



Criptografie și Securitate

- Prelegerea 12 - Coduri de autentificare a mesajelor

Adela Georgescu, Ruxandra F. Olimid

Facultatea de Matematică și Informatică
Universitatea din București

Cuprins

1. Necesitatea autentificării
2. Definiție MAC
3. Securitate MAC
4. CBC-MAC

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 aceasta 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 văzut cum putem obține aceasta cu ajutorul **schemelor de criptare**;
- ▶ Însă, 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 aceasta cu ajutorul **schemelor de criptare**;
- ▶ Însă, 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 aceasta cu ajutorul **schemelor de criptare**;
- ▶ Însă, 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 considerăm 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 considerăm 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 considerăm 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 considerăm 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

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

Comunicare sigură și integritatea mesajelor

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

Comunicare sigură și integritatea mesajelor

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

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

Criptare vs. autentificarea mesajelor

- ▶ Criptarea, în general, **NU** oferă integritatea 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 aibă sens în textul clar;

Criptare vs. autentificarea mesajelor

- ▶ Criptarea, în general, **NU** oferă integritatea 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

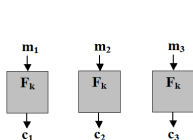
► Criptarea folosind sisteme fluide

- $Enc_k(m) = G(k) \oplus m$, 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.

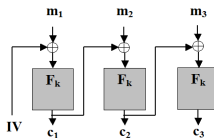
Criptare vs. autentificarea mesajelor

► Criptarea folosind sisteme bloc

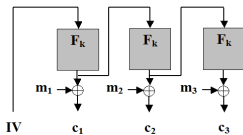
- **Intrebare:** Atacul de mai sus se poate aplica și pentru sistemele bloc cu modurile de operare studiate?



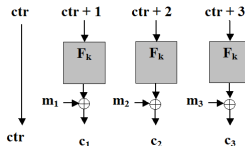
(a) ECB



(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 și c_{i-1} nu au fost modificate).

Coduri de autentificare a mesajelor - MAC

- ▶ Așa cum am văzut, criptarea nu rezolvă problema autentificării 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)**;

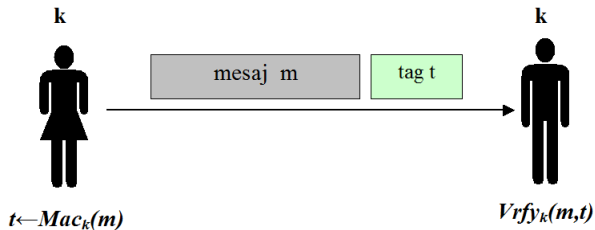
Coduri de autentificare a mesajelor - MAC

- ▶ 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;

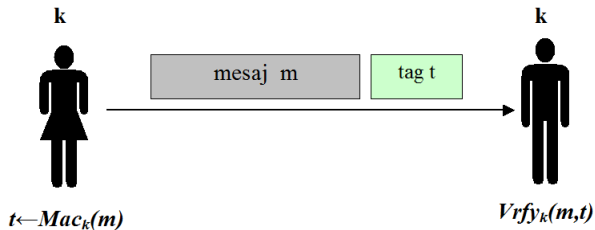
Coduri de autentificare a mesajelor - MAC

- ▶ 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

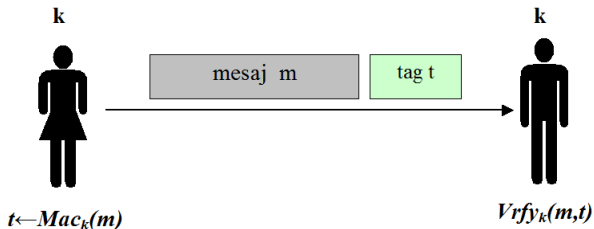


MAC - Definiție



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

MAC - Definiție



- ▶ Alice și Bob stabilesc 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 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 ;
- ▶ În 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-o pereche de algoritmi polinomiali $(\text{Mac}, \text{Vrfy})$ unde:

1. $\text{Mac} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{T}$ este algoritmul de generare a tag-urilor
 $t \leftarrow \text{Mac}_k(m)$;
2. $\text{Vrfy} : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \rightarrow \{0, 1\}$
este algoritmul de verificare ce întoarce un bit
 $b = \text{Vrfy}_k(m, t)$ cu semnificația că:
 - ▶ $b = 1$ înseamnă valid
 - ▶ $b = 0$ înseamnă invalid

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

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 împreună 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 împreună cu tag-urile aferente.
- ▶ Adversarul poate influența conținutul mesajelor (direct sau indirect), fiind deci un adversar *activ*.

Securitate MAC - formalizare

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

Securitate MAC - formalizare

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

Securitate MAC - formalizare

- ▶ Formal, îi dăm adversarului acces la un *oracol* $\text{Mac}_k(\cdot)$;
- ▶ Adversarul poate trimite orice mesaj m dorit către oracol și primește înapoi un tag corespunzător $t \leftarrow \text{Mac}_k(m)$;
- ▶ 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* $\text{Mac}_k(\cdot)$;
- ▶ Adversarul poate trimite orice mesaj m dorit către oracol și primește înapoi un tag corespunzător $t \leftarrow \text{Mac}_k(m)$;
- ▶ 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 : $\text{Vrfy}_k(m, t) = 1$;

Securitate MAC - formalizare

- ▶ Formal, îi dăm adversarului acces la un *oracol* $\text{Mac}_k(\cdot)$;
- ▶ Adversarul poate trimite orice mesaj m dorit către oracol și primește înapoi un tag corespunzător $t \leftarrow \text{Mac}_k(m)$;
- ▶ 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 : $\text{Vrfy}_k(m, t) = 1$;
 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

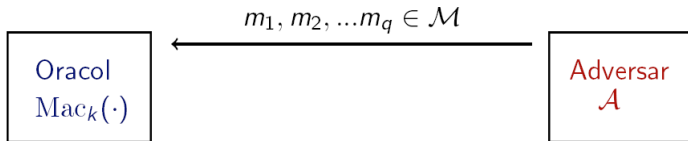
- ▶ Despre un MAC care satisface nivelul de securitate de mai sus spunem că *nu poate fi falsificat printr-un atac cu mesaj ales*;
- ▶ Aceasta înseamnă că un adversar nu este capabil să falsifice un tag valid pentru nici un mesaj ...
- ▶ ... deși poate obține tag-uri pentru orice mesaj ales de el, chiar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

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

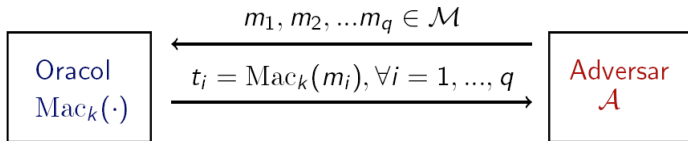
Oracol
 $\text{Mac}_k(\cdot)$

Adversar
 \mathcal{A}

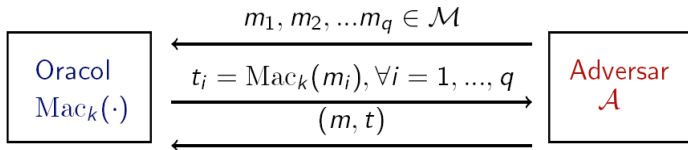
Experimentul $\text{Mac}_{\mathcal{A},\pi}^{\text{forge}}(n)$



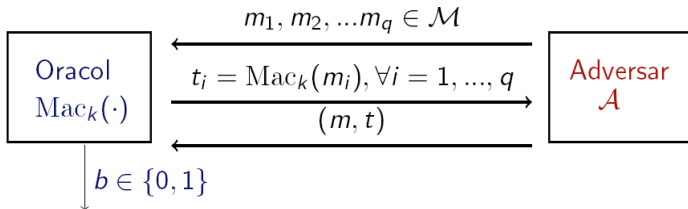
Experimentul $\text{Mac}_{\mathcal{A}, \pi}^{\text{forge}}(n)$



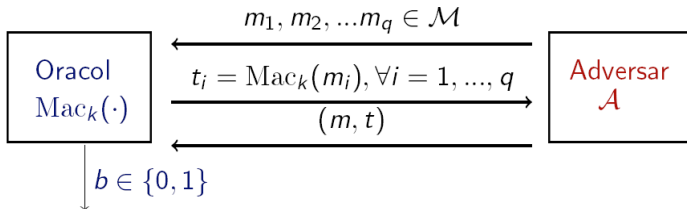
Experimentul $\text{Mac}_{\mathcal{A}, \pi}^{\text{forge}}(n)$



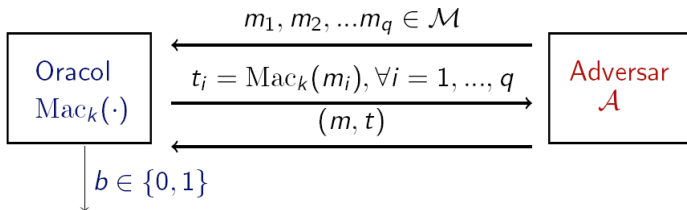
Experimentul $\text{Mac}_{\mathcal{A}, \pi}^{\text{forge}}(n)$



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

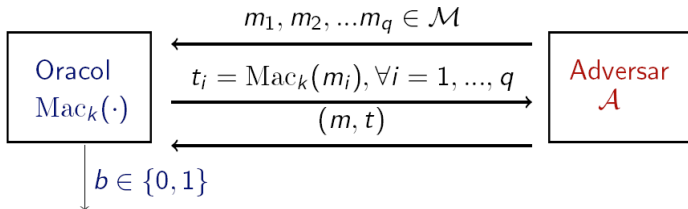


Experimentul $\text{Mac}_{\mathcal{A}, \pi}^{\text{forge}}(n)$



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

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



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

Definiție

Un cod de autentificare al mesajelor $\pi = (\text{Mac}, \text{Vrfy})$ 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

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

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 prin replicare în 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 prin replicare în care un adversar copiază un mesaj împreună cu tag-ul aferent trimise de părțile comunicante;
- ▶ **Intrebare:** Definiția MAC oferă protecție la atacurile prin replicare efectuate chiar 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 prin replicare în care un adversar copiază un mesaj împreună cu tag-ul aferent trimise de părțile comunicante;
- ▶ **Intrebare:** Definiția MAC oferă protecție la atacurile prin replicare efectuate chiar de părțile comunicante?
- ▶ **Răspuns:** NU! MAC-urile nu oferă nici un fel de protecție la atacurile prin replicare efectuate de părțile comunicante.

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 algoritmul 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 algoritmul de verificare;
- ▶ Mai degrabă, protecția împotriva replicării trebuie făcută la nivel înalt de către aplicațiile care folosesc MAC-uri;

Atacuri prin replicare

- ▶ Un MAC nu protejează împotriva unui atac prin replicare pentru că definiția nu încorporează nici o noțiune de *stare* în algoritmul de verificare;
- ▶ Mai degrabă, protecția împotriva replicării trebuie făcută la nivel înalt de către aplicațiile care folosesc MAC-uri;
- ▶ Două tehnici comune de protejare împotriva atacurilor prin replicare folosesc *secvențe de numere* sau *ștampilă de timp*;

Atacuri prin replicare

- ▶ Un MAC nu protejează împotriva unui atac prin replicare pentru că definiția nu încorporează nici o noțiune de *stare* în algoritmul de verificare;
- ▶ Mai degrabă, protecția împotriva replicării trebuie făcută la nivel înalt de către aplicațiile care folosesc MAC-uri;
- ▶ Două tehnici comune de protejare împotriva atacurilor prin replicare folosesc *secvențe de numere* sau *ștampilă de timp*;
- ▶ Pentru secvențe de numere, fiecare mesaj m are asignat un număr i iar tag-ul este calculat pe mesajul $i||m$;

Atacuri prin replicare

- ▶ Un MAC nu protejează împotriva unui atac prin replicare pentru că definiția nu încorporează nici o noțiune de *stare* în algoritmul de verificare;
- ▶ Mai degrabă, protecția împotriva replicării trebuie făcută la nivel înalt de către aplicațiile care folosesc MAC-uri;
- ▶ Două tehnici comune de protejare împotriva atacurilor prin replicare folosesc *secvențe de numere* sau *ștampilă de timp*;
- ▶ Pentru secvențe de numere, fiecare mesaj m are asignat un număr i iar tag-ul este calculat pe mesajul $i||m$;
- ▶ **Întrebare:** De ce această tehnică protejează împotriva atacurilor prin replicare?

Atacuri prin replicare

- **Răspuns:** Pentru că orice nouă replicare a lui m trebuie să construiască un tag pentru mesajul $i' || m$, unde i' nu a mai fost folosit niciodată;

Atacuri prin replicare

- ▶ **Răspuns:** Pentru că orice nouă replicare a lui m trebuie să construiască un tag pentru mesajul $i' || m$, unde i' nu a mai fost folosit niciodată;
- ▶ Dezavantajul este că trebuie stocată o listă cu numerele folosite anterior

Atacuri prin replicare

- ▶ **Răspuns:** Pentru că orice nouă replicare a lui m trebuie să construiască un tag pentru mesajul $i' || m$, unde i' nu a mai fost folosit niciodată;
- ▶ Dezavantajul este că trebuie stocată o listă cu numerele folosite anterior
- ▶ O alternativă ar fi folosirea ștampilelor de timp: la recepția mesajului, destinatarul verifică dacă ștampila de timp inclusă se află într-un interval de timp acceptabil;

Atacuri prin replicare

- ▶ **Răspuns:** Pentru că orice nouă replicare a lui m trebuie să construiască un tag pentru mesajul $i' || m$, unde i' nu a mai fost folosit niciodată;
- ▶ Dezavantajul este că trebuie stocată o listă cu numerele folosite anterior
- ▶ O alternativă ar fi folosirea ștampilelor de timp: la recepția mesajului, destinatarul verifică dacă ștampila de timp inclusă se află într-un interval de timp acceptabil;
- ▶ Aceasta presupune ca părțile să aibă ceasuri sincronizate;

Atacuri prin replicare

- ▶ **Răspuns:** Pentru că orice nouă replicare a lui m trebuie să construiască un tag pentru mesajul $i' || m$, unde i' nu a mai fost folosit niciodată;
- ▶ Dezavantajul este că trebuie stocată o listă cu numerele folosite anterior
- ▶ O alternativă ar fi folosirea ștampilelor de timp: la recepția mesajului, destinatarul verifică dacă ștampila de timp inclusă se află într-un interval de timp acceptabil;
- ▶ Aceasta presupune ca părțile să aibă ceasuri sincronizate;
- ▶ În plus, dacă un atac prin replicare este suficient de rapid, el se poate desfășura cu succes chiar și în aceste condiții.

Constructia MAC-urilor sigure

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

Construcție

Fie $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 o funcție aleatoare, construcția de mai sus reprezintă un cod de autentificare a mesajelor sigur (nu poate fi falsificat prin atacuri cu mesaj ales).

Demonstrație intuitivă

- ▶ Dacă un tag t este obținut prin aplicarea unei funcții pseudoaleatoare pe un mesaj m , atunci falsificarea unui tag aferent unui mesaj ne-autentificat anterior presupune ca adversarul să ghicească valoarea funcției într-un "punct nou" (i.e. mesaj);

Demonstrație intuitivă

- ▶ Dacă un tag t este obținut prin aplicarea unei funcții pseudoaleatoare pe un mesaj m , atunci falsificarea unui tag aferent unui mesaj ne-autentificat anterior presupune ca adversarul să ghicească valoarea funcției într-un "punct nou" (i.e. mesaj);
- ▶ Probabilitatea de a ghici valoarea unei funcții aleatoare într-un punct nou este 2^{-n} (unde n este lungimea ieșirii funcției);

Demonstrație intuitivă

- ▶ Dacă un tag t este obținut prin aplicarea unei funcții pseudoaleatoare pe un mesaj m , atunci falsificarea unui tag aferent unui mesaj ne-autentificat anterior presupune ca adversarul să ghicească valoarea funcției într-un "punct nou" (i.e. mesaj);
- ▶ Probabilitatea de a ghici valoarea unei funcții aleatoare într-un punct nou este 2^{-n} (unde n este lungimea ieșirii funcției);
- ▶ Prin urmare, probabilitatea de a ghici valoarea într-un punct nou pentru o funcție pseudoaleatoare nu poate fi decât neglijabil mai mare.

MAC-uri pentru mesaje de lungime variabilă

- ▶ Construcția prezentată anterior funcționează doar pe mesaje de lungime fixă;

MAC-uri pentru mesaje de lungime variabilă

- ▶ Construcția prezentată anterior funcționează doar pe mesaje de lungime fixă;
- ▶ Însă 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;
- ▶ Însă 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;
- ▶ Însă 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 $(\pi' = (\text{Mac}', \text{Vrfy}'))$ 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;
- ▶ Însă 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 $(\pi' = (\text{Mac}', \text{Vrfy}'))$ 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, \dots, m_d și autentificăm blocurile folosind π' ;

MAC-uri pentru mesaje de lungime variabilă

- ▶ Construcția prezentată anterior funcționează doar pe mesaje de lungime fixă;
- ▶ Însă 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 $(\pi' = (\text{Mac}', \text{Vrfy}'))$ 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, \dots, m_d și autentificăm 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:*

$$t = \text{Mac}'_k(\oplus_i m_i)$$

MAC-uri pentru mesaje de lungime variabilă

1. *XOR pe toate blocurile cu autentificarea rezultatului:*

$$t = \text{Mac}'_k(\oplus_i m_i)$$

- **Intrebare:** Este sigură această metodă?

MAC-uri pentru mesaje de lungime variabilă

1. *XOR pe toate blocurile cu autentificarea rezultatului:*

$$t = \text{Mac}'_k(\oplus_i m_i)$$

- ▶ **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:*

$$(t_1, \dots, t_d), \text{ unde } t_i = \text{Mac}'_k(m_i)$$

MAC-uri pentru mesaje de lungime variabilă

2. *Autentificare separată pentru fiecare bloc:*

$$(t_1, \dots, t_d), \text{ unde } t_i = \text{Mac}'_k(m_i)$$

► **Intrebare:** Este sigură această metodă?

MAC-uri pentru mesaje de lungime variabilă

2. *Autentificare separată pentru fiecare bloc:*

$$(t_1, \dots, t_d), \text{ unde } t_i = \text{Mac}'_k(m_i)$$

- ▶ **Intrebare:** Este sigură această metodă?
- ▶ **Răspuns:** NU! Un adversar poate schimba ordinea blocurilor în mesajul m , el obținând un tag valid pentru un mesaj nou;

MAC-uri pentru mesaje de lungime variabilă

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

$$(t_1, \dots, t_d), \text{ unde } t_i = \text{Mac}'_k(i || m_i)$$

MAC-uri pentru mesaje de lungime variabilă

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

$$(t_1, \dots, t_d), \text{ unde } t_i = \text{Mac}'_k(i || m_i)$$

► **Intrebare:** Este sigură această metodă?

MAC-uri pentru mesaje de lungime variabilă

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

$$(t_1, \dots, t_d), \text{ unde } t_i = \text{Mac}'_k(i || m_i)$$

- **Intrebare:** Este sigură această metodă?
- **Răspuns:** NU! Un adversar poate scoate blocuri de la sfârșitul mesajului: (t_1, \dots, t_{d-1}) este un tag valid pentru mesajul (m_1, \dots, m_{d-1}) ;
Mai mult, dacă (t_1, \dots, t_d) și (t'_1, \dots, t'_d) sunt tag-uri valide pentru mesajele $m = m_1, \dots, m_d$ și $m' = m'_1, \dots, m'_d$, atunci $(t_1, t'_2, t_3, t'_4, \dots)$ este un tag valid pentru mesajul $m_1, m'_2, m_3, m'_4, \dots$

MAC-uri pentru mesaje de lungime variabilă

- ▶ O soluție pentru atacurile anterioare o reprezintă adăugarea de informație suplimentară în fiecare bloc, în afara numărului de secvență:
 - ▶ un identificator aleator de mesaj - previne combinarea blocurilor din mesaje diferite
 - ▶ lungimea mesajului - previne modificarea lungimii mesajelor

CBC-MAC

- ▶ Soluția este ineficientă și greu de folosit în practică;

CBC-MAC

- ▶ Soluția este ineficientă și greu de folosit în practică;
- ▶ Însă, 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;
- ▶ Însă, 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;
- ▶ Însă, 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

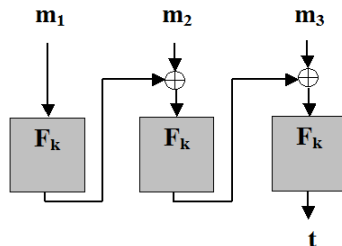
- ▶ O soluție mult mai eficientă este să folosim CBC-MAC;

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.



CBC-MAC

Definiție

Fie F o funcție pseudoaleatoare. Un CBC-MAC este format dintr-o pereche de algoritmi polinomiali probabiliști (Mac , Vrfy):

1. Mac : pentru o cheie $k \in \{0,1\}^n$ și un mesaj m de lungime l :

- ▶ Sparge m în $m = m_1, \dots, m_l$, $|m_i| = n$ și notează $t_0 = 0^n$;
- ▶ Pentru $i = 1, \dots, l$, calculează $t_i = F_k(t_{i-1} \oplus m_i)$;

Întoarce t_l ca tag-ul rezultat;

2. Vrfy : pentru o cheie $k \in \{0,1\}^n$, un mesaj m de lungime l , și un tag t de lungime n :

întoarce 1 dacă și numai dacă $t = \text{Mac}_k(m)$.

Rămâne valabilă condiția de corectitudine:

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

Securitatea CBC-MAC

Teoremă

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 $l \cdot n$.

Securitatea CBC-MAC

Teoremă

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 $l \cdot n$.

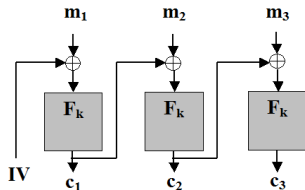
- Construcția prezentată este sigură numai pentru autentificarea mesajelor de lungime fixă;

Teoremă

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 $l \cdot n$.

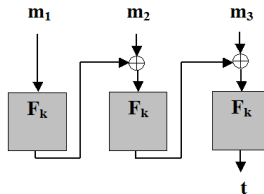
- ▶ Construcția prezentată este sigură numai pentru autentificarea mesajelor de lungime fixă;
- ▶ Avantajul acestei construcții față de cea anterioară este că ea poate autentifica mesaje de lungime mult mai mare;

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)

CBC-MAC pentru mesaje de lungime variabilă

Putem modifica construcția anterioară în diverse moduri ca să obținem o versiune de CBC-MAC pentru mesaje de lungime variabilă. Iată trei dintre ele care pot fi demonstrate ca fiind sigure:

1. Calculează $k_I = F_k(I)$; Apoi folosește CBC-MAC cu cheia k_I ; aceasta asigură faptul că sunt folosite chei diferite pentru a autentifica mesaje de lungimi diferite;
2. Se adaugă un bloc de mesaj (în fața primului bloc) care conține $|m|$ și se aplică CBC-MAC pe mesajul rezultat.
3. Se poate modifica schema așa încât să se aleagă două chei $k_1, k_2 \in \{0, 1\}^n$; se autentifica mesajul m cu CBC-MAC folosind cheia k_1 și se obține t iar tag-ul rezultat va fi $t' = F_{k_2}(t)$.

Important de reținut!

- ▶ MAC-urile oferă două proprietăți importante de securitate: integritatea mesajelor și autentificarea mesajelor;
- ▶ Pentru construcția lor se folosesc funcții pseudoaleatoare (în practică, sisteme bloc) fiind destul de rapide.