**TEMA 1**

**3.1 Partea teoretică:**

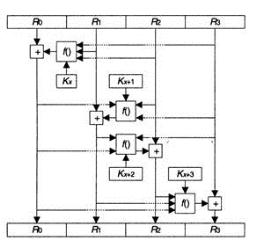
Algoritmul de criptare RC2 a fost dezvoltat la sfârșitul anilor '80. (1987 [395] și 1989 [224] sunt indicate în diverse surse) de Ronald Rivest, care, în special, a dezvoltat algoritmul RC5, considerat în sec. 3.42 este proprietatea RSA Data Security [390]; este cunoscut faptul că dezvoltarea acestui algoritm a fost inițiată și parțial sponsorizată de Lotus, ceea ce a necesitat un algoritm de criptare puternic (dar nu răspândit) pentru utilizarea ulterioară în sistemul software Lotus Notes. Mai mult decât atât, puterea criptografică a algoritmului ar fi trebuit să fie verificată de Agenția Națională de Securitate a SUA (NSA). De asemenea, ANS a contribuit la dezvoltarea algoritmului prin propunerea unor detalii de implementare încorporate în algoritmul Rivest [395]. O altă caracteristică interesantă este că RC2 a fost permis pentru export în afara Statelor Unite [395]. Există surse în care se afirmă că, în schimbul unei astfel de bunăvoințe din partea autorităților de reglementare, Rivest a introdus un anumit algoritm de slăbire în procedura cheie de extindere. Autorul acestei cărți nu a putut, totuși, să găsească informații detaliate despre acest lucru, astfel încât algoritmul RC2 va fi considerat în continuare în versiune fără astfel de slăbiciuni. Nu mai puțin interesant este faptul că, timp îndelungat după dezvoltarea RC2, structura acestui algoritm a rămas un secret, dar la începutul anului 1996, un autor anonim a publicat sci.crypt (a se vedea [320]) în codul sursă al algoritmului într-o conferință Usenet. Mai mult, nu este clar dacă aceasta a fost o scurgere de informații de la RSA Data Security sau rezultatul ingineriei inverse a oricărei implementări software a algoritmului RC2 [395]. Nu se știe cât timp va rămâne secret algoritmul dacă nu ar fi pentru această publicație anonimă, dar în martie 1998, Rivest a publicat complet algoritmul sub forma RFC 2268 [327]. A fost utilizat pe scară largă, cel puțin datorită faptului că este algoritmul principal și obligatoriu de criptare în conformitate cu standardul S / MIME pentru protecția mesajelor de e-mail (a se vedea [142]).

**3.1.2 Structura algoritmului**

Criptează datele în blocuri de 64 de biți folosind taste de dimensiuni variabile: de la 8 la 1024 de biți inclusiv; dimensiunea cheii recomandată este de 64 biți [327]. Algoritmul este o rețea Feistel, realizează 18 runde de transformări. Mai mult, runde ale algoritmului sunt împărțite în 2 tipuri: runde de amestecare (amestecare) și runde combinate (ochiuri). Structura generală a algoritmului este următoarea:  
1. Există 5 runde de amestecare.  
2. Există 1 rundă combinată.  
3. Există 6 runde de amestecare.  
4. Există 1 rundă combinată.  
5. Se efectuează 5 runde de amestecare.  
Structura rundei de amestecare este prezentată în Fig. 3.152. Se presupune  
că blocul de date criptat este împărțit în 4 cuvinte de 16 biți Rq ... R3 \* onD la ~  
Runda de amestec într-un ciclu în i de la 0 la 3 realizează următoarele

operații (componente ale funcției / () prezentate în Fig. 3.152):[http://crypto.pp.ua/img/shifr_image824-min-vs.jpg](http://crypto.pp.ua/img/shifr_image824.jpg)

în cazul în care:  
- Kj - fragmentul de cheie extins definit de variabila globală j \ această variabilă este inițial zero și crește cu 1 (așa cum se arată mai sus) în fiecare rundă de amestecare; procedura de extindere a cheilor este descrisă în detaliu mai jos;  
- & - operație logică bit-bit "și";

- ~ x - complement de biți la x \[](http://crypto.pp.ua/img/shifr_image826.jpg)

http://crypto.pp.ua/img/shifr_image850.jpg

http://crypto.pp.ua/img/shifr_image852.jpg

http://crypto.pp.ua/img/shifr_image854.jpg

*:*

**3.1.3 Decriptarea**

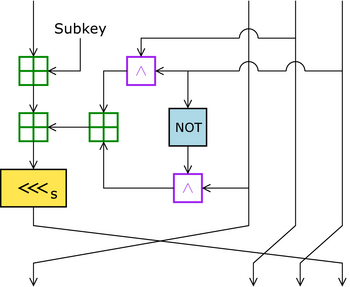
Decriptarea se realizează în același mod general ca și criptarea. Cu toate acestea, decriptarea utilizează alte operațiuni efectuate în amestecurile și combinarea runde.  
Runda de amestecare a decriptarea într-un ciclu de la 3 la 0 realizează următoarele operații:  
în cazul în care:  
- - - o deplasare ciclică spre dreapta prin numărul de biți determinat de valoarea lui St (vezi tabelul 3.93);  
- valoarea inițială a lui j este setată nu la 0, ci la 63.  
Runda de unificare în timpul decriptării este de asemenea modificată în comparație cu criptarea. În el, într-un ciclu de-a lungul i de la 3 la 0, se efectuează următoarea operație:

http://crypto.pp.ua/img/shifr_image856.jpg

**3.2 Partea practică:**

RIVEST CODE 2

Mesajul de criptare m= “ingineri” Subkey GX



**CRIPTARE**

**01101001 01101110 01100111 01101001 01101110 01100101 01110010 01101001**

a => (**01101001 01101110**) b => (**01100111 01101001**)

c => (**01101110 01100101**) d => (**01110010 01101001**)

Subkey =GX=>(01000111 01011000)

1. a + subkey=>(**01101001 01101110**+01000111 01011000) mod 2^32=

(26990+18264) mod 2^32 = 45254 = 1011000011000110

1. c^d=> **01101110 01100101** &

**01110010 01101001** =

**01100010 01100001**

1. (not d) ^ b= **10001101 10010110** &

**01100111 01101001**=

00000101 00000000

1. (c ^d) + (not d ^b) => (**01100010 01100001**+

00000101 00000000)mod2^32 =>

(25185+1280)mod2^32=>( 110011101100001)

1. (pt1)+(pt4) 1011000011000110 +110011101100001 mod2^32 =>

(45254+ 26465) mod 2^32 = mod 2^32 =>71719 => (10001100000100111)

1. Deplasament la stanga cu 4 (<<<4): 1100000100111 1000
2. b => (**01100111 01101001**) c => (**01101110 01100101**)

d => (**01110010 01101001**) a=>(**10001100000100111**)

1. Concatenare b,c,d,a => (**01100111 01101001 01101110 01100101 01110010 01101001 10001100000100111**) “gineri” „18278”

**DECRIPTARE**

1. Deplasamentul a la dreapta cu 4 (>>>4):

**10001100000100111** (>>>4):

**1110 1000110000010**

1. c^d=> **01101110 01100101**&

**01110010 01101001**=

**01100010 01100001**

1. (not d) ^ b= 00000101 00000000 &

**01100111 01101001** =

**00000101 00000000**

1. (c ^d) + (not d ^b) => (**01100010 01100001** +

**00000101 00000000**) mod 2^32 =>

(25185+1280)mod2^32 =>

110011101100001

1. (a - (b ^ (not d) + (c ^ d) )) =>

( **10001100000100111** –

110011101100001) mod2^32=>

(71719 - 26465)mod2^32=>  
45254=> (1011000011000110)

1. a = p5- GX=>(1011000011000110 - 01000111 01011000) mod 2^32 =>(45254-18264) mod2^32=26990=>

**01101001 01101110**

Concatenare a ,b,c,d => (**01101001 01101110 01100111 01101001 01101110 01100101 01110010 01101001**

“ ingineri”

**Concluzii:** In aceasta lectie s-a efectuat criptarea unui mesaj care constă din 8 simboluri, s-a utilizat algoritmul RC2 în care s-a criptat și decriptat un mesaj de 64 de biți. Pentru aceasta mesajul intial se convertește în cod binar dupa care efectuam treptat cele 8 etape de criptare s-a criptat mesajul - ingineri. Pentru decriptarea mesajului este necesar la fel de efectuat 6 etape si la sfârșit de concatinat mesajul binar în text.

Securitatea informaţională este o problemă care devine tot mai stringentă şi mai actuală odată cu dezvoltarea reţelelor şi industriei sistemelor de calcul. Una din metodele de bază de asigurare a securităţii informaţionale este metoda criptografică.

Un sistem criptografic este eficient atunci cînd ţine echilibrul între ceea ce este necesar şi ceea ce este posibil.

În practică, cînd apare problema implementării unui astfel de sistem, se poate de mers pe două căi: alegerea unui sistem existent sau crearea unuia nou.

În această lucrare am analizat un sistem existent bazat pe blocuri RC2. După cum se poate vedea mesajul ales a fost criptat și decriptat cu succes.