STUDIA UNIVERSITATIS
Revistă științifică a Universității de Stat din Moldova, 2012, nr.7(57)

MATRICELE DE RELAȚII MULTI-ARE ŞI NUMERELE PRIME ÎN CRIPTAREA INFORMAȚIEI

Sergiu CATARANCIUC, Aureliu ZGUREANU*

Universitatea de Stat din Moldova
**Academia de Transporturi, Informatică şi Comunicații*

Se propune un sistem simetric de criptare a informației, numit *Crypto 2*. Acest sistem se bazează pe utilizarea numerelor prime mari şi pe dezvoltarea lor polinomială. Un rol deosebit revine matricelor multidimensionale, folosite pentru reprezentarea mulțimilor de relații multi-are. Se descrie algoritmul de criptare/decriptare a informației, Cripto 2, şi se demonstrează că acest algoritm are o complexitate liniară.

Cuvinte-cheie: *securitatea informației, metode de criptare, relații multi-are, numere prime, matrice multidimensională, complexitatea algoritmului.*

MULTI-ARY RELATIONS MATRIXES AND PRIME NUMBERS IN ENCRYPTING OF INFORMATION

It is proposed a symmetric system for encryption information, called *Crypto 2*. This system is based on using large prime numbers and their polynomial development. A special role lays multidimensional arrays which represent the sets of multi-ary relations. The encryption/decryption information algorithm for the system Cripto 2 is described. This algorithm has linear complexity.

Keywords: *information security, encryption methods, multi-ary relations, prime numbers, multidimensional matrix, algorithm complexity.*

Printre problemele de ordin superior din secolul informaticii se înscrie problema privind protecția informației şi lupta contra spărgătorilor sistemelor de criptare. Elaborarea unor metode eficiente de criptare are drept scop reducerea accesului neautorizat la informație. La baza elaborării unor sisteme de criptare se află numerele prime mari [1]. Descifrarea mesajelor devine o problemă complicată în cazul în care numerele indicate sunt mari, deoarece ea este relaționată cu problema factorizării numerelor. Cu cat mai mari sunt aceste numere, cu atât mai dificilă devine problema spargerii cifrului bazat pe utilizarea numerelor prime. Majoritatea metodelor de generare a numerelor mari prime sunt probabilistice [2]. În acest scop se folosesc teste probabilistice de primalitate.

Unul dintre astfel de teste este testul Fermat [1, 3], care are la bază mica teoremă Femat: *pentru n >1 se alege a >1 şi se calculează* a^{n-1} *mod n; dacă rezultatul este diferit de 1, atunci numărul n este compus, în caz contrar n este considerat număr pseudoprim*. Însă, acest test poate considera un număr compus ca fiind prim (în special numerele Carmichael, care sunt compuse, iar testul Fermat le consideră prime indiferent de numărul de iterații). Probabilitatea erorii în testul Fermat este de ε^t, unde *t* reprezintă numărul de iterații ale testului şi $\varepsilon \le \frac{\varphi(n)}{n}$, unde $\varphi(n)$ este funcția Euler. Pentru realizarea acestui test se recomandă de a efectua circa 50 de iterații. În prezent, testul Fermat în forma sa originală nu se mai foloseşte, el nu are un control al erorii, dar poate fi utilizat în fazele incipiente la verificarea primalității unor numere foarte mari.

Testul Solovay–Strassen [3, 5] este un alt test de primalitate şi se bazează pe diferența dintre simbolurile Jacoby şi Legendre, aceste simboluri pentru numerele prime *n* fiind identice. Probabilitatea erorii în testul Fermat este de ε^t, unde *t* reprezintă numărul de iterații ale testului, iar $\varepsilon \le \frac{\varphi(n)}{2n} < \frac{1}{2}$. În scopul utilizării acestui test, în criptografie este necesar de a lua o valoare suficient de mare a lui *t*, aproape de 100. În acest caz, şansa de a obține un număr compus este atât de mică încât putem aplica aceste numere în scopuri criptografice. De menționat că acest algoritm determină numerele Carmichael ca fiind compuse.

Un test mai performant este testul Miller-Rabin [3, 4], cel mai utilizat test de generare a numerelor pseudoprime pentru algoritmii de criptare. De obicei, pentru verificarea unui număr prim, folosind testul Miller-Rabin, este necesară o singură iterație (numărul recomandabil de iterații este 5). Probabilitatea erorii în testul

Miller-Rabin este considerabil mai mică decât în primele două teste şi la o iteraţie ea este $\varepsilon \le \frac{\varphi(n)}{4n} < \frac{1}{4}$. Adică, limita de sus a erorii la o singură iteraţie este de 2 ori mai mică decât la testul Solovay–Strassen şi de 4 ori mai mică decât la testul Fermat. Astfel, testul Miller-Rabin este mai eficient decât celelalte astfel de teste.

Cu toate că testul Miller-Rabin este considerat suficient de bun pentru generarea numerelor prime aplicate în criptografie, există o foarte mică probabilitate ca acest test să genereze un număr compus.

Această situaţie ne face să căutăm algoritmi eficienţi de criptare a informaţiei, bazaţi pe alte principii şi idei. Printre astfel de algoritmi se numără algoritmul *Cripto* 2 [6-8], care este un algoritm *simetric* ce funcţionează în baza numerelor prime mari şi a mulţimilor de relaţii multi-are. Algoritmul este alcătuit din:

- **generator de chei,** care construieşte numerele mari prime $y_1,...,y_a$ (cheile private) cu ajutorul generatorului descris în [7], mărimea lor fiind în funcţie de cerinţele concrete:
  a) securitatea sporită – necesită chei mai mari;
  b) viteza mai mare de prelucrare a informaţiei – necesită chei mai mici;
  c) volumul de informaţie la fel necesită diferite lungimi de chei;
  d) modul de funcţionare a sistemului – dacă se utilizează *a* chei, ele pot fi mai mici, dacă mai puţine sau una singură, acestea trebuie să fie mai mari etc.

Toate aceste cerinţe sunt interdependente;

- **codificator,** care construieşte cortegiul $\overline{c}$ în baza coordonatelor cortegiului $\overline{m}$, a parametrilor *a*, *b*, *t* şi cheilor private $y_1,...,y_a$. Acestea se criptează aplicând unul din sistemele sigure de criptare;
- **decodificator,** care construieşte cortegiul $\overline{m}$ în bază cortegiilor $\overline{c} = (c_1,...,c_i,...,c_u)$ şi $\overline{y} = (y_1,...,y_a)$, decodifică cortegiul $\overline{m}$ şi imprimă textul decodificat, restabilindu-l după codurile ASCII obţinute în $\overline{m}$.

Să examinăm un caz particular al relaţiilor pe mulţimi. Fie o familie de mulţimi $X = \{X_1,\ X_2,\ ...,\ X_n\}$, unde $X_i = \{x_{i1},\ x_{i2},...,x_{i\lambda_i}\}$, $i = \overline{1,n}$, şi o mulţime $\Omega = \{\omega_1,...,\omega_r\}$ cu elemente arbitrare (în cazul nostru – numere întregi). Definim *k* relaţii

$$R_j = R_{X_{j_1}....X_{j_{d_j}}}\ (2 \le d_j \le n,\ j = \overline{1,\ k},\ j_1,\ j_2\ ,...,\ j_{d_j} \in \{1,\ 2,\ ...,\ n\}),$$

ca submulţimi ale produselor carteziene $X_{j_1} \times X_{j_2} \times ... \times X_{j_{d_j}}$. Matricele acestor relaţii sunt matrice d_j- dimensionale cu elemente din mulţimea Ω. Notăm cu $\overline{R}$ cortegiul cu componentele R_j, adică $\overline{R} = (R_1,...,R_j,...,R_k)$. Acestui cortegiu îi punem în corespondenţă o matrice *n*-dimensională $A_R = \Phi(\overline{R})$, ale cărei elemente sunt notate cu $a_{s_1...s_\tau...s_n}$. Să explicăm obţinerea acestor elemente.

Construim produsul cartezian

$$X_1 \times X_2 \times ... \times X_n = \{x_{11},...,x_{1\lambda_1}\} \times ... \times \{x_{n1},...,x_{n\lambda_n}\},$$

care, evident, conţine $u = \lambda_1 \cdot ... \cdot \lambda_n$ elemente. Cu aceste elemente compunem o matrice formată din *u* linii şi *n* coloane, ce corespunde familiei de mulţimi *X* (Fig.1, partea stângă).

$$A_R = \begin{array}{c} \\ 1 \\ \vdots \\ i \\ \vdots \\ u \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & \cdots & X_\tau & \cdots & X_n \end{array} \\ \begin{pmatrix} x_{11} & \cdots & x_{\tau 1} & \cdots & r_{n1} \\ & & \ddots & & \\ x_{1s_1} & \cdots & x_{\varpi_\tau} & \cdots & x_{ns_n} \\ & & \ddots & & \\ x_{1\lambda_1} & \cdots & x_{\tau\lambda_\tau} & \cdots & x_{n\lambda_n} \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{ccccc} R_1 & \cdots & R_j & \cdots & R_k \end{array} \\ \begin{pmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1k} \\ & & \ddots & & \\ r_{i1} & \cdots & r_{ij} & \cdots & r_{ik} \\ & & \ddots & & \\ r_{u1} & \cdots & r_{uj} & \cdots & r_{uk} \end{pmatrix} \end{array}$$

Fig.1. Reprezentarea matricei n-dimensionale, corespunzătoare cortegiului $\overline{R}$.

Compunem o altă matrice bidimensională $\| r_{ij} \|$ cu *u* linii şi *k* coloane care corespunde relațiilor din $\overline{R}$ (Fig.1, partea dreaptă), unde $r_{ij} = r_{s_{j_1} \ldots s_{j_d}}$, cu elementele s_{j1}, ..., s_{j_d} selectate în linia *i* pe locurile j_1, ..., j_d. Aceste locuri indică mulțimile X_{j_1}, ..., X_{j_d} pe care este definită relația R_j.

Pentru simplitate substituim elementele x_{ϖ_τ} cu indicele respectiv s_τ, aşa cum e arătat în Figura 2. Liniile matricei din partea stângă în Figura 2 reprezintă indicii elementelor matricei A_R. Liniile matricei din partea dreaptă formează elementele matricei A_R:

$$a_{s_1 \ldots s_\tau \ldots s_n} = (r_{i1}, \ldots, r_{ij}, \ldots, r_{ik}). \tag{1}$$

Cortegiului (1) i se asociază un număr c_i în baza *y,* care verifică condiția $y > \max \omega_h$, $h = \overline{1,r}$:

$$c_i = r_{i1} y^{k-1} + \ldots + r_{ij} y^{k-j} + \ldots + r_{ik} = \sum_{j=1}^{k} r_{ij} y^{k-j}, \; i = \overline{1,u}. \tag{2}$$

Astfel, obținem cortegiul

$$\overline{c} = (c_1, \ldots, c_i, \ldots, c_u). \tag{3}$$

$$A_R = \begin{array}{c} \\ 1 \\ \vdots \\ i \\ \vdots \\ u \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & \cdots & X_\tau & \cdots & X_n \end{array} \\ \begin{pmatrix} 1 & \cdots & 1 & \cdots & 1 \\ & & \ddots & & \\ s_1 & \cdots & s_\tau & \cdots & s_n \\ & & \ddots & & \\ \lambda_1 & \cdots & \lambda_\tau & \cdots & \lambda_n \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{ccccc} R_1 & \cdots & R_j & \cdots & R_k \end{array} \\ \begin{pmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1k} \\ & & \ddots & & \\ r_{i1} & \cdots & r_{ij} & \cdots & r_{ik} \\ & & \ddots & & \\ r_{u1} & \cdots & r_{uj} & \cdots & r_{uk} \end{pmatrix} \end{array} \begin{array}{c} c \\ \begin{pmatrix} c_1 \\ \vdots \\ c_i \\ \vdots \\ c_u \end{pmatrix} \end{array}$$

Fig.2. Reprezentarea simplificată a matricei n-dimensionale ce corespunde cortegiului $\overline{R}$.

Astfel, prin transformarea $A_R = \Phi(\overline{R})$, cortegiul $\overline{c}$ (2), (3) este pus în corespondență cortegiului $\overline{R}$. Transformarea inversă:

$$\overline{R} = \Phi^{-1}(\vec{c}). \tag{4}$$

În unele cazuri particulare putem găsi coordonatele vectorului cortegiului $\overline{R}$ după coordonatele cortegiului $\overline{c}$ (acest lucru a fost realizat la investigarea distribuției numerelor prime în mulțimea numerelor naturale, în rezultat fiind elaborat un algoritm de generare a numerelor prime).

Dacă transformarea (4) este dificilă, putem folosi acest lucru la elaborarea sistemelor de criptare a informației.

Fie că trebuie să cifrăm coordonatele cortegiului $\vec{m} = (m_1, m_2, \ldots, m_t)$, care reprezintă codificarea numerică a unui text dat în unul din sistemele de codificare dorite (fie ASCII). Considerăm coordonatele acestui cortegiu ca fiind elemente ale mulțimii Ω. În aşa caz, aceste coordonate trebuie sa fie elemente ale matricelor R_j. Dacă coordonatele vectorului nu se repetă conform unei legi determinate, atunci toate matricele A_{R_j} trebuie să fie de ordinul *n*.

Fie

$$t = a \cdot b \tag{5}$$

Atunci matricea A_R poate avea forma din Figura 3, unde $n = \lceil \log_2 a \rceil$, $k = b$, iar coordonatele cortegiului $\overline{c}$ se calculează după formula:

$$c_i = \sum_{j=1}^{b} m_{(i-1)b+j} \cdot y_i^{b-j}, \; i = \overline{1,a}, \; y_i > \max \omega_h, \; h = \overline{1,t}.$$

Descrierea algoritmului *Cripto* 2

Fie ***T*** un text arbitrar ce conţine ***N*** caractere (inclusiv spaţiile libere). Numărul ***N*** se numeşte lungime a textului ***T***. Vom considera că textul ***T*** este reprezentat în codurile *ASCII*. Prin urmare, textul ***T*** poate fi privit ca un cortegiu numeric de lungime ***N***:

$$T = (m_1, ..., m_i, ..., m_N).$$

Fie acest text trebuie criptat cu ajutorul unui set de chei ***Y***. În aceste condiţii criptarea textului ***T*** poate fi efectuată în modul următor:

Pasul 1. Reprezentăm numărul *N* ca un produs de două numere $N=a\cdot b$. În acest scop e suficient a lua în calitate de *a* primul divizor mai mare decât unu al numărului *N*.

Pasul 2. Formăm matricea multidimensională A_R şi o reprezentăm cu ajutorul unei matrice bidimensionale [8] cu numărul de linii *a* şi numărul de coloane *b*. Elementele matricei A_R în formă bidimensională se calculează după formula:

$$A_R(i, j) = m_{(i-1)\cdot b+j}.$$

Această matrice conţine *N* elemente ($N=a\cdot b$).

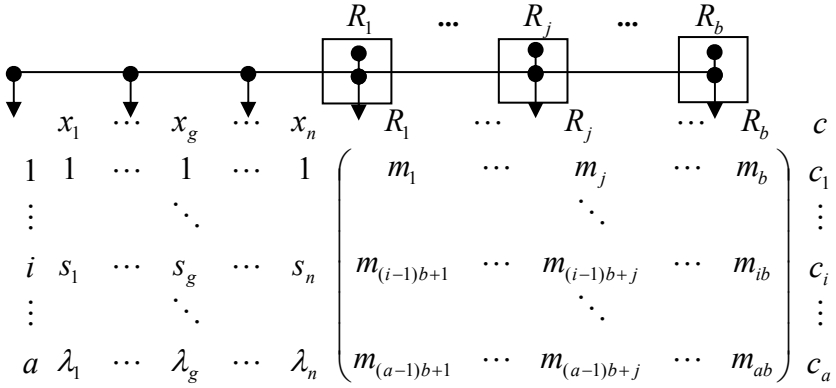

Fig.3. O formă de reprezentare a matricei A_R.

Pasul 3. Se generează setul de chei $Y = (y_1, y_2, ..., y_l),$ unde $l \le a.$

Pentru acesta folosim generatorul de numere prime „BUZGUCIBI PRIM 1" [7].

Pasul 4. Pentru fiecare linie *i* din *A* se calculează un număr c_i în conformitate cu formula

$$c_i = \sum_{j=1}^{b} m_{(i-1)b+j} \cdot y_i^{b-j},\ i = \overline{1,a},\ y_i > \max \omega_h,\ h = \overline{1,t}. \tag{6}$$

Scrisă în formă desfăşurată, obţinem sistemul de ecuaţii:

$$\begin{cases} c_1 = m_1 y_1^{b-1} + m_2 y_1^{b-2} + ... + m_b \\ c_2 = m_{b+1} y_2^{b-1} + m_{b+2} y_2^{b-2} + ... + m_{2b} \\ c_3 = m_{2b+1} y_3^{b-1} + m_{2b+2} y_3^{b-2} + ... + m_{3b} \\ \cdots \\ c_a = m_{(a-1)b+1} y_a^{b-1} + m_{(a-1)b+2} y_a^{b-2} + ... + m_{ab} \end{cases}, \tag{7}$$

unde $y_1, y_2, ..., y_a$ sunt cheile private. După cum s-a menţionat, numărul cheilor poate varia de la 1 până la *a*.

Pasul 5. Stop. Cortegiul $C = (c_1, ..., c_i, ..., c_a)$ conţine textul iniţial *T* în varianta criptată.

Remarcă. Dacă numărul *N* este prim, atunci la textul iniţial *T* adăugăm un simbol oarecare pentru a obţine text de lungime, care nu se exprimă printr-un număr prim.

Din descrierea algoritmului observăm că numărul de linii *a* şi numărul de coloane *b* ale matricei A_R se estimează prin $O(N)$.

Pentru a găsi reprezentarea numărului N ca produs a două numere a şi b ($N=a\cdot b$) e suficient să alegem în calitate de a primul divizor al lui N. Pentru aceasta sunt necesare $O(N)$ operații.

Realizarea pasului 2 se face în timp constant $O(1)$ şi nu poate depinde de lungimea N a textului T. Calcularea matricei A_R, evident, necesită $O(N)$ operații (se calculează $a\cdot b$ elemente ale matricei). Generarea unei chei se face prin intermediul algoritmului „BUZGUCIBI PRIM 1", a cărui complexitate nu depinde de lungimea textului T. Deci, generarea unei chei se face în timp constant $O(1)$. Pentru criptarea textului, algoritmul *Cripto* 2 foloseşte un set de chei $Y=(y_1,y_2,...,y_l)$, $l\le a=O(N)$. Prin urmare, generarea celor l chei se face în timp $O(N)$.

Menționăm că, deoarece $a\ge 2$, setul Y ar putea fi limitat la un număr fixat de chei, de exemplu $l=2$ (sau $l=3$), ceea ce ar însemna că timpul necesar pentru generarea setului de chei nu depinde de N.

Pentru a obține textul criptat, la pasul 5 se calculează cortegiul de numere $C=(c_1,...,c_i,...,c_a)$, pentru care se folosesc $O(N)$ operații. Astfel, obținem estimarea complexității algoritmului *Cripto* 2, adică este adevărată următoarea

Teoremă. *Algoritmul Cripto 2 rezolvă problema criptării/decriptării unui text T de lungimea N în timp O(N).*

Referințe:

1. Adleman L., Pomerance C. and Rumley R., On Distinguishing Prime Numbers from Composite Numbers // Annals of Mathematics, 1983, no.117 (1), p.173-206.
2. Koblitz N.A. Course in Number Theory and Cryptography. - Springer-Verlag, 1987.
3. Kranakis E. Primality and Cryptography. - Stuttgart: Teubner, Wiley, 1986.
4. Minuț P. Teoria numerelor. Vol.1. - Iași: Crenguța Gâldău, 1997.
5. Pomerance C. Recent Developments in Primality Testing // The Mathematical Intelligencer, 1981, vol.3, no.3, p.97-105.
6. Zgureanu A., Cataranciuc S. Encryption systems based on multidimensional matrixes. „Tiberiu Popoviciu seminar", Cluj-Napoca 6-7 september, 2010, p.99-110.
7. Zgureanu A. Securitatea informațională şi metode de criptare bazate pe mulțimi de relații multi-are: Teză de doctor în științe fizico-matematice. - Chişinău, 2011.
8. Булат М.С., Згуряну А.Ф., Чобану Я.И., Бивол Л.Г. Криптосистемы на базе n-арных отношений. - В: Системы управления, контроля и измерений (УКИ-08), Российская Конференция с международным участием, Москва ИПУ РАН, 2008, с.66-67.

Prezentat la 20.11.2012