APLICAREA CODURILOR MATROIDE CORECTOARE DE ERORI IN CRIPTOGRAFIE

Larisa Dunai

Universitatea Tehnică a Moldovei

Referat

În articol sunt tratate problemele de generare, codare şi decodare a codurilor matroide şi aplicarea acestora în criptografie. Generarea codului matroid de lungimea *n*, capacitatea corectoare *t* raportate la secvenţa de intrare de *k* simboluri este o problemă de căutare a matroidului uniform în spaţiul vectorial asupra câmpului Galois extins cu caracteristica 2^m, unde *m* este binaritatea simbolurilor de intrare-ieşire. Este prezentată corespondenţa dintre parametrii *n*, *k* şi *t*. Au fost delimitate graniţele de existenţă a matroidelor uniforme. Sunt analizate particularităţile codării şi decodării codurilor matroide. De asemenea, sunt analizate caracteristicile informaţionale ale textelor generate de codurile matroide.

Abstract

This work treats the matroid code generation, encoding and decoding and their application in cryptography. Generation of matroid code of the length *n* and error ability *t* related to *k* original symbols is a task of the uniform matroid searching in the vector space over extended Galois field with characteristic 2^m, where *m* is the symbol length. The relation between parameters *n*, *k* and *t* are obtained. The bounds of the uniform matroid were found. The particularities of the matroid codes encoding and decoding are analyzed. Also, the informational parameters of the generated texts are analyzed from the cryptography point of view.

I. Introducere

”Criptografie” în traducere din limba greacă înseamnă ”scriere secretă”. Metodele primitive de criptare sunt cunoscute încă din vremuri străvechi şi analizate timp îndelungat, mai degrabă ca un vicleşug, decât o disciplină ştiinţifică exactă. Sarcina clasică a criptării este transformarea reversibilă a mesaj iniţial (original) într-o succesiune aleatoare numită criptogramă. În acest mod textul cifrat poate conţine atît elemente noi, cât şi elemente deja existente în mesajul iniţial. Cantitatea simbolurilor în criptogramă şi în mesajul iniţial poate fi diferită. O cerinţă indispensabilă este că făcând careva substituţii logice asupra simbolurilor mesajului cifrat se poate univoc (şi totalmente) de restabilit mesajul iniţial. Păstrarea informaţiei depindea pe vremuri de metoda de transformare.

Una din lucrările teoretice fundamentale în criptografie a fost publicată de K. Shannon în anul 1949. Această lucrare [1,2] dedicată analizei teoretice a sistemelor secrete, a pus temelia criptografiei contemporane şi a devenit baza elaborarea noilor sisteme criptografice. După Shannon cifrarea este aplicaţie a mesajului iniţial într-o criptogramă:

$$C=F_i(M),$$

unde C – criptograma, F_i – aplicaţia, M – mesajul iniţial, indicatorul i corespunde cheii de cifrare. Pentru o decriptare univocă este necesar ca F_i să aibă o funcţie inversă unică astfel încât $F_iF_i^{-1}=I$, unde I este o transformare echivalentă:

$$M=F_i^{-1}(C).$$

Se presupune că generatorul cheii este un proces statistic sau un dispozitiv, care definește transformările $F_1, F_2, ..., F_{N1}$, cu probabilitățile $P_1, P_2, ..., P_{N1}$, iar numărul mesajelor posibile N_2 este finit şi mesajele $M_1, M_2, ..., M_{N2}$ au probabilitățile apriori $q_1, q_2, ..., q_{N2}$.

Fie un cifru, alfabetul primar al căruia coincide cu mulțimea simbolurilor cheii şi ale criptogramei, iar criptarea se efectuiază prin înlocuirea secvențială a simbolurilor mesajului original cu simbolurile criptogramei în dependență de semnificația curentă a simbolurilor cheii. În acest caz mesajul, cheia şi criptograma reprezintă o secvență de litere ale unuia şi aceluiaşi alfabet: $M=(m_2,m_2,...,m_n)$, $K=(k_1,k_2,...,k_n)$, $C=(c_1,c_2,...,c_n)$. Pasul curent al cifrării este descris de relația:

$$C_i=f(m_i,k_i).$$

În sistemele criptografice aplicate în practică, de obicei, lungimea cheii este mai mică decât lungimea mesajului cifrat. De aceea deseori secvența $k_1,k_2,...,k_n$ este calculată pe baza cheii primare de proporții mai mici sau poate fi periodică.

Sarcina criptoanalistului este calculul mesajului original cunoscând criptograma şi mulțimea transformărilor $F_1, F_2, ..., F_{N1}$. Există criptosisteme, pentru care orice volum de informație interceptată este insuficientă pentru a găsi transformările de criptare şi aceasta nu depinde de performanțele de calcul disponibile. Criptarea de acest tip este numită **sigur necondiționată** (conform K. Shannon **absolut secretă**). Conform definiției sigur necondiționate sunt acelea cifruri pentru care criptoanaliticul (chiar dispune de resurse de calcul infinite) nu poate îmbunătăți estimarea mesajului original M pe baza cunoaşterii criptogramei C în comparație cu cea a criptogramei necunoscute. Aceasta este posibil doar în acel caz în care M şi C sînt statistic independente, adică este satisfăcută condiția:

$$P(M=M_i/C=C_i)= P(M=M_i)$$

pentru toate mesajele posibile M.

Criptosisteme sigur necondiționate există. Fie că în cifrul analizat mai sus se utilizează alfabetul din L simbolur, iar simbolurile criptogramei se determină după formula:

$$c_i=f(m_i, k_i)= (m_i+k_i) \textbf{ mod } L,$$

unde în fiecărui simbol c_i, m_i şi k_i îi corespunde numărul de ordine din alfabet.

Alegem în calitate de cheie o serie, compusă din n simboluri aleatoare $k_1,k_2,...,k_n$, adică o cheie aleatoare lungimea căreia este egală cu lungimea mesajului. Pentru generarea cheii poate fi utilizt un generator de numere aleatoare, care asigură la ieşire o apariție echiprobabilă pentru orice simbol din mulțimea de numere {1, 2, ..., L}. În acest caz sursa va permite selectarea echiprobabilă a cheii de lungimea n, iar probabiitatea de selectare a cheii va fi:

$$P(k=k_i)= L^{-n}.$$

Din această relație rezultă că pentru oricare M şi C are loc egalitatea:

$$P(M=M_i/C=C_i)= L^{-n}.$$

Din ultima relație rezultă că unei criptograme de lungimea n îi corespunde cu probabilitatea L^{-n} un mesaj original de lungimea n. La cifrarea unui mesaj nou se va alege o cheie nouă. În acest mod se va induce o aleatorizare pe simbolurile de ieşire care va asigura o incertitudine maximală pe alfabetul textului criptat. Astfel, procedeul de criptare descris este sigur necondiționat.

Cheia este combinația de cod secretă cu ajutorul căreia se efectuiază criptarea textlui inițial. Parametrul de bază al cheii este lungimea ei L_k. Această mărime caracterizează complexitatea de descifrare a criptogramei. Dacă alfabetul de codificare a cheii conține A simboluri, atunci numărul total de chei N_k este egal cu:

$$N_k = A^{L_k}$$

algoritmul de criptare este sigur dacă spargerea acesteia poate fi efectuată într-un singur mod – trierea cheilor. Dacă criptogramele sunt statistic uniforme atunci numărul mediu de trieri este egal cu:

$$\overline{N}_k = N_k/2 = A^{L_k}/2.$$

În cazul unui cod binar avem:

$$\overline{N}_k = 2^{L_k-1}.$$

Un criptoalgoritm este caracterizat de complexitatea internă şi externă. Complexitatea externă este determinată de lungimea cheii şi alţi parametri. Complexitatea internă este determinată de performanţele criptoalgoritmului, posibilitatea de modificare a acestuia. În cazul unei substituţii, complexitatea interioară creşte brusc odată cu creşterea lungimii alfabetului. Una din metodele de ridicare a complexităţii este extinderea alfabetului.

În lucrarea prezentă se propune introducerea codecului codului corector de erori în schema sistemului de transmitere a informaţiei secrete. Aceasta va permite ridicarea complexităţii algoritmului de criptare. În primul rînd, prin codarea textului original se va obţine un text, alfabetul caruia are un cardinal (numărul de smboluri) mai mare decît alfabetul primar. Astfel va creşte complexitatea internă (*confuzia*) a criptoalgoritmului. În al doilea rînd, codarea va creşte difuzia simbolurilor textului secundar şi, prin aceasta, se va micşora interdependenţa simbolurilor.

Pentru realizarea codării se propune aplicarea codurilor matroide – o clasă nouă de coduri liniare corectoare de erori. În compartimentele 2 şi 3 sînt prezentate caracteristicile codului matroid, în compartimentele 4 şi 5 – algoritmii de codare şi decodare a codului matroid, iar în compartimentul 6 – softul experimental şi rezultatele de statistică (statistice) obţinute.

II. Caracteristicile codului matroid

Teoria codurilor corectoare de erori a avut o dezvoltare ascendentă în anii 60-70 ai secolului trecut. În această perioadă au fost propuse şi anlizate majoritatea codurilor cnoscute în prezent. Însă, pentru necesităţile practicii, au fost selectate codurile cu parametrii cei mai optimali. Conform definiţiei *optimal* este codul cu redondanţă minimală şi capacitate corectoare maximală. Redondanţa, adică numărul de simboluri suplimentare adăugate la combinaţia originală, depinde de capacitatea corectoare a codului. Modul de formare (generare) a cuvintelor de cod specifică tipul codului. Sunt cunoscute trei clase de coduri corectoare: liniare, ciclice şi convoluţionale. Codurile liniare se obţin prin transformarea (liniară) a unei secvenţe de k simboluri originale (informaţionale) într-o secvenţă, numită bloc, de n simboluri, unde $n>k$. Cuvintele codului ciclic se obţin prin înmulţirea blocului de intrare la aşa numitul polinom generator. În codurile convoluţionale simbolurile suplimentare ale unui cuvînt de cod controlează simbolurile informaţionale ale altor cuvinte de cod.

În practică, cea mai “crunt” exploatată metodă a devenit codificarea *Reed-Solomon*. Această metodă a fost aleasă pentru corectarea erorilor în sistemele de înregistrare a inormaţiei pe compact-discuri şi, nu întimplător. La acel moment codurile ciclice Reed-Solomon (RS) erau cele mai acceptabile din punct de vedere al criteriului de optimalitate. Pe de altă parte tehnologia circuitelor integrate a permis implementarea schemelor complexe de codare şi decodare.

Recent, în lucrarea de pionerat [3], au fost sugerate idei importante referitoare la construirea unei clase noi de coduri corectoare, numite *coduri matroide* (M-coduri). Pentru construirea codurilor matroide se folosesc *matroidele uniforme* [4]. Căutarea matroidelor uniforme este o problemă complexă (polinomială). În lucrarea [5] a fost propusă o metodă de căutare a matroidelor uniforme bazată pe analiza cicloclaselor, generate în cămpul Galois extins. Aplicarea acestei metode în practică permite găsirea într-o perioadă de timp acceptabilă a matroidelor uniforme de caracteristicile solicitate.

În această lucrare vor fi prezentate inedit modul de estimare a performanţelor codului matroid, graniţele de existenţă a codurilor matroide de caracteristicile specificate. De asemenea, vor fi analizate “mecanismele” de codare şi decodare a codurilor matroide.

Pentru generarea codului matroid se foloseşte matricea care urmează structura matroidului uniform. Conform definiţiei construcţia matematică $U_{k,n}=(G, B)$ este un matroid uniform de rangul k, dacă toate submulţimile B din G sunt baze de cardinalitatea k, unde G este o mulţime de vectori asupra câmpului Galois extins $\mathbf{GF}(2^m)$, $|G|=n$, $k<n$ şi $m= 2,3,\ldots$Atunci matroidul uniform $U_{k,n}$ se prezintă printr-o

matrice de forma $\mathbf{G}_{k\times n}=[g_{ij}]_{k\times n}$, unde $g_{ij}\in$ $\mathbf{GF}(2^m)$. în această interpretare vectorii M-codului vor fi generați prin aplicarea operației matriciale:

$$\mathbf{v}=<v_1,...,v_n>=\mathbf{x}\cdot\mathbf{G}, \qquad (1)$$

unde $\mathbf{x}=<x_1,...,x_n>$ este vectorul original, $x\in$ $\mathbf{GF}(2^m)$.

Precum e binecunoscut, la codarea în bloc, simbolurile redondante ale vectorului **v** sunt destinate pentru corectarea pînă la *t* simboluri. Parametrul (caracteristica) *t* specifică capacitatea (abilitatea) corectoare a codului. Între parametrii *n*, *k* şi *t* există o relație, de regulă, liniară. La construirea codului matroid apare întrebarea: pentru *k* declarat ce lungime *n* trebuie să aibă vectorul **v** ca să asigure capacitatea corectoare *t* ?

Însă, înainte de a răspunde la această întrebare, trebuie de menționat următoarele. Spre deosebire de codarea clasică, codarea matroidală, definită de aplicația $\mathbf{x}\xrightarrow{\mathrm{G}}\mathbf{v}$, generează vectori (cuvinte de cod) **v** în care nu pot fi distinct evidențiate partea de control şi cea informațională. Şi mai mult, în general simbolurile vectorului original **x** "nu se conțin" printre simbolurile vectorului rezultat **v**. Numai în cazul unei structuri alese specific ai matricei **G** poate fi asigurată "tranziția" simbolurilor din **x** în **v**.

Decodarea vectorilor codului matroid, la fel, se deosebeşte vădit de decodarea clasică: simbolurile originale sunt calculate (restabilite) din simbolurile (neeronate) ale vectorului recepționat **v′**. De aceea, este mai oportun de a numi (clasifica) acest proces: restabilire.

Acum, referitor la dependența dintre *k*, *n* şi *t*. Avem următoarea **propoziție**: pentru a corecta *t* erori, cuvintele codului matroid trebuie să fie de lungimea:

$$n=2t+k \text{ sau } n=2t+k+1. \qquad (2)$$

Imediat după această propoziție vine următoarea: *codul matroid de lungimea* (2) *are distanța minimă egală cu*

$$d_{min}=(n-k)\ \mathbf{div}\ 2, \qquad (3)$$

unde **div** este împărțirea fără rest a numerelor întregi.

Acum esența codării M-codului poate fi exprimată de următoarea

Teoremă. Codul matroid construit în $\mathbf{GF}^k(2^m)$ de lungimea (2) poate corecta *t* erori, unde $\mathbf{GF}^k(2^m)$ este spațiul vectorial de dimensiunea *k* asupra câmpului Galois extins de caracteristica 2^m, $m=2,3,\ldots$

Într-adevăr, orice matrice **G** a matroidului uniform $U_{k,n}$ poate fi adusă la forma canonică:

$$\mathbf{G}=[\mathbf{I}\ \mathbf{P}],$$

unde **I** – matricea-unitate de dimensiunea $k\times k$; $\mathbf{P}=[p_{ij}]$ – matrice arbitrară de dimensiunea $k\times(n-k)$, $p_{ij}\in$ $\mathbf{GF}(2^m)$ şi $p_{ij}\neq 0$.

Ca urmare, ponderea minimală *w* a vectorilor codului nu-i mai mare decât: $w\le n(k-1)$. Iar, conform teoremei 3.2.2 din [6], distanța minimă este:

$$d_{min}=w \quad \text{or} \quad d_{min}=n-k+1.$$

Deoarece $d_{min}\ge 2t+1$ (după Hamming), avem $n-k+1=2t+1$ ori $n-k=2t$. De exemplu, pentru cazul trivial $k=2$, $t=1$ şi $m=2$ avem M-codul cu $n=4$. Una din matricele matroidului uniform asupra $\mathbf{GF}(2^2)$ este:

$$\mathbf{G}=\begin{bmatrix}1 & 0 & 1 & 1\\ 0 & 1 & 1 & 2\end{bmatrix}.$$

Încercarea de a construi M-codul de caracteristicile $(n, k, t)=(6, 2, 2)$ va eşua. Asupra câmpului $\mathbf{GF}(2^2)$ nu există matroid uniform de tipul $U_{2,6}$. "Resursele" câmpului s-au epuizat! Imediat apare întrebarea: care-s limitele de existență ale matroidelor uniforme $U_{k,n}$ cu $n=2t+k$ în spațiile vectoriale asupra câmpurilor $\mathbf{GF}(2^m)$?

III. Limitele de existență a codurilor matroide

În preambulul acestui articol a fost menționat că matroidele uniforme trebuie căutate printre cicloclasele generate în spațiul vectorial $\mathbf{GF}^k(2^m)$. Reamintim că cicloclasa **N** este un set de elemente conjugate ale câmpului Galois extins: $\mathbf{N}= \{N, N^2, N^4,\ldots\}$, unde $N\in \mathbf{GF}^k(2^m)$. Pentru căutarea matroidelor $U_{k,n}$ se vor folosi cicloclasele de lungimea $n= 2t+k$.

Dezvoltarea $\mathbf{N}_n$ este generată prin selectarea unui element (vector) nenul al câmpului $\mathbf{GF}^k(2^m)$. Cât selectarea elementelor generatoare ale cicloclaselor, atît şi căutarea cicloclaselor este rațional să se facă printre vectorii liniar independenți ai câmpului $\mathbf{GF}^k(2^m)$. Se poate arăta că numărul vectorilor liniar independenți în spațiul vectorial de dimensiunea k asupra câmpului $\mathbf{GF}(2^m)$ este egal cu

$$j(k,m)=\frac{2^{k\cdot m}-1}{2^m-1},$$

iar rangul maximal, $rank(U_{k,n})$, pe care îl poate atinge un matroid uniform, construit în spațiul $\mathbf{GF}^k(2^m)$, este determinat de mărimea:

$$n_{\max}=\frac{2^{2m}-1}{2^m-1} \quad \text{or} \quad n_{\max}=2^m+1. \tag{4}$$

Mărimea (4) delimitează "resursele" câmpului. Astfel, pentru $m= 2$ avem $n_{max}= (2^4-1) / (2^2-1)= 5$. Iar matroidul $U_{2,6}$ pentru construirea sa necesită nu mai puțin de 6 elemente!

Deci, se poate afirma că valoarea maximală pe care o poate atinge lungimea cuvîntului de cod în $\mathbf{GF}(2^m)$ este delimitată de inegalitatea:

$$n\le 2^m+1. \tag{5}$$

Pe de altă parte, între caracteristicile codului matroid n, k şi t există corespondența (2). Prin urmare, utilizatorul poate impune valori lui t şi k, verificând apoi inegalitatea (5). În tabelul 1 sunt prezentate valorile capacității corectoare ce pot fi atinse de codul matroid cu valoarea parametrului k asupra câmpului Galois de caracteristica 2^m.

Tabelul 1

Capacitatea corectoare t a codului matroid în dependență de k şi m

m \ k	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	…
2	1	1	0	0																	
3	3	3	2	2	1	1	0	0													
4	7	7	6	6	5	5	4	4	3	3	2	2	1	1	0	0					
5	15	15	14	14	13	13	12	12	11	11	10	10	9	9	8	8	7	7	6	6	…
6	31	31	30	30	29	29	28	28	27	27	26	26	25	25	24	24	23	23	22	22	…
7	63	63	62	62	61	61	60	60	59	59	58	58	57	57	56	56	55	55	54	54	…

Este uşor de observat că parametrul t îşi atinge valoarea maximală în cazul $k= 2$ şi-i egală cu 2^m-1. Codul matroid optimal, adică codul cu cel mai mare t şi cea mai mică redondanță $n-k$, va fi codul cu parametrii k şi t egali. În acest caz rata $k/n= 1/3$.

Relațiile (2) şi (5) delimitează granițele de existență a codurilor matroide, în particular, şi a codurilor liniare, în general. Astfel, în tabelul 3.1 din [6] sunt prezentate codurile Hamming asupra $\mathbf{GF}(2^2)$ cu parametrii $(n, k)= (5, 3)$, în $\mathbf{GF}(2^3)$ cu $(n, k)= (9, 7)$, în $\mathbf{GF}(2^4)$ cu $(n, k)= (17, 15)$. Aceasta se acordă perfect cu datele prezentate în tabelul 1. Însă existența codurilor liniare Hamming de capacitate corectoare arbitrară în câmpurile Galois extinse $\mathbf{GF}(2^m)$ a rămas o problemă nerezolvată pînă în prezent. În acest context codurile matroide reprezintă o alternativă codurilor liniare Hamming.

Codurile liniare sunt atractive prin simplitatea implementării "mecanismelor" de codare şi decodare. Dar ar fi binevenită o comparare între codurile matroide şi concurentul său "cel mai înverşunat", şi

anume, codul Reed-Solomon. Conform celor cunoscute (vezi, de exemplu [6]) parametrii codului RS sunt caracterizați de relațiile:

$$n\text{-}\,k = 2t \tag{6}$$

şi

$$n \le 2^m - 1. \tag{7}$$

Relațiile (2) şi (6) sunt într-o concordanță perfectă! Prin extrapolare se pot extinde rezultatele Teoremei 7.3.1 [6] şi asupra codurilor matroide. Aceasta înseamnă că nu numai codul Reed-Solomon are cea mai mare valoare a distanței minimale.

Din compararea inegalităților (5) şi (7) s-ar părea că codul matroid ar avea un avans față de codul Reed-Solomon. Însă introducerea codurilor RS extinse [6] echilibrează situația.

Deci, parametric codurile matroide şi codurile Reed-Solomon sunt identice?! Da, anume această concluzie rezultă din analiza scurtă făcută anterior. Atunci, are oare sens de a introduce un cod nou, dacă deja există un cod cu aceleaşi caracteristici ? Mai degrabă este o întrebare retorică. Este cert că orice "candidat" trebuie să aibă alternativă. Precum motoarele în patru tacte (pe benzină) au ca alternativă motoarele de tip Diezel (pe motorină).

Parametrii n, k şi t sunt indicatori importanți ai performanțelor codului corector. Alt aspect al analizei comparative vizează caracteristicile algoritmilor de codare şi decodare.

IV. Codarea codului matroid

Imediat, după definirea parametrilor codului matroid, se trece la generarea matroidului uniform respectiv $U_{k,n}$. Căutarea matroidului uniform în spațiul vectorial asupra câmpului Galois extins este o problemă de complexitate polinomială. În [5] a fost prezentată o metodă de căutare a matroidelor uniforme bazată pe testarea cicloclaselor generate de elementele primitive ale câmpului.

Matricea de control **H** a codului Reed-Solomon la fel constă din cicloclase generate de un element primitiv α al câmpului [6]:

$$\mathbf{H} = \begin{bmatrix} a^0 & a^1 & a^2 & \mathbf{K} & a^{n-1} \\ a^0 & a^3 & a^6 & \mathbf{K} & a^{3(n-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a^0 & a^{2t-1} & a^{(2t-1)2} & \mathbf{K} & a^{(2t-1)(n-1)} \end{bmatrix}. \tag{8}$$

De aceea, este logic de presupus că structura matricei de control de dimensiunea $k \times k$ poate să reprezinte o matrice a matroidului uniform asupra $\mathbf{GF}(2^m)$. Într-adevăr, numeroasele experimente au demonstrat că majoritatea matricelor de tipul (8), să le numim *clasice*, pot reprezenta un matroid uniform $U_{k,n}$.

Exemplu de matrice de control al RS-codului care nu este un matroid. Pentru M-codul (n, k, t)= (4, 2, 1) asupra $\mathbf{GF}(2^2)$ avem:

$$\mathbf{H} = \begin{bmatrix} a^0 & a^1 & a^2 & a^3 \\ a^0 & a^3 & a^6 & a^9 \end{bmatrix}.$$

Deoarece $\alpha^0=\alpha^3=\alpha^6=\alpha^9$, atunci

$$\mathbf{H} = \begin{bmatrix} a^0 & a^1 & a^2 & a^0 \\ a^0 & a^0 & a^0 & a^0 \end{bmatrix}. \tag{9}$$

În matricea (9) sunt doi vectori liniar dependenți (vezi prima şi ultima coloană).

În acest mod poate fi testată apriori orice matrice **H**. Dacă matricea de control **H** trece cu succes testul de "coincidență" a coloanelor, atunci se efectuează verificarea propriu-zisă a "candidatului" în matroid.

Matricea matroidului astfel obținută este "skeletul" coderului codului matroid. Coderul M-codului (sau M-coderul) realizează operația de înmulțire matriceală (1). În această operație matriceală sunt implicate operațiile de adunare şi înmulțire în câmpul Galois extins **GF**(2^m). Operația de adunare este binecunoscuta operație XOR bit-cu-bit. Operația de înmulțire se face *modulo* polinomul $p(x)$, unde $p(x)=\sum_{i=0}^{m} a_i x^i$ - un polinom ireductibil, $a_i \in \{0, 1\}$. Un factor al acestei operații rămîne constant. Aceasta permite de a optimiza cheltuielile de hard la realizarea M-coderului: multiplicatorul la constantă în câmpul Galois are o structură mai simplă decât multiplicatorul universal. În [7] este descrisă tehnica de generare a schemei multiplicatorului la constantă, dacă este cunoscut polinomul câmpului $p(x)$.

Exemplu. Fie k=2, n= 6 asupra **GF**(2^3) cu $p(x)$= 1+ x+ x^3. Matricea matroidului uniform este:

$$\mathbf{G}=\begin{bmatrix}1 & 3 & 5 & 4 & 7 & 2\\ 1 & 2 & 4 & 3 & 6 & 7\end{bmatrix}. \qquad (10)$$

Coderul M-codului realizează transformarea:

$$\mathbf{v}=\mathbf{x}\cdot\mathbf{G}=\langle x_1, x_2\rangle\begin{bmatrix}1 & 3 & 5 & 4 & 7 & 2\\ 1 & 2 & 4 & 3 & 6 & 7\end{bmatrix}$$

or

$$\begin{cases} x_1 + x_2 = v_1,\\ 3x_1 + 2x_2 = v_2,\\ 5x_1 + 4x_2 = v_3,\\ 4x_1 + 3x_2 = v_4,\\ 7x_1 + 6x_2 = v_5,\\ 2x_1 + 7x_2 = v_6. \end{cases} \qquad (11)$$

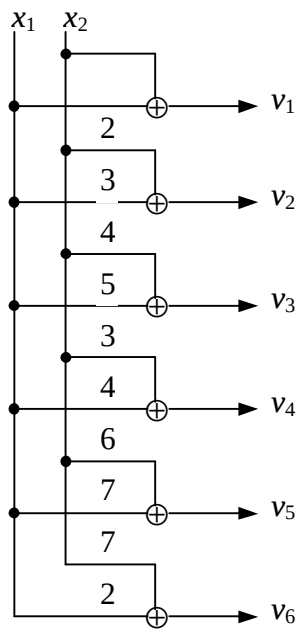


Fig.1. Bloc-diagrama M-coderului

În figura 1 este prezentată diagrama bloc a coderului codului matroid definit de sistemul de ecuații liniare (11). Semnul '+' specifică operația XOR, iar marcajul liniilor, adică cifrele de asupra liniilor, semnifică operația de înmulțire la coeficientul respectiv *mod* $p(x)$.

Din textul acestui compartiment rezultă că vectorii codului matroid sunt generați asincron, într-un singur tact. Comparați: cuvintele RS-codului, de regulă, sunt generate secvențial; numărul de tacte este egal cu lungimea codului n.

Schema de decodare a M-codului este mult mai complexă decât cea de codare. Structura ei depinde de algoritmul propus (ales). În celea ce urmează se va analiza unul din algoritmii de decodare a codului matroid.

V. Decodarea codului matroid

În momentul când matroidele uniforme se propuneau în calitate de bază constructivă pentru generarea codurilor liniare, se presupunea că, datorită proprietății lor excepționale, va fi suficient de avut k simboluri corecte din n recepționate pentru a restabili datele originale **x**. Da, aceasta este valabil, dacă decoderul apriori "ştie" valorile aşteptate ale celor k simboluri ! Dat fiind faptul că aceasta n-are loc, atunci decoderul trebuie să aibă un criteriu după care să ia decizia respectivă. În continuare va fi analizat principiul *decodării majoritare*, aplicat pentru codurile matroide.

Decodarea cu *logică majoritară* (sau *de prag*) se bazează pe un sistem de verificări care pot fi realizate separat sau integral. Logica majoritară conține un element de decizie (voter) care are un set de intrări şi o ieşire. Ieşirea voterului va fi în starea activă unu, dacă mai mult de jumătate de intrări (majoritatea) sunt activizate şi inactivă (starea zero) – în caz contrar.

Decoderul codului matroid are un sistem complex de verificări. Complexitatea este cauzată de numărul (volumul) de sisteme de ecuații liniare care trebuie să fie testate. Numărul total (exhaustiv) de ecuații este egal cu C_n^k. De exemplu, pentru k= 4 şi t= 4 avem $C_{12}^4 = 495$, iar pentru k= t= 8 - $C_{24}^8 = 735471$. Nu este rezonabil de efectuat verificarea exhaustivă ! Care atunci este numărul necesar de testări (încercări) pentru a lua o decizie adecvată ?

Reamintim că la început decoderul trebuie să recunoască combinația eronată. Iar o combinație eronată poate să conțină pînă la t simboluri eronate. S-ar părea că sarcina decoderului se complică semnificativ: el trebuie să recunoască erorile singulare, duble, triple etc. Însă situația nu-i atît de drastică precum s-ar părea la prima vedere. În acest context cităm sursa [8]:"…este suficient de avut grijă de erorile t-uple, deoarece oricare combinație, ce constă dintr-un număr mai mic de erori, la fel va fi acoperită (detectată)" (v. pag. 109). Acceptînd această ipoteză, decidem că decoderul M-codului trebuie să găsească soluția adecvată pentru eroarea cea mai gravă, adică pentru eroarea t-uplă.

Un vector **v** de lungimea n generează un set de C_n^k sisteme de ecuații liniare. De exemplu, din mulțimea de ecuații (11) pot fi compuse $C_6^2 = 15$ sisteme de ecuații liniare. Dacă vectorul recepționat **v′** este egal cu cel transmis **v**, adică **v′**= **v**, atunci toate celea 15 soluții vor fi identice. Dacă e vorba de M-codul (12, 4, 4), atunci verificarea celor 495 soluții devine anevoioasă. Câte sisteme de ecuații este necesar şi suficient de rezolvat pentru a lua decizia: soluția găsită este combinația corectă (originală) ?

Căutarea răspunsului la întrebarea pusă vom începe cu analiza mecanismului de recunoaştere a erorii duble în vectorul recepționat al M-codului (6, 2, 2) . Avem 6 ecuații liniare în (11). Numerotăm aceste ecuații de la 1 la 6. Precum s-a menționat anterior, pot fi generate 15 sisteme de ecuații liniare, şi anume, S= {1,2; 1,3; 1,4; 1,5; 1,6; 2,3; 2,4; 2,5; 2,6; 3,4; 3,5; 3,6; 4,5; 4,6; 5,6}.

Tabelul 2

Diagrama încercărilor sistemelor de ecuații liniare pentru erorile duble în codul matroid (6,2,2)

Erori duble	Sisteme de ecuații liniare														
	1,2	1,3	1,4	1,5	1,6	2,3	2,4	2,5	2,6	3,4	3,5	3,6	4,5	4,6	5,6
(1,2)										1	1	1	1	1	1
(1,3)							1	1	1				1	1	1
(1,4)						1		1	1		1	1			1
(1,5)						1	1		1	1		1		1	
(1,6)						1	1	1		1	1		1		
(2,3)			1	1	1								1	1	1
(2,4)		1		1	1						1	1			1
(2,5)		1	1		1					1		1		1	
(2,6)		1	1	1						1	1		1		
(3,4)	1			1	1			1	1						1
(3,5)	1		1		1		1		1					1	
(3,6)	1		1	1			1	1					1		
(4,5)	1	1			1	1			1			1			
(4,6)	1	1		1		1		1			1				
(5,6)	1	1	1			1	1			1					

În vectorul recepționat îs posibile $C_n^t = C_6^2 = 15$ erori duble. Dacă numerotăm pozițiile simbolurilor în vectorul **v** de la 1 la 6, atunci, evident, mulțimea perechilor de poziții eronate *E* va coincide cu mulțimea *S*. Asupra mulțimilor *E* şi *S* definim aplicația ϕ: $E \rightarrow S$ care pune în corespundere fiecărei erori setul de sisteme de ecuații care trebuie să genereze soluții identice corecte. De exemplu, pentru perechea eronată (1,2) setul de ecuații {3,4; 3,5; 3,6; 4,5; 4,6; 5,6} este seul de încercări care defineşte pragul de decizie: soluții identice – eroarea este recunoscută, în caz contrar – trierea variantelor continuă.

Pentru reprezentarea aplicației ϕ poate fi construit un tabel , în care liniile corespund erorilor *t*-uple, iar coloanele – sistemelor de ecuații liniare. Tabelul este completat în modul următor: la intersecția liniei şi coloanei în pătrățel se înscrie unitatea, dacă sistemul de ecuații, indicat în coloana respectivă, se foloseşte la identificarea erorii specificate în linie. În tabelul 2 este prezentată aplicația ϕ analizată.

Tabelul 2 poate fi folosit pentru optimizarea numărului de sisteme de ecuații rezolvate. Se observă că la trecerea de la o linie la alta rezultatele obținute la pașii precedenți pot fi folosite în pasul curent. Astfel, pentru analiza prezenței perechii de erori (1,3) vor fi rezolvate numai 3 ecuații noi, şi anume, {2,3; 2,5; 2,6}, iar celelalte soluții vor fi luate din pasul precedent, adică obținute la verificarea sistemelor de ecuații corespunzătoare perechii (1,2).

O analiză generală a procesului de testare a sistemelor de ecuații liniare pentru erorile *t*-uple duce la următoarea concluzie: ***pentru luarea deciziei corecte*** într-o încercare este necesar de rezolvat un număr de

$$N_x(k,t) = C_{n-t}^k \qquad (12)$$

sisteme de ecuații liniare, iar numărul total de încercări la care se poate ajunge în procesul de testare (verificare), este egal cu

$$N_\Sigma(k,t) = C_n^t . \qquad (13)$$

Mărimea (13) caracterizează complexitatea algoritmului de decodare a codului matroid.

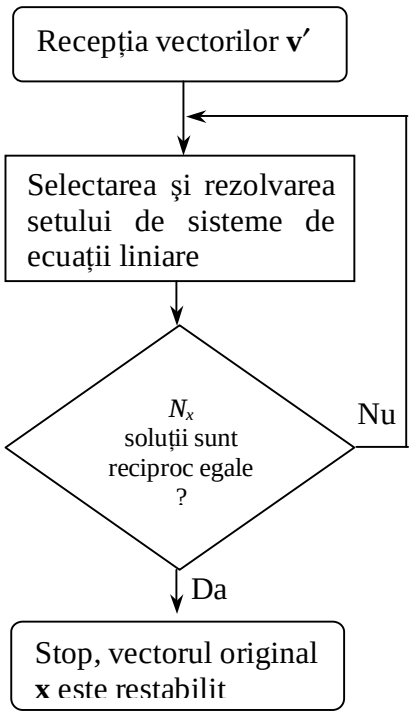


Fig. 2. Diagrama algoritmului M-decodării

Valorile caracteristicilor (12) şi (13) pentru codurile matroide (12, 4, 4) şi (24, 8, 8) vor fi respectiv $N_x(4, 4)= 70$, $N_\Sigma(4, 4)= 495$ şi $N_x(8, 8)= 12870$, $N_\Sigma(8, 8)= 735471$.

Din celea analizate mai sus rezultă schema algoritmului de decodare a codului matroid. În figura 2 este prezentată diagrama acestui algoritm. Trebuie menționat că pentru rezolvarea sistemelor de ecuații liniare vor fi utilizate scheme de tipul celor prezentate în figura 1. Alt aspect al decodării este numărul mare de încercări N_Σ care îl poate necesita corecția unei erori *t*-uple. (Însă nici decodarea Reed-Solomon nu-i mai simplă !)

VI. Softul experimental şi rezultatele experimentelor de statistică

În conformitate cu cele analizate anterior a fost realizat softul experimental al codecului matroid. Acest soft citeşte conținutul fişierelor textuale şi generează textul codificat. Interfața softului este prezentată în figura 3.

La tastarea butonului **Load** apare o fereastră de dialog standard (vezi fig. 4), prin intermediul căreia are loc încărcarea fişierului textual în componenta *Memo*, situată în colțul din stînga-sus al ferestrei. Tastînd butonul **Encode**, textul inițial (original) se codifică într-un text de ieşire, care se înscrie în componenta *Memo* din stînga-jos a interfeței (vezi fig. 5).

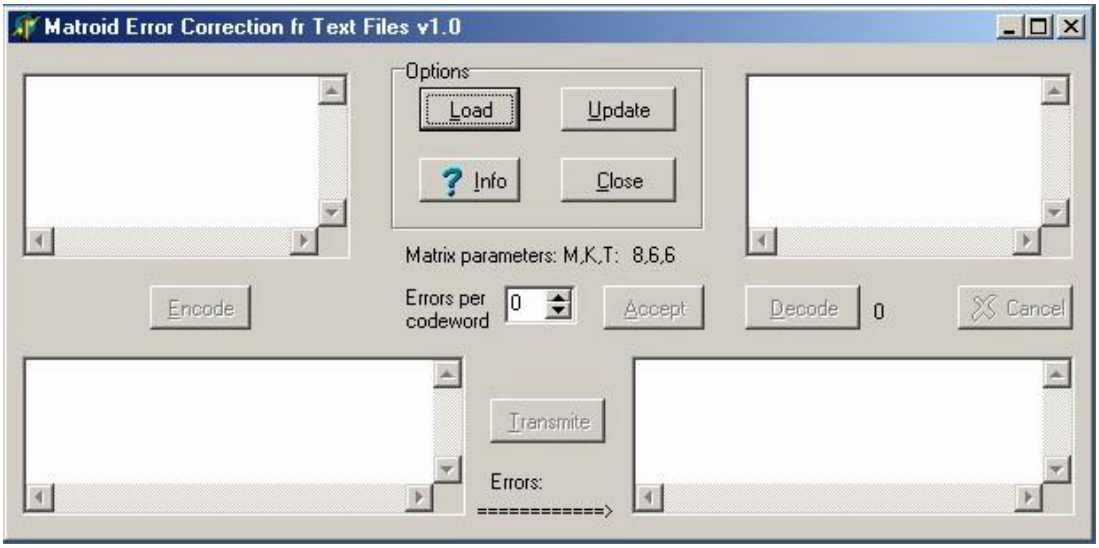


Fig. 3. Interfața codecului soft experimental.

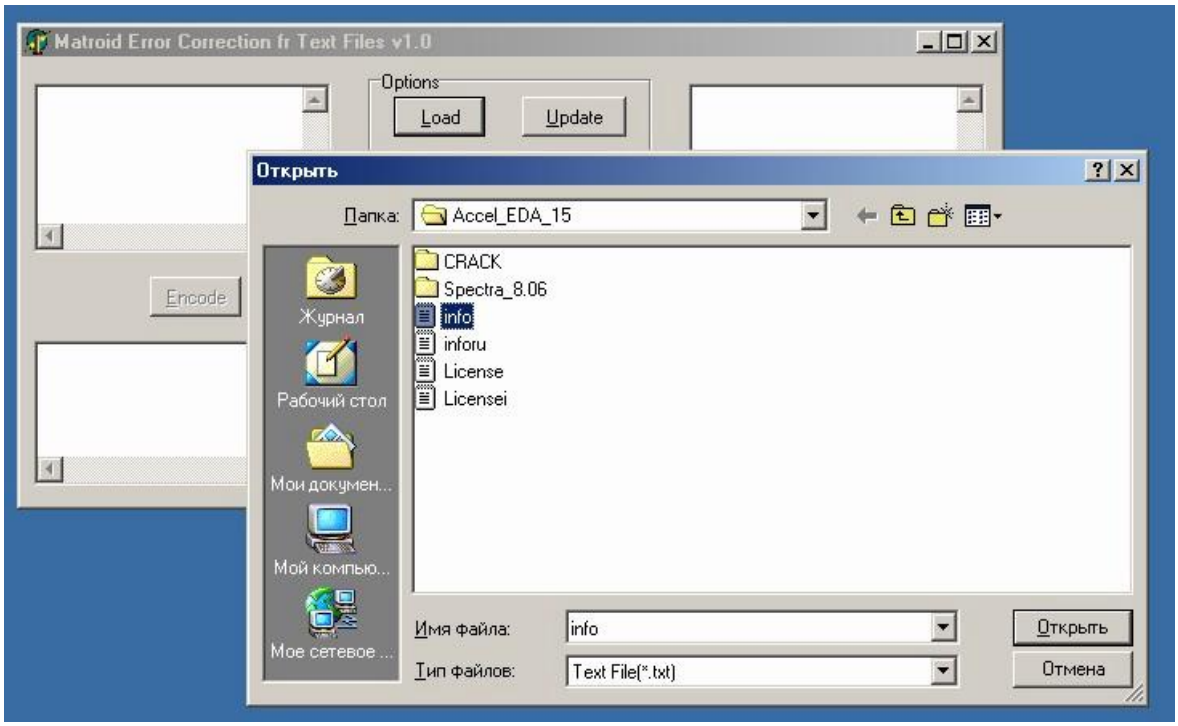


Fig.4. Incarcarea fişierului textual inițial.

Codificarea are loc conform structurii matroidului download-at cu ajutorul tastei **Update**. În experimentele de analiză statistică-informațională a fost folosit matroidul cu arametrii $(n, k, t)=(18,6,6)$ asupra cîmpului **GF**(2^8). Textul codificat este apoi transcris în procesorul textual MS Word cu ajutorul căruia se alcătuieşte alfabetul secundar şi frecvența (absolută) literelor. Ca exemplu în calitate de text inițial a fost selectat următorul aliniat:

” Accel-EDA Version 15.0 CRACKED 09-03-99 Win32
ACCEL EDA ... a complete tool box for electronic design. From conception
to design archive, ACCEL EDA covers the range of capability required for

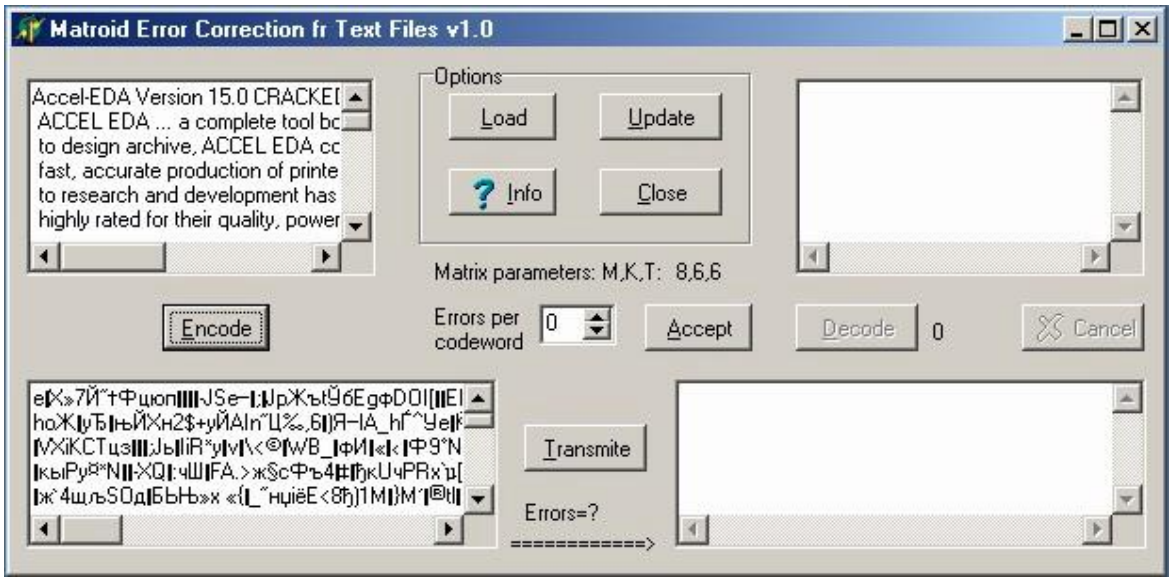


Fig. 5. Codificarea textului inițial.

fast, accurate production of printed circuit boards. ACCEL's commitment to research and development has resulted in a suite of design tools highly rated for their quality, power, and ease-of-use.

ACCEL EDA Version Summary of V15 Features is now available. Check out the information about this major release you'll find dozens of NEW features and enhancements packed into this update!

Installation Notes:

Unzip, Unrar, then install normally the program.

The program is fully repackaged and already cracked, I decided to do not include crack, because it is really complex, these are the motivations.

1) All executables files are shrinked, and must be deshrinked before any kind of path crack, and this is not a easy things for not skilled people. To do it I used procdump V1.4 using Deshrink V3.x script.
2) During installation a large amount of serial number is requested.
3) Many licenses file are needed after installation.

The already cracked program have this benefit:

1) During install all serial number / password / license key are already typed you need only to press continue->.
2) After install all executable are already patched and 100% working.
3) All licenses files xxxx.lic are already in program dir after install.

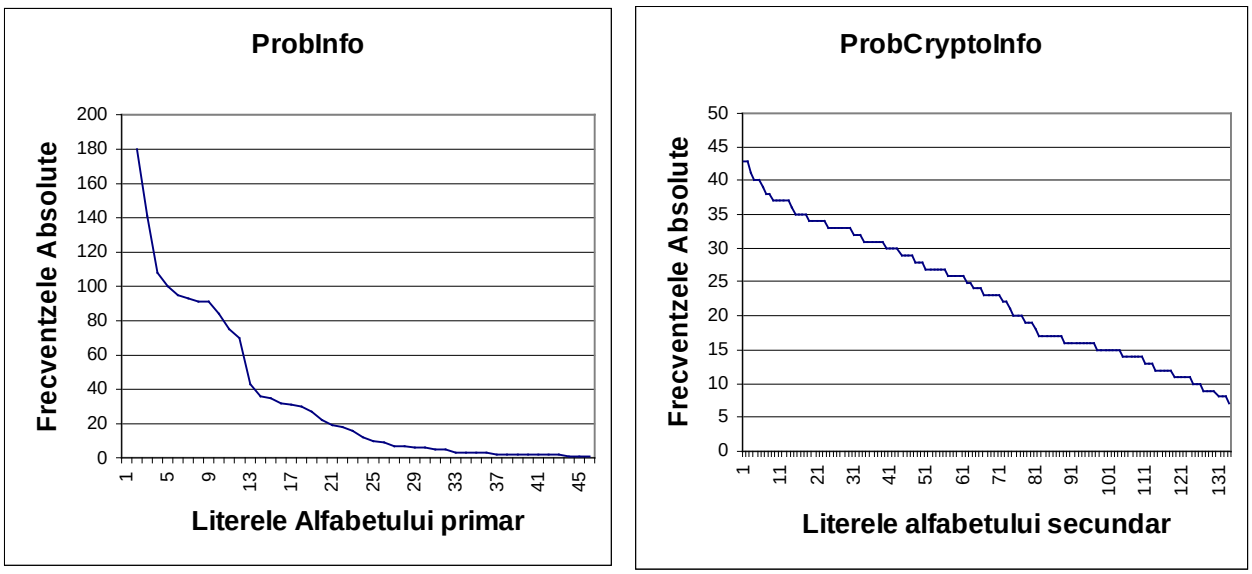
For make this program 100% great it need SPECCTRA router installed. Specctra V8.0.6 is previously released by RBS some mount ago, in any case you can get it here:

ftp.acceltech.com\pub\download\SPECCTRA\SPECCTRA806.EXE and use the included license.dat as valid license.

\blastsoft

Thanks fly out to Blast Soft on yet another great program that is cracked. P.S. Have a Nice vacation blastie you earned it.”.

Fig. 6. Textul codificat.



a) b)

Fig. 7. repartiția frecvențelor literelor alfabetului primar pînă la codificarea matroidală (a) şi a celui secundar – după codificare (b).

Pentru textul selectat „matroidul” a generat mesajul codificat, prezentat în figura 6. Pentru ambele texte au fost alcătuite alfabetele respectiv primar şi secundar, pentru care s-au estimat repartițiile frecvențelor literelor. În figura 7 sînt prezentate graficile repartițiilor frecvențelor literelor alfabetului primar şi secundar. Se observă uşor că repartiția literelor alfabetului secundar este mai aplatizată decât repartiția alfabetului primar, adică este o repartiție mai aproape de cea echiprobabilă!

Cantitativ această afirmare poate fi estimată foarte simplu – prin calculul redondanțelor informaționale ale textelor analizate. Pentru aceasta trebuie de calculat entropiile repartițiilor literelor alfabetelor primar şi secundar, care sînt respectiv egale cu:

H_{prim}= 4,5 [bit] şi H_{sec}= 6,94 [bit].

Redondanța μ se va calcula după formula binecunoscută: $\mu = (H^* - H)/ H^*$, unde: $H^* = \mathrm{Log}_2 L$ – valoarea maximală a entropiei, L – numărul de litere in alfabet.

Avem H^*_{prim}= 5,52 [bit] şi H^*_{sec}= 7,07 [bit]. Deci, μ_{prim}= 0,185 şi μ_{sec}= 0,018; câştigul este de aproape 10,3 ori!

Astfel, codarea matroidală a permis micşorarea redondanței informaționale mai bine de 10 ori, ceea ce ridică, cel puțin, de 10 ori siguranța criptării.

Bibliografia

[1] Шеннон К.Э. *Теория связи в секретных системах* // В кн.: Шеннон К.Э. Работы по теории информации и кибернетике. М.: ИЛ, 1963. – С. 333-402.

[2] Shannon C.E. *Communication theory of secrecy systems*, Bell System Technical Journal, Vol. 28, 1949, P.656-715.

[3] V.Borshevich, W.Oleinik. *A new approach to information coding and protection based on the theory of matroids* // Computer Science Journal of Moldova, vol.2, N.1, 1994. - pp.113-116.

[4] В.Борщевич, В.Олейник. *Матроидные коды – новый подход к защите информации в компьютерных системах* //Acta Academia, 1997. – pp. 40-50.

[5] G.Bodean. *O metodă de construirea codului corector de pachete de erori* //Acta Academia, 2002. – pp. 77-85.

[6] Р. Блейхут. *Теория и практика кодов, контролирующих ошибки*. – М.: Мир, 1986.

[7] G.Bodean. *Coding and decoding of the matroid codes* // ELECO 03: The 3d International Conference on Electrical and Electronics Engineering, Bursa, Turkey, 2003. – 5 p.

[8] Дж. Кларк, Дж. Кейн. *Кодирование с исправлением ошибок в системах цифровой связи*. – М.: Радио и связь, 1987.

Informație despre autor

Larisa DUNAI, lector asistent
Universitatea Tehnică a Moldovei, catedra Construirea şi Producerea Aparatajului Electronic
MD2012, Kishinau, Stefan Cel Mare, 168
email: laryka@mail.md