METODE CRIPTOGRAFICE DE PROTECȚIE A INFORMAȚIEI

Tema 6: Coduri de autentificare a mesajelor

OBIECTIVELE TEMEI

În această secțiune vom examina codurile de autentificare a mesajelor (MAC), care sunt utilizate între două entități ce partajează o cheie secretă cu scopul de a valida informația transmisă între ele. Obiectivele de bază sunt următoarele:

A prezenta conceptul de autentificare a sursei mesajului;

A defini conceptul funcțiilor MAC și a examina unele proprietăți ale acestora;

A analiza procedeele generale de construcție a funcțiilor MAC;

A examina algoritmi de construcție a MAC-urilor bazați pe utilizarea funcțiilor hash fără cheie;

A analiza algoritmul HMAC;

A examina algoritmi de construcție a MAC-urilor bazați pe utilizarea cifrurilor bloc;

A analiza algoritmii CBC-MAC și CMAC.

SUBIECTELE ABORDATE ÎN CADRUL TEMEI

1.1 Conceptul funcţiilor MAC

1.2 Procedee de construcţie a MAC-urilor

1.2.1 MAC-uri bazate pe funcţii hash fără cheie (MDC-uri)

1.2.1.1 Algoritmul HMAC

1.2.2 MAC-uri bazate pe cifruri bloc

1.2.2.1 Algoritmul CBC-MAC

1.2.2.2. Algoritmul CMAC

INTRODUCERE

În unele probleme aplicative, cum ar fi tranzacțiile financiare, pe lângă asigurarea confidențialității datelor este importantă și verificarea integrității și autenticității acestora.

Dacă informația despre tranzacțiile financiare se va scurge către public, aceasta poate să conducă la diminuarea numărului de clienți ai băncii, dar și mai dezastruoasă poate fi situația atunci când tranzacțiile frauduloase nu au fost detectate.

Este important ca tranzacțiile să fie efectuate de către o entitate autentificată și ca acestea să nu fie falsificate.

Două entități ce comunică într-un canal nesecurizat necesită o metodă prin care informația transmisă de o parte poate fi validată ca și autentică (sau nemodificată) de cealaltă.

INTEGRITATEA ȘI AUTENTIFICAREA

Integritatea datelor este o proprietate prin care se asigură că datele nu au fost modificate într-o manieră neautorizată din momentul creării acestora, pe parcursul transmiterii și stocării de către o sursă autorizată.

Autentificarea sursei datelor este un tip de autentificare prin care o entitate este confirmată ca și sursă a unor date specifice create la un moment de timp anterior.

În asigurarea autenticității sursei datelor se folosesc următoarele tehnici criptografice:

Codurile de autentificare a mesajelor MAC.

Schemele de semnătură digitală.

Integritatea datelor nu poate fi separată de autenticitatea sursei datelor. Datele care au fost alterate au o nouă sursă, iar dacă sursa nu poate fi determinată, atunci nici nu poate fi investigată integritatea datelor.

CODURI DE AUTENTIFICARE A MESAJELOR

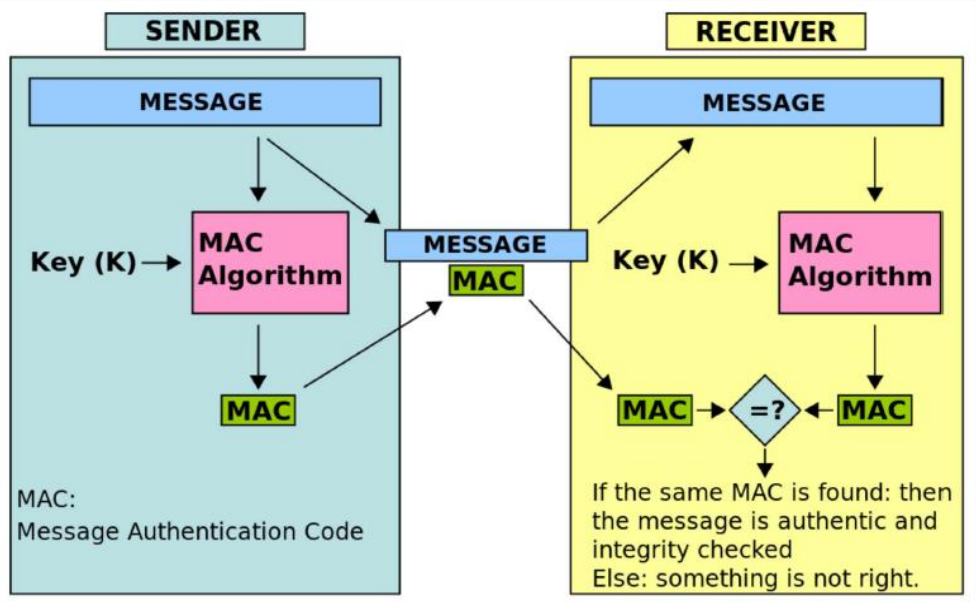
Funcțiile hash cu cheie, care au ca scop autentificarea mesajelor transmise între două entități ce partajează aceeași cheie secretă, sunt numite coduri de autentificare a mesajelor (MAC).

Valoarea MAC este un șir binar relativ scurt, construit în baza mesajului și a cheii secrete, și care este utilizat pentru autentificarea mesajului.

Pentru ca MAC-ul să fie sigur, trebuie să fie computațional dificil de calculat o valoare MAC pentru un mesaj dat fără a cunoaște cheia.

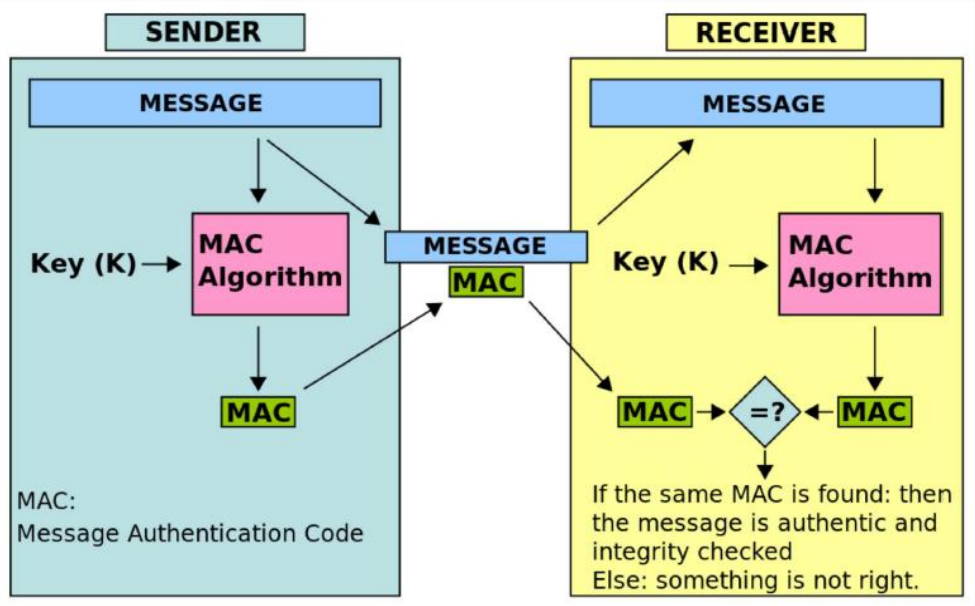
CODURI DE AUTENTIFICARE A MESAJELOR

Valoarea MAC este anexată la mesajul ce urmează a fi transmis. Recipientul mesajului determină identitatea sursei, în informația recepționată separă MAC-ul de mesaj, apoi în baza cheii secrete partajate calculează MAC-ul pentru mesajul recepționat și îl compară cu MAC-ul recepționat. În cazul în care acestea coincid, recipientul consideră că datele recepționate sunt autentice și integre.



CODURI DE AUTENTIFICARE A MESAJELOR

MAC-ul reprezintă un șir scurt $MAC_k(x)$, calculat în baza mesajului de autentificat x și a cheii secrete k. Expeditorul transmite perechea $(x, MAC_k(x))$, iar recipientul, care recepționează perechea (x', σ'), verifică faptul că $\sigma' = MAC_k(x')$.



CODURI DE AUTENTIFICARE A MESAJELOR

Un algoritm MAC constă din 3 algoritmi:

Algoritmul de generare a cheii, care selectează aleator o cheie din spațiul de chei.

Algoritmul de generare a valorii MAC în baza cheii secrete și a mesajului.

Algoritmul de verificare a valorii MAC ce verifică autenticitatea mesajului în baza cheii și a valorii MAC a acestuia, și întoarce „acceptat” dacă mesajul și valoarea MAC nu sunt falsificate și „respins” – în caz contrar.

CODURI DE AUTENTIFICARE A MESAJELOR

Codul de autentificare a mesajelor MAC este o familie de funcții h_k parametrizate de cheia secretă k și care posedă următoarele proprietăți:

1. *Eficiență de calcul* – fiind date mesajul x, valoarea k și funcția h_k, se poate calcula eficient valoarea MAC $h_k(x)$.
2. *Compresie* - h_k transformă șirul binar x de lungime arbitrară într-un șir binar $h_k(x)$ de lungime apriori dată.
3. *Rezistență de calcul* – fiind cunoscute perechile MAC $(x_i, h_k(x_i))$, este computațional dificil de calculat cheia corespunzătoare k sau de determinat perechea MAC $(x, h_k(x))$ pentru un mesaj $x \neq x_i$ (chiar dacă $h_k(x) = h_k(x_i)$ pentru un careva i).

CODURI DE AUTENTIFICARE A MESAJELOR

Pentru a avea posibilitate de a verifica codul MAC este necesar ca două entități ce au încredere reciprocă să convină în prealabil asupra cheii secrete.

Un dezavantaj al acestui mod de abordare este că cele două părți implicate trebuie să procedeze corect, ceea ce înseamnă că algoritmul MAC nu promovează mecanismul de non-repudiere. Ambele entități pot să modifice mesajul și MAC-ul, iar în cazul unei dispute, o a treia parte, considerată în calitate de judecător, nu va putea să soluționeze problema.

CODURI DE AUTENTIFICARE A MESAJELOR

Termenul de non-repudiere a originii reprezintă un serviciu prin care recipientul mesajului obține garanția autenticității mesajului, în sensul că recipientul poate să demonstreze unei terțe părți că mesajul este autentic chiar dacă expeditorul acestuia neagă acest fapt.

Atunci când entitățile ce comunică nu au încredere reciprocă, este necesară o abordare specifică ce promovează non-repudierea. În cazul unei dispute, autoritatea de încredere (judecătorul) trebuie să ia o decizie în favoarea uneia dintre părțile participante. O soluție tehnică elegantă pentru această problemă este oferită de conceptul de semnătură digitală, bazat pe funcțiile unidirecționale cu trapă. Schemele de semnătură digitală vor fi examinate la tema următoare.

CODURI DE AUTENTIFICARE A MESAJELOR

Obiective urmărite de către adversari în atacul valorilor MAC

În atacurile pe care le poate efectua un adversar asupra valorilor MAC obiectivul de bază va fi următorul:

Pentru una sau mai multe perechi MAC $\left(x_i, h_k\left(x_i\right)\right)$ cunoscute și cheia k necunoscută, se calculează o nouă pereche MAC $\left(x, h_k\left(x\right)\right)$ pentru un mesaj distinct $x \neq x_i$.

CODURI DE AUTENTIFICARE A MESAJELOR

Vom menționa următoarele scenarii de atac asupra codurilor MAC (expuse în ordinea creșterii avantajelor pentru adversar):

1. *Atac cu text cunoscut.* Adversarul cunoaște una sau mai multe perechi MAC $\left(x_i, h_k\left(x_i\right)\right)$.
2. *Atac cu text ales.* Pentru x_i alese de către adversar, sunt accesibile una sau mai multe perechi MAC $\left(x_i, h_k\left(x_i\right)\right)$.
3. *Atac cu text adaptiv ales.* Ca și în cazul precedent, mesajele x_i pot fi alese de către adversar, dar sunt posibile alegeri bazate pe precedentele rezultate.

CODURI DE AUTENTIFICARE A MESAJELOR

Tipul funcției hash	Obiectivul de proiectare	Obiectivul adversarului	Lungimea binară cerută pentru un nivel practic de securitate n - lungimea în biți a valorii hash (sau MAC)
cod de autentificare a mesajelor	non-recuperarea cheii; rezistență de calcul	să deducă cheia MAC; să genereze o nouă valoare MAC	$n \geq 64$ și cu o cheie MAC pe 64-80 biți

CODURI DE AUTENTIFICARE A MESAJELOR

În raport cu schemele MDC, există un număr mai redus de scheme MAC. Aceasta se explică prin aceea că schemele inițial propuse au fost acceptate pe larg.

În construcția algoritmilor MAC sunt cunoscute 3 abordări de bază:

- Utilizarea cifrurilor bloc, cea mai populară construcție fiind CBC-MAC și extensiile acesteia
- Utilizarea funcțiilor hash criptografice, cu cea mai populară schemă HMAC. Se consideră că schemele MAC construite în baza funcțiilor hash criptografice sunt eficiente din punctul de vedere al vitezei de calcul.
- Utilizarea funcțiilor hash universale, cele mai populare scheme fiind UMAC și VMAC. Algoritmul UMAC poate fi privit ca și o alternativă pentru HMAC, doar că UMAC este mai rapid, dar mai complex.

MAC-URI BAZATE PE FUNCȚII HASH FĂRĂ CHEIE (MDC-URI)

În practică a fost observat faptul că viteza de lucru a funcțiilor hash fără cheie (a MDC-urilor) este, de regulă, una mai mare decât viteza de lucru a cifrurilor bloc cu cheie simetrică. Aceasta a condus la ideea de a utiliza funcțiile hash fără cheie în construcția codurilor de autentificare a mesajelor MAC.

Vom examina coduri de autentificare MAC ce utilizează în construcția lor cheia secretă criptografică în combinație cu o funcție hash aprobată, cum ar fi, MD5, SHA-1, SHA-2, RIPEMD-160 ș.a. Algoritmul respectiv este numit HMAC și a fost definit în standardele mai multor organizații, precum ANSI, IETF, ISO, NIST.

Securitatea schemelor HMAC depinde direct de proprietățile funcției hash utilizate în construcția lor.

Algoritmii MAC ce utilizează o funcție hash nu necesită careva modificări în codul acesteia, deoarece posibilele modificări pot să conducă la diminuarea eficienței și a securității criptografice.

MAC-URI BAZATE PE FUNCȚII HASH FĂRĂ CHEIE (MDC-URI)

Printre obiectivele urmărite de schemele MAC bazate pe utilizarea funcţiilor hash sunt următoarele:

A utiliza fără modificări funcţiile hash fără cheie a căror cod este disponibil și care sunt eficiente la nivel de realizare software.

A realiza algoritmi MAC cu eficienţă comparabilă cu cea a funcţiei hash utilizate în construcţia lor.

A utiliza și a gestiona simplu cheile secrete.

A dispune de informație privind siguranța mecanismului de autentificare bazată pe cea a funcţiilor hash utilizate.

A permite înlocuirea simplă a funcţiei hash utilizate în construcţia algoritmului MAC în cazul în care este necesară o funcție hash mai rapidă sau mai sigură.

Chiar dacă pentru unele funcţii hash au fost depistate unele vulnerabilităţi, cum ar fi, de exemplu, pentru MD5, acestea nu se răsfrâng și asupra schemei HMAC ce utilizează MD5 (HMAC-MD5).

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

Algoritmul HMAC este utilizat de către expeditorul mesajului pentru a realiza o valoare MAC formată prin comprimarea cheii secrete și a mesajului transmis. Valoarea MAC este transmisă destinatarului împreună cu mesajul. Destinatarul, în baza aceleași chei secrete și a algoritmului HMAC, calculează valoarea MAC a mesajului recepționat și compară rezultatul cu valoarea MAC recepționată. Dacă valorile coincid, atunci este asigurată integritatea mesajului, iar destinatarul poate fi sigur că expeditorul mesajului este un membru al comunității de utilizatori ce partajează cheia secretă.

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

Definirea schemei HMAC necesită o funcție hash criptografică (fără cheie), pe care o vom nota prin H, și o cheie secretă k. Vom considera că funcția de compresie a funcției hash H iterează pe blocuri de lungime B octeți ale mesajului, iar valoarea hash rezultată este de L octeți. De exemplu, pentru MD5 avem $B = 64,\ L = 16$, iar pentru SHA-1 - $B = 64,\ L = 20$. Schema HMAC este definită astfel încât funcția hash H poate fi utilizată fără modificări în codul său.

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

Cheia de autentificare k se va alege aleator (periodic se va reînnoi) de lungime mai mică sau egală cu B. Însă aceasta se va extinde până la o cheie k_0 de lungime B octeți.

În cazul în care o aplicație folosește o cheie secretă k de lungime mai mare ca B octeți, se calculează valoarea hash a acesteia în baza funcției H, după care șirul de L octeți rezultat se va completa cu biți de zero până lungimea devine egală cu B octeți. Acest șir de lungime B octeți este considerat drept cheie k_0 pentru HMAC.

Dacă cheia k are lungime mai mică ca B, se concatenează biți de zero până se atinge lungimea B octeți. Lungimea minimă recomandată pentru cheia k este de L octeți (din considerente de securitate).

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

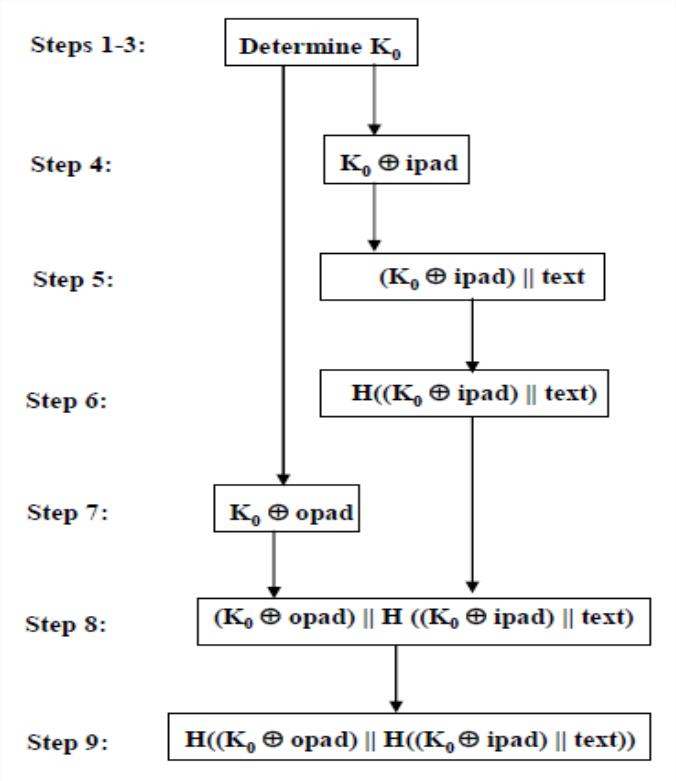
Algoritmul HMAC generează valoarea MAC a mesajului *text*, folosind funcția HMAC:

$$MAC(text) = HMAC(K, text) = H\left((k_0 \oplus opad) \,||\, H\left((k_0 \oplus ipad) \,||\, text\right)\right).$$

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

Algoritmul HMAC generează valoarea MAC a mesajului *text*, folosind funcția HMAC:

$$MAC(text) = HMAC(K, text) = H\left((k_0 \oplus opad) \| H\left((k_0 \oplus ipad) \| text\right)\right).$$



MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

Algoritmul HMAC

Date de intrare:

k - cheia secretă partajată între expeditorul mesajului și destinatar

H - specificarea funcției hash aprobate

B - lungimea cheii extinse k_0 (lungimea blocului de mesaj)

L - lungimea blocului de ieșire în octeți pentru funcția hash H

ipad - octetul $0x36$ repetat de B ori („i" de la „inner")

opad - octetul $0x5c$ repetat de B ori („o" de la „outer")

text - mesajul pentru care este generată valoarea MAC, de lungime n biți, unde $n \in \left[0, 2^B - 8B\right)$

Date de ieșire:

T - valoarea MAC de L octeți a mesajului *text*

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

Pașii algoritmului:

1. Dacă lungimea lui k, $length(k) = B$, atunci $k_0 := k$ și se trece la pasul 4.
2. Dacă $length(k) > B$, atunci se calculează valoarea hash pentru k, astfel obținându-se un șir de L octeți, după care, se concatenează la acesta $B - L$ biți de zero pentru a realiza un șir de B octeți k_0 (cheia modificată): $k_0 := H(k) \| 00...00$. Se trece la pasul 4.
3. Dacă $length(k) < B$, atunci se concatenează octeți nuli $0x00$ la sfârșitul șirului k pentru a realiza șirul k_0 de B octeți.
4. Se generează șirul de B octeți $k_0 \oplus ipad$.

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

5. Se concatenează șirul *text* la șirul rezultat la pasul 4: $(k_0 \oplus ipad) \| text$
6. Se aplică funcția hash la șirul generat la pasul 5: $H((k_0 \oplus ipad) \| text)$
7. Se generează șirul de B octeți $k_0 \oplus opad$.
8. Se concatenează șirurile obținute la pașii 6 și 7:
$$(k_0 \oplus opad) \| H((k_0 \oplus ipad) \| text).$$
9. Se aplică funcția hash H la șirul obținut la pasul 8:
$$T := H((k_0 \oplus opad) \| H((k_0 \oplus ipad) \| text)).$$

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

Primii 3 pași ai algoritmului au ca scop, pornind de la cheia secretă k, obținerea cheii extinse k_0 de lungime B octeți egală cu lungimea blocului de mesaj. Șirurile *ipad* și *opad* sunt definite astfel încât distanța Hamming între ele (numărul de poziții de biți în care biții corespunzători sunt diferiți) să fie suficient de mare. În Figura 1.5.4 este dată ilustrarea algoritmului HMAC.

Uneori valoarea MAC este trunchiată până la un anumit număr de biți. Când este necesară aplicarea operației de trunchiere, sunt utilizați t cei mai din stânga biți ai șirului de la ieșirea funcției HMAC. Se recomandă ca t să fie cel puțin $L/2$ octeți și, la fel, nu mai puțin de 80 biți. Prin trunchierea valorii MAC, mai puțină informație este disponibilă unui adversar, dar, simultan, mai puțini biți urmează a fi precizați.

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

În calitate de funcție hash în algoritmul HMAC poate fi utilizată orice funcție hash fără cheie. În acest caz, se vor genera noi valori pentru $ipad$ și $opad$ în baza lungimii de bloc asociate noii funcții hash H'.

Șirurile intermediare pe B octeți $k_0 \oplus ipad$ și $k_0 \oplus opad$ pot fi stocate și apoi utilizate de fiecare dată când este necesară autentificarea mesajului în baza aceleiași chei k. Ca și cheia secretă șirurile intermediare trebuie să fie protejate.

MAC-URI BAZATE PE MDC-URI. ALGORITMUL HMAC

Valori pentru testarea algoritmului HMAC

Denumirea algoritmului HMAC	Cheia	Mesajul	Valoarea MAC
HMAC-MD5	key	The quick brown fox jumps over the lazy dog	80070713463e7749b90c2dc24911e275
HMAC-SHA1	key	The quick brown fox jumps over the lazy dog	de7c9b85b8b78aa6bc8a7a36f70a90701c9 db4d9
HMAC-SHA256	key	The quick brown fox jumps over the lazy dog	f7bc83f430538424b13298e6aa6fb143ef4 d59a14946175997479dbc2d1a3cd8

MAC-URI BAZATE PE CIFRURI BLOC

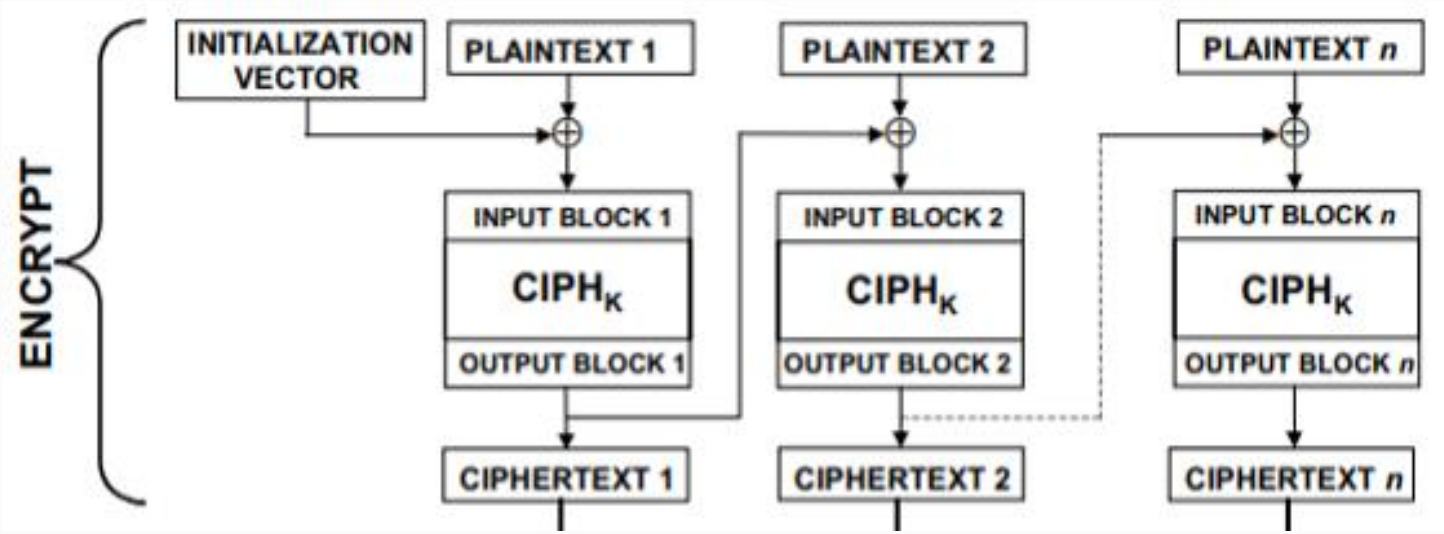
În această secțiune sunt examinați algoritmi pentru construcția valorilor MAC, bazați pe cifrurile bloc cu cheie simetrică. Procedeul general de calcul al valorii MAC în baza unui cifru bloc folosește modul de implementare CBC (Cipher-Block-Chaining).

Acest mod de implementare a cifrurilor bloc a fost propus în anul 1976 în lucrarea [17] și este cel mai frecvent utilizat. Fiecare bloc de text clar, înainte de a fi criptat, este combinat printr-un XOR cu blocul precedent de text criptat. Astfel, fiecare bloc de text criptat depinde de toate blocurile de text criptat procesate până la moment. Pentru a face unic fiecare mesaj criptat, se va utiliza o valoare de inițializare IV pe n biți la criptarea primului bloc al mesajului. Valoarea de inițializare nu e necesar să fie secretă, dar trebuie să fie distinctă pentru fiecare mesaj (uneori se alege aleator).

MAC-URI BAZATE PE CIFRURI BLOC

Textul clar este divizat pe blocuri (șiruri de biți de o anumită lungime), iar condițiile impuse asupra lungimii blocului variază în dependență de modul de operare utilizat. Unele moduri de operare a cifrurilor bloc (cum ar fi, ECB și CBC) folosesc blocuri complete (lungimea textului clar este un multiplu al lungimii blocului pentru sistemul de criptare utilizat) și necesită, eventual, ca ultimul bloc de date să fie completat (prin procedura de padding) până la un bloc complet. Există mai multe scheme de padding, cea mai simplă fiind cea în care șirul de biți ce reprezintă textul clar este completat cu biți de 0, până atunci când lungimea șirului devine un multiplu al lungimii blocului. O altă variantă de padding completează șirul inițial cu un bit de 1, după care urmează biți de 0.

MAC-URI BAZATE PE CIFRURI BLOC



ALGORITMUL CBC-MAC

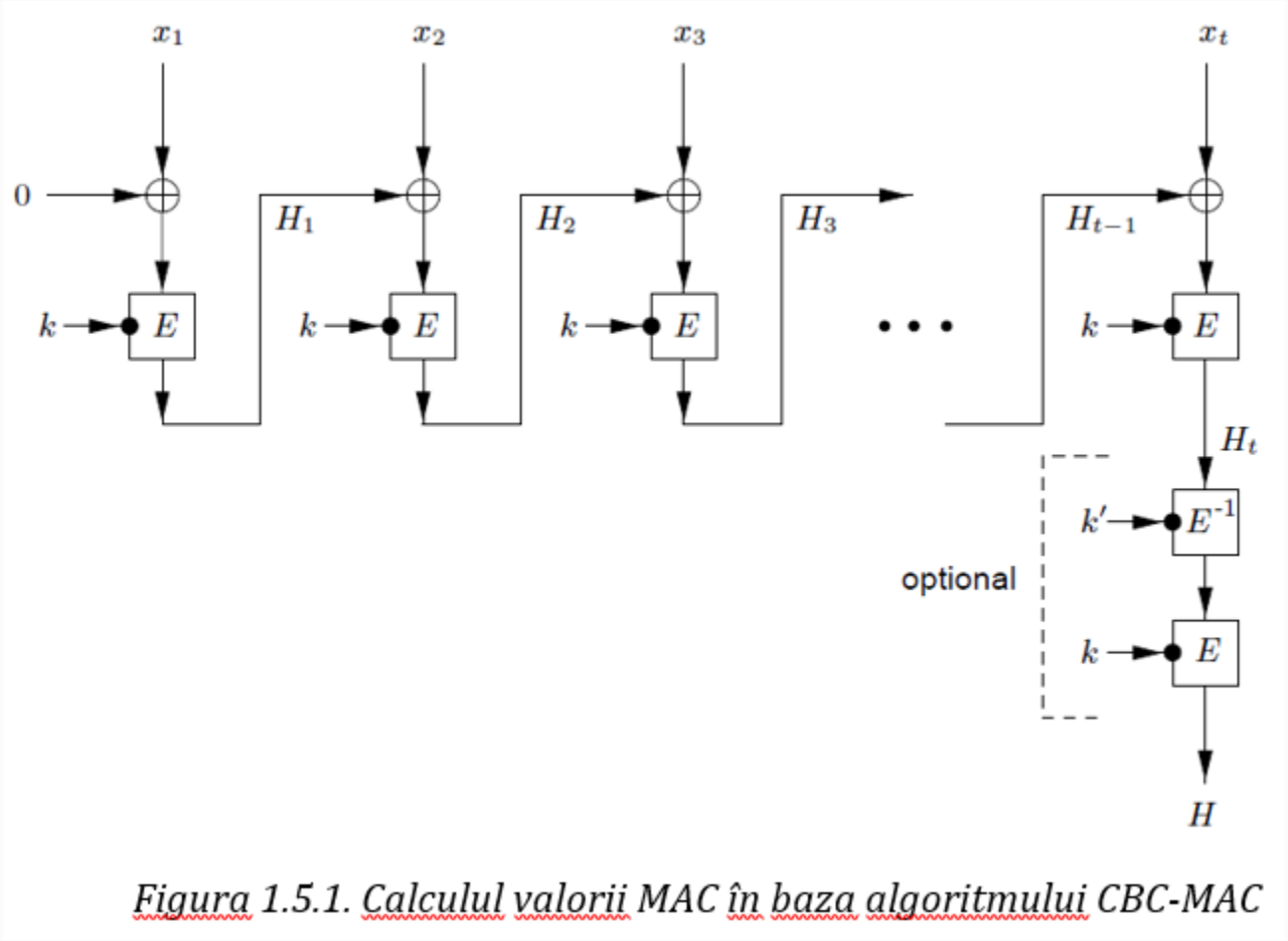


Figura 1.5.1. Calculul valorii MAC în baza algoritmului CBC-MAC

ALGORITMUL CBC-MAC

Pentru a calcula valoarea CBC-MAC a mesajului x, acesta este criptat în modul CBC cu vectorul de inițializare nul. Mesajul x este divizat în t blocuri $x = x_1 \| x_2 \| \ldots \| x_t$, fiecare bloc având lungimea n biți, și sunt precizate cheia k și cifrul bloc E. Valoarea CBC-MAC a mesajului x, $CBC_k(x)$, se determină ca și valoarea H_t generată iterativ în baza relației $H_i = E_k(x_i \oplus H_{i-1})$, $i = \overline{1,t}$, $H_0 = \{0\}^n$.

Algoritmul CBC-MAC

Date de intrare:
- x - mesajul pentru care este generată valoarea MAC
- E - specificarea cifrului bloc utilizat
- k - cheia MAC secretă pentru E
- n - lungimea blocului prelucrat de către cifrul E

Date de ieșire:
- H_t - valoarea MAC pe n biți a mesajului x

ALGORITMUL CBC-MAC

Pașii algoritmului:

1. *Procedura de padding și divizarea pe blocuri.* Se aplică procedura de padding a mesajului x: la reprezentarea binară a mesajului se concatenează bitul 1, urmat de un anumit număr de biți 0, până lungimea mesajului devine un multiplu al lui n. În continuare, șirul completat este divizat pe blocuri de n biți, notate prin $x_1,\ldots,x_t$.
2. *Procesarea CBC.* Fie E_k algoritmul de criptare cu cifrul bloc E și cheia k. Se determină blocul H_t în modul următor: $H_1 := E_k(x_1), H_i := E_k(H_{i-1} \oplus x_i), i = \overline{2,t}$.
3. *Transformarea de ieșire (opțională) pentru amplificarea securității MAC.* Folosind o a doua cheie secretă $k' \neq k$, se calculează: $H'_t := E_{k'}^{-1}(H_t), H_t := E_k(H'_t)$.
4. *Completarea.* Valoarea MAC este blocul pe n biți H_t.

ALGORITMUL CBC-MAC

Pe când în criptarea CBC se consideră o valoare inițială IV aleatoare, care servește pentru a preveni atacul codebook asupra primului bloc din mesajul criptat, în algoritmul CBC-MAC valoarea inițială este una nulă $IV = \{0\}^n$. Algoritmul nu este sigur în cazul în care IV este selectată aleator.

Atunci când cifrul bloc generează un text criptat pseudoaleator, algoritmul CBC-MAC este sigur pentru mesaje de lungime fixă de $t \cdot n$ biți, unde n - lungimea blocului cifrului E. Însă, este necesar de întreprins măsuri suplimentare în cazul în care sistemul admite mesaje de lungime variabilă.

ALGORITMUL CBC-MAC

Pentru mesaje de lungime variabilă au fost propuse câteva variante modificate ale algoritmului CBC-MAC. În cazul în care se știe lungimea mesajului, o soluție ar fi includerea reprezentării binare pe n biți a acesteia în primul bloc, dar cea mai simplă variantă (care nu necesită cunoașterea lungimii mesajului) este schema ECBC-MAC (Encrypt-last-block CBC-MAC), care constă în criptarea valorii obținute cu algoritmul CBC-MAC în baza cifrului bloc E și a unei noi chei k_2, $ECBC_{k_1,k_2}(x) = E_{k_2}\left(CBC_{k_1}(x)\right)$, x - mesajul, k_1 - cheia CBC-MAC, $CBC_{k_1}(x)$ - valoarea CBC-MAC a mesajului x. Se știe că schema ECBC-MAC este sigură dacă lungimea mesajului este divizibilă prin n. Dezavantajul schemei ține de utilizarea a două chei pentru cifrul bloc E.

ALGORITMUL CMAC

Există algoritmul One-Key CBC-MAC (OMAC), care este sigur pentru mesaje de orice lungime.

Oficial există două versiuni ale algoritmului OMAC (OMAC1 și OMAC2), care diferă nesemnificativ.

De fapt, OMAC1 este echivalent cu algoritmul CMAC (Cipher-based MAC), care în anul 2005 a fost recomandat pentru utilizare de către NIST (US National Institute for Standards and Technology).

CMAC este bazat pe utilizarea unui cifru bloc standardizat (de exemplu, AES).

ALGORITMUL CMAC

Cheia secretă k a cifrului bloc este utilizată pentru a determina două valori adiționale secrete, de lungime în biți egală cu cea a blocului, numite subchei și notate prin k_1 și, respectiv, k_2. Subcheile pot fi calculate în prealabil și pot fi stocate împreună cu cheia, pentru utilizare multiplă, sau pot fi recalculate la fiecare apel. Fiecare valoare intermediară utilizată în calculul subcheilor urmează să rămână secretă.

Pe parcursul generării subcheii este folosit șirul de biți R_b, care este determinat de numărul b de biți ai blocului de mesaj. În particular, pentru cazul în care este folosit cifrul bloc AES, avem $R_{128} = \left(0^{120}10000111\right)_2$, iar în cazul cifrului bloc TDEA - $R_{64} = \left(0^{59}11011\right)_2$ ($\{0\}^s$ - șir de biți format din s zerouri).

ALGORITMUL CMAC

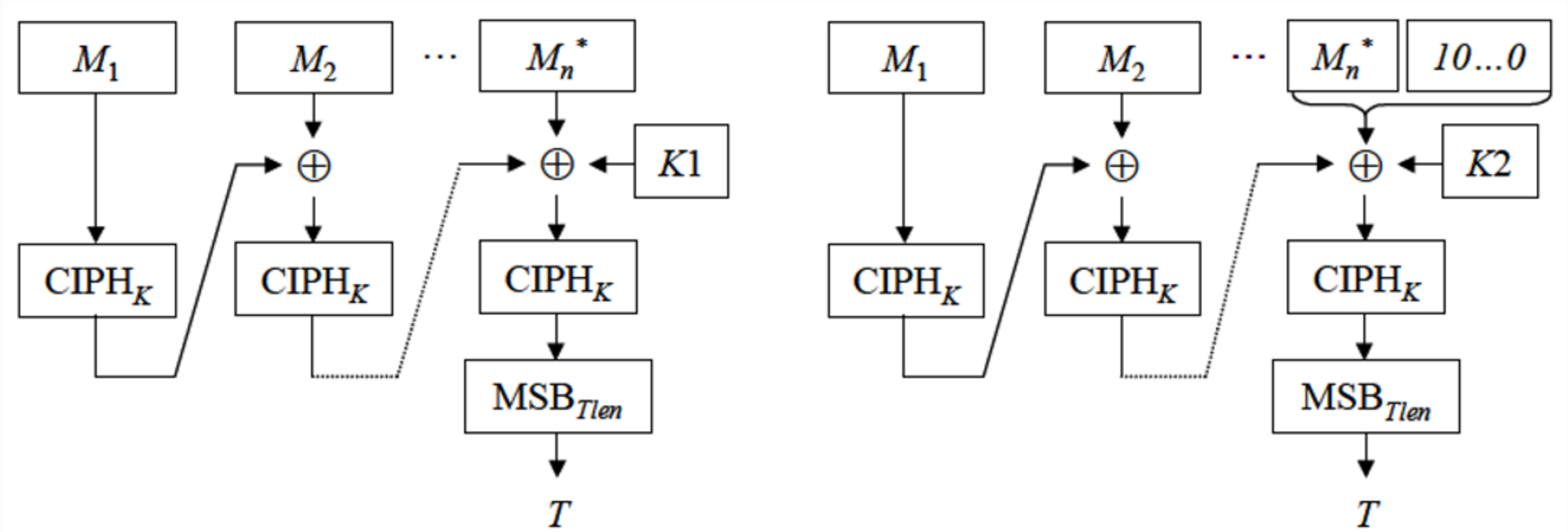


Figura 1.5.3. Ilustrarea celor două cazuri ale algoritmului CMAC

ALGORITMUL CMAC

Algoritmul CMAC

Date de intrare:

M - mesajul pentru care este generată valoarea MAC

$lenM$ - lungimea în biți a mesajului M

E - specificarea cifrului bloc utilizat

k - cheia MAC secretă pentru E

b - lungimea blocului lui E

$lenT$ - lungimea în biți a valorii MAC

Date de ieșire:

T - valoarea MAC pe $lenT$ biți a mesajului M (se va nota $CMAC(k, M, lenT)$)

ALGORITMUL CMAC

Pașii algoritmului:

1. *Se aplică la cheia k procedura de generare a subcheilor* k_1, k_2.

   1.1. Se determină șirul $L = E_k\left(\{0\}^b\right)$.

   1.2. Dacă $msb_1(L) = 0$, atunci $k_1 := rol(L,1)$

   altfel $k_1 := rol(L,1) \oplus R_b$ ($R_b = R_{128} = \left(0^{120}10000111\right)_2$ sau în hexazecimal $R_b = 0x0...087$ $(n = 128)$ dacă se aplică AES).

   1.3. Dacă $msb_1(k_1) = 0$, atunci $k_2 := rol(k_1,1)$

   altfel $k_2 := rol(k_1,1) \oplus R_b$.

   1.4. Se transmit subcheile k_1, k_2 către procedura de generare a valorii MAC.

ALGORITMUL CMAC

2. *Generarea valorii MAC.*

2.1. Dacă $lenM = 0$, atunci $n := 1$
altfel $n := ceil(lenM / b)$.

2.2. Fie $M_1, M_2, ..., M_{n-1}, M_n^*$ o secvență de șiruri de biți ce satisfac $M = M_1 \| M_2 \| ... \| M_{n-1} \| M_n^*$. Blocurile $M_1, M_2, ..., M_{n-1}$ sunt complete (dacă $lenM \le b$, atunci $M = M_1^*$).

2.3. Dacă M_n^* este un bloc complet, atunci $M_n := k_1 \oplus M_n^*$,
altfel $M_n := k_2 \oplus \left(M_n^* \| 1\, 0^j\right)$, $j = nb - lenM - 1$.

2.4. Fie $C_0 := 0^b$.

2.5. Pentru $i = \overline{1,n}$ se determină $C_i := E_k\left(C_{i-1} \oplus M_i\right)$.

2.6. Se returnează $T := msb_{lenT}\left(C_n\right)$.

ALGORITMUL CMAC

Pentru a verifica perechea MAC (M, T') destinatarul mesajului, mai întâi, generează valoarea MAC T pentru mesajul recepționat M, după care o compară cu valoarea recepționată T'.

Verificarea valorii MAC generate cu algoritmul CMAC (vom nota $Ver(k, M, T')$)

Date de intrare:
- M - mesajul pentru care este generată valoarea MAC
- $lenM$ - lungimea în biți a mesajului M
- E - specificarea cifrului bloc utilizat
- k - cheia MAC secretă pentru E
- b - lungimea blocului lui E
- $lenT$ - lungimea în biți a valorii MAC
- k_1, k_2 - subcheile
- T' - valoarea MAC recepționată

Date de ieșire:
- *Result* - variabilă ce conține rezultatul verificării: *Valid* sau *Invalid*

ALGORITMUL CMAC

Pașii algoritmului:

1. Se aplică algoritmul MAC $CMAC(k, M, lenT)$ și se obține valoarea MAC T.
2. Dacă $T = T'$, atunci se returnează în *Result* valoarea *Valid*; altfel, se returnează *Invalid*.

În cazul în care algoritmul CMAC folosește AES în calitate de cifru bloc, algoritmul de construcție a valorii MAC corespunzător este numit AES-CMAC [15].

ALGORITMUL CMAC

Exemple pentru testarea algoritmului CMAC

AES-128 ($lenT = 128$ ***, 32 de simboluri în hexazecimal)***	
k (cheie pe 128 biți)	2b7e1516 28aed2a6 abf71588 09cf4f3c
Generarea subcheilor	
$E_k\left(0^{128}\right)$	7df76b0c 1ab899b3 3e42f047 b91b546f
k_1	fbeed618 35713366 7c85e08f 7236a8de
k_2	f7ddac30 6ae266cc f90bc11e e46d513b
Exemplul 1 ($lenM = 0$)	
M	mulțimea vidă
T	bb1d6929 e9593728 7fa37d12 9b756746
Exemplul 2 ($lenM = 128$)	
M	6bc1bee2 2e409f96 e93d7e11 7393172a
T	070a16b4 6b4d4144 f79bdd9d d04a287c

ALGORITMUL CMAC

Exemplul 3 ($lenM = 320$)	
M	6bc1bee2 2e409f96 e93d7e11 7393172a ae2d8a57 1e03ac9c 9eb76fac 45af8e51 30c81c46 a35ce411
T	dfa66747 de9ae630 30ca3261 1497c827
AES-192 ($lenT = 128$, 32 de simboluri în hexazecimal)	
k (cheie pe 192 biți)	8e73b0f7 da0e6452 c810f32b 809079e5 62f8ead2 522c6b7b
Generarea subcheilor	
$E_k(0^{128})$	22452d8e 49a8a593 9f7321ce ea6d514b
k_1	448a5b1c 93514b27 3ee6439d d4daa296
k_2	8914b639 26a2964e 7dcc873b a9b5452c

ALGORITMUL CMAC

Exemplul 1 ($lenM = 0$)	
M	mulțimea vidă
T	d17ddf46 adaacde5 31cac483 de7a9367
Exemplul 2 ($lenM = 128$)	
M	6bc1bee2 2e409f96 e93d7e11 7393172a
T	9e99a7bf 31e71090 0662f65e 617c5184
Exemplul 3 ($lenM = 320$)	
M	6bc1bee2 2e409f96 e93d7e11 7393172a ae2d8a57 1e03ac9c 9eb76fac 45af8e51 30c81c46 a35ce411
T	8a1de5be 2eb31aad 089a82e6 ee908b0e

ALGORITMUL CMAC

AES-256 ($lenT = 128$ ***, 32 de simboluri în hexazecimal)***	
k (cheie pe 256 biți)	603deb10 15ca71be 2b73aef0 857d7781 1f352c07 3b6108d7 2d9810a3 0914dff4
Generarea subcheilor	
$E_k\left(0^{128}\right)$	e568f681 94cf76d6 174d4cc0 4310a854
k_1	cad1ed03 299eedac 2e9a9980 8621502f
k_2	95a3da06 533ddb58 5d353301 0c42a0d9
Exemplul 1 ($lenM = 0$)	
M	mulțimea vidă
T	028962f6 1b7bf89e fc6b551f 4667d983

ALGORITMUL CMAC

Exemplul 2 ($lenM = 128$)	
M	6bc1bee2 2e409f96 e93d7e11 7393172a
T	28a7023f 452e8f82 bd4bf28d 8c37c35c
Exemplul 3 ($lenM = 320$)	
M	6bc1bee2 2e409f96 e93d7e11 7393172a ae2d8a57 1e03ac9c 9eb76fac 45af8e51 30c81c46 a35ce411
T	aaf3d8f1 de5640c2 32f5b169 b9c911e6

BIBLIOGRAFIE RECOMANDATĂ

1. A. Menezes, P. van Oorschot and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.
2. J. Pieprzyk, T. Hardjono, J. Seberry, Fundamentals of Computer Security, Springer, 2003.
3. D. Stinson, M. Paterson, Cryptography: Theory and Practice, 4th ed., Boca Raton, FL: CRC Press, 2019.
4. Federal Information Processing Standards Publication 198-1, "The Keyed-Hash Message Authentication Code (HMAC)," National Institute of Standards and Technology, 2008.
5. J. Black and P. Rogaway, "CBC MACs for arbitrary-length messages: The three-key constructions," in *Advances in Cryptology—Crypto 2000, Lecture Notes in Computer Science*, 2000.
6. M. Dworkin, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication," NIST Special Publication 800-38B, 2005.