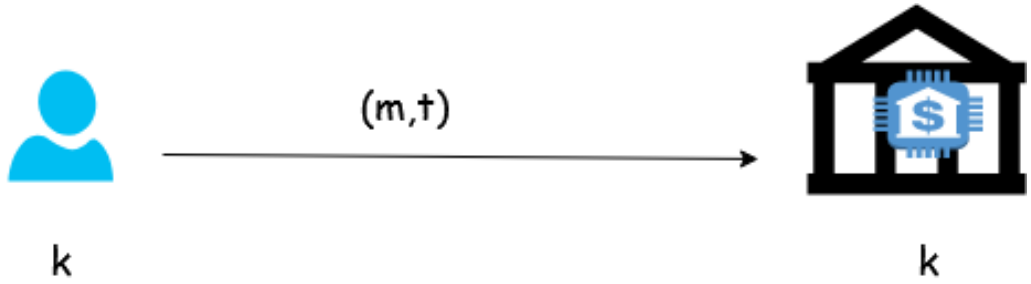
a.î : $\forall m \in \mathcal{M},\ k \in \mathcal{K}\ \mathrm{Vrfy}_k(m, \mathrm{Mac}_k(m)) = 1.$

Cazuri de folosire ale MAC

1. când cele două părti care comunică partajează o cheie secretă în avans

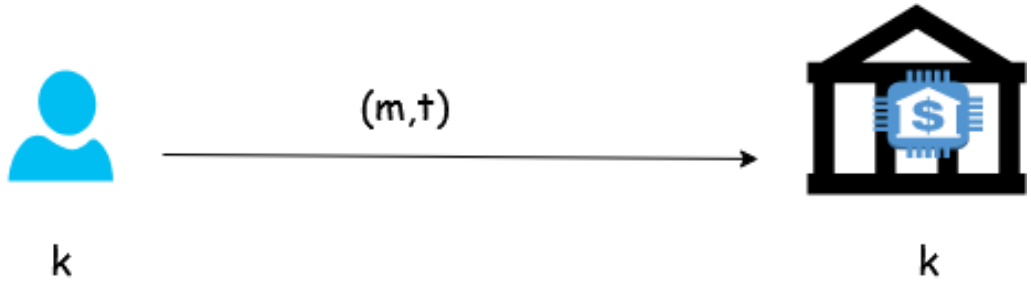
Cazuri de folosire ale MAC

1. când cele două părti care comunică partajează o cheie secretă în avans

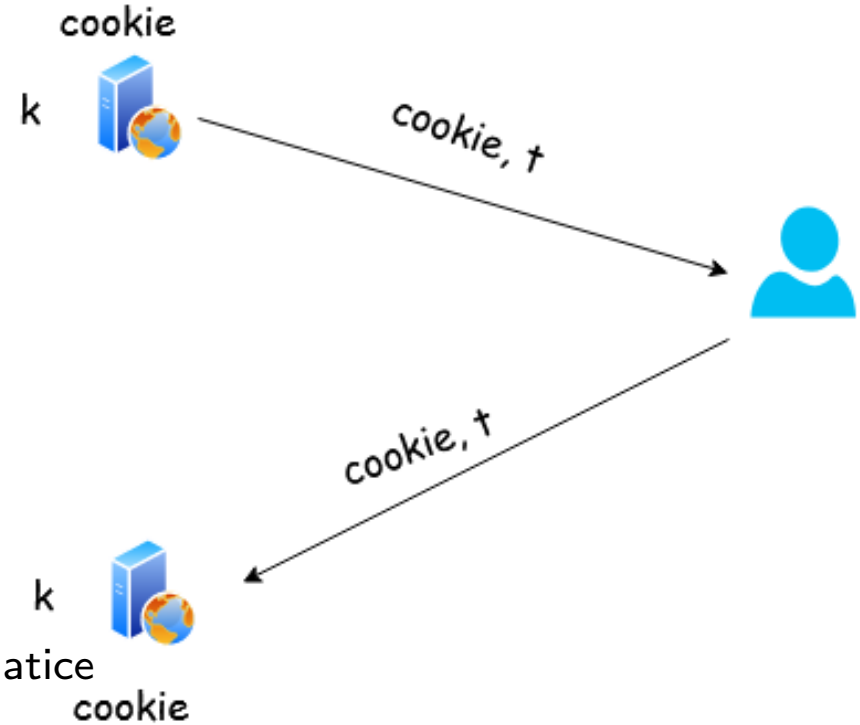


Cazuri de folosire ale MAC

1. când cele două părti care comunică partajează o cheie secretă în avans



2. când avem o singură parte care autentifică o comunicație, interacționând cu ea însăși după un timp



Securitate MAC - discuție

- Intuiție: nici un adversar polinomial nu ar trebui să poată genera un tag valid pentru nici un mesaj "nou" care nu a fost deja trimis (și autentificat) de părțile care comunică;

Securitate MAC - discuție

- Intuiție: nici un adversar polinomial nu ar trebui să poată genera un tag valid pentru nici un mesaj "nou" care nu a fost deja trimis (și autentificat) de părțile care comunică;
- Trebuie să definim puterea adversarului și ce înseamnă spargerea sau un atac asupra securității;

Securitate MAC - discuție

- Intuiție: nici un adversar polinomial nu ar trebui să poată genera un tag valid pentru nici un mesaj "nou" care nu a fost deja trimis (și autentificat) de părțile care comunică;
- Trebuie să definim puterea adversarului și ce înseamnă spargerea sau un atac asupra securității;
- Adversarul lucrează în timp polinomial și are acces la mesajele trimise între părți împreuna cu tag-urile aferente.

Securitate MAC - discuție

- Intuiție: nici un adversar polinomial nu ar trebui să poată genera un tag valid pentru nici un mesaj "nou" care nu a fost deja trimis (și autentificat) de părțile care comunică;
- Trebuie să definim puterea adversarului și ce înseamnă spargerea sau un atac asupra securității;
- Adversarul lucrează în timp polinomial și are acces la mesajele trimise între părți împreuna cu tag-urile aferente.
- Adversarul este *activ*, poate influența autentificarea unor mesaje alese de el...

Securitate MAC - discuție

- Intuiție: nici un adversar polinomial nu ar trebui să poată genera un tag valid pentru nici un mesaj "nou" care nu a fost deja trimis (și autentificat) de părțile care comunică;
- Trebuie să definim puterea adversarului și ce înseamnă spargerea sau un atac asupra securității;
- Adversarul lucrează în timp polinomial și are acces la mesajele trimise între părți împreuna cu tag-urile aferente.
- Adversarul este *activ*, poate influența autentificarea unor mesaje alese de el...
- ...dar nu trebuie să poată falsifica tag-ul aferent unor mesaje care nu au fost autentificate de expeditor

Securitate MAC - formalizare

- Formal, îi dăm adversarului acces la un *oracol* $\mathrm{Mac}_k(\cdot)$;

Securitate MAC - formalizare

- Formal, îi dăm adversarului acces la un *oracol* $\mathrm{Mac}_k(\cdot)$;
- Adversarul poate trimite orice mesaj m dorit către oracol și primește înapoi un tag corespunzător $t \leftarrow \mathrm{Mac}_k(m)$;

Securitate MAC - formalizare

- Formal, îi dăm adversarului acces la un *oracol* $\mathrm{Mac}_k(\cdot)$;
- Adversarul poate trimite orice mesaj m dorit către oracol și primește înapoi un tag corespunzător $t \leftarrow \mathrm{Mac}_k(m)$;
- Considerăm că securitatea este impactată dacă adversarul este capabil să producă un mesaj m împreună cu un tag t așa încât:

Securitate MAC - formalizare

- Formal, îi dăm adversarului acces la un *oracol* $\mathrm{Mac}_k(\cdot)$;
- Adversarul poate trimite orice mesaj m dorit către oracol și primește înapoi un tag corespunzător $t \leftarrow \mathrm{Mac}_k(m)$;
- Considerăm că securitatea este impactată dacă adversarul este capabil să producă un mesaj m împreună cu un tag t așa încât:
  1. t este un tag valid pentru mesajul m: $\mathrm{Vrfy}_k(m, t) = 1$;

Securitate MAC - formalizare

- Formal, îi dăm adversarului acces la un *oracol* $\mathrm{Mac}_k(\cdot)$;
- Adversarul poate trimite orice mesaj m dorit către oracol și primește înapoi un tag corespunzător $t \leftarrow \mathrm{Mac}_k(m)$;
- Considerăm că securitatea este impactată dacă adversarul este capabil să producă un mesaj m împreună cu un tag t așa încât:
  1. t este un tag valid pentru mesajul m: $\mathrm{Vrfy}_k(m, t) = 1$;
  2. Adversarul nu a solicitat anterior (de la oracol) un tag pentru mesajul m.

Securitate MAC - formalizare

- Despre un MAC care satisface nivelul de securitate de mai sus spunem că *nu poate fi falsificat printr-un atac cu mesaj ales*;

Securitate MAC - formalizare

- Despre un MAC care satisface nivelul de securitate de mai sus spunem că *nu poate fi falsificat printr-un atac cu mesaj ales*;
- Aceasta înseamnă că un adversar nu este capabil să falsifice un tag valid pentru nici un mesaj ...

Securitate MAC - formalizare

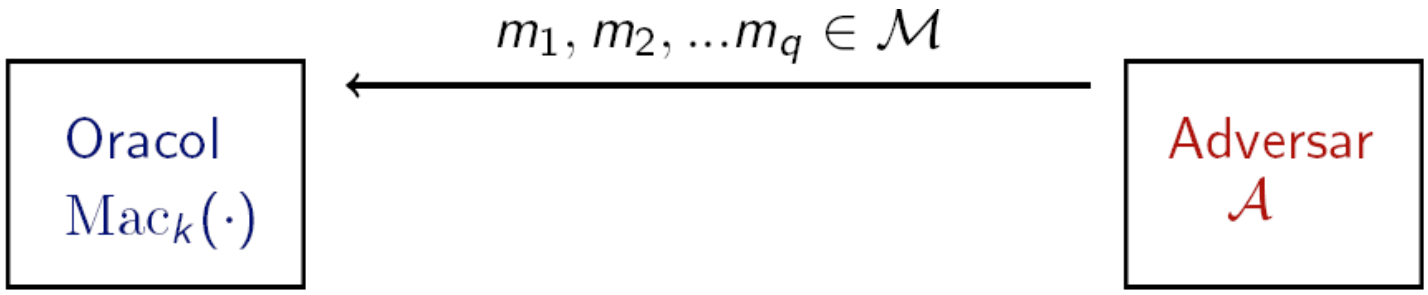
- Despre un MAC care satisface nivelul de securitate de mai sus spunem că *nu poate fi falsificat printr-un atac cu mesaj ales*;
- Aceasta înseamnă că un adversar nu este capabil să falsifice un tag valid pentru nici un mesaj ...
- ... deși poate obține tag-uri pentru orice mesaj ales de el, chiar *adaptiv* în timpul atacului.

Securitate MAC - formalizare

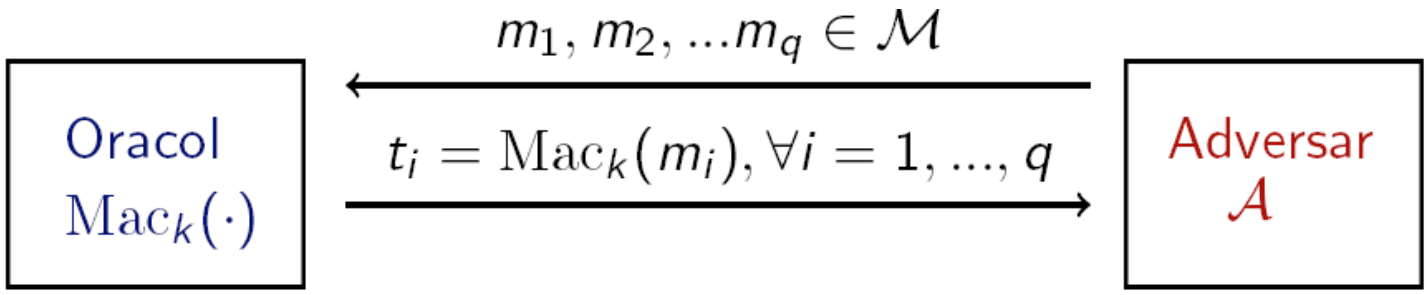
- Despre un MAC care satisface nivelul de securitate de mai sus spunem că *nu poate fi falsificat printr-un atac cu mesaj ales*;
- Aceasta înseamnă că un adversar nu este capabil să falsifice un tag valid pentru nici un mesaj ...
- ... deși poate obține tag-uri pentru orice mesaj ales de el, chiar *adaptiv* în timpul atacului.
- Pentru a da definiția formală, definim mai întâi un experiment pentru un MAC $\pi = (\mathrm{Mac}, \mathrm{Vrfy})$, în care considerăm un adversar $\mathcal{A}$ și parametrul de securitate n;

Experimentul $\mathrm{Mac}^{forge}_{\mathcal{A},\pi}(n)$

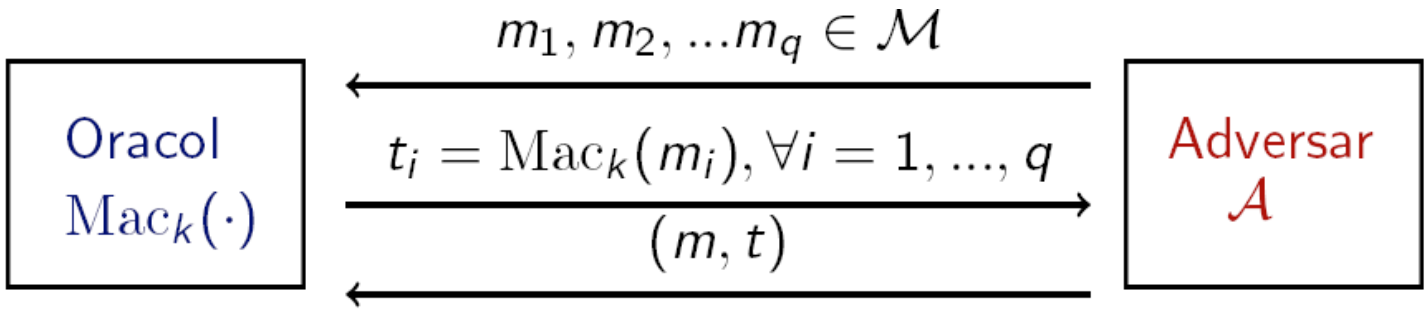
Experimentul $\mathrm{Mac}^{forge}_{\mathcal{A},\pi}(n)$



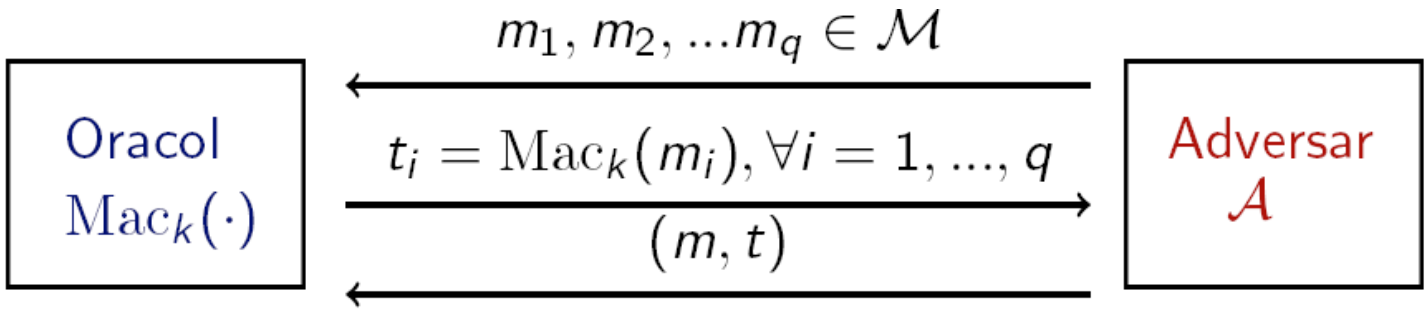
Experimentul $\mathrm{Mac}^{forge}_{\mathcal{A},\pi}(n)$



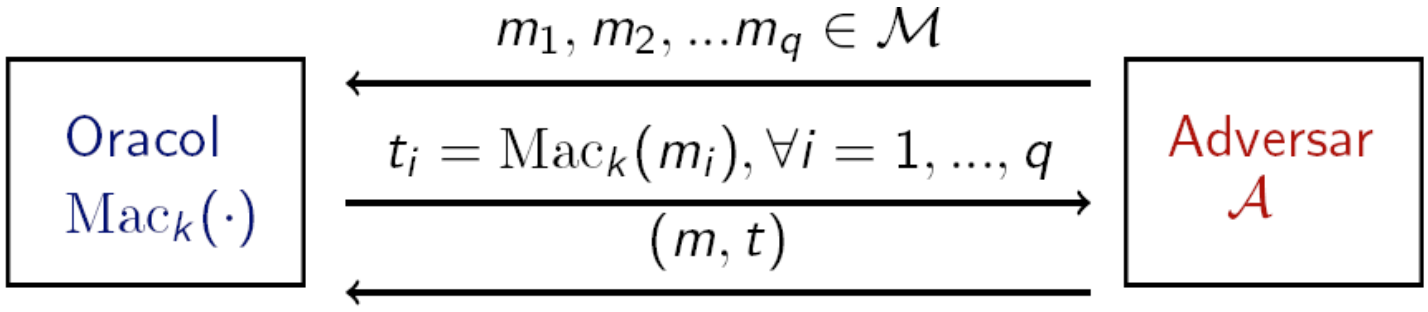
Experimentul $\mathrm{Mac}^{forge}_{\mathcal{A},\pi}(n)$



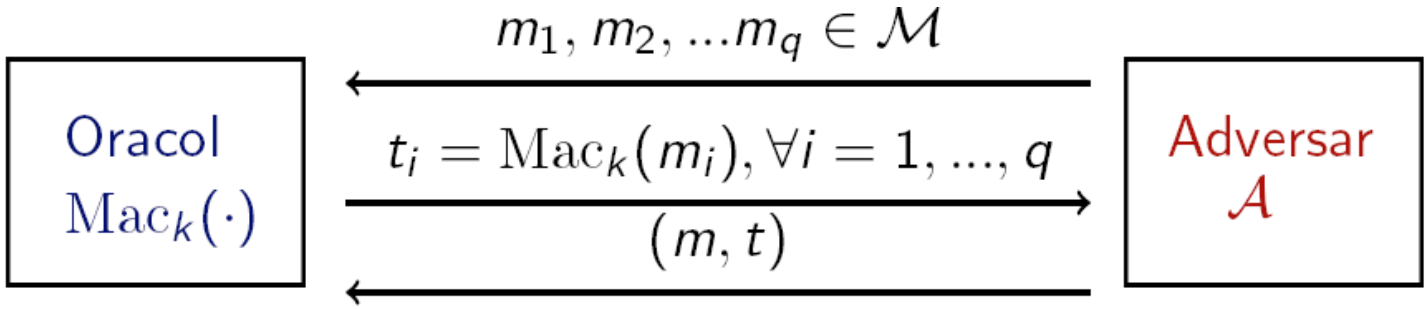
Experimentul $\mathrm{Mac}^{forge}_{\mathcal{A},\pi}(n)$



Experimentul $\mathrm{Mac}_{\mathcal{A},\pi}^{forge}(n)$

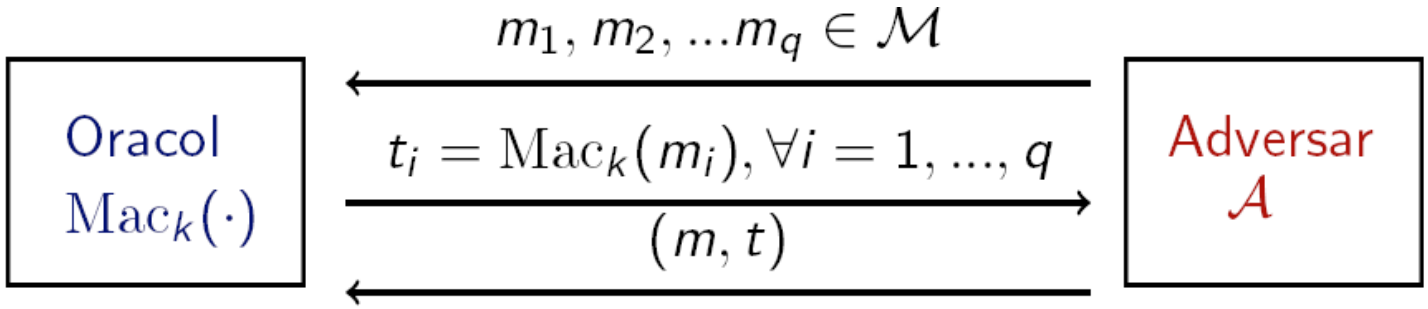


Experimentul $\mathrm{Mac}^{forge}_{\mathcal{A},\pi}(n)$



- Output-ul experimentului este 1 dacă și numai dacă:
  (1) $\mathrm{Vrfy}_k(m, t) = 1$ si (2) $m \notin \{m_1, ..., m_q\}$;

Experimentul $\mathrm{Mac}^{forge}_{\mathcal{A},\pi}(n)$



- Output-ul experimentului este 1 dacă și numai dacă:
  (1) $\mathrm{Vrfy}_k(m, t) = 1$ si (2) $m \notin \{m_1, ..., m_q\}$;
- Dacă $\mathrm{Mac}^{forge}_{\mathcal{A},\pi}(n) = 1$, spunem că $\mathcal{A}$ a efectuat experimentul cu succes.

Securitate MAC

Definiție

Un cod de autentificare al mesajelor $\pi = (\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$ *este sigur (nu poate fi falsificat printr-un atac cu mesaj ales) dacă pentru orice adversar polinomial* $\mathcal{A}$ *există o funcție neglijabilă* $negl$ *așa încât*

$$Pr[\mathrm{Mac}^{forge}_{\mathcal{A},\pi}(n) = 1] \leq negl(n).$$

Atacuri prin replicare

- Întrebare: De ce este necesară a doua condiție de la securitatea MAC (un adversar nu poate întoarce un mesaj pentru care anterior a cerut un tag)?

Atacuri prin replicare

- Întrebare: De ce este necesară a doua condiție de la securitatea MAC (un adversar nu poate întoarce un mesaj pentru care anterior a cerut un tag)?
- Răspuns: Pentru a evita atacurile triviale în care un adversar cere tag-ul aferent unui mesaj și apoi întoarce chiar acel mesaj împreună cu tag-ul primit.

Atacuri prin replicare

- Întrebare: De ce este necesară a doua condiție de la securitatea MAC (un adversar nu poate întoarce un mesaj pentru care anterior a cerut un tag)?
- Răspuns: Pentru a evita atacurile triviale în care un adversar cere tag-ul aferent unui mesaj și apoi întoarce chiar acel mesaj împreună cu tag-ul primit.
- Intrebare: Definiția MAC oferă protecție la atacurile prin replicare (in care un adversar copiază un mesaj împreună cu tag-ul aferent trimise de părțile comunicante)?

Atacuri prin replicare

- Întrebare: De ce este necesară a doua condiție de la securitatea MAC (un adversar nu poate întoarce un mesaj pentru care anterior a cerut un tag)?
- Răspuns: Pentru a evita atacurile triviale în care un adversar cere tag-ul aferent unui mesaj și apoi întoarce chiar acel mesaj împreună cu tag-ul primit.
- Intrebare: Definiția MAC oferă protecție la atacurile prin replicare (in care un adversar copiază un mesaj împreună cu tag-ul aferent trimise de părțile comunicante)?
- Răspuns: NU! MAC-urile nu oferă nici un fel de protecție la atacurile prin replicare.

Atacuri prin replicare

- De exemplu: Alice trimite către banca sa un ordin de transfer a 1.000$ din contul ei în contul lui Bob;

Atacuri prin replicare

- De exemplu: Alice trimite către banca sa un ordin de transfer a 1.000$ din contul ei în contul lui Bob;
- Pentru aceasta, Alice calculează un tag MAC și îl atașază mesajului așa încât banca știe că mesajul este autentic;

Atacuri prin replicare

- De exemplu: Alice trimite către banca sa un ordin de transfer a 1.000$ din contul ei în contul lui Bob;
- Pentru aceasta, Alice calculează un tag MAC și îl atașază mesajului așa încât banca știe că mesajul este autentic;
- Dacă MAC-ul este sigur, Bob nu va putea intercepta mesajul și modifica suma la 10.000$;

Atacuri prin replicare

- De exemplu: Alice trimite către banca sa un ordin de transfer a 1.000$ din contul ei în contul lui Bob;
- Pentru aceasta, Alice calculează un tag MAC și îl atașază mesajului așa încât banca știe că mesajul este autentic;
- Dacă MAC-ul este sigur, Bob nu va putea intercepta mesajul și modifica suma la 10.000$;
- Dar Bob poate intercepta mesajul și îl poate replica de *zece* ori către bancă;

Atacuri prin replicare

- De exemplu: Alice trimite către banca sa un ordin de transfer a 1.000$ din contul ei în contul lui Bob;
- Pentru aceasta, Alice calculează un tag MAC și îl atașază mesajului așa încât banca știe că mesajul este autentic;
- Dacă MAC-ul este sigur, Bob nu va putea intercepta mesajul și modifica suma la 10.000$;
- Dar Bob poate intercepta mesajul și îl poate replica de *zece* ori către bancă;
- Dacă banca îl acceptă, Bob va avea în cont 10.000$.

Atacuri prin replicare

- Un MAC nu protejează împotriva unui atac prin replicare pentru că definiția nu încorporează nici o noțiune de *stare* în algortimul de verificare;

Atacuri prin replicare

- Un MAC nu protejează împotriva unui atac prin replicare pentru că definiția nu încorporează nici o noțiune de *stare* în algortimul de verificare;
- Mai degrabă, protecția împotriva replicării trebuie făcută la nivel înalt de către aplicatiile care folosesc MAC-uri;

Construcția MAC-urilor sigure

- Avem nevoie de o funcție cu cheie, pentru care, dându-se $Mac_k(m_1), Mac_k(m_2)...$

Construcția MAC-urilor sigure

- Avem nevoie de o funcție cu cheie, pentru care, dându-se $Mac_k(m_1), Mac_k(m_2)...$
- ... să nu fie ușor (în timp polinomial) a găsi $Mac_k(m)$ pentru orice $m \notin \{m_1, m_2, ...\}$

Construcția MAC-urilor sigure

- Avem nevoie de o funcție cu cheie, pentru care, dându-se $Mac_k(m_1), Mac_k(m_2)...$
- ... să nu fie ușor (în timp polinomial) a găsi $Mac_k(m)$ pentru orice $m \notin \{m_1, m_2, ...\}$
- Funcția MAC ar putea fi un PRF

Construcția MAC-urilor sigure

- *Funcțiile pseudoaleatoare* (PRF) sunt un instrument bun pentru a construi MAC-uri sigure;

Construcție

Fie $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ *o PRF. Definim un MAC în felul următor:*

- Mac : *pentru o cheie* $k \in \{0,1\}^n$ *și un mesaj* $m \in \{0,1\}^n$, *calculează tag-ul* $t = F_k(m)$ *(dacă* $|m| \neq |k|$ *nu întoarce nimic);*
- Vrfy : *pentru o cheie* $k \in \{0,1\}^n$, *un mesaj* $m \in \{0,1\}^n$ *și un tag* $t \in \{0,1\}^n$, *întoarce 1 dacă și numai dacă* $t = F_k(m)$ *(dacă* $|m| \neq |k|$, *întoarce 0).*

Construcția MAC-urilor sigure

Teoremă

Dacă F este PRF, construcția de mai sus reprezintă un cod de autentificare a mesajelor sigur (nu poate fi falsificat prin atacuri cu mesaj ales).

MAC-uri pentru mesaje de lungime variabilă

- Construcția prezentată anterior funcționează doar pe mesaje de lungime fixă (PRF-urile sunt instanțiate cu sisteme de criptare bloc care, cel mai adesea, acceptă input-uri de 128 biți);

MAC-uri pentru mesaje de lungime variabilă

- Construcția prezentată anterior funcționează doar pe mesaje de lungime fixă (PRF-urile sunt instanțiate cu sisteme de criptare bloc care, cel mai adesea, acceptă input-uri de 128 biți);
- Insă în practică avem nevoie de mesaje de lungime variabilă;

MAC-uri pentru mesaje de lungime variabilă

- Construcția prezentată anterior funcționează doar pe mesaje de lungime fixă (PRF-urile sunt instanțiate cu sisteme de criptare bloc care, cel mai adesea, acceptă input-uri de 128 biți);
- Insă în practică avem nevoie de mesaje de lungime variabilă;
- Arătăm cum putem obține un MAC de lungime variabilă pornind de la un MAC de lungime fixă;

MAC-uri pentru mesaje de lungime variabilă

- Construcția prezentată anterior funcționează doar pe mesaje de lungime fixă (PRF-urile sunt instanțiate cu sisteme de criptare bloc care, cel mai adesea, acceptă input-uri de 128 biți);
- Insă în practică avem nevoie de mesaje de lungime variabilă;
- Arătăm cum putem obține un MAC de lungime variabilă pornind de la un MAC de lungime fixă;
- Fie $(\pi' = (\mathrm{Mac}', \mathrm{Vrfy}'))$ un MAC sigur de lungime fixă pentru mesaje de lungime n;

MAC-uri pentru mesaje de lungime variabilă

- Construcția prezentată anterior funcționează doar pe mesaje de lungime fixă (PRF-urile sunt instanțiate cu sisteme de criptare bloc care, cel mai adesea, acceptă input-uri de 128 biți);
- Insă în practică avem nevoie de mesaje de lungime variabilă;
- Arătăm cum putem obține un MAC de lungime variabilă pornind de la un MAC de lungime fixă;
- Fie $(\pi' = (\mathrm{Mac}', \mathrm{Vrfy}'))$ un MAC sigur de lungime fixă pentru mesaje de lungime n;
- Pentru a construi un MAC de lungime variabilă, putem sparge mesajul m în blocuri $m_1, ..., m_d$ și autentificam blocurile folosind π';

MAC-uri pentru mesaje de lungime variabilă

- Construcția prezentată anterior funcționează doar pe mesaje de lungime fixă (PRF-urile sunt instanțiate cu sisteme de criptare bloc care, cel mai adesea, acceptă input-uri de 128 biți);
- Insă în practică avem nevoie de mesaje de lungime variabilă;
- Arătăm cum putem obține un MAC de lungime variabilă pornind de la un MAC de lungime fixă;
- Fie $(\pi' = (\mathrm{Mac}', \mathrm{Vrfy}'))$ un MAC sigur de lungime fixă pentru mesaje de lungime n;
- Pentru a construi un MAC de lungime variabilă, putem sparge mesajul m în blocuri $m_1, ..., m_d$ și autentificam blocurile folosind π';
- Iată câteva modalități de a face aceasta:

MAC-uri pentru mesaje de lungime variabilă

1. *XOR pe toate blocurile cu autentificarea rezultatului*:
$$t = \mathrm{Mac}'_k(\oplus_i m_i)$$

MAC-uri pentru mesaje de lungime variabilă

1. *XOR pe toate blocurile cu autentificarea rezultatului*:
$$t = \mathrm{Mac}'_k(\oplus_i m_i)$$

   - Intrebare: Este sigură această metodă?

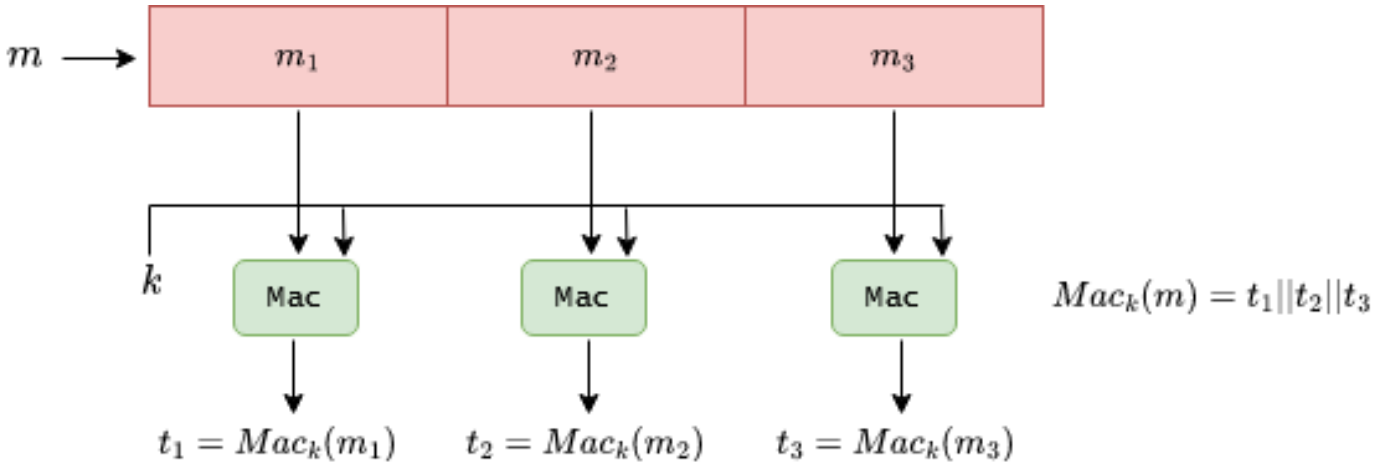
MAC-uri pentru mesaje de lungime variabilă

1. *XOR pe toate blocurile cu autentificarea rezultatului*:
$$t = \mathrm{Mac}'_k(\oplus_i m_i)$$

   - Intrebare: Este sigură această metodă?
   - Răspuns: NU! Un adversar poate modifica mesajul original m a.î. XOR-ul blocurilor nu se schimbă, el obținând un tag valid pentru un mesaj nou;

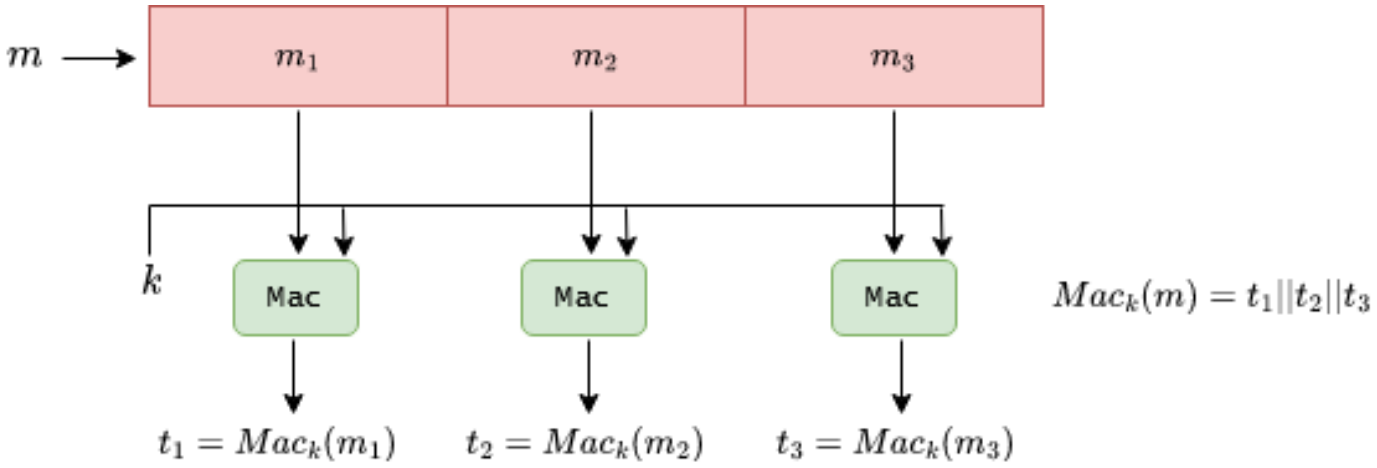
MAC-uri pentru mesaje de lungime variabilă

2. *Autentificare separată pentru fiecare bloc*:



MAC-uri pentru mesaje de lungime variabilă

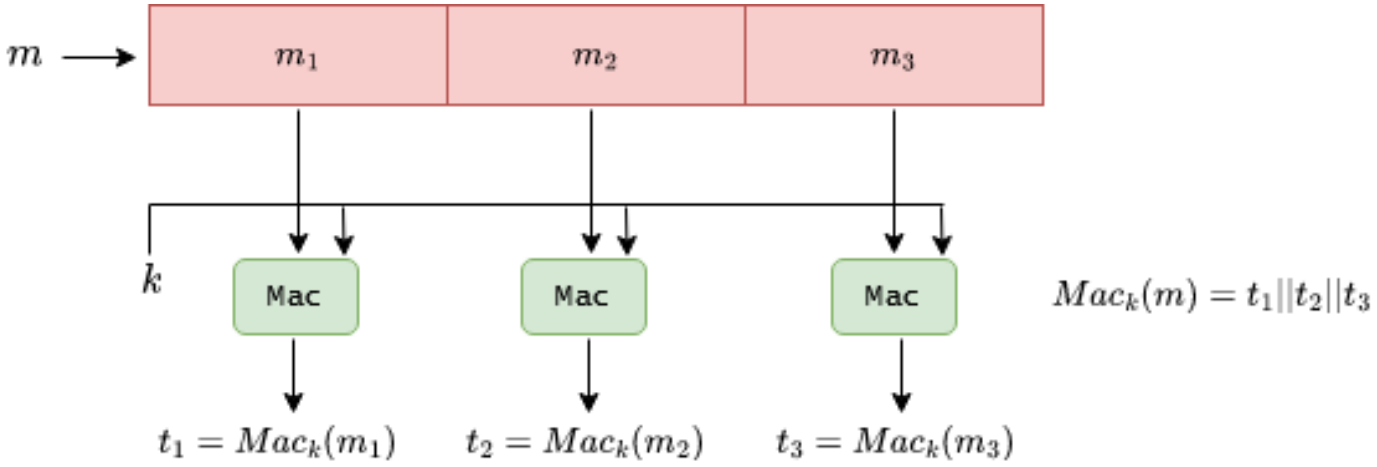
2. *Autentificare separată pentru fiecare bloc*:



- Intrebare: Este sigură această metodă?

MAC-uri pentru mesaje de lungime variabilă

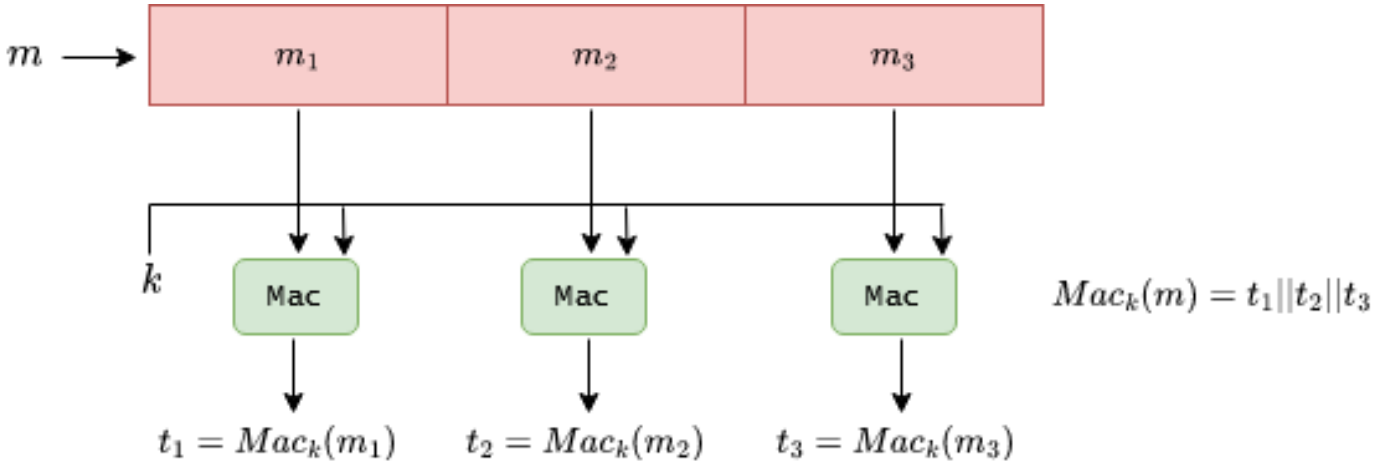
2. *Autentificare separată pentru fiecare bloc*:



- Intrebare: Este sigură această metodă?
- Răspuns: NU!

MAC-uri pentru mesaje de lungime variabilă

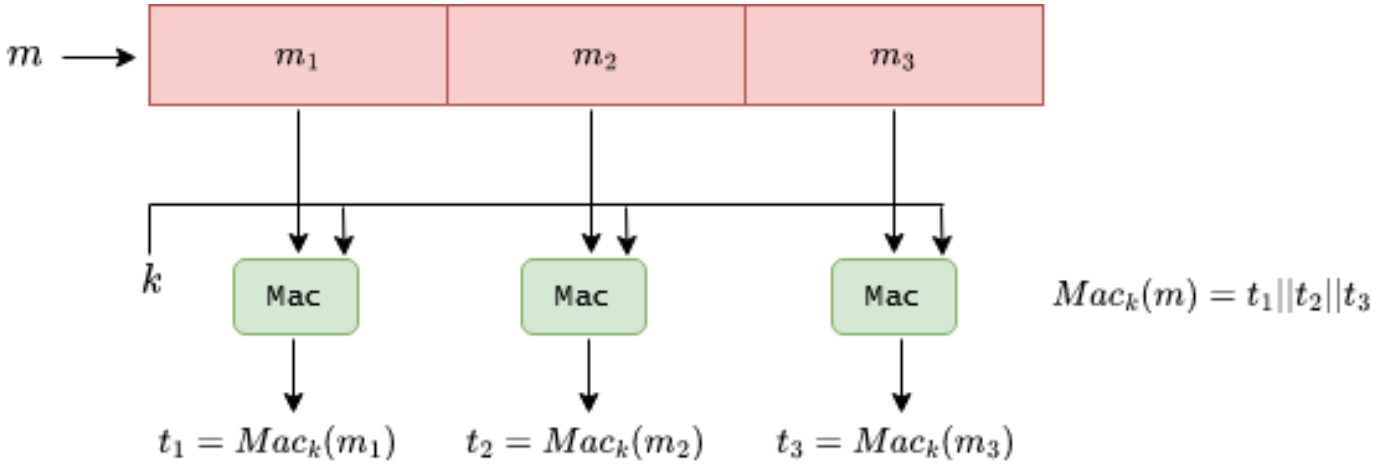
2. *Autentificare separată pentru fiecare bloc*:



- Intrebare: Este sigură această metodă?
- Răspuns: NU!
- Fiind dat (m, t) cu $m = m_1||m_2||m_3$ și $t = t_1||t_2||t_3$

MAC-uri pentru mesaje de lungime variabilă

2. *Autentificare separată pentru fiecare bloc*:

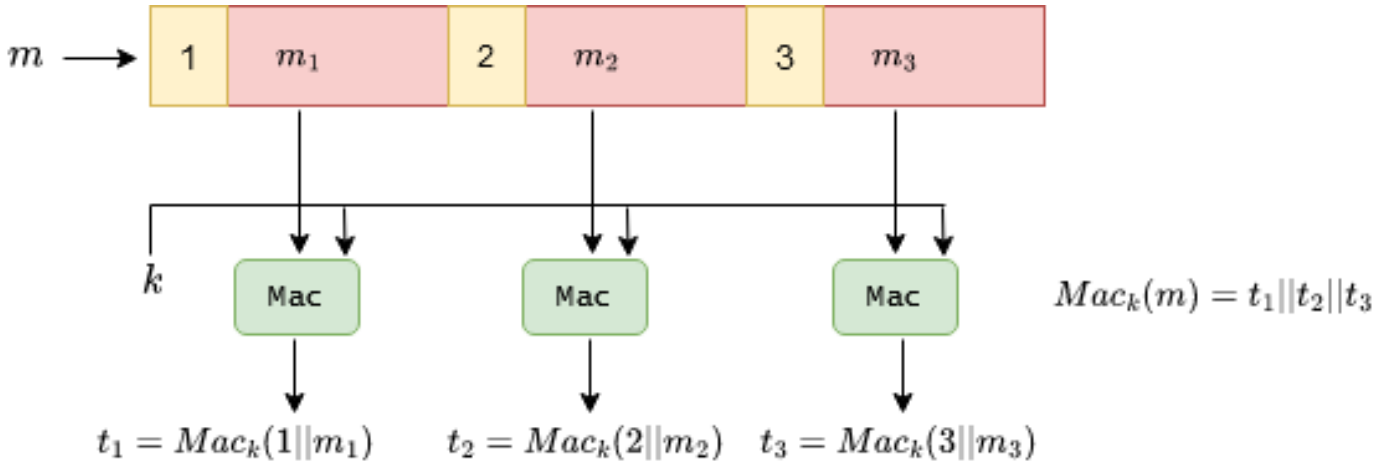


- Intrebare: Este sigură această metodă?
- Răspuns: NU!
- Fiind dat (m, t) cu $m = m_1||m_2||m_3$ și $t = t_1||t_2||t_3$
- Atunci (m', t') este o pereche validă cu $m' = m_2||m_1||m_3$ și $t' = t_2||t_1||t_3$

MAC-uri pentru mesaje de lungime variabilă

3. *Autentificare separată pentru fiecare bloc folosind o secvență de numere*:

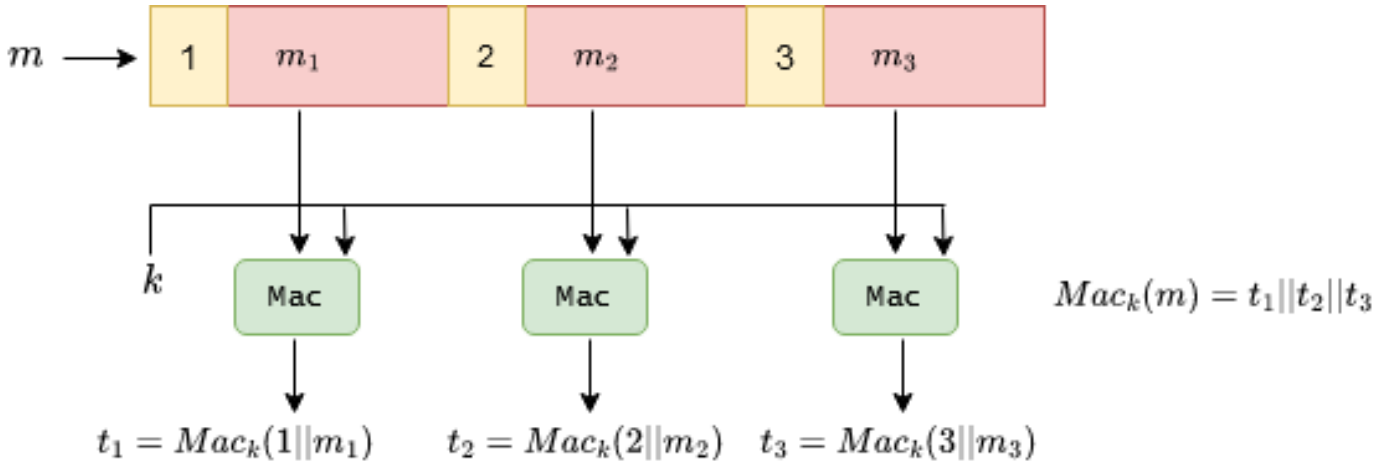
In felul acesta prevenim atacul anterior



MAC-uri pentru mesaje de lungime variabilă

3. *Autentificare separată pentru fiecare bloc folosind o secvență de numere*:

In felul acesta prevenim atacul anterior

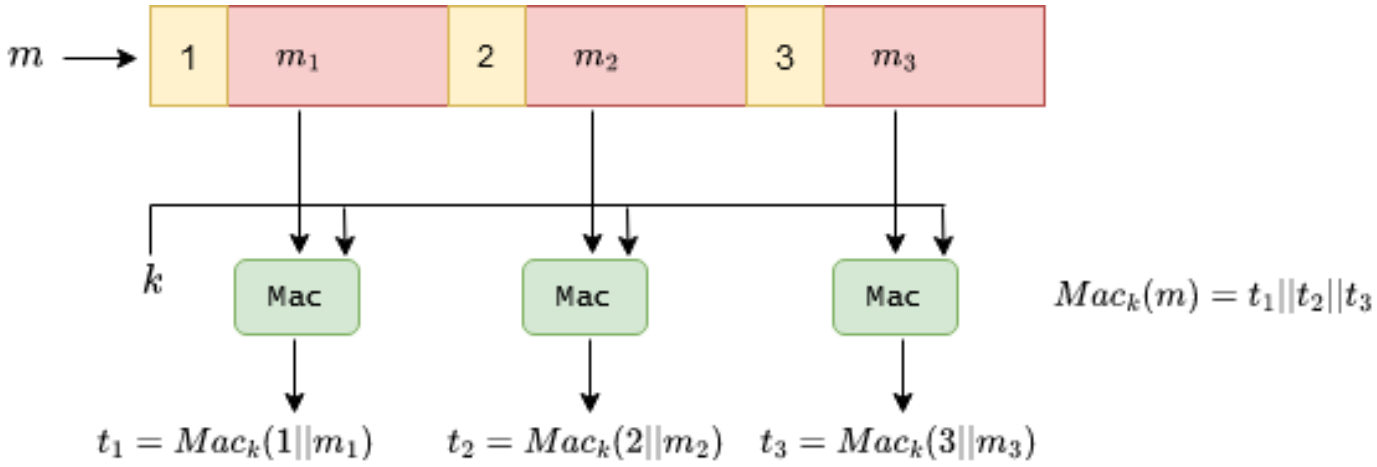


- Intrebare: Este sigură această metodă?

MAC-uri pentru mesaje de lungime variabilă

3. *Autentificare separată pentru fiecare bloc folosind o secvență de numere*:

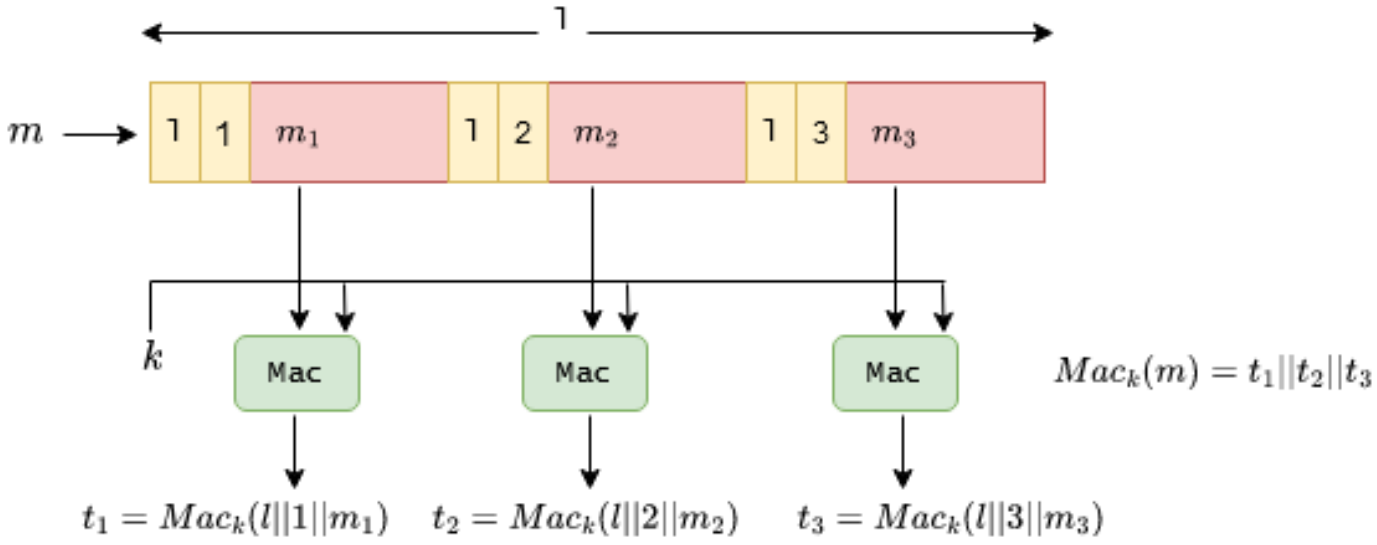
In felul acesta prevenim atacul anterior



- Intrebare: Este sigură această metodă?
- Răspuns: NU! Un adversar poate scoate blocuri de la sfârșitul mesajului: $t = t_1||t_2$ este un tag valid pentru mesajul $m = m_1||m_2$;

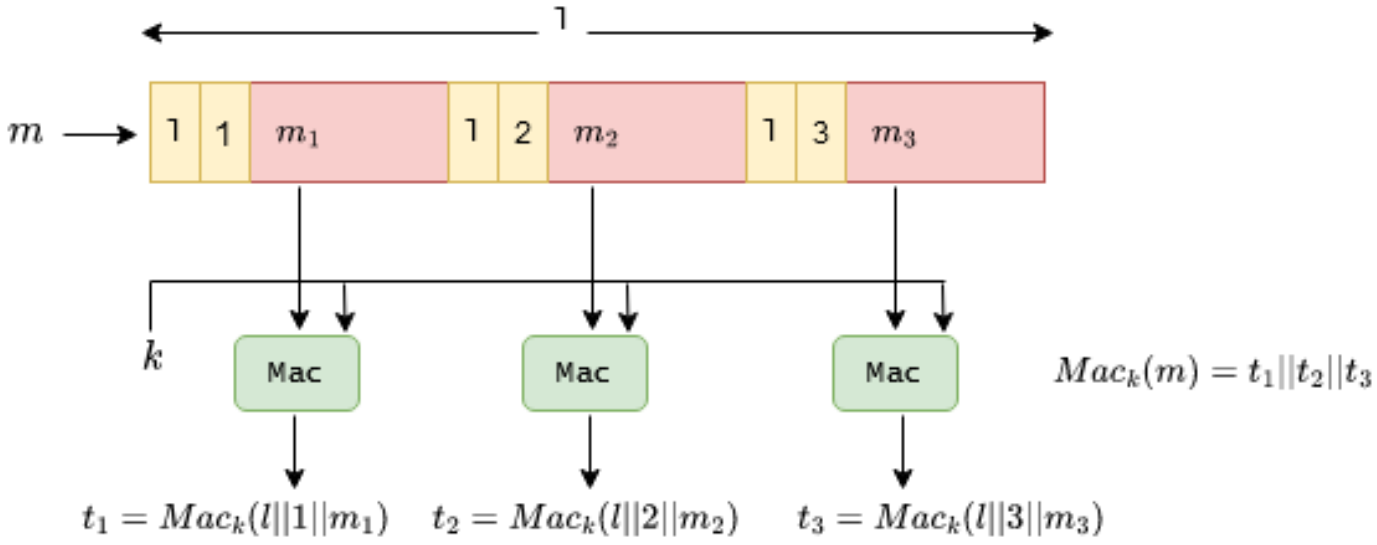
MAC-uri pentru mesaje de lungime variabilă

- Soluție pentru atacurile anterioare: adăugarea de informație suplimentară în fiecare bloc



MAC-uri pentru mesaje de lungime variabilă

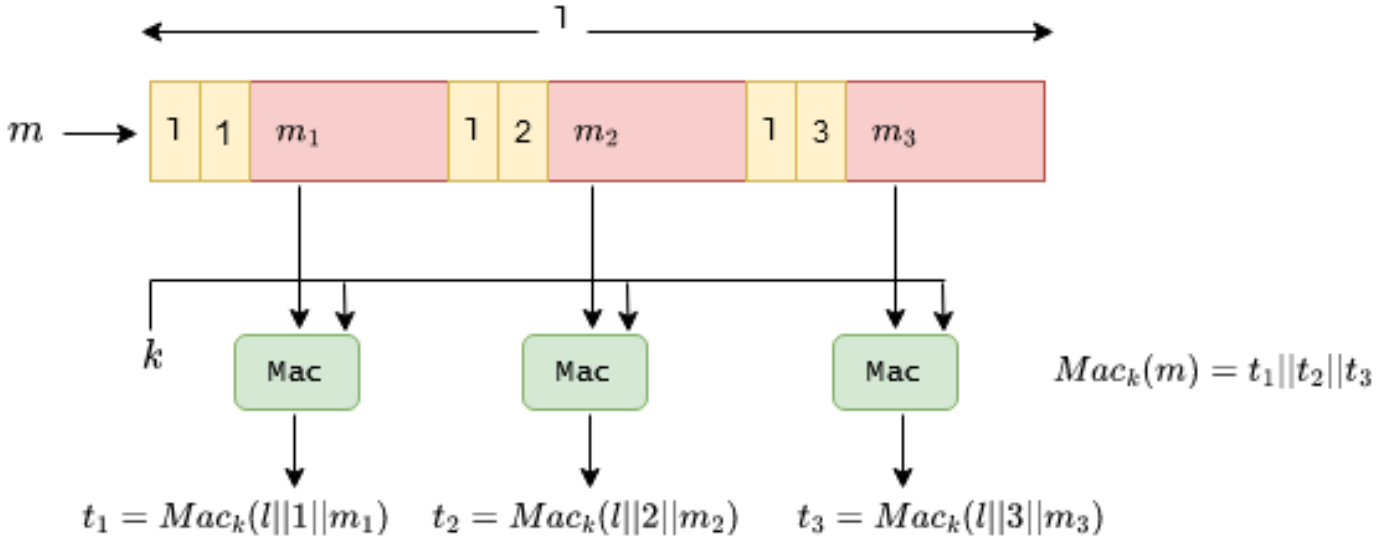
- Soluție pentru atacurile anterioare: adăugarea de informație suplimentară în fiecare bloc



- Intrebare: Este sigură această metodă?

MAC-uri pentru mesaje de lungime variabilă

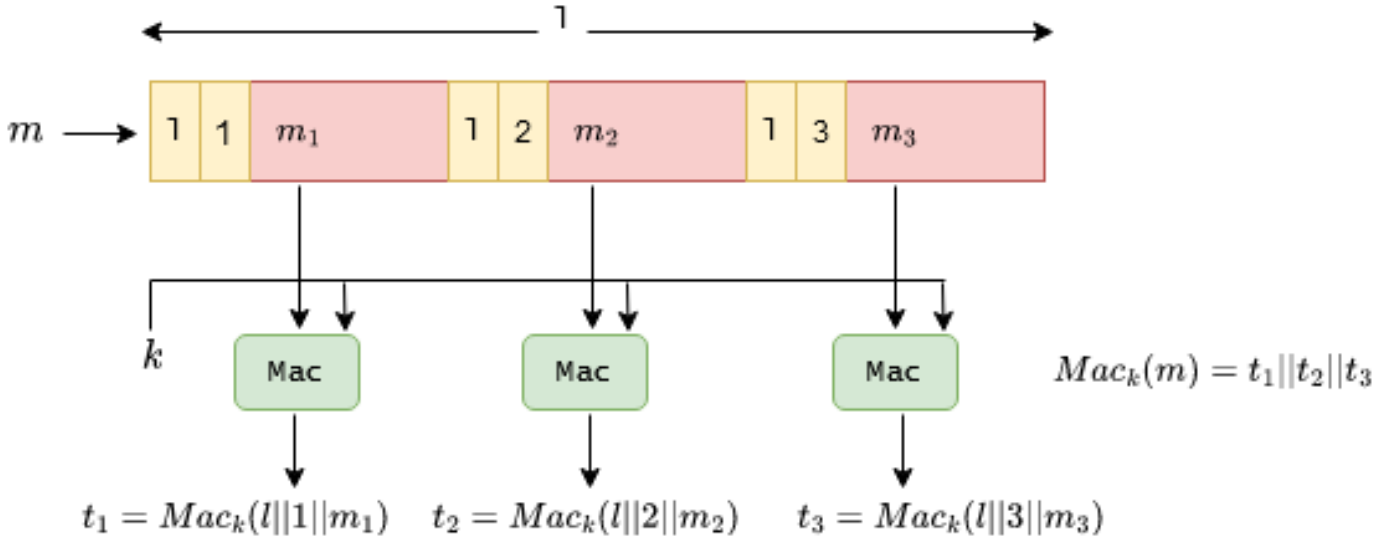
- Soluție pentru atacurile anterioare: adăugarea de informație suplimentară în fiecare bloc



- Intrebare: Este sigură această metodă?
- Răspuns: NU!
- Pentru perechea (m, t) cu $m = m_1||m_2||m_3$ și $t = t_1||t_2||t_3$ și perechea (m', t') cu $m' = m'_1||m'_2||m'_3$ și $t' = t'_1||t'_2||t'_3$ iar $|m| = |m'|$

MAC-uri pentru mesaje de lungime variabilă

- Soluție pentru atacurile anterioare: adăugarea de informație suplimentară în fiecare bloc



- Intrebare: Este sigură această metodă?
- Răspuns: NU!
- Pentru perechea (m, t) cu $m = m_1||m_2||m_3$ și $t = t_1||t_2||t_3$ și perechea (m', t') cu $m' = m'_1||m'_2||m'_3$ și $t' = t'_1||t'_2||t'_3$ iar $|m| = |m'|$
- perechea $(m_1||m'_2||m_3, t_1||t'_2||t_3)$ este una validă

CBC-MAC

- Soluția este ineficientă și greu de folosit în practică;

CBC-MAC

- Soluția este ineficientă și greu de folosit în practică;
- Insă, am văzut că putem construi MAC-uri sigure (chiar pentru mesaje de lungime variabilă) pe baza funcțiilor pseudoaleatoare (intrare de lungime fixă);

CBC-MAC

- Soluția este ineficientă și greu de folosit în practică;
- Insă, am văzut că putem construi MAC-uri sigure (chiar pentru mesaje de lungime variabilă) pe baza funcțiilor pseudoaleatoare (intrare de lungime fixă);
- Ceea ce înseamnă că putem construi MAC-uri sigure pornind de la cifruri bloc;

CBC-MAC

- Soluția este ineficientă și greu de folosit în practică;
- Insă, am văzut că putem construi MAC-uri sigure (chiar pentru mesaje de lungime variabilă) pe baza funcțiilor pseudoaleatoare (intrare de lungime fixă);
- Ceea ce înseamnă că putem construi MAC-uri sigure pornind de la cifruri bloc;
- Dar, cu construcția de mai sus, rezultatul e foarte ineficient: pentru un tag aferent unui mesaj de lungime $l \cdot n$, trebuie să aplicăm sistemul bloc de $4l$ ori iar tag-ul rezultat are $(4l + 1)n$ biți;

CBC-MAC

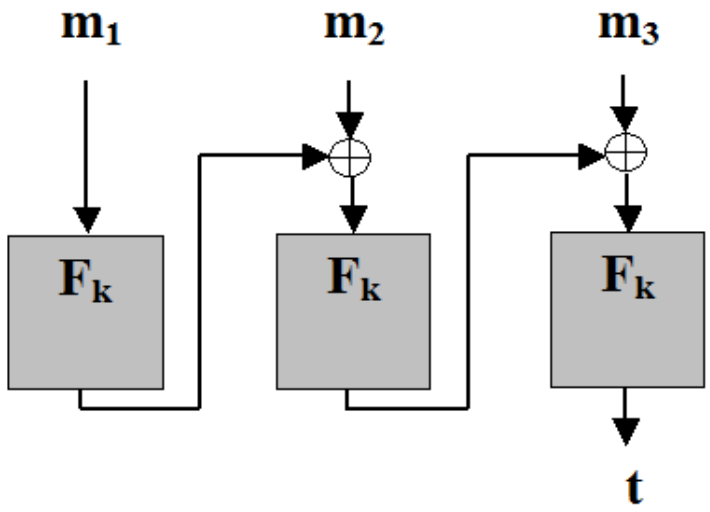
- O soluție mult mai eficientă este să folosim CBC-MAC;

CBC-MAC

- O soluție mult mai eficientă este să folosim CBC-MAC;
- CBC-MAC este o construcție similară cu modul CBC folosit pentru criptare;

CBC-MAC

- O soluție mult mai eficientă este să folosim CBC-MAC;
- CBC-MAC este o construcție similară cu modul CBC folosit pentru criptare;
- Folosind CBC-MAC, pentru un tag aferent unui mesaj de lungime $l \cdot n$, se aplică sistemul bloc doar de l ori, iar tag-ul are numai n biți.



Securitatea CBC-MAC

- Dacă F este o funcție pseudoaleatoare, construcția de mai sus reprezintă un cod de autentificare a mesajelor sigur (nu poate fi falsificat prin atacuri cu mesaj ales) pentru mesaje de lungime $\ell \cdot n$ pentru ℓ fixat.

Securitatea CBC-MAC

- Dacă F este o funcție pseudoaleatoare, construcția de mai sus reprezintă un cod de autentificare a mesajelor sigur (nu poate fi falsificat prin atacuri cu mesaj ales) pentru mesaje de lungime $\ell \cdot n$ pentru ℓ fixat.
- Construcția prezentată este sigură numai pentru autentificarea mesajelor de lungime fixă;

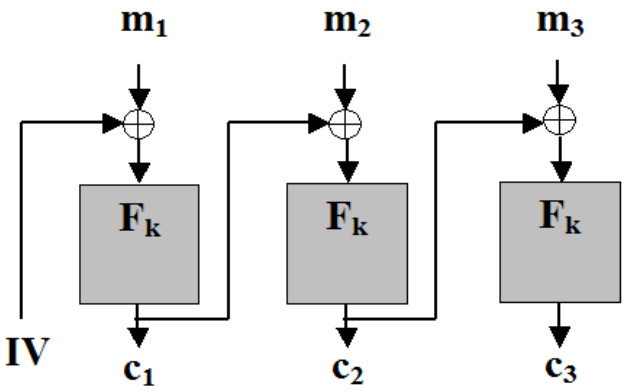
Securitatea CBC-MAC

- Dacă F este o funcție pseudoaleatoare, construcția de mai sus reprezintă un cod de autentificare a mesajelor sigur (nu poate fi falsificat prin atacuri cu mesaj ales) pentru mesaje de lungime $\ell \cdot n$ pentru ℓ fixat.
- Construcția prezentată este sigură numai pentru autentificarea mesajelor de lungime fixă;
- ℓ fixat înseamnă că cele două părți care comunică trebuie să partajeze ℓ în avans

Securitatea CBC-MAC

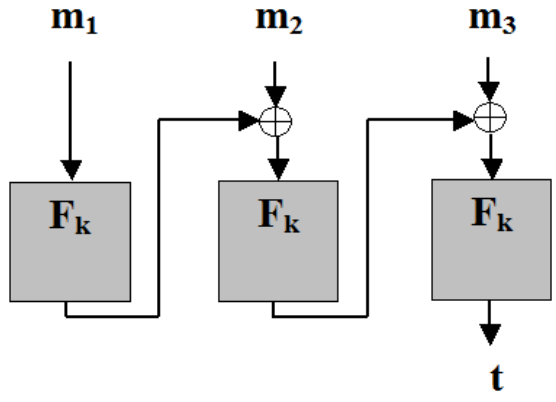
- Dacă F este o funcție pseudoaleatoare, construcția de mai sus reprezintă un cod de autentificare a mesajelor sigur (nu poate fi falsificat prin atacuri cu mesaj ales) pentru mesaje de lungime $\ell \cdot n$ pentru ℓ fixat.
- Construcția prezentată este sigură numai pentru autentificarea mesajelor de lungime fixă;
- ℓ fixat înseamnă că cele două părți care comunică trebuie să partajeze ℓ în avans
- Avantajul acestei construcții față de cea anterioară este că ea poate autentifica mesaje de lungime mult mai mare;

CBC-MAC vc. Criptare în modul CBC



Criptare în mod CBC

- IV este aleator pentru a obține securitate;
- toate blocurile c_i constituie mesajul criptat.



CBC-MAC

- $IV = 0^n$ este fixat pentru a obține securitate;
- doar ieșirea ultimului bloc constituie tag-ul (întoarcerea tuturor blocurilor intermediare duce la pierderea securității)

Construcții MAC

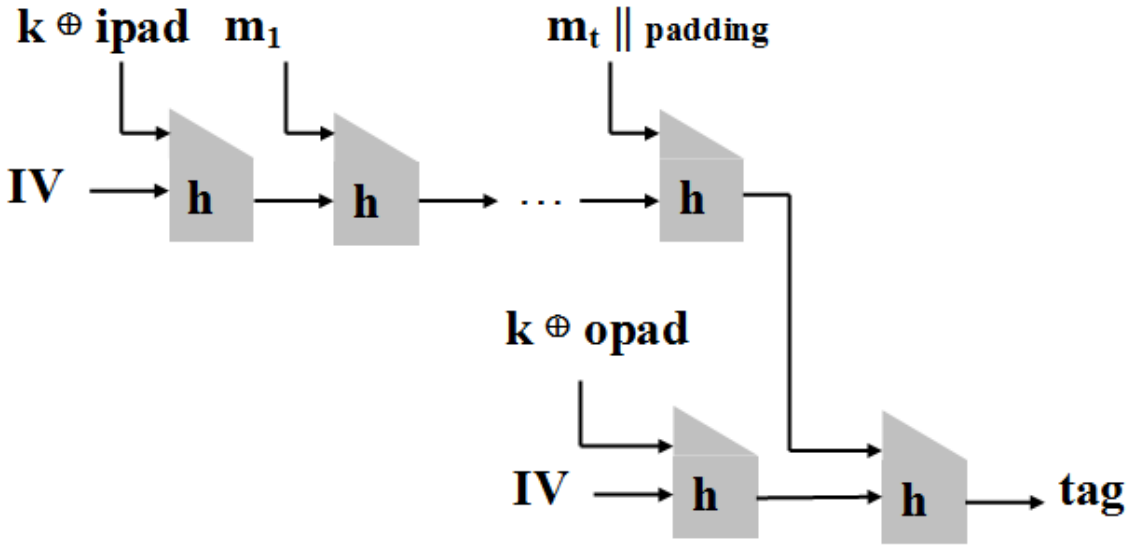
- Există două construcții de bază care se folosesc în practică:
  - CBC-MAC - folosit pe larg în industria bancară

Construcții MAC

- Există două construcții de bază care se folosesc în practică:
  - CBC-MAC - folosit pe larg în industria bancară
  - HMAC - pentru protocoale pe Internet: SSL, IPsec, SSH...
- Am studiat CBC-MAC mai devreme în curs;
  In continuare trecem în revistă HMAC

HMAC

- Folosim metoda standardizată HMAC (Hash MAC):



- HMAC se definește astfel:
- $\mathrm{Mac}(k, m)$: $t = H((k \oplus \texttt{opad}) \,||\, H((k \oplus \texttt{ipad}) \,||\, m))$;
- $\mathrm{Vrfy}(k, m, t) = 1 \iff t = \mathrm{Mac}(k, m)$.

Notații

- `ipad` și `opad` sunt două constante de lungimea unui bloc m_i
- `ipad` constă din byte-ul $0x5C$ repetat de atâtea ori cât e nevoie;
- `opad` constă din byte-ul $0x36$ repetat de atâtea ori cât e nevoie;
- IV este o constantă fixată.

Confidențialitate și integritate - criptare autentificată

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

Confidențialitate și integritate - criptare autentificată

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

Confidențialitate și integritate - criptare autentificată

- 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 adica criptare autentificată - authenticated encryption

Confidențialitate și integritate - criptare autentificată

- 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 adica criptare autentificată - authenticated encryption
- Nu orice combinație de schemă de criptare sigură și MAC sigur oferă cele două proprietăți de securitate!

Confidențialitate și integritate - criptare autentificată

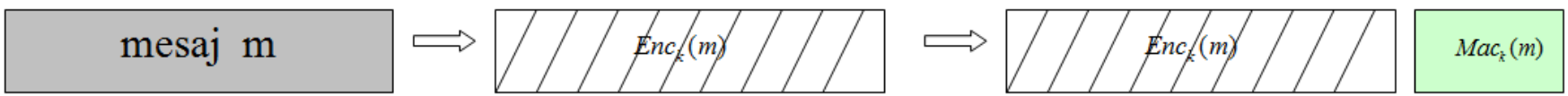
- Iată trei abordari uzuale pentru a combina criptarea și autentificarea mesajelor:

Confidențialitate și integritate - criptare autentificată

- 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

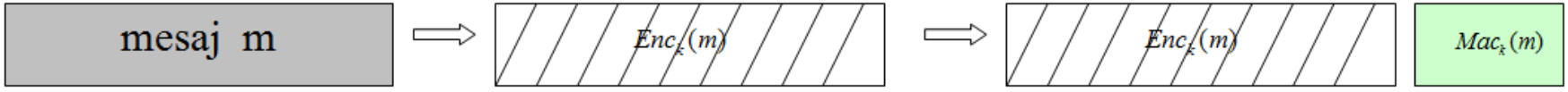
     $$c \leftarrow \mathrm{Enc}_{k_1}(m) \text{ și } t \leftarrow \mathrm{Mac}_{k_2}(m)$$

     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 - criptare autentificată

- 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 $\perp$.



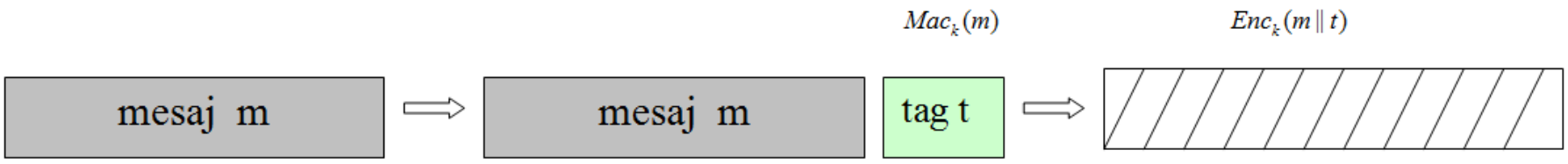
- Combinația aceasta **nu** este neaparat sigură; un MAC sigur nu implică nici un fel de confidențialitate;

Confidențialitate și integritate - criptare autentificată

- *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 $\perp$.

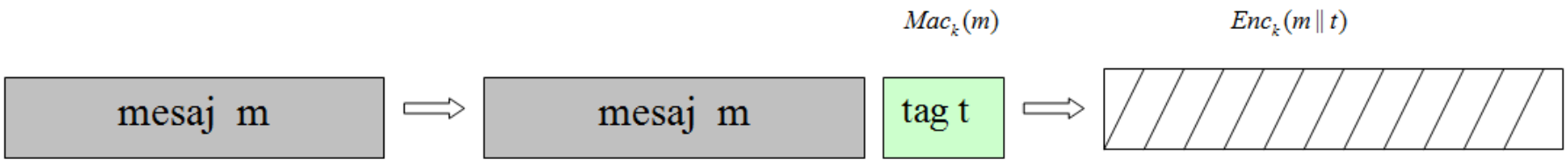


Confidențialitate și integritate - criptare autentificată

- *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 $\perp$.



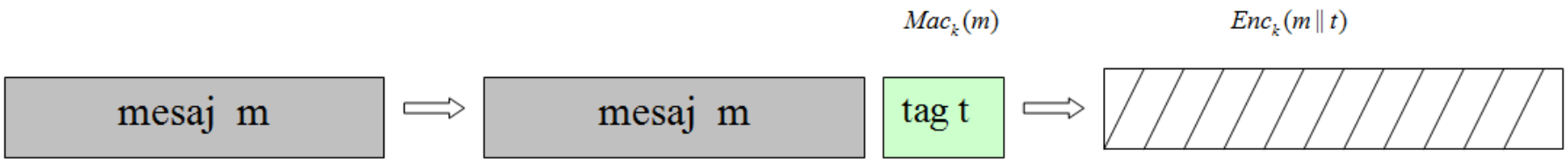
- Combinația aceasta **nu** este neapărat sigură;

Confidențialitate și integritate - criptare autentificată

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

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

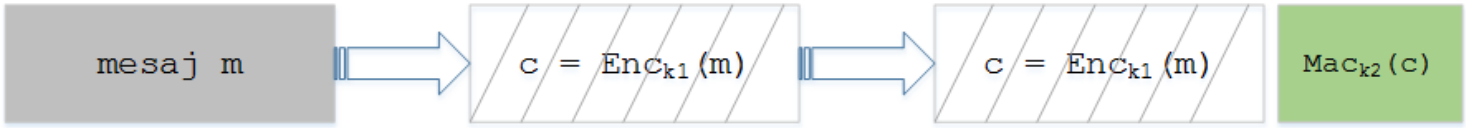
La recepție, $m||t = \mathrm{Dec}_{k_1}(c)$ și dacă $\mathrm{Vrfy}_{k_2}(m,t) = 1$, atunci întoarce m; altfel întoarce $\perp$.



- 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;

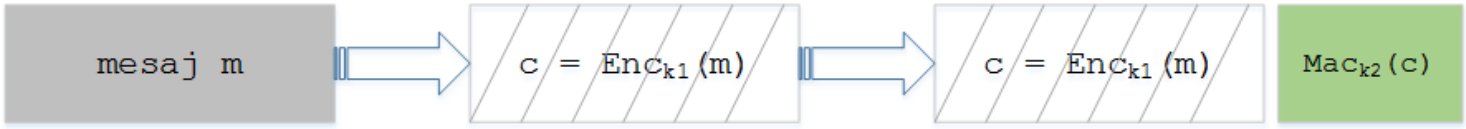
Securitate Criptare-apoi-autentificare

3. *Criptare-apoi-autentificare*



Securitate Criptare-apoi-autentificare

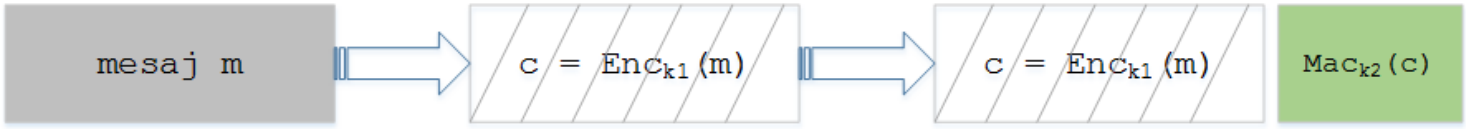
3. *Criptare-apoi-autentificare*



- 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 cu 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 cu aceeași cheie k atât pentru criptare cât și pentru autentificare;
- Definim $\mathrm{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0,1\}^{n/2}$, $r \leftarrow \{0,1\}^{n/2}$ aleator, 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 cu aceeași cheie k atât pentru criptare cât și pentru autentificare;
- Definim $\mathrm{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0,1\}^{n/2}$, $r \leftarrow \{0,1\}^{n/2}$ aleator, 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 cu aceeași cheie k atât pentru criptare cât și pentru autentificare;
- Definim $\mathrm{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0,1\}^{n/2}$, $r \leftarrow \{0,1\}^{n/2}$ aleator, 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 cu aceeași cheie k atât pentru criptare cât și pentru autentificare;
- Definim $\mathrm{Enc}_k(m) = F_k(m||r)$, pentru $m \in \{0,1\}^{n/2}$, $r \leftarrow \{0,1\}^{n/2}$ aleator, 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...
- $\mathrm{Enc}_k(m), \mathrm{Mac}_k(\mathrm{Enc}_k(m)) = F_k(m||r), F_k^{-1}(F_k(m||r)) = F_k(m||r), m||r$.

Securitate CCA vs. Criptare autentificată

- Orice schemă de criptare autentificată este și CCA-sigură dar există și scheme de criptare CCA-sigure care nu sunt scheme de criptare autentificată

Securitate CCA vs. Criptare autentificată

- Orice schemă de criptare autentificată este și CCA-sigură dar există și scheme de criptare CCA-sigure care nu sunt scheme de criptare autentificată
- Totusi, cele mai multe aplicații de scheme de criptare cu cheie secretă în prezența unui adversar activ necesită integritate

Criptare autentificată in practică

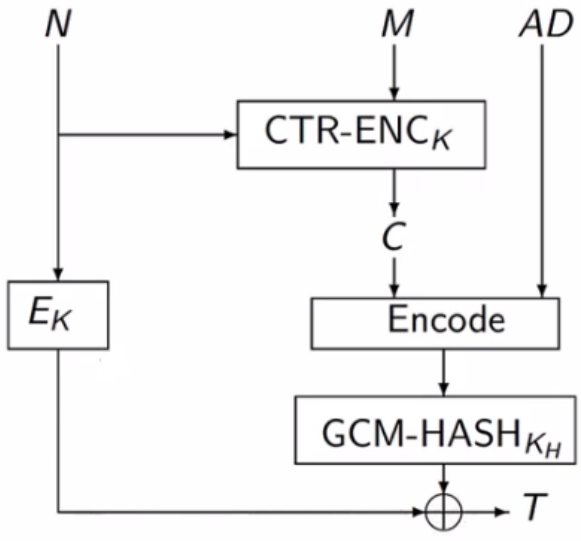
CA in	bazata pe	care in general este	dar in acest caz este
SSH	C_si_A	nesigura	sigura
SSL	A_apoi_C	nesigura	nesigura
IPSec	C_apoi_A	sigura	sigura
WinZip	C_apoi_A	sigura	nesigura

- CA = criptare autentificată;
  C-apoi-A = criptare-apoi-autentificare;
  C-si-A = criptare-si-autentificare

Criptare autentificată in practică

- Scheme de criptare autentificată: OCB, GCM, CCM, EAX
- TLS folosește GCM
- Competiția CAESAR pentru standardizarea de noi scheme de criptare autentificată
  http://competitions.cr.yp.to/caesar.html

Exemplu: GCM



- CTR-ENC reprezintă criptare în modul CTR
- GCM-HASH$_{K_H}$ - funcție hash bazată pe polinoame (peste corp Galois binar)
- K_H este derivată din E și K