კრიპტოგრაფიის სიმეტრიული სისტემის ზოგიერთი მეთოდის რეალიზაციის საკითხების შესახებ

ვალერიან კეკელია, გულნარა კოტრიკაძე

საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

განხილულია კრიპტოგრაფიის ზოგიერთი მეთოდების რეალიზაციის პრაქტიკული საკითხები. კერძოდ, იგი ეძღვნება ცნობილი სიმეტრიული მეთოდების (ცეზარის, ვიჟინერისა და ვერნამის) მარეალიზებელი ალგორითმების აბსტრაქტული მოდელის სახით წარმოდგენას და მათი აპარატურული რეალიზაციის საკითხებს. შემოთავაზებულია მათემატიკური აპარატი, რომელსაც საფუძვლად უდევს ალგორითმული (მიკროპროგრამული) ალგებრის სისტემის – ოპერატორული ალგებრის და პირობის ალგებრის ცნება, რომელთა ტერმინებშიც შეიძლება იყოს აღწერილი სხვადასხვა სახის ალგორითმული პროცესები.

საკვანძო სიტყვები: კრიპტოგრაფია. ოპერატორული ალგებრა. პირობების ალგებრა. მიკროპროგრამა. იტერაცია. კონიუნქცია. დიზიუნქცია. ცეზარი. ვიჟინერი. ვერნამი.

1. შესავალი

ნაშრომში განხილულია მათემატიკური აპარატი, რომელსაც საფუძვლად უდევს ორი ერთმანეთთან ურთიერთდაკავშირებული ალგებრის – ოპერატორული-$G(G_1,G_2,\dots)$ და პირობის-$P(\alpha,\beta,\dots)$ ალგებრის ცნება. აღნიშნული ალგებრები წარმოდგენილნი არიან ერთრეგისტრიანი ან მრავალრეგისტრიანი პერიოდულად განსაზღვრული გარდასახვების სახით, რომელთა ტერმინებშიც აღიწერება ზოგიერთი ალგორითმული პროცესების მიკროპროგრამები. ცნობილია, რომ ოპერატორული ალგებრის ელემენტებს როგორც ბაზურს, ასევე მისგან წარმოებულს, უწოდებენ ოპერატორებს და განსაზღვრულნი არიან ისინი M ინფორმაციულ სიმრავლეზე, სადაც M იმ რეგისტრების მდგომარეობათა საერთო რიცხვია, რომლების მონაწილეობას ღებულობენ სისტემაში მიმდინარე გამოთვლით პროცესებში. დაუშვათ, რომ $X^R=\{\dots, x_{-1}^R, x_0^R, x_1^R, \dots\}$(სადაც R=1,2,3,…) ორმხრივ უსასრულო რეგისტრების ერთობლიობაა და მათი ყოველი n-ური ($-\infty<n>\infty$) ელემენტი (ე.წ. ტრიგერი) ღებულობს ერთ-ერთ მნიშვნელობას სიმრავლიდან $E_2=\{0,1\}$. პირობის ალგებრის ელემენტები (როგორც ბაზურის, ასევე მისგან წარმოებულის) განისაზღვრებიან M ინფორმაციულ სიმრავლეზე, როგორც პირობები, რომლებსაც შეუძლიათ მიიღონ ერთ-ერთი მნიშვნელობა თავისი სამი მნიშვნელობიდან <T(true), F(false), U(unknown)>. ოპერატორულ ალგებრაში ძირითად ოპერაციად მიღებულია გამრავლების ოპერაცია ანუ ოპერატორების თანმიმდევრული შესრულება, ხოლო პირობის ალგებრაში – ოპერაციები: კონიუნქცია, დიზიუნქცია, ინვერსია. განვიხილოთ ოპერაციები, რომელთა მეშვეობითაც ხორციელდება G და P ალგებრების ურთიერთდაკავშირება [1,3]:

1. α - დიზიუნქცია არის ოპერაცია, რომლის მიხედვითაც განისაზღვრება შესასრულებელი ოპერატორი ორი მოცემული ოპერატორიდან:

$$Q = (_{\alpha}\ G_1 \cup G_2\ )$$

სადაც $Q=G_1$ თუ α=true და $Q=G_2$ თუ α=false, ხოლო თუ α=unknown-ს ეს არის შემთხვევა, რაც იწვევს error-ს. აღნიშნული ოპერაცია პროგრამირებაში ცნობილია როგორც ‘‘პირობითი გადასვლის’’ ოპერატორი ანუ ოპერატორი, რომელიც გამოიყენება განშტოებადი ალგორითმების სარეალიზაციოდ.

2. α - იტერაცია, არის ოპერაცია, რომლის მიხედვითაც განისაზღვრება შესასრულებელი ოპერატორის მრავალჯერადი გამეორება:

$$Q=\{_{\alpha}\ G\}$$

სადაც Q ოპერატორი ღებულობს G შესასრულებელი ოპერატორის მნიშვნელობებს მანამ, სანამ α=true. ოპერატორი Q არ არის განსაზღვრული, თუ ლოგიკური პირობა α=unknown-ს. იმ შემთხვევაში, თუ α=false ოპერატორი G არ სრულდება. α-იტერაციის ოპერაცია პროგრამირებაში გამოიყენება "ციკლური პროცესების" სარეალიზაციოდ. აღწერილი ოპერაციების სახესხვაობები შეიძლება წარმოვადგინოთ შემდეგი გამოსახულებების სახით:

$$Q=\{G\ \alpha\}=G\{_{\alpha}\ G\}$$

$$(G_1 \cup G_2\ )=(_{\alpha}\ \ G_2 \cup G_1\ )$$

$$Q=\{_{F}\ G\}=e$$

სადაც e ცარიელი ოპერატორია.

$\beta=G\times\alpha$ - არის ლოგიკური პირობა, რომელიც ღებულობს იმავე მნიშვნელობას რასაც α, ოღონდ G ოპერატორის შესრულების შემდეგ [3].

2. ძირითადი ნაწილი

ცნობილია, რომ ნებისმიერი ოპერატორის წარმოდგენას ალგორითმული ალგებრის სისტემაში უწოდებენ ამ ოპერატორის რეგულარულ მიკროპროგრამას [1,3]. მაგალითის სახით ქვემოთ მოყვანილია რეგულარული მიკროპროგრამა - Σ^{c}, რომლის შესრულების შედეგი ორი მთელი რიცხვის ჯამია:

$$\Sigma^{c}=\begin{matrix} 0^{i} & Z^{i}_{r1} \\ 0^{j} & Z^{j}_{r2} \end{matrix}\ \ \Sigma_{R}^{\,i\,j} \qquad (1)$$

$$\Sigma_{R}^{\,i,j}=\left\{ \begin{matrix} & mod_2(X^{i}_{n},X^{j}_{n}) & \\ \alpha & \&(X^{i}_{n},X^{j}_{n}) & L^{j}_{1} \end{matrix} \right\} \qquad (2)$$

სადაც, $\Sigma_{R}^{i,j}$ – ორი მთელი რიცხვის ($r1$ და $r2$) შეკრების მიკროპროგრამაა. იგულისხმება, რომ რიცხვები $r1$ და $r2$ შესაბამისად შეტანილია X^{i} და X^{j} რეგისტრებში, ხოლო მიკროპროგრამის $\Sigma_{R}^{i,j}$ შესრულების შედეგი ფიქსირდება X^{i} რეგისტრში, იმ შემთხვევაში თუ $R=i$, ხოლო როცა $R=j$ – X^{j} რეგისტრში.

0^{R} – X^{R} $(R=i,j,..)$ რეგისტრის ნულოვან მდგომარეობაში გადაყვანის ოპერატორია, ხოლო Z^{R}_{r} – r რიცხვის მნიშვნელობის X^{R} – რეგისტრში შეტანის ოპერატორია.

$mod_2(X^{i}_{n}, X^{j}_{n})$ – წარმოებული ლოგიკური ოპერატორია, მარეალიზებელი $f(x^{i}_{n}, x^{j}_{n})$ გადამრთველი ფუნქციის $f(x^{i}_{n},x^{j}_{n})=\sim x^{i}_{n}\&x^{j}_{n}\cup x^{i}_{n}\&\sim x^{j}_{n}$ (ფუნქცია აღწერს X^{i} და X^{j} რეგისტრების n-ური თანრიგების $(-\infty<n>\infty)$ მნიშვნელობების ორის მოდულით შეკრებას).

$\&(X^{i}_{n}, X^{j}_{n})$ – ბაზური ლოგიკური ოპერატორია მარეალიზებელი $f(x^{i}_{n}, x^{j}_{n})$ გადამრთველი ფუნქციის $f(x^{i}_{n}, x^{j}_{n})=x^{i}_{n}\&x^{j}_{n}$ (ფუნქცია აღწერს X^{i} და X^{j} რეგისტრების n-ური თანრიგების მნიშვნელობების ლოგიკურ გამრავლებას – კონიუნქციას).

L^{j}_{1} – X^{j} რეგისტრში შეტანილი რიცხვის ერთი თანრიგით მარცხნივ დაძვრის ოპერატორია.

α - ლოგიკური პირობაა, სადაც α= false , თუ ლოგიკური ოპერატორის $\&(X^{i}_{n}, X^{j}_{n})$ შესრულების შედეგად X^{j} რეგისტრის ყველა ელემენტი მიიღებს ნულის მდგომარეობას, წინააღმდეგ შემთხვევაში α= true.

შევნიშნოთ, რომ მიკროპროგრამებში ერთ სვეტში შეტანილი ოპერატორები სრულდება პარალელურად. აღნიშნულიდან გამომდინარე იგულისხმება, რომ Z^{i} და Z^{j} ასევე 0^{i} და 0^{j} ოპერატორები სრულდება ერთდროულად. ერთდროულად სრულდება აგრეთვე - $mod_2(X^{i}_{n}, X^{j}_{n})$ და $\&(X^{i}_{n}, X^{j}_{n})$ ოპერატორები.

განვიხილოთ Σ^{c} მიკროპროგრამის შესრულების პროცედურა კონკრეტულ მაგალითზე. ვთქვათ, შესაკრებია ორი მთელი დადებითი რიცხვი: 87 და 78, რომელთა ჯამი უდრის 165-ს. შევნიშნოთ,

რომ 87 და 78, W და N სიმბოლოების კოდების მნიშვნელობებია შესაბამისად, ათობით ათვლის სისტემაში.

01010111 01001110	შეესაბამება რიცხვს 87, ორობით ათვლის სისტემაში შეესაბამება რიცხვს 78, ორობით ათვლის სისტემაში	X^1 X^2
00011001 01000110	ჯამი mod_2 &-კონიუნქცია (ლოგიკური ნამრავლი)	X^1 X^2
10001100	L_1^2 (X^2 –ის ერთი თანრიგით მარცხნივ დაძვრა)	X^2
00011001 10001100	ოპერანდების ფორმირება	X^1 X^2
10010101 10001000	ჯამი mod_2 &-კონიუნქცია (ლოგიკური ნამრავლი)	X^1 X^2
00010000	L_1^2 (X^2 –ის ერთი თანრიგით მარცხნივ დაძვრა)	X^2
10010101 00010000	ოპერანდების ფორმირება	X^1 X^2
10000101 00010000	ჯამი mod_2 &-კონიუნქცია (ლოგიკური ნამრავლი)	X^1 X^2
00100000	L_1^2 (X^2 –ის ერთი თანრიგით მარცხნივ დაძვრა)	X^1
10000101 00100000	ოპერანდების ფორმირება	X^1 X^2
10100101 00000000	ჯამი mod_2. (1*128+32+4+1=165) & - კონიუნქცია (ლოგიკური ნამრავლი)	X^1 X^2

რადგან & - კონიუნქციის შედეგი გახდა ნულის ტოლი. ცხადია, რომ მიკროპროგრამის შესრულება დამთავრებულია.

განვიხილოთ შეკრების ოპერაციის შებრუნებული ოპერაცია "გამოკლება", კონკრეტულ მაგალითზე და შემდეგ შევადგინოთ "გამოკლების" ოპერაციის მარეალიზებელი მიკროპროგრამა.

ვთქვათ გამოსათვლელია სხვაობა ორი (165 და 78) დადებით რიცხვებს შორის, რეზულტატი იქნება 165-78=87. აღნიშნული ოპერაციის შესასრულებლად საჭიროა მაკლები (78) გადავიყვანოთ შებრუნებულ კოდში და მიღებულ შედეგს ბოლო თანრიგში დაუმატოთ 1. აღნიშნული მანიპულაციების შესრულების შედეგად მიიღება მაკლები გადაყვანილი დამატებით კოდში. ამრიგად, გვექნება: ~(01001110)+00000001=10110010. აღწერილი პროცედურების შესრულების შემდეგ, ვასრულებთ მიკროპროგრამას "შეკრება" - Σ^c.

10100101 10110010	საკლები, ანუ 165-ის ორობითი კოდი მაკლები, ანუ 78-ის დამატებითი კოდი	X^1 X^2
00010111 10100000	ჯამი mod_2 &-კონიუნქცია (ლოგიკური ნამრავლი)	X^1 X^2
01000000	L_1^2 (X^2 –ის ერთი თანრიგით მარცხნივ დაძვრა)	X^2
00010111 01000000	ჯამი mod_2 ოპერანდების ფორმირება &-კონიუნქცია (ლოგიკური ნამრავლი)	X^1 X^2
01010111 00000000	ჯამი mod_2 . (1*64+1*16+1*4+1*2+1=87) &-კონიუნქცია (ლოგიკური ნამრავლი)	X^1 X^2

რადგან &-კონიუნქციის (ლოგიკური ნამრავლის) შედეგი გახდა ნულის ტოლი ცხადია, რომ ოპერაციის შემდგომი შესრულება დამთავრებულია.

აღწერილი პროცედურის (ორი მთელი რიცხვის ოპერაცია "გამოკლება": r1-r2) მა-რეალიზებელი მიკროპროგრამას - Σ^s აქვს შემდეგი სახე:

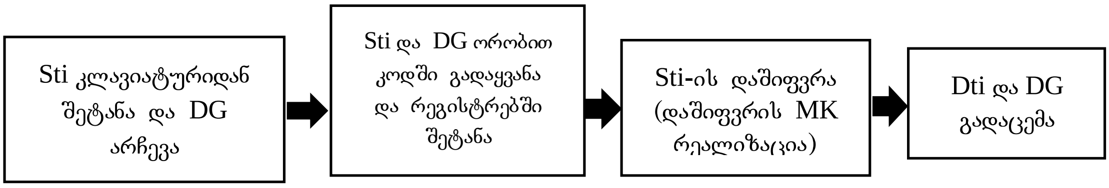
$$\Sigma^s = \begin{matrix} 0^i & Y^{i}{}_1 \\ 0^j & Z^{j}{}_{r2} \end{matrix} \quad \sim X^j \Sigma_i{}^{i,j}\ 0^j Z^{j}{}_{r1} \Sigma_i{}^{i,j} \tag{3}$$

სადაც, Y_1^j – არის ოპერატორი, რომლის შესრულების შედეგად X^1 რეგისტრის ბოლო თანრიგი გადადის ერთის (ანუ – 0000....0001) მდგომარეობაში [3].

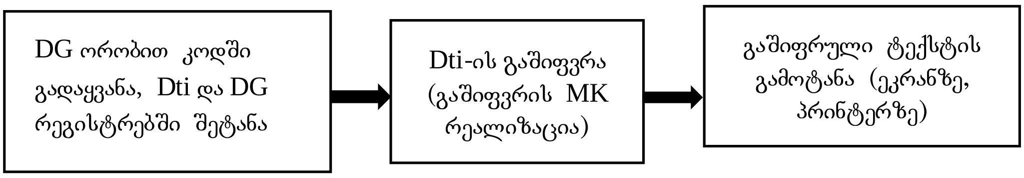
შემოთავაზებული მათემატიკური აპარატის გამოყენებით შედგენილია მიკროპროგრამები და მათი სქემური რეალიზაციის აბსტრაქტული მოდელები, კრიპტოგრაფიაში ფართოდ ცნობილი სიმეტრიული სისტემის მეთოდებისა, რომელთა რიცხვსაც მიეკუთვნება ცეზარის, ვიჟინერისა და ვერნამის მეთოდები.

აღნიშნული მეთოდების მიხედვით ტექსტური ინფორმაციის დაშიფვრა-გაშიფვრა ხორციელდება ერთი ან რამოდენიმე სიმბოლოს ე.წ. დაშიფვრის დახურული გასაღების - DG გამოყენებით. DG შეირჩევა საწყისი ტექსტური ინფორმაციის დამშიფრავის (**სტიდ**) მიერ, რომელსაც იგი პირადად (საიდუმლო გზით) გადასცემს დაშიფრული ტექსტური ინფორმაციის გამშიფრავის (**დტიგ-ს**). ავღნიშნოთ, რომ დახურული გასაღები - DG, შერჩეული სტიდ-ის მიერ გამოიყენება, როგორც ტექსტის დასაშიფრად ასევე მის გასაშიფრადაც. რაც შეეხება დაშიფრულ ტექსტურ ინფორმაციას (Dti), მისი გადაცემა შეიძლება და როგორც წესი, ხორციელდება ხელთარსებული ტექნიკური საშუალებებით, ღია არხებით (გლობალური ქსელით), რაც ყველასათვის ხელმისაწვდომია [2].

საწყისი ტექსტური ინფორმაციის (Sti) დაშიფვრისა და გაშიფვრის პროცედურების რეალი-ზაციის ეტაპები, შესაბამისად, ნაჩვენებია 1-ელ და მე-2 ნახაზებზე.



ნახ.1. საწყისი ტექსტური ინფორმაციის დაშიფვრის პროცედურა.



ნახ.2. დაშიფრული ტექსტური ინფორმაციის გაშიფვრის პროცედურა.

მომავალში იგულისხმება, რომ Sti აიკრიფება კლავიატურიდან იმ სიმბოლოების გამოყენებით, რომლებიც განსაზღვრულია ASCII სტანდარტით. ამ სტანდარტის მიხედვით ნებისმიერი S_i სიმ-ბოლოს რიცხვითი კოდის მნიშვნელობა, ათობით ათვლის სისტემაში, ეკუთვნის სასრულო სიმრავლეს U, სადაც U€{32,33,......,106,106}. ცხადია, რომ S_i სიმბოლოს რიცხვითი კოდი, რომ წარმოვადგინოთ ორობით ათვლის სისტემაში, საჭიროა შვიდი ბიტი. დაშიფვრა-გაშიფვრის პროცედურების განხორ-ციელებისა და მათი მარეალიზებელი სქემის აღწერის მიზნით, განვიხილოთ სამი ორმხრივ უსას-რულო ორობითი რეგისტრი X^1, X^2 და X^3 (შევნიშნოთ, რომ X^3 რეგისტრი გამოიყენება შუალედური რეზულტატების დასამახსოვრებლად). დავყოთ ეს რეგისტრები ტოლ ნაწილებად. თითოეულ ნაწილში

N^z (z=1,2,3,…,Z-სიმბოლოების რაოდენობა Sti-ში) გავაერთიანოთ რეგისტრის რვა ელემენტი ანუ რვა ტრიგერი - x^Z_j (სადაც j=7,6,5,4,3,2,1,0). X^1 რეგისტრის N^z ნაწილში შევიტანოთ დასაშიფრი ტექსტის S_z სიმბოლოს კოდის ორობითი მნიშვნელობა, ხოლო X^2 მდგომარეობა განისაზღვრება შემდეგი წესით:

ცეზარის მეთოდი. X^2 რეგისტრის ყოველ N^z (z=1,2,3,…) ნაწილში შეიტანება დამშიფრავი სიმბოლოების კოდების ორობითი მნიშვნელობა. აღნიშნულ მეთოდში Sti დასაშიფრად (ანუ დამშიფ-რავი სტრიქონის ფორმირებისათვის), როგორც ცნობილია, გამოიყენება ერთი სიმბოლო ან სიმბო-ლოების მიმდევრობა დაძრულნი ერთმანეთისგან გარკვეული წესით, რომელთა რაოდენობა Z-ის ტოლია.

ვიჟინერის მეთოდი. როგორც ცნობილია, წინასწარ შერჩეული (ვთქვათ, ABC) სიმბოლოებისგან ფორმირდება დამშიფრავი სტრიქონი (შემდეგი წესით: ABCABCABC…), რომელშიც სიმბოლოების რაოდენობა უნდა იყოს ტოლია (და არა მეტი და არც ნაკლები) Sti-ში შემავალი სიმბოლოების რაოდენობაზე. ამგვარად ფორმირებული სტრიქონში შემავალი ყოველი სიმბოლოს რიცხვითი კოდი შეიცვლება მისთვის შესაბამისი ორობითი კოდით და შეიტანება X^2 რეგისტრში.

ვერნამის მეთოდი. სტიდ-ი ირჩევს დამშიფრავ ტექსტს, რომელშიც სიმბოლოების რაოდენობა Sti-ში შემავალი სიმბოლოების რაოდენობის ტოლია. X^1 რეგისტრში Sti-ის შეტანის ანალოგიურად X^2 რეგისტრში შეიტანება დამშიფრავი ტექსტის სიმბოლოების კოდების ორობითი მნიშვნელობები. შევნიშნოთ, რომ X^2 რეგისტრში ნაკლები სიმბოლოების შემცველი ტექსტის შეტანის შემთხვევაში, ამ რეგისტრის ბოლო თანრიგები იქნება შევსებული ნულებით [2].

განვიხილოთ ტექსტური ინფორმაციის დაშიფვრა-გაშიფვრის პროცედურების რეალიზაციის ორი ვარიანტი. ავღნიშნოთ, რომ თითოეული მათგანი სრულდება ორ ეტაპად. პირველ ეტაპზე სრულ-დება Sti დაშიფვრა, ხოლო მეორეზე - Dti-გაშიფვრა. ორივე შემოთავაზებული ვარიანტისათვის ამ ეტაპების თანმიმდევრული შესრულება იწვევს ანალოგიურ გარდასახვებს.

ვარიანტი 1. Sti დაშიფვრისა მარეალიზებელ მიკროპროგრამას M^1 აქვს შემდეგი სახე:

$$M^1=\mathrm{mod}_2(X^1_n, X^2_n), \tag{4}$$

სადაც (-∞<n>∞), ხოლო $\mathrm{mod}_2(X_i^1,X_i^2)$ – წარმოებული ლოგიკური ოპერატორია (იხ.ფორმულა 2). M^1 მიკროპროგრამის რეალიზაციის შედეგად X^1 რეგისტრში დაფიქსირდება შიფროტექსტი (Dti).

M^1 მიკროპროგრამის განმეორებითი შესრულების შედეგად, X^1 რეგისტრში დაფიქსირებული შიფროტექსტი Dti, გარდაისახება Sti, რომელიც დაფიქსირდება აგრეთვე X^1 რეგისტრში.

ვარიანტი 2. X^1 რეგისტრის ყოველი N^z ნაწილისათვის თუ განხორციელდება Σ^c მიკ-როპროგრამის მარეალიზებელ გარდასახვები და მიეთითება, რომ R=1, X^1 რეგისტრში დაფიქსირდება შიფროტექსტი (Dti).

Dti-იდან Sti აღდგენის მიზნით საკმარისია X^1, X^2 და X^3 რეგისტრებზე განხორციელდეს Σ_1^s მიკროპროგრამით (5) განსაზღვრული გარდასახვები:

$$\Sigma_1^s=X_1^3 \mathop{}\limits^{Y_1^{1,z}}_{Z_{r2}^2} \sim X^2\Sigma_2^{1,2}X_3^1\,\Sigma_1^{1,2} \tag{5}$$

სადაც $Y_1^{1,z}$ არის მიკროოპერაცია, რომლის შესრულების შედეგია X^1 რეგისტრის ყოველ N^z (z=1,2,3,..,Z) ნაწილში x_0^1 ელემენტის ნულის მდგომარეობაში გადაყვანა.

Σ_1^s მიკროპროგრამის შესრულების შედეგი - Sti დაფიქსირდება X^1 რეგისტრში.

დასასრულს ავღნიშნოთ, რომ გაერთიანებული სიმბოლოების (ინგლისური, ქართული, რუსული ენების ალფაბიტის) ნაკრების ბაზაზე აგრეთვე შედგენილია კრიპტოგრაფიაში ფართოდ გავრცე-ლებული სიმეტრიული სისტემების ალგორითმების მიკროპროგრამები და შემოთავაზებულია მათი სქემოტექნიკური რეალიზაციის ოპტიმალური ვარიანტები, ასევე დამუშავებულია Microsoft Visual studio 2010 გარემოში დაპროგრამების ენის C# ბაზაზე ტექსტური ინფორმაციის დაშიფვრა-გაშიფვრის მარეალიზებელი პროგრამები.

3. დასკვნა

ავღნიშნოთ, რომ კრიპტოგრაფიის სიმეტრიული სისტემების მეთოდების შემოთავაზებული მიკროპროგრამული რეალიზაცია, მიკროელექტრონიკაში თანამედროვე მიღწევების გათვალისწინებით არ უნდა წარმოადგენდეს დიდ სირთულეს და არ უნდა მოითხოვდეს დიდ დანახარჯებს. ანუ შეიძლება იყოს ეკონომიკური, ფინანსური თვალსაზრისითაც.

ლიტერატურა:

1. Глушков В. М., Теория автоматов и формальные преобразования микропрограмм, журн. „Кибернетика“ 5, К. 1965;
2. კოტრიკაძე გ. (2010). ინფორმაციის დაცვის მოცულობითი მატრიცის მეთოდის დამუშავება და მისი შედარება ასიმეტრიულ მეთოდებთან. შრომები, მართვის ავტომატიზირებული სისტემები, სტუ №1(8), გვ.45-51.
3. კეკელია ვ. (2010). ალგორითმული ალგებრის საშუალებათა გამოყენება მიკროპროგრა-მირების საკითხებში. ილ. ჭავჭავაძის სახ. თბილისის სასწ. უნივერსიტეტი, „სამეცნიერო ძიებანი“, ტ.6, თბილისი.

SYMMETRIC CRYPTOGRAPHY SYSTEM ON THE METHOD OF IMPLEMENTATION ISSUES

Kekelia Valer, Kotrikadze Gulnara

Georgian Technical University

Summary

The article deals with some of the methods for the realization of practical issues in cryptography. In particular, it is dedicated to the well-known symmetric methods (Cezar, Vijiner and Vernam) algorithms selling model in the form of an abstract idea and the realization of their hardware. The proposed mathematical apparatus, which is based on algorithmic system algebra - terms of camerawork algebra and concepts algebra, which may be described the term different of algorithmic processes.

О ВОПРОСАХ РЕАЛИЗАЦИИ НЕКОТОРЫХ МЕТОДОВ СИМЕТРИЧНОЙ СИСТЕМЫ КРИПТОГРАФИИ

Кекелия В., Котрикадзе Г.

Грузинский Технический Университет

Резюме

В работе рассмотрены практические вопросы реализации некоторых методов криптографии. В частности, она посвящена представлению ввиде абстрактной модели алгоритмов, реализующие известные симетрические методы (Цезария, Видженера и Вернама) и вопросам их аппаратной реализации. Предложен математический аппарат, в основе которого лежит понятие системы алгоритмической (микропрограмной) алгебры - алгебры операторов и алгебры условий, в терминах которых может быть описаны разного рода алгоритмические процессы.