Acta et Commentationes, Sciences of Education, nr. 3(25), 2021 ISSN 1857-0623
p. 7-23 E-ISSN 2587-3636

CZU: 37.016:51 DOI: 10.36120/2587-3636.v25i3.7-23

# ABORDĂRI METODICE ÎN STUDIEREA SISTEMULUI CRIPTOGRAFIC ASIMETRIC MERKLE–HELLMAN

**Liubomir CHIRIAC**, dr. hab., prof. univ.
https://orcid.org/0000-0002-5786-5828
**Aureliu DANILOV**, doctorand
https://orcid.org/0000-0003-0859-7043
Universitatea de Stat Tiraspol

**Rezumat**. În acest articol, în baza sistemului criptografic Merkle–Hellman, este examinat procesul de criptare/decriptare și conexiunea cu conceptele și noțiunile de bază din algebra abstractă. Totodată se evidențiează interconexiunea dintre algebra abstractă – critptografiere - algoritmică – programare. Din punct de vedere metodologic sunt examinate detaliat etapele de criptare și decriptare utilizare la aplicarea sistemului criptografic Merkle–Hellman.

**Cuvinte cheie**: sistem criptografic, număr prim, algoritmul Euclid extins, numere binare, problema rucsac, șir cu creștere mare, algoritmul Merkle–Hellman.

# METHODICAL APPROACHES IN STUDYING THE MERKLE – HELLMAN ASYMMETRIC CRYPTOGRAPHIC SYSTEM

**Abstract**. This article, based on the Merkle – Hellman cryptographic system, examines the encryption /decryption process and the connection with the concepts and basics of abstract algebra. At the same time, the interconnection between abstract algebra - cryptography - algorithmic - programming is highlighted. From a methodological point of view, the stages of encryption and decryption used in the application of the Merkle – Hellman cryptographic system are examined in detail.

**Keywords**: cryptographic system, prime number, extended Euclid algorithm, binary numbers, knapsacks problem, high rise set, Merkle – Hellman algorithm.

## 1. Sisteme criptografice asimetrice. Noțiuni de bază

O bună înțelegere a conceptelor din domeniile informaticii și criptografie, presupune o bună înțelegere a noțiunilor și afirmațiilor din matematică care se utilizează pe larg la soluționarea problemelor din informatică. Interconexiunea dintre informatică și matematică este vitală pentru dezvoltarea noilor direcții a informaticii. Astfel, dezvoltarea conceptului de criptografie cu chei publice se bazează pe noțiuni fundamentale din matematică, inclusiv algebra abstractă.

Conceptul de criptografie cu chei publice a fost elaborat și promovat de Whitfield Diffie şi Martin Hellman. Ideea revoluționară lansată de cercetătorii respectivi constă în propunerea de a utiliza un nou criptosistem în care cheile de criptare şi decriptare sunt diferite, iar cheia de decriptare (care este secretă) nu poate fi dedusă din cheia de criptare (care este publică).

În continuare vom examina un algoritm similar - sistemul criptografic asimetric Merkle–Hellman, luând în considerare noțiunile de mai jos.

*Criptografia* este o ramură a matematicii care se ocupă cu securizarea informației precum și cu autentificarea și restricționarea accesului într-un sistem informatic [1, 2, 4, 6, 9].

*Criptarea* este transformarea reversibilă a informațiilor pentru a le ascunde de persoanele neautorizate, oferind, în același timp, acces la acestea pentru utilizatorii autorizați [1, 3].

*Sistem criptografic* este un set de algoritmi criptografici necesari pentru a implementa un anumit serviciu de securitate, cel mai frecvent pentru obținerea confidențialității (criptare) [5, 9].

Scopul principal al criptării este de a menține confidențialitatea informațiilor transmise. O proprietate specială a algoritmelor de criptare este folosirea așa numitor chei de criptare, cu ajutorul cărora sunt selectați parametri specifici pentru transformarea/criptarea textului și restabilirea/decriptarea lui.

*Cheie* - informații secrete folosite de algoritmul criptografic la criptarea/decriptarea textelor, setarea și verificarea semnăturilor digitale, calcularea codurilor de autentificare [4, 3, 5, 6, 9].

În criptografie se folosesc *chei simetrice* și *chei asimetrice*. Proprietatea principală a cheilor simetrice. La efectuarea transformărilor criptografice directe cât și inverse (criptare/decriptare, calculul/verificarea codurilor de autentificare) trebuie să se folosească una și aceeași cheie.

Cheile simetrice asigură o confidențialitate mai mare a textelor secrete, însă creează dificultăți la răspândirea (difuzarea) cheii, atunci când sunt implicați un număr mare de utilizatori în procesul de criptare/decriptare.

Sistemele criptografice care folosesc cheile simetrice se numesc *sistemele criptografice simetrice* [3, 4, 6].

Proprietatea principală a cheilor asimetrice. Cheile asimetrice sunt o pereche de chei cu următoarele caracteristici:

- *Cheia privată* este o cheie cunoscută doar de creatorul cheii. Numai utilizatorul care păstrează secretul cheii sale private garantează imposibilitatea falsificării unui document criptat sau a unei semnături digitale.
- *Cheia publică* este o cheie care poate fi publicată și folosită pentru a verifica autenticitatea documentului semnat. Cheia publică este calculată ca valoarea unei funcții a cheii private. Important este să știm că cunoașterea cheii publice nu face posibilă determinarea cheii private.

Sistemele criptografice care folosesc cheile asimetrice se numesc *sistemele criptografice asimetrice* [3, 4, 6].

*Cifru* este un sistem de transformări reversibile care depind de un parametru secret (cheie) și este conceput pentru a asigura secretul informațiilor transmise [3, 4, 5]. În

continuare vom reaminti câteva noțiuni algebrice pe care le vom utiliza ulterior. Sistemul de *numere binare* este un sistem de numere cu baza 2 [7]. *Bit* este unitate de măsură a cantității de informații.

Un bit de informații - un semnal care poate lua două valori: pornit sau oprit, da sau nu, ridicat sau scăzut, încărcat sau neîncărcat; în sistemul de numerație binar, este 1 (unul) sau 0 (zero).

Prin *secvență/șir* vom înțelege un set de obiecte numerotat, printre care sunt permise repetări, unde ordinea obiectelor contează. Numerotarea se realizează cel mai des cu ajutorul numerelor naturale. Dacă pentru numerele întregi **a** și **b** există un număr întreg **q** astfel încât **bq** = **a**, atunci numărul **a** este divizibil cu **b**. Numărul **b** se numește *divizorul* numărului **a**[8].

Se numește *număr prim*, un număr natural (întreg pozitiv) care are exact doi divizori naturali diferiți - unu și el însuși [1].

Un exemplu de aplicare a numerelor prime în criptografie este funcția IsPrim(n) , scrisă în limbajul C++, care testează dacă numărul **n** este sau nu prim, descrisă în punctul 9. Soluție pentru algoritmul de criptare Merkle–Hellman, în continuare (9-SMH).

Două numere întregi **a** și **b** sunt *relativ prime* sau *reciproc prime*, dacă singurul număr întreg pozitiv care le împarte pe amândouă este 1. Acest lucru este echivalent cu: cmmdc(**a**, **b**) = 1 [8].

Funcția **g**: **Y** → **X** se numește *inversă de stânga* pentru funcția **f**: **X** → **Y**, dacă: **g**(**f**(**x**)) = **x**, ∀ **x** ∈ **X**.

Funcția **g** : **Y** → **X** se numește *inversă de dreapta* pentru funcția **f**: **X** → **Y**, dacă: **f**(**g**(**y**)) = **y**, ∀ **y** ∈ **Y**.

Funcția inversă este o funcție care inversează dependența exprimată de o funcție dată. De exemplu, dacă o funcție de **x** returnează **y**, atunci funcția sa inversa pentru valoarea **y** returnează **x**. Funcția inversă funcției **f** de obicei se notează $\mathbf{f}^{-1}$ sau $\mathbf{f}^{inv}$.

Funcția **g**: **Y** → **X** se numește *inversa funcției* **f**: **X** → **Y**, dacă se îndeplinesc următoarele relații: **f**(**g**(**y**)) = **y**, ∀ **y** ∈ **Y**; **g**(**f**(**x**)) = **x**, ∀ **x** ∈ **X** [8].

*Inversul multiplicativ modulo* **m** al unui număr întreg **a** este un număr întreg **x** astfel încât **a** · **x** este congruent cu 1 în raport cu modulo **m** [1, 4].

În matematică această congruență este scrisă ca: **a** · **x** = 1 (mod **m**), unde **a** - număr întreg, **x** - inversul multiplicativ modulo **m**, 1 - restul după împărțirea **a** · **x** cu numărul întreg **m**.

## 2. Algoritmul lui Euclid extins

### 2.1. Algoritmul lui Euclid. CMMDC

Cel mai mare divizor comun (CMMDC) a două numere este cel mai mare număr care le divide pe ambele. Algoritmul lui Euclid folosește proprietatea, că cel mai mare

divizor comun al două numere nu se modifică dacă numărul cel mai mic este scăzut din cel mai mare.

Fie **a** și **b** sunt numere întregi diferite de zero concomitent și succesiunea de numere **a** > **b** > $\mathbf{r}_1$ > $\mathbf{r}_2$ > $\mathbf{r}_3$ > … > $\mathbf{r}_n$, unde $\mathbf{r}_i$ - este restul împărțirii după cum urmează:

**a** = **b** · $\mathbf{q}_0$ + $\mathbf{r}_1$; **b** = $\mathbf{r}_1$ · $\mathbf{q}_1$ + $\mathbf{r}_2$; $\mathbf{r}_1$ = $\mathbf{r}_2$ · $\mathbf{q}_2$ + $\mathbf{r}_3$; ...; $\mathbf{r}_{n-2}$ = $\mathbf{r}_{n-1}$ · $\mathbf{q}_{n-1}$ + $\mathbf{r}_n$, $\mathbf{r}_{n-1}$ = $\mathbf{r}_n$ · $\mathbf{q}_n$.

Atunci cmmdc(**a**, **b**), cel mai mare divizor comun al lui **a** și **b**, este egal cu $\mathbf{r}_n$, ultimul rest nenul al acestei secvențe.

## 2.2. Algoritmul lui Euclid extins

Formulele pentru $\mathbf{r}_i$ pot fi scrise după cum urmează:

$\mathbf{r}_1$ = **a** – **b** · $\mathbf{q}_0$ = **a** + **b** · (- $\mathbf{q}_0$),

$\mathbf{r}_2$ = **b** - $\mathbf{r}_1$ · $\mathbf{q}_1$ = **a** · (-$\mathbf{q}_1$) + **b** · (1 + $\mathbf{q}_1$ · $\mathbf{q}_0$),

...

cmmdc(**a**, **b**) = $\mathbf{r}_n$ =**a** · **s** + **b** · **t= d**, unde **s**, **t** sunt numere întregi.

Această reprezentare a celui mai mare divizor comun se numește identitatea Bezout, iar numerele **s** și **t** se numesc coeficienții Bezout.

**EXEMPLU 1.** *Aplicarea algoritmului Euclid extins*. Fie date două numere prime **a**=7 și **b**=11. Este necesar de  a găsi inversul 7 (mod 11).

*Soluție.* Pentru a găsi inversul 7 (mod 11), trebuie să rezolvăm echivalența 7 x ≡ 1 (mod 11).

Folosim *algoritmul euclidian extins.*

**P1.** Realizăm descompunerile liniare până la restul 0:

$\mathbf{a}_1$=**a**=7, $\mathbf{b}_1$=**b**=11;

$\mathbf{a}_2$=$\mathbf{b}_1$=11, $\mathbf{b}_2$=$\mathbf{a}_1$ mod $\mathbf{b}_1$=7 mod 11=7;

$\mathbf{a}_3$=$\mathbf{b}_2$=7, $\mathbf{b}_3$=$\mathbf{a}_2$ mod $\mathbf{b}_2$=11 mod 7=4;

$\mathbf{a}_4$=$\mathbf{b}_3$=4, $\mathbf{b}_4$=$\mathbf{a}_3$ mod $\mathbf{b}_3$=7 mod 4=3;

$\mathbf{a}_5$=$\mathbf{b}_4$=3, $\mathbf{b}_5$=$\mathbf{a}_4$ mod $\mathbf{b}_4$=4 mod 3=1;

$\mathbf{a}_6$=$\mathbf{b}_5$=1, $\mathbf{b}_6$=$\mathbf{a}_5$ mod $\mathbf{b}_5$=3 mod 1=0;

**P2.** Efectuăm calculele inverse (înapoi).

$\mathbf{s}_6$= 1, $\mathbf{t}_6$= 0;

$\mathbf{s}_5$= $\mathbf{t}_6$=0, $\mathbf{t}_5$= $\mathbf{s}_6$ - int($\mathbf{a}_5$/$\mathbf{b}_5$) · $\mathbf{t}_6$=1- int(3/1) · 0=1;

$\mathbf{s}_4$= $\mathbf{t}_5$=1, $\mathbf{t}_4$= $\mathbf{s}_5$ - int($\mathbf{a}_4$/$\mathbf{b}_4$) · $\mathbf{t}_5$=0- int(4/3) · 1=-1;

$\mathbf{s}_3$= $\mathbf{t}_4$=-1, $\mathbf{t}_3$= $\mathbf{s}_4$ - int($\mathbf{a}_3$/$\mathbf{b}_3$) · $\mathbf{t}_4$=1- int(7/4) · (-1)=2;

$\mathbf{s}_2$= $\mathbf{t}_3$=2, $\mathbf{t}_2$= $\mathbf{s}_3$ - int($\mathbf{a}_2$/$\mathbf{b}_2$) · $\mathbf{t}_3$=-1- int(11/7) · 2=-3;

$\mathbf{s}_1$= $\mathbf{t}_2$=-3, $\mathbf{t}_1$= $\mathbf{s}_2$ - int($\mathbf{a}_1$/$\mathbf{b}_1$) · $\mathbf{t}_2$=2- int(7/11) · (-3)=2;

**P3.** Conform relației **cmmdc**(**a**, **b**) = **a** · **s** + **b** · **t** = **d** obținem:

**cmmdc**(7, 11) = 7 · (-3) + 11 · 2=1

este o descompunere liniară formată de 7 și 11, și

7 · (-3) = 1 – 11 · 2 => 7 · (-3) ≡ 1 (mod 11),

prin urmare -3 este inversul lui 7 (mod 11). Pentru a exprima inversul ca un număr pozitiv efectuăm (-3+11) (mod 11) = 8. Prin urmare inversul 7 (mod 11) este 8, sau $7^{-1} \equiv 1 \pmod{11}$.

Un exemplu de aplicare a algoritmului lui Euclid extins în criptografie este funcția cmmdcEuclidExt(a, b, s, t), scrisă în limbajul C++ (9-SMH), care întoarce cmmdc a două numere **a**, **b** și coeficienții **s**, **t**, aducem aminte că pe baza coeficientului **s** se calculează inversul multiplicativ al numărului **a** (mod **b**).

## 3. Notația binară a textului

În numărarea binară se folosește aceeași procedură ca în numărarea zecimală, cu excepția faptului că sunt disponibile doar două simboluri 0 și 1. De exemplu:

$$(0)_2 + (1)_2 = (1)_2; \quad (1)_2 + (1)_2 = (10)_2; \quad (10)_2 + (1)_2 = (11)_2$$

### 3.1. Conversia numerelor zecimale în binare

Fie dat un număr zecimal **a**. Pentru transferul numărului **a** în sistemul de numerație binar putem folosi algoritmul [7]:

1. $\mathbf{a} / 2 = \mathbf{q}_0$ cu restul $\mathbf{r}_0$;
2. $\mathbf{q}_0 / 2 = \mathbf{q}_1$ cu restul $\mathbf{r}_1$;

...

n. $\mathbf{q}_n / 2 = 0$ cu restul $\mathbf{r}_{n-1}$.

Ca rezultat, obținem numărul $(\mathbf{a})_{10}$ în notație binară este $(\mathbf{r}_{n-1} \ldots \mathbf{r}_1 \ \mathbf{r}_0)_2$, unde **n** – numărul de cifre binare, $\mathbf{q}_i$ – câtul de la împărțire, $\mathbf{r}_i$ – restul de la împărțire, iar **i** = (0, 1, 2, ..., **n**-1) [7].

**EXEMPLU 2**. *Conversia numărului binar în zecimal*. Fie că este necesar să convertim numărul 97 în binar. Folosind algoritmul de mai sus obținem:

Pasul 1. 97 / 2 = 48 cu restul 1;
Pasul 2. 48 / 2 = 24 cu restul 0;
Pasul 3. 24 / 2 = 12 cu restul 0;
Pasul 4. 12 / 2 = 6 cu restul 0;
Pasul 5. 6 / 2 = 3 cu restul 0;
Pasul 6. 3 / 2 = 1 cu restul 1;
Pasul 7. 1 / 2 = 0 cu restul 1.

După cum se vede am împărțit numărul inițial 97 la 2 iar ulterior fiecare cât la 2. Procesul respectiv se stopează atunci când se obține câtul egal cu 0. Scriem rezultatul final, evidențiind resturile obținute de la împărțire la 2, de la stânga la dreapta, începând cu restul de la ultima împărțire, pasul 7. Adică, restul (1) de la pasul 7 va fi prima cifra din stânga, apoi restul (1) de la pasul 6, urmează restul (0) de la pasul 5 și așa mai departe. Ca rezultat, obținem numărul

$(97)_{10} = (1100001)_2$. Un exemplu de aplicare a numerelor binare în criptografie este funcția IntToBin(a, n, b), scrisă în limbajul C++ (9-SMH), care întoarce vectorul binar **b** de o lungime **n** al unui caracter ASCII care are codul **a**.

### 3.2. Conversia numerelor binare în zecimale

În sistemul binar numărul natural, poate fi scris ca

$$(\mathbf{a_{n-1}\, a_{n-2} \ldots a_1 a_0})_2 = \sum_{\mathbf{k=0}}^{\mathbf{n-1}} \mathbf{a_k\, 2^k}$$

Unde **n** – numărul de cifre în numărul binar examinat iar $\mathbf{a_k}$ – valoarea cifrelor din mulțimea (0, 1), iar **k** – numărul de ordine al cifrei [7].

Astfel, fiecare cifră reprezintă o putere a lui 2, prima cifră din dreapta reprezintă $2^0$, a doua cifră $2^1$, apoi $2^2$ și așa mai departe. Valoarea unui număr binar este suma puterilor.

**EXEMPLU 3**. *Conversia numărului zecimal în binar*. Numărul binar $(01100001)_2$ conține n = 8 cifre de 0 și 1. Deci, k va lua valori de la 0 până la n-1 =7. Astfel, numărul binar $(01100001)_2$ este convertit în formă zecimală după cum urmează: $(01100001)_2 = 0\cdot2^7 + 1\cdot2^6 + 1\cdot2^5 + 0\cdot2^4 + 0\cdot2^3 + 0\cdot2^2 + 0\cdot2^1 + 1\cdot2^0 = 0 + 64 + 32 + 0 + 0 + 0 + 0 + 1 = 97$.

Dacă folosim tabelul ASCII (American Standard Code for Information Interchange) ca o relație biunivocă între simboluri și coduri zecimale, atunci putem afirma că am realizat conversia literei **a** în număr binar și invers.

## 4. Algoritmul de generare a șirului cu creștere mare

*Șir cu creștere mare* se numește șirul, în care fiecare element următor este mai mare decât suma tuturor elementelor precedente. Cu alte cuvinte, un șir de **n** numere întregi pozitive $s_1, s_2, s_3, \ldots, s_n$ este considerat șir cu creștere mare dacă: $s_{n+1} > \sum_{i=1}^{n} s_i\,, \forall n > 1$ [1, 2].

**EXEMPLU 4**. Șirul ( 5, 7, 15, 31, 63, 127, 255, 511, 1023) este considerat șir cu creștere mare; șirul ( 5, 5, 10, 20, 40, 80, 160, 320, 640) nu este considerat șir cu creștere mare.

Un exemplu de aplicare a șirurilor cu creștere mare în criptografie este funcția CrestereMare_1(n, w[nn]), scrisă în limbajul C++ (9-SMH), care generează vectorul cu creștere mare **w** de **n** elemente.

## 5. Algoritmul problemei rucsacului

În anul 1978 Ralph Merkle și Martin Hellman au propus ca problema rucsacului să fie utilizată pentru criptarea asimetrică. Astfel, criptosistemul **Merkle-Hellman** (**MH**) este considerat unul dintre primele criptosisteme cu cheie publică. Sistemul Merkle-Hellman se bazează pe problema sumelor de submulțimi (un caz special al problemei rucsacului). În general, această problemă este considerată a fi NP-completă; dar există niște cazuri ușoare care pot fi rezolvate eficient. Schema Merkle-Hellman este bazată pe transformarea unui caz ușor într-unul dificil, și invers. În orice caz, schema respectivă a

fost spartă de Adi Shamir, nu prin atacarea problemei rucsacului, ci prin coruperea conversiei de la rucsacul ușor la cel dificil.

*Formularea clasică a problemei rucsacului.* Fie că avem **n** obiecte, fiecare dintre aceștia având doi parametri - *greutate* $\mathbf{w_i}$ și *valoare* $\mathbf{v_i}$, unde **i**=1, 2, ..., **n**. Există, de asemenea, un rucsac cu o limită de greutate **W**. Sarcina este să umpleți rucsacul cu obiecte astfel încât suma valorilor obiectelor selectate să fie maximă [3, 6]:

$$\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{v_i}\,\mathbf{x_i}$$

respectând în același timp limita de greutate a rucsacului

$$\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{w_i}\,\mathbf{x_i} \leq \mathbf{W}, \text{unde } \mathbf{x_i} \in \{\mathbf{0}, \mathbf{1}\}$$

**Notă.** Valoarea $x_i$ = 1 denotă că fiecare obiect cu indicile **i** este pus în rucsac, iar $x_i$ = 0 denotă că obiectul respectiv nu este inclus în rucsac.

*Criptarea cu ajutorul problemei rucsacului.* Datorită faptului că problema rucsacului nu poate fi rezolvată în timp rezonabil folosind greutatea sumară a obiectelor, ea poate fi folosită în criptografice. Prezentăm textul ca un set de obiecte în rucsac, pentru care trimitem doar greutatea sumară.

Fie dat un șirul cu creștere mare **A** = ($\mathbf{a}_1$, $\mathbf{a}_2$, ..., $\mathbf{a}_n$), *elementele posibile în rucsac*, care este o mulțime ordonată de **n** elemente, unde $\mathbf{a}_i$ – greutatea elementului ***i***, iar ***i****=(1, 2, ..., n)*.

Șirul **A** este *cheia publică*. Pentru a cripta textul clar, acesta este împărțit în blocuri de lungime **n** biți, fiecare bit indicând dacă elementul se află în rucsac. De exemplu, fie **n**=8, **A** = ($\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$, $\mathbf{a}_4$, $\mathbf{a}_5$, $\mathbf{a}_6$, $\mathbf{a}_7$, $\mathbf{a}_8$), atunci textul deschis $(11111000)_2$ indică că primele cinci obiecte { $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$, $\mathbf{a}_4$, $\mathbf{a}_5$}, din opt posibile, sunt în rucsac. Se consideră 1 - prezența obiectului în rucsac , iar 0 - indică absența.

După aceasta se calculează greutatea totală a obiectelor din rucsac, suma se calculează pentru textul clar (de exemplu, o literă, un simbol) $\mathbf{a}_1 + \mathbf{a}_2 + \mathbf{a}_3 + \mathbf{a}_4 + \mathbf{a}_5$ și este transmisă ca text cifrat.

**EXEMPLU 5.** Fie dat un șirul cu creștere mare **A** = (5, 7, 15, 31, 63, 127, 255, 511), cu lungimea **n** = 8. Să se cripteze textul „BAC” după algoritmul descris.

**Soluție.** În șirul A fiecare număr are un anumit indice conform tabelei 1:

**Tabelul 1. Indicii șirului cu creștere mare**

| **Șirul** | 5 | 7 | 15 | 31 | 63 | 127 | 255 | 511 |
|---|---|---|---|---|---|---|---|---|
| **Indice** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

În cazul codului ASCII, pentru informația textuală, fiecărui caracter îi corespunde un anumit cod - un șir finit format din opt cifre binare. Șirul respectiv se numește octet (în engleză byte). În total sunt posibile $2^8$=256 de șiruri distincte, fapt ce permite

reprezentarea literelor mari și mici ale alfabetului latin, a cifrelor, semnelor de punctuație etc. Corespondența dintre caractere și octeți se definește cu ajutorul unui tabel, numit tabel de codificare sau, pur și simplu, cod. Mai jos sunt redate codurile ASCII pentru literele mari ale alfabetului latin utilizat în calculatoarele personale, Tabelul 2.

**Tabelul 2. Codul ASCII (binar și zecimal) pentru litere mari ale alfabetului latin**

| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** |
|---|---|---|---|---|---|---|---|---|
| 01000001<br>65 | 01000010<br>66 | 01000011<br>67 | 01000100<br>68 | 01000101<br>69 | 01000110<br>70 | 01000111<br>71 | 01001000<br>72 | 01001001<br>73 |
| **J** | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** |
| 01001010<br>74 | 01001011<br>75 | 01001100<br>76 | 01001101<br>77 | 01001110<br>78 | 01001111<br>79 | 01010000<br>80 | 01010001<br>81 | 01010010<br>82 |
| **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** | |
| 01010011<br>83 | 01010100<br>84 | 01010101<br>85 | 01010110<br>86 | 01010111<br>87 | 01011000<br>88 | 01011001<br>89 | 01011010<br>90 | |

Corespondența dintre codurile binare ale simbolurilor alfabetului și șirul cu creștere mare este prezentă în tabelul 3.

**Tabelul 3. Corespondența dintre codurile binare ale simbolurilor alfabetului și șirul A**

| **Indice** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Șirul** | **5** | **7** | **15** | **31** | **63** | **127** | **255** | **511** |
| **Codul binar al simbolului A** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| **Codul binar al simbolului B** | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| **Codul binar al simbolului C** | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

Respectiv în tabelul 4 se descriu parametrii procesului de criptare a textului „BAC”.

**Tabelul 4. Descrierea procedurii de criptare a textului ”BAC”**

| **Text deschis** | **B** | **A** | **C** |
|---|---|---|---|
| **ASCII cod** | **66** | **65** | **67** |
| **Cod binar** | 01000010 | 01000001 | 0100011 |
| **Numere din șirul A pe poziția respectivă conform codului binar** | Pe poziția 2 și 7 sunt numerele:<br>7; 255 | Pe poziția 2 și 8 sunt numerele:<br>7; 511 | Pe poziția 2, 7 și 8 sunt numerele:<br>7; 255; 511 |
| **Suma rucsacului** | 7+255=262 | 7+511=518 | 7+255+511=773 |
| **Text criptat** | 262 | 518 | 773 |

## 6. Algoritmul de criptare Merkle–Hellman

Algoritmul de criptare Merkle–Hellman are la bază ideea descrisa în **Exemplul 5** și constă din 3 părți: generarea cheilor; criptare; decriptare [3, 6].

*Generarea cheilor*

1. Se alege un număr întreg **n**, astfel încât toate codurile alfabetului să poată fi reprezentate în numere de până la **n** biți (număr binar de **n** simboluri).
2. Se calculează aliator un șir **W** de numere cu creștere mare (elementele posibile în rucsac), de lungime **n**, $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n)$.
3. Se alege un număr întreg aliator **q** astfel încât:

$$\mathbf{q} > \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{w_i}$$

4. Se alege un număr întreg aliator **r** astfel încât **cmmdc**(**r**, **q**)=1.
5. Se calculează șirul **B** = ($\mathbf{b}_1$, $\mathbf{b}_2$, ..., $\mathbf{b}_n$), unde $\mathbf{b}_i$=**r** · $\mathbf{w}_i$ mod **q**, **i**=(1, 2, ..., **n**).
6. Se determină cheile: a) cheia privată este (**W**, **q**, **r**); b) Cheia publică este **B**.

*Criptarea textului*

7. Fie **M** un text. Textul **M** transformat în număr de **n**-biți este ($\mathbf{m}_1$ $\mathbf{m}_2$ ... $\mathbf{m}_n$)$_2$. Se calculează

$$\mathbf{c} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{m_i} \cdot \mathbf{b_i}$$

8. Textul criptat este **c**.

*Decriptarea textului criptat*

9. Se calculează inversul **r** modulo **q** folosind algoritmul Euclid extins. Inversul va exista deoarece **r** și **q** sunt reciproc prime.

$$\mathbf{r}' = \mathbf{r}^{-1} \pmod{\mathbf{q}}.$$

10. Se calculează **c**' = **c** · **r**' (mod **q**), apoi determinăm șirul **X** după cum urmează:
    a. Se inițializează **X**, șirul unde v-om plasa ordinul unităților în reprezentarea binară a textului decriptat;
    b. Se alege cel mai mare element $\mathbf{w}_j \leq \mathbf{c}'$, unde $\mathbf{w}_j \in \mathbf{W}$, $\mathbf{j} \in$(1, 2, ..., **n**).
    c. Se calculează **c**' = **c**' - $\mathbf{w}_j$;
    d. Se adaugă indicele **j** în lista **X**;
    e. Dacă **c**' este mai mare decât zero, reveniți la **pasul b**; În caz contrar procesul itrerativ se stopează și se trece la următorii pași.
11. Fie că **X** = ($\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_k$). Se verifică relația:

$$\mathbf{c}' = \sum_{\mathbf{i=1}}^{\mathbf{k}} \mathbf{w_{x_i}}$$

12. Se calculează textul **M** după cum urmează:

$$\mathbf{M} = \sum_{\mathbf{i=1}}^{\mathbf{k}} \mathbf{2^{n-x_i}}$$

13. Textul **M** este textul decriptat.

## 7. Aplicarea algoritmului Merkle–Hellman

**PROBLEMA 1.** Folosind algoritmul sistemului de criptare Merkle–Hellman să se cripteze apoi să se decripteze textul **M**= „C”. Să se folosească în calitate de alfabet literele mari ale alfabetului latin din tabelul ASCII cu reprezentările zecimale ale codurilor.

**Soluție.**

***Generarea cheilor***

1. Toate codurile tabelului ASCII pot fi reprezentate în numere de până la 8 biți. Se alege un număr întreg. Fie **n**=8.
2. Se calculează aliator un șir **W** = (**w**$_1$, **w**$_2$, ..., **w**$_n$) de numere cu creștere mare (rucsacul), de lungimea **n**.

   **w**$_1$=5;

   **w**$_2$=2+5=7;

   **w**$_3$=3+5+7=15;

   **w**$_4$=4+5+7+15=31;

   **w**$_5$=5+5+7+15+31=63;

   **w**$_6$=6+5+7+15+31+63=127;

   **w**$_7$=7+5+7+15+31+63+127=255;

   **w**$_8$=8+5+7+15+31+63+127+255=511;

   Obținem șirul **W** = (5, 7, 15, 31, 63, 127, 255, 511).
3. Se alege un număr întreg aliator *q* astfel încât:

$$\mathbf{q} > \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{w_i} = 5 + 7 + 15 + 31 + 63 + 127 + 255 + 511 = 1014$$

   Fie **q**=1020.
4. Se alege un număr întreg aliator astfel încât **cmmdc**(**r**, **q**)=1. Fie **r** = 77, deoarece cmmdc(77, 1020) = 1.
5. Se calculează șirul **B** = (**b**$_1$, **b**$_2$, ..., **b**$_n$), unde $\mathbf{b_i}=\mathbf{r} \times \mathbf{w_i} \bmod \mathbf{q}$, **i** = (1, 2, ..., **n**).

   Reamintim că operația modulo (mod) a numerelor întregi a și b „**a mod b**” returnează restul după împărțirea lui **a** la **b.** Astfel,

   **b**$_1$=77 × 5 mod 1020 = 385 mod 1020 = 385; Deoarece, 385=1020 × 0 +385.

   **b**$_2$=77 × 7 mod 1020 = 539 mod 1020 = 539; Deoarece, 539=1020 × 0 +539.

   **b**$_3$=77 × 15 mod 1020 = 1155 mod 1020 = 135; Deoarece, 1155=1020 × 1 +135.

   **b**$_4$=77 × 31 mod 1020 = 2387 mod 1020 = 347; Deoarece, 2387=1020 × 2 +347.

   **b**$_5$=77 × 63 mod 1020 = 4851 mod 1020 = 771; Deoarece, 4851=1020 × 4 +771.

   **b**$_6$=77 × 127 mod 1020 = 9779 mod 1020 = 599; Deoarece, 9779=1020 × 9 +599.

   **b**$_7$=77 × 255 mod 1020 = 19635 mod 1020 = 255; Deoarece, 19635= =1020 × 19 +255.

   **b**$_8$=77 × 511 mod 1020 = 39347 mod 1020 = 587; Deoarece, 39347= =1020 × 38 +587.

   **B** = (385, 539, 135, 347, 771, 599, 255, 587).
6. Astfel, am determinat cheile:

   *Cheia privată* este (**W**, **q**, **r**), unde **W** = (5, 7, 15, 31, 63, 127, 255, 511); **q** = 1020; **r** = 77.

*Cheia publică* este **B**, unde **B** = (385, 539, 135, 347, 771, 599, 255, 587).

***Criptarea textului***

7. Fie **M** = „C” un text. Textul **M** = 67 (codul ASCII al simbolului „C”) transformat în număr de **n**-biți este $(\mathbf{m}_1\ \mathbf{m}_2\ ...\ \mathbf{m}_n)_2$. = $(01000011)_2$ . Se calculează:

$c= m_1\times b_1+ m_2\times b_2+ m_3\times b_3+ m_4\times b_4+ m_5\times b_5+ m_6\times b_6+ m_7\times b_7+ m_8\times b_8 =$

$= 0\times385+1\times539+0\times135+0\times347+0\times771+0\times599+1\times255+1\times587 =539+255+587=1381$

8. Textul criptat este **c**=*1381*.

***Decriptarea textului***

9. Se calculează inversul **r** modulo **q** folosind algoritmul Euclid extins. Inversul va exista deoarece **r** și **q** sunt reciproc prime.

$$\mathbf{r}' = \mathbf{r}^{-1} \pmod{\mathbf{q}} = (77)^{-1} \pmod{1020} = [(77*53 \bmod 1020)=1] =53.$$

10. Se calculează $\mathbf{c}' = \mathbf{c} \times \mathbf{r}' \pmod{\mathbf{q}}$.

Reamintim că două numere întregi, **a** și **b**, sunt congruente după (modulo **n**) dacă și numai dacă au același rest atunci când sunt împărțite la **n**. Cu alte cuvinte, pentru un număr întreg **k** (pozitiv sau negativ): **a = b + kn** sau **a − b = kn**. Când două numere **a** și **b** sunt congruente după modulo n se notează cu: **a ≡ b (mod n)**.

Astfel, calculăm $\mathbf{c}' = \mathbf{c} \times \mathbf{r}' \pmod{\mathbf{q}} = 1381 \times 53 \pmod{1020} = 773$ și ulterior se determină **X** după cum urmează:

a. Se inițializează **X**, vectorul unde v-om plasa ordinul unităților în reprezentarea binară a textului decriptat;

**I iterație**

b. Se alege cel mai mare element $\mathbf{w}_j \le c'$, $\mathbf{w}_8 =511 \le 773$.
c. Se calculează $\mathbf{c}' = \mathbf{c}' - \mathbf{w}_j = 773 – 511 = 262$;
d. Se adaugă indicele **j=8** în lista **X**=(8);
e. Deoarece **c**' = 262>0, se revine la **pasul b**;

**II iterație**

b. Se alege cel mai mare element $\mathbf{w}_j \le \mathbf{c}'$, $\mathbf{w}_7 =255 \le 262$.
c. Se calculează $\mathbf{c}' = \mathbf{c}' - \mathbf{w}_j = 262 – 255 = 7$;
d. Se adaugă indicele **j=7** în lista **X**=(8, 7);
e. Deoarece **c**' = 7>0, se revine la **pasul b**;

**III iterație**

b. Se alege cel mai mare element $\mathbf{w}_j \le c'$, $\mathbf{w}_2 =7 \le 7$.
c. Se calculează $\mathbf{c}' = \mathbf{c}' - \mathbf{w}_j = 7 – 7 = 0$;
d. Se adaugă indicele **j=2** în lista **X**=(8, 7, 2);
e. Deoarece **c**' = 0, procesul iterațional se stopează.
Se trece la îndeplinirea Pașilor 11-13.

11. Am obținut că $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = (8, 7, 2)$.

Luând în considerare că W = (5, 7, 15, 31, 63, 127, 255, 511) și ținând cont de elemenele vectorului X avem că $w_8$=511, $w_7$=255 și $w_2$=7.

Verificăm dacă **c**'= 773.

Calculăm **c**'= $w_{x1} + w_{x2} + w_{x3} = w_8 + w_7 + w_2 = 511+255+7 = 773$.

Deci, **c**'= 773.

Rezultă că până la acest pas toate calculele au fost efectuate corect.

12. Luând în considerare că n=8 și k=3, calculăm textul **M** după cum urmează:

$$\mathbf{M} = \sum_{\mathbf{i=1}}^{\mathbf{k}} \mathbf{2^{n-x_i}} = 2^{8-8} + 2^{8-7} + 2^{8-2} = 2^0 + 2^1 + 2^6 = 67$$

13. Astfel, obținem că textul **M** =„C”, deoarece codul ASCII 67 corespunde simbolului „C”. Textul **M** =„C” este decriptat corect.

**PROBLEMA 2**. În condițiile Problemei 1, folosind sistemul de criptare Merkle–Hellman să se cripteze apoi să se decripteze textul **M** = „LAC”. Să se folosească în calitate de alfabet literele mari ale alfabetului latin din tabelul ASCII cu reprezentările zecimale ale codurilor.

**Soluție**. Pentru a soluționa problema și în acest caz trebuie să realizăm 3 etape de bază:

Etapa 1. Generarea cheilor;

Etapa 2. Criptarea textului;

Etapa 3. Decriptarea textului.

**Etapa 1.** *Generarea cheilor*, toți pașii se realizează fără nici o modificare exact ca și în cazul Problemei 1. Astfel, obținem că:

Cheia privată este (**W**, **q**, **r**), unde **W** = (5, 7, 15, 31, 63, 127, 255, 511); **q** = 1020; **r** = 77.

Cheia publică este **B**, unde **B** = (385, 539, 135, 347, 771, 599, 255, 587).

**Etapa 2.** *Criptarea textului* procedăm că și în cazul Problemei 1. Astfel, pentru: punctul 7 avem:

| **Textul M** | **L** | **A** | **C** |
|---|---|---|---|
| **Codul ASCII** | **76** | **65** | **67** |
| **Codul binar** | 01001100 | 01000001 | 0100011 |
| **Calcularea textului criptat $c_1$, $c_2$, $c_3$** | $c_1$= $m_1$· $b_1$+ $m_2$· $b_2$+ $m_3$· $b_3$+ $m_4$· $b_4$+ $m_5$· $b_5$+ $m_6$· $b_6$+ $m_7$·$b_7$+$m_8$·$b_8$= 0·385+1·539+0·135+0·347 +1·771+1·599+0·255+0·58 7=539+771+599==1909 | $c_2$= $m_1$· $b_1$+ $m_2$· $b_2$+ $m_3$· $b_3$+ $m_4$· $b_4$+ $m_5$· $b_5$+ $m_6$· $b_6$+ $m_7$·$b_7$+$m_8$·$b_8$= 0·385+1·539+0·135+0·347+0 ·771+0·599+0·255+1·587= 539+587=1126 | $c_3$= $m_1$· $b_1$+ $m_2$· $b_2$+ $m_3$· $b_3$+ $m_4$· $b_4$+ $m_5$· $b_5$+ $m_6$·$b_6$+$m_7$·$b_7$+$m_8$·$b_8$= 0·385+1·539+0·135+0·347+0 ·771+0·599+1·255+1·587=53 9+255+587=1381 |

8. Textul criptat este **C**= ( $c_1$, $c_2$, $c_3$)

| **C** | $c_1$ = 1909 | $c_2$ = 1126 | $c_3$ = 1381 |
|---|---|---|---|

**Etapa 3.** *Decriptarea textului* obținem:

9. Se calculează inversul **r** modulo **q** folosind algoritmul Euclid extins. Inversul va exista deoarece **r** și **q** sunt reciproc prime.

$$\mathbf{r}' = \mathbf{r}^{-1} \pmod{\mathbf{q}} = (77)^{-1} \pmod{1020} = [(77*53 \bmod 1020)=1] = 53.$$

10. Se calculează: $\mathbf{c}' = \mathbf{c_i} \cdot \mathbf{r}' \pmod{\mathbf{q}}$, unde i = 1, 2, 3.

| **c'** | $\mathbf{c_1}' = \mathbf{c_1} \cdot \mathbf{r}' \pmod{\mathbf{q}}$ | $\mathbf{c_2}' = \mathbf{c_2} \cdot \mathbf{r}' \pmod{\mathbf{q}}$ | $\mathbf{c_3}' = \mathbf{c_3} \cdot \mathbf{r}' \pmod{\mathbf{q}}$ |
|---|---|---|---|
| | $\mathbf{c_1}' = \mathbf{1909} \cdot \mathbf{53} \pmod{\mathbf{1020}}$ = 101177 mod 1020 = 197 | $\mathbf{c_2}' = \mathbf{1126} \cdot \mathbf{53} \pmod{\mathbf{1020}}$ = 59678 mod 1020 = 518 | $\mathbf{c_3}' = \mathbf{1381} \cdot \mathbf{53} \pmod{\mathbf{1020}}$ = 101177 mod 1020=773 |

| a) Se determină șirurile: $\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}$. | | | |
|---|---|---|---|
| **Iterația I** | | | |
| **W** = (5, 7, 15, 31, 63, 127, 255, 511) | $\mathbf{c_1}'$ =197 | $\mathbf{c_2}'$=518 | $\mathbf{c_3}'$=773 |
| b) Se alege cel mai mare element $\mathbf{w_j} \le$ c' | $w_6$ =127 ≤ 197 | $\mathbf{w_8}$ =511 ≤ 518 | $\mathbf{w_8}$ =511 ≤ 773 |
| c) Se calculează $\mathbf{c}' = \mathbf{c}' - \mathbf{w_j}$ | $\mathbf{c_1}' = \mathbf{c_1}' - \mathbf{w_6}$ = 197 – 127 = 70; | $\mathbf{c_2}' = \mathbf{c_2}' - \mathbf{w_8}$ = 518 – 511 = 7; | $\mathbf{c_3}' = \mathbf{c_3}' - \mathbf{w_8}$ = 773 – 511 = 262; |
| d) Se adaugă indicele **j** în lista **X** | **j=6** se adaugă în lista $\mathbf{X_1}$=(6) | **j=8** se adaugă în lista $\mathbf{X_2}$=(8) | **j=8** se adaugă în lista $\mathbf{X_3}$=(8) |
| e) Dacă **c**'>0, se revine la **pasul b** în caz contrar STOP. | Deoarece $\mathbf{c_1}'$=70>0, se revine la **pasul b** | Deoarece $\mathbf{c_2}'$=7>0, se revine la **pasul b** | Deoarece $\mathbf{c_3}'$=262>0, se revine la **pasul b** |
| **Iterația II** | | | |
| **W** = (5, 7, 15, 31, 63, 127, 255, 511) | $\mathbf{c_1}'$ =70 | $\mathbf{c_2}'$=7 | $\mathbf{c_3}'$=262 |
| b) Se alege cel mai mare element $\mathbf{w_j} \le$ c' | $\mathbf{w_5}$ =63 ≤ 70 | $\mathbf{w_2}$ =7≤7 | $\mathbf{w_7}$ =255 ≤ 262 |
| c) Se calculează $\mathbf{c}' = \mathbf{c}' - \mathbf{w_j}$ | $\mathbf{c_1}' = \mathbf{c_1}' - \mathbf{w_5}$ = 70 – 63 = 7; | $\mathbf{c_2}' = \mathbf{c_2}' - \mathbf{w_2}$ = 7 – 7 = 0; | $\mathbf{c_3}' = \mathbf{c_3}' - \mathbf{w_7}$ = 262 – 255 =7; |
| d) Se adaugă indicele **j** în lista **X** | **j=5** se adaugă în lista $\mathbf{X_1}$=(6,5) | **j=2** se adaugă în lista $\mathbf{X_2}$=(8, 2) | **j=7** se adaugă în lista $\mathbf{X_3}$=(8,7) |
| e) Dacă **c**'>0 se revine la **pasul b**, în caz contrar STOP. | Deoarece $\mathbf{c_1}'$=7>0, se revine la **pasul b** | Deoarece $\mathbf{c_2}'$=7=0, procesul se stopează | Deoarece **c**'= 7>0, se revine la **pasul b** |
| **Iterația III** | | | |
| **W** = (5, 7, 15, 31, 63, 127, 255, 511) | $\mathbf{c_1}'$ =7 | - | $\mathbf{c_3}'$=7 |
| b) Se alege cel mai mare element $\mathbf{w_j} \le$ c' | $\mathbf{w_2}$ =7 = $\mathbf{c_1}'$ =7 | - | $\mathbf{w_2}$ =7 = $\mathbf{c_3}'$=7 |
| c) Se calculează $\mathbf{c}' = \mathbf{c}' - \mathbf{w_j}$ | $\mathbf{c_1}' = \mathbf{c_1}' - \mathbf{w_2}$ = 7 – 7 = 0; | - | $\mathbf{c_3}' = \mathbf{c_3}' - \mathbf{w_2}$ = 7 – 7 = 0; |
| d) Se adaugă indicele **j** în lista **X** | **j=5** se adaugă în lista $\mathbf{X_1}$=(6,5,2) | - | **j=2** se adaugă în lista $\mathbf{X_3}$=(8,7,2) |
| e) Dacă **c**'>0 se revine la **pasul b**, în caz contrar STOP. | Deoarece $\mathbf{c_1}'$=0, procesul se stopează | - | Deoarece $\mathbf{c_3}'$=0, procesul iterațional se stopează |
| Am deteminat șirul **X** | $\mathbf{X_1}$=(6,5,2) | $\mathbf{X_2}$=(8, 2) | $\mathbf{X_3}$=(8,7,2) |

Efectuăm pașii 11-13.

| Scriem șirul **X** | $\mathbf{X_1} = (\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}) = (6, 5, 2)$. | $\mathbf{X_2} = (\mathbf{x_1}, \mathbf{x_2}) = (8, 2)$. | $\mathbf{X_3} = (\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}) = (8, 7, 2)$. |
|---|---|---|---|
| Luând în considerare W= (5, 7, 15, 31, 63, 127, 255, 511) scriem $w_i$ | $w_6$=127, $w_5$=63 și $w_2$=7 | $w_8$=511 și $w_2$=7 | $w_8$=511, $w_7$=255 și $w_2$=7 |
| Dacă $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2},\ldots,\mathbf{x_k})$, atunci verificăm $\mathbf{c'} = \sum_{i=1}^{k} \mathbf{w_{x_i}}$ | **c'**= $w_{x1} + w_{x2} + w_{x3} = w_6 + w_5 + w_2 =$ 127+63+7 = 197. **Deci, c'**= 197. Rezultă că calculele s-au efectuat corect. | **c'**= $w_{x1} + w_{x2} =$ $w_8 + w_2 =$ 511+7 = 518. **Deci, c'**= 773. Rezultă că calculele s-au efectuat corect. | **c'**= $w_{x1} + w_{x2} + w_{x3} =$ $w_8 + w_7 + w_2 =$ 511+255+7 = 773. **Deci, c'**= 773. Rezultă că calculele s-au efectuat corect. |
| Luând în considerare n și k determinăm: $\mathbf{M} = \sum_{i=1}^{k} \mathbf{2^{n-x_i}}$ | Deoarece n=8, k=3 și $(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}) = (6, 5, 2)$ avem: $M=2^{8-6} + 2^{8-5}+2^{8-2} =$ =4+8+64=76. | Deoarece n=8, k=2 și $(\mathbf{x_1}, \mathbf{x_2}) = (8, 2)$ avem: $M=2^{8-8} + 2^{8-2} = 65$. | Deoarece n=8, k=3 și $(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}) = (8, 7, 2)$ avem: $M=2^{8-8} + 2^{8-7}+2^{8-2} = 67$. |
| | Obținem că textul $\mathbf{M_1}$ =„L”, deoarece codul ASCII 76 corespunde simbolului „L”. | Obținem că textul $\mathbf{M_2}$ =„A”, deoarece codul ASCII 65 corespunde simbolului „A”. | Obținem că textul $\mathbf{M_3}$ =„C”, deoarece codul ASCII 67 corespunde simbolului „C”. |
| Textul decriptat | L | A | C |

## 8. Sarcini individuale

Folosind algoritmul sistemului de criptare Merkle–Hellman să se cripteze apoi să se decripteze textul **M** cu datele inițiale respective **n**, **W**, **q**, **r**. Folosiți în calitate de alfabet literele tabelului ASCII cu reprezentările zecimale ale codurilor.

| **Ex.** | **Date inițiale** | | | | |
|---|---|---|---|---|---|
| | **n** | **W** | **q** | **r** | **Textul M** |
| 1. | 8 | 5, 7 , 15, 31, 63, 127, 255, 511 | 1785 | 1528 | ABEL |
| 2. | 8 | 5, 11, 23, 47, 95, 191, 383, 767 | 1588 | 1111 | PLAN |

## 9. Algoritmul de criptare Merkle–Hellman în C++

```
//C-Free 5.0
#include<iostream.h>
#include <string.h>
const int nn = 100;
int IsPrim(int n) {
   int i, e = 1;
   for (i = 2; i <= n / 2; i++) {
      if (n % i == 0) { e = 0; break; }
   }
   return e;
}
int cmmdcEuclidExt(int a, int b, int& s,
int& t) {
   int d = 0, x = 0, y = 0;
   if (b == 0) {
      s = 1; t = 0;
      return a;
   }
   d = cmmdcEuclidExt(b, a % b, x, y);
   s = y;
   t = x - (a / b) * y;
   return d;
```

```
}
void IntToBin(int a, int n, int b[nn]) {
   int i, k = a;
   for (i = 0; i <= n; i++) { b[i] = 0; }
   for (i = 0; k > 0; i++) {
      b[n - i] = k % 2;
      k = k / 2;
   }
}
void CrestereMare_1(int n, int w[nn]) {
   int i, s;
   w[1] = 5; s = w[1];
   for (i = 2; i <= n; i++) {
      w[i] = s + i;
      s += w[i];
   }
}
int ProdusMod(int a, int b, int m) {
   int i, s = 0, a1, b1;
   a1 = a % m; b1 = b % m;
   if (a1 > b1) { a1 = b % m; b1 = a % m;
}
   for (i = 1; i <= a1; i++)
      if (s < m) s += b1; else { s += (b1 -
m); }
   return (s % m);
}
void CriptareMH(int PublicKey[nn], int
n, char Text[nn], int TextCrypt[nn]) {
   int i, j, aBin[nn], m = strlen(Text);
   for (j = 1; j <= m; j++) {
      IntToBin((int)Text[j - 1], n, aBin);
      TextCrypt[j] = 0;
      for (i = 1; i <= n; i++) {
         TextCrypt[j] += aBin[i] *
PublicKey[i];
      }
   }
}
void DecriptareMH(int q, int r, int w[nn],
int n, int c[nn], int m, int d2[nn]) {
   int i, j, aBin2[nn][nn], d1[nn], iv = 0,
p, d, t = 0;
   d = cmmdcEuclidExt(r, q, iv, t);
   iv = (iv + q) % q;
   for (j = 1; j <= m; j++) {
      d1[j] = ProdusMod(c[j], iv, q);
      for (i = n; i > 0; i--) {
         aBin2[j][i] = 0;
         if (w[i] <= d1[j]) {
            aBin2[j][i] = 1;
            d1[j] -= w[i];
         }
      }
   }
   for (j = 1; j <= m; j++) {//binToInt
      p = 1; d2[j] = 0;
      for (i = n; i > 0; i--) {
         d2[j] += p * aBin2[j][i];
         p *= 2;
      }
   }
}
int main() {
   int i, j, k, s = 0, aBin2[nn][nn];
   int w[nn];//sir crestere mare
   int n = 8;//numar in sit crestere mare
   int q, r;
   int b[nn];//cheia publica
   char a[nn];//textul
   int m;//lungime mesaj
   int c[nn];//textul criptat
   cout << "SISTEMUL
CRIPTOGRAFIC MERKLE-
HELLMAN\n";
   CrestereMare_1(n, w);
   cout << "Sirul cu crestere mare\n";
   cout << "w = {";
```

```
    for (i = 1; i <= n; i++) { cout << w[i] << " "; } cout << "}" << endl;
    for (i = 1; i <= n; i++) { s += w[i]; }
    cout << "Suma sirului s = " << s << ". Selectati un numar >s. q = "; cin >> q;
    cout << "Alegeti un numar de la 1 la " << q << " unde cmmdc(q,r)=1. r = "; cin >> r;
    for (i = 1; i <= n; i++) { b[i] = ProdusMod(w[i], r, q); }
    cout << "\nCRIPTARE\n";
    cout << "Cheia publica este b = {";
    for (i = 1; i <= n; i++) { cout << b[i] << " "; } cout << "}" << endl;
    cout << "mesajul a = "; cin.getline(a, nn, '\n');
    cin.getline(a, nn, '\n');
    CriptareMH(b, n, a, c);
    m = strlen(a);
    cout << "Mesajul criptat este c = ";
    for (i = 1; i <= m; i++) { cout << c[i] << " "; } cout << endl;
    cout << "\nDECRIPTARE\n";
    cout << "Cheia secreta este (w, q, r):\n";
    cout << "w = {";
    for (i = 1; i <= n; i++) { cout << w[i] << " "; } cout << "}" << endl;
    cout << "q = " << q << endl;
    cout << "r = " << r << endl;
    int d2[nn];
    DecriptareMH(q, r, w, n, c, m, d2);
    cout << "Mesajul decriptat este d2 = ";
    for (j = 1; j <= m; j++) { cout << (char)d2[j]; } cout << endl;
    return 0;
}
```

## 10. Rezultate obținute

**Ex. 1**

| | | | | |
|---|---|---|---|---|
| **B** | (500, 1771, 1500, 958, 1659, 1276, 510, 763) | | | |
| **M** | A | B | E | L |
| **Cod** | 65 | 66 | 69 | 76 |
| **Binar** | 01000001 | 01000010 | 01000101 | 01001100 |
| **c** | 2534 | 2281 | 3810 | 4706 |
| **r'** | 382 | | | |
| **c'** | 518 | 262 | 645 | 197 |
| **X** | (8, 2) | (7, 2) | (8, 6, 2) | (6, 5, 2) |
| $w_{x_i}$ | (511, 7) | (255, 7) | (511, 127, 7) | (127, 63, 7) |
| $2^{n-x_i}$ | $2^{8-8}$, $2^{8-2}$ | $2^{8-7}$, $2^{8-2}$ | $2^{8-8}$, $2^{8-6}$, $2^{8-2}$ | $2^{8-6}$, $2^{8-5}$, $2^{8-2}$ |
| **Cod** | 65 | 66 | 69 | 76 |
| **M** | A | B | E | L |

**Ex. 2**

| | | | | |
|---|---|---|---|---|
| **B** | (791, 1105, 145, 1401, 737, 997, 1517, 969) | | | |
| **M** | P | L | A | N |
| **Cod** | 80 | 76 | 65 | 78 |
| **Binar** | 01010000 | 01001100 | 01000001 | 01001110 |
| **c** | 2506 | 2839 | 2074 | 4356 |
| **r'** | 263 | | | |
| **c'** | 58 | 297 | 778 | 680 |
| **X** | (4, 2) | (6, 5, 2) | (8, 2) | (7, 6, 5, 2) |
| $w_{x_i}$ | (47, 11) | (191, 95, 11) | (767, 11) | (383, 191, 95, 11) |
| $2^{n-x_i}$ | $2^{8-4}$, $2^{8-2}$ | $2^{8-6}$, $2^{8-5}$, $2^{8-2}$ | $2^{8-8}$, $2^{8-2}$ | $2^{8-7}$, $2^{8-6}$, $2^{8-5}$, $2^{8-2}$ |
| **Cod** | 80 | 76 | 65 | 78 |
| **M** | P | L | A | N |

**Concluzii**. Abordarea respectivă, în opinia noastră, permite înțelegerea în profunzime aplicarea sistemelor criptografice asimetrice, totodată evidențiază conexiunea puernică dintre algebra abstractă și criptografiere, cu accente pe fundamentele algebrice necesare la elaborarea formarea algoritmilor de calcul și programarea lor. Sistemul criptografic Merkle–Hellman în procesul de criptare/decriptare folosește următoarele concepte și noțiuni algebrice: numere reciproc prime, înmulțirea a două numere (modulo m), inversul multiplicativ al unui număr (modulo m), algoritmul lui Euclid extins, conversia numerelor binare în zecimal și invers, șirului cu creștere mare, problema rucsacului.

Cunoașterea algoritmilor de calcul al acestor noțiuni fundamentale din toeria numerelor oferă posibilitatea viitorului informatician să cunoască în profunzime procesul de criptare/decriptare, inclusiv sistemul criptografic Merkle–Hellman. În așa mod studentul își îmbunătățește cunoștințele în domeniu și este convins de faptul că cunoștințele temeinice din algebra abstractă facilitează enorm înțelegerea funcționării sistemelor criptografie și ulterior perfecționarea și elaborarea unor noi sisteme în acest sens. Din punct de vedere procedural studentului i se cultivă abilități și deprinderi în gestionarea cheilor criptografice.

*Articol elaborat în cadrul proiectului de cercetări științifice „Metodologia implementării TIC în procesul de studiere a științelor reale în sistemul de educație din Republica Moldova din perspectiva inter/transdisciplinarității (concept STEAM)”, inclus în „Program de Stat” (2020-2023), Prioritatea IV: Provocări societale, cifrul 20.80009.0807.20.*

## Bibliografie

1. MOLLIN, R. A. An introduction to cryptography. Second Edition. Taylor & Francis Group, LLC, 2007. 394 p. ISBN-13: 978-1-58488-618-1.
2. TILBORG van, H. C.A. Fundamentals of cryptology. A Professional Reference and Interactive Tutorial. Kluwer Academic Publishers, 2002. 508 p.
3. ROBLING DENNING, D. E. Cryptography and data Security. Addison-Wesley Publishing Company, 1982. 419 p. ISBN 0-201-10150-5.
4. EASTTOM, C. Modern Cryptography, Applied Mathematics for Encryption and Information Security. McGraw-Hill Education, 2016. 505 p.
5. ЯЩЕНКО, В.В. Введение в криптографию. Москва: Издательство МЦНМО, 2012. 342 p. ISBN 978-5-4439-0026-1.
6. УРБАНОВИЧ, П. П. Защита информации методами криптографии, стеганографии и обфускации. Минск, 2016. 220 p. ISBN 978-985-530-562-1.
7. КОВРИЖЕНКО, Г. А. Системы счисления и двоичная арифметика. Киев, 1984. 22.12/К56, 82 p.
8. GALLIAN, J. A. Contemporary Abstract Algebra. Cengage Learning, 2017. 631 p. ISBN: 978-1-305-65796-0.
9. ATANASIU, A. Securitatea informației (criptografie). vol. 1. Cluj: INFODATA, 2007. ISBN: 978-973-1803-16-6.