ვალერიან კეკელია, გულნარა კოტრიკაძე

კრიპტოგრაფიის სიმეტრიული სისტემის მეთოდები და მოდელები

ნაწილი I

„ტექნიკური უნივერსიტეტი“

2016

საქართველოს ტექნიკური უნივერსიტეტი

ვალერიან კეკელია, გულნარა კოტრიკაძე

კრიპტოგრაფიის სიმეტრიული სისტემის მეთოდები და მოდელები

რეკომენდებულია საქართველოს
ტექნიკური უნივერსიტეტის
სარედაქციო-საგამომცემლო საბჭოს
მიერ, 30.03.2016, ოქმი №1

თბილისი

2016

უაკ 681.142.001

დამხმარე სახელმძღვანელო, რომელშიც განხილულია ინფორმაციის დაცვის მეთოდები, ეძღვნება როგორც კრიპტოგრაფიის თეორიული მასალის შესწავლას, ისე პრაქტიკული ამოცანების გადაწყვეტასა და ლაბორატორიული სამუშაოების შესრულების საკითხებს. კერძოდ, შესრულებულია კრიპტოგრაფიის პრაქტიკული ამოცანების Microsoft Visual Studio 2010 გარემოში დაპროგრამების ენა C#-ის ბაზაზე ფუნქციონირებადი პროგრამული მოდულების (ალგორითმების) კომპიუტერული რეალიზაცია და შემოთავაზებულია მათი პრაქტიკული გამოყენების ძირითადი ასპექტები.

განკუთვნილია საქართველოს ტექნიკური უნივერსიტეტის ბაკალავრიატისა და მაგისტრატურის სტუდენტებისათვის, რომლებსაც ასწავლიან სასწავლო პროგრამით გათვალისწინებულ საგანს - ბაკალავრებს - „ინფორმაციის დაცვის მეთოდები და სისტემები“, ხოლო მაგისტრანტებს - „ინფორმაციის უსაფრთხოება“, აგრეთვე მსურველებისათვის ვინც დაინტერესებულია მიიღოს საწყისი, მაგრამ ამომწურავი ცოდნა, კრიპტოგრაფიის კვლევის საკითხებში.

რეცენზენტები: პროფესორი თ. კაიშაური,
პროფესორი რ. სამხარაძე

ISBN 978-9941-20-653-5 (ყველა ნაწილი)
ISBN 978-9941-20-654-2 (პირველი ნაწილი)
http://www.gtu.ge

სარჩევი

შესავალი

თანამედროვე მიღწევებმა კომპიუტერულ ტექნოლოგიებში წამოჭრა რიგი პრობლემები, რომლებმაც მოიცვა ადამიანთა მიღწევების თითქმის ყველა სფერო. ამ პრობლემების ძირითად ნაწილს შეადგენს კომპიუტერული უსაფრთხოების უზრუნველყოფა, რომელიც მეცნიერების (სპეციალისტების) კვლევის აქტუალური საგანია და ამასთან ერთად მილიონობით გამოთვლითი სისტემების მომხმარებლების მოთხოვნაცაა.

აღნიშნული პრობლემიდან განსაკუთრებულ ყურადღებას იმსახურებს ორი ამოცანა:

1. გამოთვლითი სისტემების დაცვა არასანქციონირებული შეღწევისაგან;

2. ინფორმაციის (მონაცემების) დაცვა, რომელიც ფუნქციონირებს გამოთვლით სისტემებში.

დღევანდელი შეფასებით „ინფორმაცია“ - ეს არის ძვირად ღირებული საქონელი, რომელიც შეიძლება იყოს შეძენილი, გაყიდული, გაცვლილი და ა.შ. მისი ღირებულება ათჯერ და უფრო მეტად ძვირია იმ სისტემებზე, რომლებშიც ის ფუნქციონირებს. აქედან გამომდინარე, ინფორმაციის დაცვა იყო, არის და რჩება აქტუალურ ამოცანად, რომლის განხილვასაც ეძღვნება წინამდებარე დამხმარე სახემძღვანელო.

მეცნიერებაში ინფორმაციის დაცვის მიმართულება პრაქტიკაში დამკვიდრდა „კრიპტოგრაფიის“ სახელწოდებით. იგი ცნობილია გასული საუკუნეებიდან და იმდროინდელ მკვლევართა განსაკუთრებულ ყურადღებას იმსახურებდა, რომელთა მიღწევები დღესაც შეიძლება ეფექტურად გამოვიყენოთ [1].

კრიპტოგრაფია ინფორმაციის დასაიდუმლოების სამეცნიერო-ტექნიკური დარგია, რომელსაც მრავალსაუკუნოვანი ისტორია აქვს. მან განვითარების განსაკუთრებულ საფეხურს გასული საუკუნის მეორე ნახევარში მიაღწია, როდესაც მისი მეთოდები მათემატიკურ სისტემებს დაეფუძნა, ხოლო 1976 წლიდან ღია გასაღებების მეთოდოლოგიამ მას თვისობრივად ახალი ხარისხი

შესძინა. კრიპტოგრაფიის გამოყენების სფერო მრავალმხრივია. მაგალითად, როგორიცაა სამხედრო-სახელმწიფოებრივი და საბანკო-საფინანსო კომერციული საქმიანობა, ლოკალური და გლობალური ქსელებში (ინტერნეტში) ფუნქციონირებადი პროგრამებისა და მონაცემების დაცვა და სხვ [12].

1949 წლამდე კრიპტოგრაფიულ „კვლევებს“ და ამ მიმართულებით ჩატარებულ სამუშაოებს, აგრეთვე მიღებულ შედეგებს განიხილავდნენ როგორც ხელოვნების ნიმუშებად და არა როგორც მეცნიერულ მიღწევებად. ცნობილია იულიუს ცეზარის და ცეზარ ავგუსტის ტექსტის დაშიფვრის მეთოდები, რომლებსაც ისინი იყენებდნენ, ჯერ კიდევ XXI საუკუნის წინ, მეგობრებისადმი გაგზავნილ შეტყობინებებში. ასე მაგალითად, იულიუს ცეზარი (ცეზარ ავგუსტი) საწყისი ტექსტის დასაშიფრად იყენებდა ლათინურ ალფაბეტს და მეთოდს, რომლის მიხედვითაც ხდებოდა დასაშიფრი ტექსტის ყოველი სიმბოლოს წანაცვლება სამი (ოთხი) პოზიციით მარჯვნივ ან მარცხნივ და მისი ჩანაცვლება სიმბოლოთი, რომელიც აღმოჩნდებოდა ალფაბეტში წანაცვლების შედეგად განსაზღვრულ პოზიციაში [9,10].

ყურადღებას იმსახურებს 1926 წელს ამერიკის ტელეფონებისა და ტელეგრაფების კომპანიის ინჟინრის გ. ვერნამის მეთოდი, რომელშიც გამოყენებული იყო ცეზარის მიერ შემოთავაზებული მეთოდი, იმ განსხვავებით, რომ ნაცვლად ლათინური ალფაბეტისა, შიფრ-ტექსტის მისაღებად ის იყენებდა ორობით ე.წ. „ბოდო“ კოდს [2].

ახალი ერა კრიპტოგრაფიაში დაიწყო 1949 წლიდან კ.შენონის ნაშრომის [3] “საიდუმლო სისტემებში კავშირის თეორია” გამოქვეყნებიდან, რომლის საფუძველსაც შეადგენდა მის მიერ 1948 წელს გამოცემული სტატია [4], რაც თავის მხრივ ინფორმაციის თეორიის განვითარების დასაწყისიც იყო. მიუხედავად კ.შენონის ასეთი დიდი წვლილისა, ჟ.დიფის და მ.ჰელმანის ნაშრომმა [5] (პარალელურად გამოცემულმა რ.მარკერის ნაშრომმა [6]) „ახალი მიმართულებები კრიპტოგრაფიაში“ 1976 წელს მეცნიერების და სპეციალისტების ფართო მასების ყურადღება მიიპყრო, რამაც განაპირობა მათი აქტიური ჩართვა კრიპტოგრაფიის პრობლემების კვლევაში.

ტერმინი „კრიპტოგრაფია“ ბერძნული სიტყვაა, რომელიც შედგება ორი ნაწილისაგან და ნიშნავს: CRIPTOS - საიდუმლო და LOGOS - სიტყვა.

კრიპტოგრაფიაში განიხილავენ ორ მიმართულებას:

1. კრიპტოგრაფია;

2. კრიპტოანალიზი.

კრიპტოგრაფიის ამოცანაა გადასაცემი (დამუშავებული) ინფორმაციის მაქსიმალური დაცვა, კერძოდ მისი საიდუმლოებისა და მთლიანობის (ნამდვილობის) უზრუნველყოფა [10].

კრიპტოანალიტიკოსი, რომლისთვისაც როგორც წესი უცნობია ინფორმაციის დაშიფვრის გასაღები, მაგრამ ხელმისაწვდომია მხოლოდ დაშიფრული ტექსტი (შიფრ-ტექსტი) და შესაძლოა დაშიფვრის ალგორითმი, მოქმედებს კრიპტოგრაფის საწინააღმდეგოდ. კერძოდ, ის ცდილობს „გატეხოს“ კრიპტოგრაფის მიერ გამოყენებული დაცვის სისტემა, გაშიფროს შიფრ-ტექსტი ანუ აღადგინოს მისი საწყისი სახე ან/და გაავრცელოს ჭეშმარიტი შეტყობინების მსგავსი სახის ინფორმაცია.

ინფორმაციის კრიპტოგრაფიული დაცვა შეიძლება ორი გზით: პროგრამული და აპარატურული. აპარატურული მიდგომა ძვირად ღირებულია, თუმცა გააჩნია დადებითი მხარეებიც. კერძოდ, მისი წარმადობა საგრძნობლად მაღალია, რეალიზაცია მარტივია, შედარებით დაცულია და ა.შ. ამიტომაც იგი ფართოდ და წარმატებულად გამოიყენება დღემდე პრაქტიკაში. ასე მაგალითად, საკმარისია აღვნიშნოთ ის, რომ აშშ-ის მთავრობის პრინციპული მოთხოვნაა სახელმწიფო ორგანიზაციებმა და კერძო სტრუქტურებმა კრიპტოგრაფიული აპარატურა გამოიყენოს, რომლის ძირითად ნაწილს შეადგენს ნაციონალური უსაფრთხოების სააგენტოს (NSA) მიერ შემოთავაზებული ალგორითმების ბაზაზე შექმნილი მოწყობილობები [7].

გამოთვლით სისტემებში ინფორმაციის დაცვის მასიური გამოყენების ალგორითმები უნდა აკმაყოფილებდეს შემდეგ ძირითად მოთხოვნილებებს:

- დაშიფრული ტექსტის წაკითხვა უნდა იყოს შესაძლებელი, მხოლოდ გამშიფრავი გასაღების გამოყენებით;

- დაშიფვრის გასაღების უმნიშვნელო ცვლილებამ უნდა გამოიწვიოს შიფრ-ტექსტის მნიშვნელოვანი ცვლილება;
- დასაშიფრი ტექსტის უმნიშვნელო ცვლილებამ უნდა გამოიწვიოს შიფრ-ტექსტის მნიშვნელოვანი ცვლილება იმ შემთხვევაშიც კი, თუ დაშიფვრა ხორციელდება ერთი და იგივე გასაღებით;
- დაშიფვრის ალგორითმის ძირითადი პრინციპები უნდა იყოს საიდუმლოდ დაცული;
- დაშიფვრის პროცესი უნდა ექვემდებარებოდეს კონტროლს;
- დასაშიფრი ტექსტისა და შიფრ-ტექსტის სიგრძე (მათში შემავალი სიმბოლოების რაოდენობა) უნდა იყოს ერთმანეთის ტოლი;
- ნებისმიერი დაშიფვრის გასაღები, გასაღებების სიმრავლიდან უნდა უზრუნველყოფდეს ინფორმაციის საიმედო დაცვას;
- გამოყენებული ალგორითმი უნდა უზრუნველყოფდეს, როგორც პროგრამულ ასევე აპარატურულ რეალიზაციის საშუალებას. ამასთან გასაღების სიგრძის ზრდამ არ უნდა გამოიწვიოს ხარისხობრივი დაშიფვრის საიმედოობის და ალგორითმის გაუარესება;
- დაშიფრული ინფორმაციის დეშიფრაცია არ უნდა ხდებოდეს იგივე გასაღებით, რომლითაც დაიშიფრა ინფორმაცია;
- დაშიფრული ინფორმაციის დეშიფრაცია არ უნდა სრულდებოდეს უკუგზით;
- ინფორმაციის დაშიფრა-დეშიფრაციის პროცესი უნდა იყოს მარტივი და სწრაფი, ხოლო კრიპტოანალიტიკოსისათვის - გაშიფვრა, რეალურ დროში, შეუძლებელი [8].

პრაქტიკაში განიხილავენ კრიპტოგრაფიული სისტემების ორ ძირითად ჯგუფს: სიმეტრიულ და ასიმეტრიულ სისტემებს. სიმეტრიულ სისტემებს მიეკუთვნება ისეთი მეთოდები, რომელთა მიხედვითაც ტექსტური ინფორმაციის დაშიფვრა/გაშიფვრა ხორციელდება ერთი ან რამდენიმე სიმბოლოს (ე.წ. დამშიფრავი სიმბოლო(ები)ს ან რომელსაც აგრეთვე უწოდებენ და-

შიფვრის დახურულ გასაღებს) გამოყენებით. აღნიშნულიდან გამომდინარეობს, რომ სიმეტრიულ სისტემებში გამოიყენება ერთი და იგივე გასაღები, ინფორმაციის როგორც დასაშიფრად, ასევე მის გასაშიფრადაც. მიუხედავად იმისა, რომ სიმეტრიული სისტემის მეთოდები ნაკლებად დაცულია, ისინი დღემდე ეფექტურად გამოიყენება პრაქტიკაში, რაც განპირობებულია მათი რეალიზაციის სიმარტივით და სასურველი შედეგების სწრაფი მიღწევით.

სიმეტრიული სისტემების რიცხვს მიეკუთვნება პრაქტიკაში ფართოდ ცნობილი ცეზარის, ვიჟინერის, ვერნამის და შებრუნებული მატრიცის მეთოდები, რომელთა დეტალურ განხილვასაც ეძღვნება შემოთავაზებული დამხმარე სახელმძღვანელო.

ასიმეტრიულ სისტემებში გაერთიანებულია მეთოდები (რომლებიც ავტორების მომავალი კვლევის საგანი იქნება): დიფი-ჰელმან-მერკლეს, კომუტატიური მატრიცული, რაივეს-შამირ-ეიდელმანის, ელ-გამალის, ბრმად ხელის მოწერის, უდავოდ ხელის მოწერის, ციფრული ხელის მოწერის მეთოდები, რიცხვთა წონა და ა.შ. ამ მეთოდების მიხედვით დაშიფვრის პროცედურები სრულდება ღია გასაღებით, ხოლო გაშიფვრა კი მისი შებრუნებული ანუ ე.წ. დახურული (საიდუმლო) - გასაღებით. ასიმეტრიული სისტემის მეთოდები არის უფრო დაცული, მაგრამ ხასიათდება შესრულების დაბალი სიჩქარით. თუმცა უნდა აღინიშნოს, რომ დიდი მოცულობის ინფორმაციის დასაშიფრად, უპირატესობა მაინც ასიმეტრიული სისტემის მეთოდებს ენიჭებათ [12].

დაშიფრული ინფორმაციის მედეგობა (საიმედოობა) დამოკიდებულია შემდეგ მახასიათებლებზე, როგორიცაა:

1. შიფრაცია-დეშიფრაციის დრო/სიჩქარე;
2. გასაღების გენერაციის (არჩევა/გამოთვლა) დრო/სიჩქარე;
3. დახურული გასაღების ამოცნობის დრო/სიჩქარე;
4. გამოყენებული ფუნქცია;
5. გასაღების სიგრძე (ბიტებში);
6. კრიპტოანალიზის სირთულე;

7. გასაღებების სიმრავლე;

8. გაშიფვრის ალბათობა.

უნდა აღინიშნოს, რომ ერთ ან რამდენიმე მახასიათებლისათვის უპირატესობის მინიჭება, ყოველთვის ხორციელდება სხვა დანარჩენი მახასიათებლების ხარჯზე.

დასასრულ შევნიშნოთ, რომ წინამდებარე დამხმარე სახელმძღვანელოს ძირითადი მიზანია დაეხმაროს ბაკალავრიატის სტუდენტებს, მაგისტრანტებს, დამწყებ სპეციალისტებს სიმეტრიული მეთოდების შესწავლით შეიძინონ საკმაოდ ღრმა ცოდნა, რაც მისცემს მათ საშუალებას დასვან და გადაჭრან კრიპტოგრაფიისათვის დამახასიათებელი თანამედროვე ამოცანები და პრობლემები. შეიმუშაონ ახალი მეთოდები და მათი რეალიზაციის საშუალებები, პრინციპები და გზები. ხოლო ინფორმაციის დაშიფვრა/გაშიფვრის მეთოდების (ალგორითმების) მარეალიზებელ პროგრამული მოდულების გამოყენება დაეხმარება მათ სხვადასხვა სახის პრაქტიკული ექსპერიმენტის ჩატარებაში.

თავი 1. სიმეტრიული სისტემის მეთოდები

ცნობილია, რომ კრიპტოგრაფიის (როგორც სიმეტრიული, ასევე ასიმეტრიული სისტემების) მეთოდები ძირითადად დაფუძნებულია ერთი და იმავე პრინციპზე, რომლის ძირითადი არსი მდგომარეობს ტექსტურ ინფორმაციაში შემავალ სიმბოლოებზე წინასწარ განსაზღვრული მათემატიკური და ლოგიკური მანიპულაციების ჩატარებაში. განიხილავენ ტექსტური ინფორმაციის (TI) წარმოდგენის სამ სახეს:

ა) დასაშიფრი TI – DasTI;

ბ) დაშიფრული TI – ShifTI (შიფრ-ტექსტი);

გ) დამშიფრავი და გამშიფრავი დახურული (საიდუმლო) გასაღები k, სადაც k იგივეა რაც TI ანუ მასში შემავალი სიმბოლოების კრებული.

სამივე სახის TI ზოგად შემთხვევაში არის α - ალფაბეტში (α={A,B,…,Z} – არის ლათინური ასომთავრული სიმბოლოების ნაკრები) შემავალი სიმბოლოებისაგან შედგენილი სტრიქონი, კერძოდ:

$DasTI = \{S_0\ S_1\ S_2 \dots S_Z\}$,

$ShifTI = \{D_0\ D_1\ D_2 \dots D_Z\}$,

$DGDG = \{K_0\ K_1\ K_2 \dots K_z\}$ (z<=Z) ,

სადაც Z და z - აღნიშნავს აღწერილ სტრიქონში შემავალი სიმბოლოების რაოდენობას ანუ მოცემული სტრიქონის სიგრძეს.

TI დაშიფვრისა და გაშიფვრის მარეალიზებელი ალგორითმების ფორმალური აღწერის მიზნით, განვიხილოთ აგრეთვე მთელი დადებითი და უარყოფითი რიცხვთა სიმრავლე - Z_0, რომლის ელემენტების გამოყენებით შეიძლება აღიწეროს დასაშიფრი, გასაშიფრი და დამშიფრავი გასაღების მოცემული სტრიქონები, კერძოდ:

$DasTIn = \{S_i^n\}$,

$ShifTIn = \{D_i^n\}$,

$K_n = \{K_j^n\}$,

სადაც i=0,1,…,Z, j=0,1,…,z<=Z; n-მაჩვენებელი მიუთითებს α-ალფაბეტში მოცემული სიმბოლოს რიგით ნომერზე, როგორც ეს ნაჩვენებია 1.1 ცხრილში.

ცხრილი 1.1

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M

13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

მოცემულ თავში შემოთავაზებულ ტერმინებში და აღნიშვნების გამოყენებით აღწერილია პრაქტიკაში ფართოდ გავრცელებული სიმეტრიული სისტემის მეთოდები, რომელთა რიცხვს მიეკუთვნება ცეზარის, ვიჟინერის, ვერნამისა და შებრუნებული მატრიცის მეთოდები [12,15,16,17].

1.1. ცეზარის მეთოდი

ცეზარის მეთოდი, რომელიც ცნობილია აგრეთვე, როგორც ცეზარის შიფრი, ცეზარის ალგორითმი, დაძვრის შიფრი, პრაქტიკული გამოყენების თვალსაზრისით ძალზე მარტივია. მიუხედავად ამისა, იგი საკმაოდ ცნობილია დაშიფვრა/გაშიფვრის სხვა მეთოდებთან შედარებით [15].

ცეზარის ალგორითმის გამოყენების არსი, რომელშიც იგი იყენებდა მხოლოდ ლათინურ α ალფაბეტს, მდგომარეობს შემდეგში: დასაშიფრი TI-ის S_i (i=1,2,…,z) სიმბოლო ჩაენაცვლება მისგან K (K=3) პოზიციით მარჯვნივ (მარცხნივ), დაშორებული D_i-იური სიმბოლოთი. ამგვარი წესით მიღებული შიფრ-ტექსტი რა თქმა უნდა არ არის მდგრადი და ადვილად შეიძლება მისი „გატეხვა“. თუმცა უნდა აღინიშნოს, რომ ცეზარის მიერ შემოთავაზებული მეთოდის ძირითადი იდეა დღემდე წარმატებით გამოიყენება კრიპტოგრაფიის შედარებით რთულ სისტემებში. მაგალითად, როგორიცაა ვიჟინერის, ვერნამის და სხვა შიფრები, რომლებიც ქვემოთ არის აღწერილი [16,17].

როგორც ცნობილია, ცეზარი მის მიერ შედგენილ წერილებში იყენებდა დაშიფვრა/გაშიფვრის, მარცხნიდან/მარჯვნივ და პირიქით მარჯვნიდან/-მარცხნივ, შემდეგ სისტემას: მან ალფაბეტიდან გამოყო სამი სიმბოლო (X,Y,Z) და გამოიყენა K=3 პოზიციით ჩანაცვლების შემდეგი მარტივი წესი (იგულისხმება, რომ დაშიფვრა ხორციელდება მარცხნიდან მარჯვნივ, ხოლო გაშიფვრა მარჯვნიდან მარცხნივ):

დაშიფვრის შემთხვევაში:	**გაშიფვრის შემთხვევაში:**
თუ $S_i^1 \cong X$, მაშინ $D_i^1 \cong A$	თუ $D_i^1 \cong A$, მაშინ $S_i^1 \cong X$
თუ $S_i^2 \cong Y$, მაშინ $D_i^2 \cong B$	თუ $D_i^2 \cong B$, მაშინ $S_i^2 \cong Y$
თუ $S_i^3 \cong Z$, მაშინ $D_i^3 \cong C$	თუ $D_i^3 \cong C$, მაშინ $S_i^3 \cong Z$

აღწერილიდან გამომდინარე, ზოგადად შემოთავაზებული ჩანაცვლება აღიწერება შემდეგი სახით:

$$X \leftrightarrow A;\ Y \leftrightarrow B;\ Z \leftrightarrow C\ . \qquad (1.1)$$

დანარჩენ შესაძლო შემთხვევებში D_i განისაზღვრება K=3 პოზიციით მისგან მარჯვნივ მდებარე სიმბოლოთი. ანალოგიურად, თუ დაშიფვრა/გაშიფვრის პროცედურებს განვახორციელებთ მარჯვნიდან მარცხნივ ჩანაწერი (1.1) მიიღებს შემდეგ სახეს:

$$A \leftrightarrow X;\ \ B \leftrightarrow Y;\ C \leftrightarrow Z\ . \qquad (1.2)$$

განვიხილოთ ცეზარის მიერ შემოთავაზებული მეთოდის რეალიზაციის პროცედურული შესრულების ორი ვარიანტი [9,11].

ვარიანტი 1. ვთქვათ, დასაშიფრია და შემდგომ გასაშიფრია ტექსტური ინფორმაცია - TI = „XZ UNIVERSIT ZY”. TI დაშიფვრა (გაშიფვრა) შევასრულოთ სიმბოლოების K=3 პოზიციით გადანაცვლებით მარცხნიდან მარჯვნივ და პირიქით. ეტაპობრივი შესრულება დაშიფვრის პროცედურისა ნაჩვენებია 1.2 ცხრილში.

ცხრილი 1.2

რიგითი №	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DasTI(S_i)	X	Z		U	N	I	V	E	R	S	I	T	Y		Z	Y
DasTIn(S_i^n)	X	Z		20	13	8	21	4	17	18	8	19	Y		Z	Y
ShifTIn(D_i^n)	A	C		23	16	11	24	7	20	21	11	22	B		C	B
შიფრ-ტექსტი	A	C		X	Q	L	Y	H	U	V	L	W	B		C	B

შევნიშნოთ, რომ 1.2 ცხრილის 1,2,13,15,16 სვეტებში შეტანილი TI-იის სიმბოლოების ჩანაცვლება ხორციელდება 1.1 ფორმულის მიხედვით, ხოლო 4-12 სვეტებში შეტანილი TI-იის სიმბოლოების ჩანაცვლება კი ჩვეულებრივი წესით (K=3 სიმბოლოთი გადანაცვლების გზით), როგორც ეს აღწერილი იყო ფორმულით 1.1.

გაშიფვრის პროცედურული ნაბიჯები ანალოგიურია დაშიფვრის აღწერილი პროცედურული ნაბიჯებისა იმ განსხვავებით, რომ ჩასანაცვლებელი სიმბოლოები განისაზღვრება K=3 პოზიციით საათის ისრის მოძრაობის საწინააღმდეგო მიმართულებით (მარჯვნიდან მარცხნივ) გადანაცვლებით (იხ. ცხრ. 1.3).

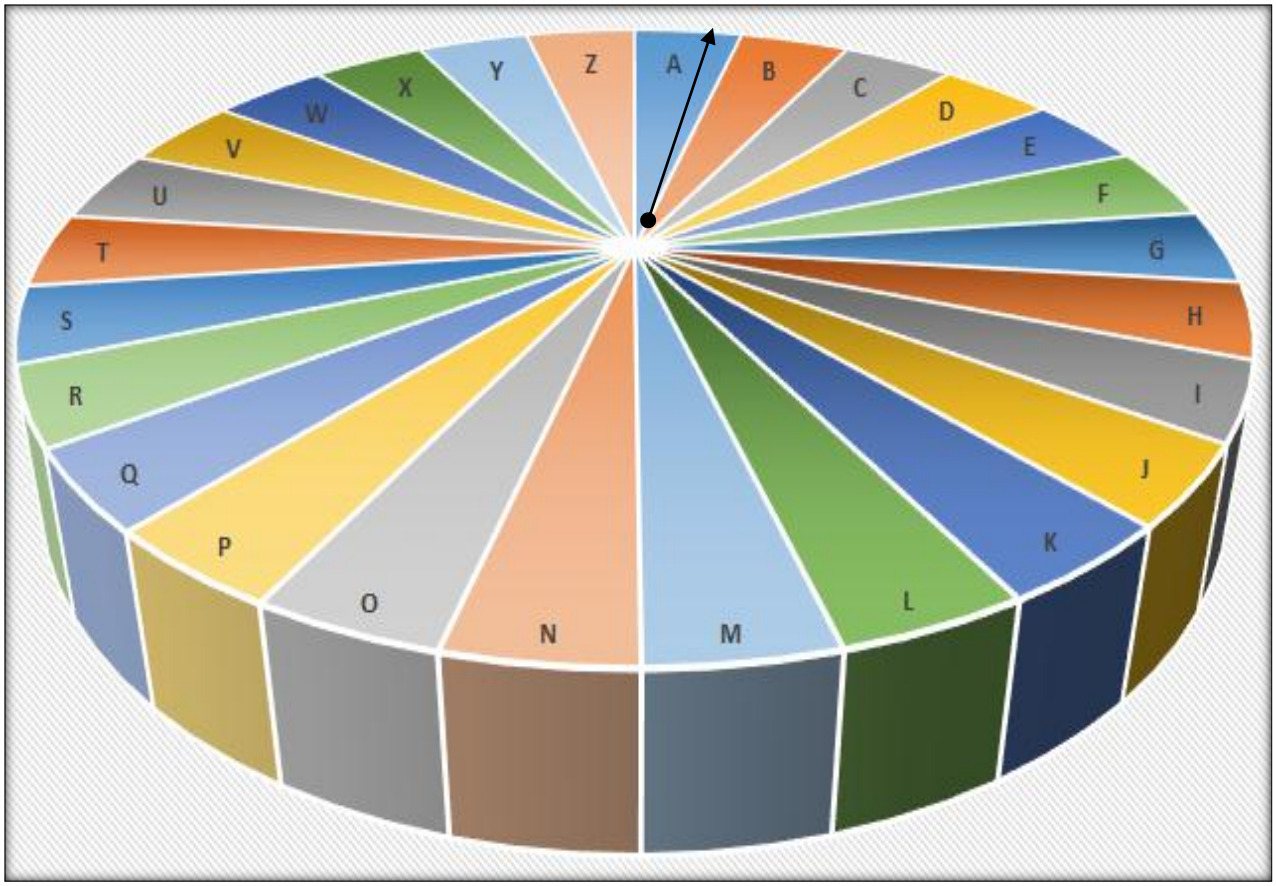
ცხრილი 1.3

რიგითი №	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
შიფრ-ტექსტი	A	C		X	Q	L	Y	H	U	V	L	W	B		C	B
ShifTIn(D_i^n)	A	C		23	16	11	24	7	20	21	11	22	B		C	B
DasTIn(S_i^n)	X	Z		20	13	8	21	4	17	18	8	19	Y		Z	Y
DasTI(S_i)	X	Z		U	N	I	V	E	R	S	I	T	Y		Z	Y

ვარიანტი 2. წარმოვიდგინოთ ბორბალი („იღბლიანი“) ან საათის ციფერბლატი (იხ.ნახ.1.1)

დავყოთ იგი n ტოლ ნაწილად (სექტორად), სადაც n ალფაბეტის სიმძლავრეა (ლათინური ალფაბეტის შემთხვევაში n=26). ყოველ ნაწილში მარცხნიდან მარჯვნივ მიმდევრობით ჩავწეროთ ალფაბეტის თითო სიმბოლო: A,B,C,…,Y,Z. ფიგურის ცენტრში დავამაგროთ რადიუსის ტოლი მოძრავი ისარი. TI-იის დასაშიფრავად, დავაყენოთ ისარი S_i - სიმბოლოზე და მივაბ-

რუნოთ იგი საათის ისრის მიმართულებით K - დანაყოფით. ისრის დაფიქსირებული ახალი მდგომარეობა მიუთითებს სიმბოლოზე - D_i (შიფრ-ტექსტის i-ურ სიმბოლოზე). შიფრ-ტექსტის გასაშიფრად საკმარისია ჩატარდეს აღწერილი პროცედურების საწინააღმდეგო მოქმედებები. კერძოდ, დავაყენოთ ისარი D_i - ურ სიმბოლოზე და მოვაბრუნოთ იგი საათის ისრის საწინააღმდეგო მოძრაობის მიმართულებით, K - დანაყოფით დაფიქსირებული ისრის მდგომარეობა მიუთითებს S_i - ურ სიმბოლოზე [12].



ნახ. 1.1. TI-იის დაშიფვრა/გაშიფვრის პროცედურების მარეალიზებელი დისკო

1.2 ვიჟინერის მეთოდი

უნდა აღინიშნოს, რომ ისტორიული მონაცემებით, რაც დღესათვის არის ცნობილი კრიპტოგრაფიაში ბ. ვიჟინერის მეთოდის სახელწოდებით, ჯერ კიდევ 1467წ. იყო შემოთავაზებული ლეონ ბატისტა ალბერტის მიერ, ხოლო 1518წ. იაგან ტრესემუსმა ნაშრომში „პოლიგრაფია“ აღწერა თავის მიერ გამოგონებული მრავალალფაბეტური ცხრილი ე.წ. tabula recta, რაც მოგვიანებით, ბ.ვიჟინერის მიერ იყო გამოყენებული, როგორც დაშიფვრის ცენტრალური კომპონენტი.

ბ.ვიჟინერის მიერ 1586წ. შემოთავაზებული მეთოდის ძირითადი არსი მსგავსია ცეზარის მეთოდისა [16]. განსხვავება ამ ორ მეთოდს შორის მდგომარეობს შემდეგში: ცეზარის მეთოდში დაძვრის კოეფიციენტის მნიშვნელობა იყო მუდმივი სიდიდე, კერძოდ K=3; ხოლო ვიჟინერი თავის მეთოდში იყენებდა ინდივიდუალურ დაძვრის კოეფიციენტებს K_i-ს TI-იის თითოეული S_i (i=1,2,…,Z) სიმბოლოს დასაშიფრავად.

ბ.ვიჟინერი, როგორც ცეზარი, მანიპულირებდა 26 სიმბოლოსაგან შემდგარ ლათინურ ალფაბეტზე, რომლის მეშვეობით ის ადგენდა 26 სტრიქონიან ცხრილს (იხ. ცხრ.1.4) რომელშიც ყოველი მომდევნო სტრიქონის სიმბოლოები იყო დაძრული მარცხნივ ერთი სიმბოლოთი. აღნიშნული წესით ფორმირებული ცხრილი იყო ეკვივალენტური ცეზარის 26 ერთმანეთისაგან განსხვავებული ალფაბეტისა (შიფრისა), რომელიც ცნობილია tabula recta-ს სახელით ან აგრეთვე ვიჟინერის 26 სვეტისა და 26 სტრიქონის მქონე კვადრატის სახელწოდებით [10].

ძირითადი მოთხოვნა ბ.ვიჟინერის მიერ შემოთავაზებული მეთოდისა არის ის, რომ დასაშიფრი და დამშიფრავი TI სიგრძე უნდა იყოს ერთმანეთის ტოლი, ვინაიდან ყოველ S_i-ურ სიმბოლოს ინდივიდუალურად შეესაბამება დაძვრის კოეფიციენტი K_i-ური. TI დაშიფვრის ამგვარი მიდგომა ქმნიდა ერთგვარ უხერხულობას იმ თვალსაზრისით, რომ როცა დასაშიფრი TI იყო დიდი მოცულობის, დამშიფრავი სტრიქონის ამავე რაოდენობის სიმბოლოების დამახსოვრება იწვევდა ერთგვარ უხერხულობას. აღნიშნულიდან

თავის დაღწევის მიზნით, მიმართავდნენ შემდეგ ხერხს: დასაშიფრი TI-დან ამოიღებდნენ (ამოჭრიდნენ) შედარებით მოკლე სიგრძის ფრაზას (დასამახსოვრებლად მოსახერხებელს) და ამ ფრაზის ციკლური გამეორების გზით აფორმირებდნენ დამშიფრავ სტრიქონს.

ცხრილი 1.4

დამშTIსს K(Kᵢ)	დასაშიფრი TI სტრიქონის სიმბოლოები (Sᵢ)																									
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

1.4 ცხრილში „დამშ-TI-სს **K(Kᵢ)**“ ნიშნავს, „დამშიფრავი TI-იის სტრიქონის სიმბოლოებს K(Kᵢ)“.

დაშიფვრის ალგორითმის არსი მდგომარეობს შემდეგში: ყოველ S_i და K_i სიმბოლოების წყვილს ჩაანაცვლებენ შიფრ-ტექსტის (ShifTI) D_i სიმბოლოთი, რომელიც მდებარეობს 1.4 ცხრილის S_i-ურ სიმბოლოს შესაბამისი სვეტისა და K_i-ური სიმბოლოს შესაბამისი სტრიქონის გადაკვეთაზე. ანალოგიურად აღიწერება შიფრ-ტექსტის გაშიფვრის ალგორითმი. კერძოდ: ყოველ D_i და K_i სიმბოლოების წყვილს ჩაანაცვლებენ S_i სიმბოლოთი, რომელიც განისაზღვრება 1.4 ცხრილის K_i-ურ სიმბოლოს სტრიქონში მდებარე D_i-ური სიმბოლოს შესაბამისი სვეტით [12].

განვიხილოთ კონკრეტული მაგალითი.

ვთქვათ, დასაშიფრია TI-ია „INFORMATIKA", დახურული გასაღებით - K=„VALERI". რადგან დასაშიფრ ტესტში შემავალი სიმბოლოების რაოდენობა აღემატება დამშიფრავ ტექსტში შემავალი სიმბოლოების რაოდენობას, ზემოაღნიშნული მოთხოვნების გათვალისწინებით, საჭიროა დამშიფრავ ტექსტს ციკლური გამეორების გზით დაემატოს საწყისი დამშიფრავი ტექსტი, ვიდრე არ გაუტოლდება დასაშიფრი და დამშიფრავი ტექსტების სიგრძეები. აღნიშნული პროცედურის შესრულების შედეგად მივიღებთ დამშიფრავ სტრიქონს (K): VALERIVALER.

ტექსტის დაშიფრა ხორციელდება შემდეგი წესით (იხ.ცხრ.1.5): ShifTI-ის ყოველი D_i სიმბოლო განისაზღვრება 1.4 ცხრილში TI-იის S_i სვეტისა და K_i სტრიქონის შესაბამისი უჯრედების გადაკვეთაზე მდებარე უჯრედში შეტანილი სიმბოლოთი. ასე მაგალითად, თუ ავირჩევთ, რომ i=4, მაშინ $D_i \cong$ **T** მდებარეობს, როგორც ეს 1.4 ცხრილშია ნაჩვენები, მდებარეობს $S_i \cong$ **O** სვეტისა $K_i \cong$ **E** სტრიქონის შესაბამისი უჯრედების გადაკვეთაზე (სიმბოლო $\cong$ აღნიშნავს ეკვივალენტობას).

ცხრილი 1.5

DasTI(S_i)	I	N	F	O	R	M	A	T	I	K	A
DGDG(K_i)	V	A	L	E	R	I	V	A	L	E	R
ShifTI(D_i)	D	N	Q	S	A	U	V	T	T	O	R

დაშიფრული ტექსტური ინფორმაციის გაშიფვრა ხორციელდება ანალოგიური წესით (ცხრ.1.6): TI-ის ყოველი S_i სიმბოლო განისაზღვრება 1.4 ცხრილში K_i სტრიქონში მდებარე D_i სიმბოლოს შესაბამისი სვეტის აღმნიშვნელი სიმბოლოთი. ასე მაგალითად, $K_i \cong E$ სიმბოლოს სვეტი განსაზღვრავს TI-იის S_i სიმბოლოს ანუ $S_i \cong O$.

ცხრილი 1.6

ShifTI(D_i)	**D**	**N**	**Q**	**S**	**A**	**U**	**V**	**T**	**T**	**O**	**R**
DGDG(K_i)	**V**	**A**	**L**	**E**	**R**	**I**	**V**	**A**	**L**	**E**	**R**
DasTI(S_i)	**I**	**N**	**F**	**O**	**R**	**M**	**A**	**T**	**I**	**K**	**A**

ნახ.1.2. TI-იის დაშიფვრა/გაშიფვრის პროცედურების მარეალიზებელი დისკური მოწყობილობა

დაშიფვრისა და გაშიფვრის პროცედურების ვიზუალური რეალიზაცია ადვილად წარმოსადგენია და განსახორციელებელია დისკოების გამოყენებით, რომლებიც ნაჩვენებია 1.2 ნახაზზე. აღნიშნული წესით ინფორმაციის

დაშიფვრას და გაშიფვრას ძირითადად მიმართავდნენ ფრანგი სამხედროები სამოქალაქო ომების წარმოების დროს [13].

ვიჟინერის შიფრი დიდი ხნის განმავლობაში ითვლებოდა ძალზე მდგრად შიფრად. თუმცა, ეს წარმოდგენა მასზე XIX საუკუნის დასაწყისში სრულიად გააქარწყლა კრიპტოანალიტიკოსმა კასისკიმ. ცნობილია აგრეთვე კრიპტოგრაფების მიერ ვიჯინერის შიფრის „გატეხვის“ შემთხვევები ჯერ კიდევ XVI საუკუნეში.

1.3. ვერნამის მეთოდი

გ.ვერნამმა, ამერიკის ტელეკომუნიკაციის და ტელეგრაფიის (AT & T) ფირმის თანამშრომელმა, 1917 წელს გამოიგონა, ხოლო 1919 წელს დააპატენტა ტელეგრაფიული შეტყობინებების ავტომატური დაშიფვრის სისტემა [17]. მან შექმნა მოწყობილობა (აპარატი), რომელიც ახორციელებდა შეტყობინებების დაშიფვრას დამშიფრავის გარეშე. ამით ჩაეყარა საფუძველი ე.წ. „წრფივი დაშიფვრის“ მიმართულების განვითარებას, რომელიც ითვალისწინებდა შეტყობინებების დაშიფვრისა და მისი გადაცემის პროცედურების ავტომატურად და ერთდროულად შესრულებას. რაც თავის მხრივ საგრძნობლად ზრდიდა კავშირის ოპერატიულობას. გ.ვერნამის აპარატის ძირითად ნაწილს შეადგენდა კვანძი, მარეალიზებელი ლოგიკური ფუნქციის შეკრება „ორის მოდულით“, რომელიც იყო აგებული რელეების ბაზაზე [13].

როგორც იყო აღნიშნული, გ.ვერნამის მიერ შემოთავაზებული კრიპტოსისტემა განკუთვნილი იყო ტელეგრაფური შეტყობინებების დასაშიფრავად, რომლებიც წარმოდგენილი უნდა ყოფილიყო ბოდოს კოდში, ხუთნიშნა ორობით (ბინარულ) კოდში (იხ. ცხრ.1.7).

ასოების და ციფრების ცხრილებში წრეწირები ტელეგრაფის ფირზე აღნიშნავენ ნახვრეტებს, რაც შეესაბამება S_i სიმბოლოს j-ური (j=1,2,3,4,5) ბიტის მნიშვნელობას-1(ერთს), რომელიც წარმოდგენილია ხუთნიშნა ორობით (ბინარულ) კოდში. ანალოგიურად, (წერტილი) შეესაბამება კოდში 0-ის მნიშვნელობას.

ცხრილი 1.7

ბოდოს ორიგინალური კოდი									
მმართველი სიმბოლოები									
○. ...	პრობელი, ასოების ცხრილზე გადასვლა								
.○ ...	პრობელი, ციფრების ცხრილზე გადასვლა								
○○ ...	ბოლო ნიშნის წაშლა								
ასოების ცხრილი					**ციფრების ცხრილი**				
.. ○..	A	○○ ○..	K		.. ○..	1	○. ○..	.	
.. ○○.	É	○○ ○○.	L		.. .○.	2	○. .○.	9/	
.. .○.	E	○○ .○.	M		.. ..○	3	○. ..○	7/	
.. .○○	I	○○ .○○	N		.. ○.○	4	○. ○.○	2/	
.. ○○○	O	○○ ○○○	P		.. ○○○	5	○. ○○○	'	
.. ○.○	U	○○ ○.○	Q		.. ○○.	1/	○. ○○.	:	
.. ..○	Y	○○ ..○	R		.. .○○	3/	○. .○○	?	
.○ ..○	B	○. ..○	S		.○ ○..	6	○○ ○..	(	
.○ ○.○	C	○. ○.○	T		.○ .○.	7	○○ .○.	)	
.○ ○○○	D	○. ○○○	V		.○ ..○	8	○○ ..○	-	
.○ .○○	F	○. .○○	W		.○ ○.○	9	○○ ○.○	/	
.○ .○.	G	○. .○.	X		.○ ○○○	0	○○ ○○○	+	
.○ ○○.	H	○. ○○.	Z		.○ ○○.	4/	○○ ○○.	=	
.○ ○..	J	○. ○..	—		.○ .○○	5/	○○ .○○	£	

ამრიგად, შიფრ-ტექსტის მისაღებად საკმარისია S_i (i=1,2,…,Z) სიმბოლოს ორობითი კოდის მნიშვნელობა შეიკრიბოს mod_2-ით წინასწარ შერჩეულ დახურული გასაღების ორობით კოდის მნიშვნელობასთან. ასე მაგალითად, თუ შერჩეული გასაღების ორობითი კოდი 10110-ია, ხოლო H სიმბოლოს კოდი 01110-ია, მაშინ დაშიფრულ D_i სიმბოლოს მიენიჭება მნიშვნელობა 10110 ⊕ 01110 = 11000 (სადაც, ⊕-აღნიშნავს ლოგიკურ ოპე-

რაციას შეკრებას mod_2-ით). თუ განმეორებით გავიმეორებთ იგივე პროცედურებს დაშიფრული D_i სიმბოლოსა და არჩეული გასაღების გამოყენებით მივიღებთ S_i-ურ სიმბოლოს კოდურ მნიშვნელობას (მაგალითად, 10110 $\oplus$ $\oplus$11000 = 01110).

გ.ვერნამმა ჩამოაყალიბა დახურული გასაღების მიმართ სამი მოთხოვნა:

1. დახურული გასაღებების შერჩევა უნდა ხდებოდეს თანაბარი ალბათობით;
2. დახურული გასაღების სიგრძე უნდა იყოს ტოლი დასაშიფრი TI-იის სიგრძის;
3. დახურული გასაღები უნდა გამოვიყენოთ ერთხელ.

გ.ვერნამი მოითხოვდა აგრეთვე ფირის, რომელზეც იყო დაშიფრული შეტყობინება, განადგურებას მისი გამოყენების შემდეგ.

ცნობილია აგრეთვე გ.ვერნამის ე.წ. TI-იის mod_2-ით დაშიფვრის მეთოდი (მას 1918 წელს ეწოდა ვერნამ-ვიჟინერის შიფრი), რომელიც იყო რეზულტატი მისი მრავალგზის მცდელობისა გაეუმჯობესებინა ვიჟინერის მეთოდი. შემოთავაზებულ მეთოდში გ.ვერნამის ძირითადი მოთხოვნა იყო ის, რომ დახურული გასაღების სიგრძე აუცილებლად უნდა ყოფილიყო დასაშიფრი TI-იის სიგრძის ტოლი, რაც მეთოდის მომხმარებლების ერთგვარ უკმაყოფილებას იწვევდა (იხ. 1.2 ქვეთავში).

როგორც ცეზარისა და ვიჟინერის მეთოდები, ასევე ვერნამის მეთოდიც მანიპულირებდა ლათინურ ალფაბეტზე (A,B,C,….,Z). გ.ვერნამის დაშიფვრისა და გაშიფვრის პროცედურების რეალიზაცია შეიძლება აღიწეროს მოდულური არითმეტიკის ფორმულებით (იგულისხმება, რომ სიმბოლოებს ალფაბეტში მინიჭებული აქვთ რიგითი ნომრები დაწყებული ნულიდან, იხ. 1 თავში, ცხრ.1.1), კერძოდ:

$$D_i = F((P_i^S+K_i)mod_n) \quad და \quad S_i = f((P_i^D-K_i+n)mod_n), \qquad (1.3)$$

$$D_i = F((P_i^S+K_i)mod_n) \quad და \quad S_i = f((P_i^D-K_i+n)mod_n), \qquad (1.4)$$

სადაც P_i^S და P_i^D არის S_i და D_i სიმბოლოების პოზიციები (რიგითი ნომრები) ალფაბეტში. შესაბამისად, n - ალფაბეტის სიმძლავრე, K_i - გასაღები სიმბოლოს რიგითი ნომერი ალფაბეტში, რომელიც მიუთითებს S_i სიმბოლოდან მარჯვნივ K_i პოზიციით გადანაცვლების რიცხვითი მნიშვნელობაზე), F, f - ჩანაცვლების ოპერაციის მარეალიზებელი ლოგიკური ფუნქციები.

განვიხილოთ კონკრეტული მაგალითი: ვთქვათ უნდა დაიშიფროს TI – S=„GULNARA" დახურული გასაღებით K=„VALERII". დაშიფვრისა და გაშიფვრის პროცედურული შესრულება ნაჩვენებია 1.8 და 1.9 ცხრილებში, შესაბამისად.

ცხრილი 1.8

DasTI (S_i)	G - 6	U - 20	L - 11	N - 13	A - 0	R - 17	A - 0
DGDG(K_i)	V - 21	A - 0	L - 11	E - 4	R - 17	I - 8	I - 8
ShifTI(D_i)	B - 1	T - 20	W - 22	R - 17	R - 17	Z - 25	I - 8

ცხრილი 1.9

ShifTI(D_i)	B - 1	T - 20	W - 22	R - 17	R - 17	Z - 25	I - 8
DGDG(K_i)	V - 21	A - 0	L - 11	E - 4	R - 17	I - 8	I - 8
DasTI(S_i)	G - 6	U - 20	L - 11	N - 13	A - 0	R - 17	A - 0

შევნიშნოთ, რიცხვები ცხრილებში მიუთითებს შესაბამისი სიმბოლოების რიგით ნომრებს (პოზიციებს) α ალფაბეტში (იხ. ცხრ.1.1).

დასასრულ აღვნიშნოთ, რომ ვერნამის შიფრი დღემდე ითვლება პრაქტიკულად მდგრად შიფრად [11].

1.4. შებრუნებული მატრიცის მეთოდი

შებრუნებული მატრიცის მეთოდით, ტექსტური ინფორმაციის, დასაშიფრად და გასაშიფრად გამოიყენება მოცემული α ალფაბეტიდან, არა რომელიმე სიმბოლო ან სიმბოლოთა კრებული, როგორც ეს შესრულებული იყო 1.1-1.3 ქვეთავებში - აღწერილ მეთოდებში, არამედ ორი ორგანზომილებიანი DM (დამშიფრავი) და GM (გამშიფრავი) კვადრატული მატრიცა,

რომელთა ყოველი წევრი განსაზღვრულია ნატურალურ მთელ რიცხვთა სიმრავლეზე. შევნიშნოთ, რომ შებრუნებული მატრიცის მეთოდით დაშიფვრისა და გაშიფვრის პროცედურების რეალიზაცია დაკავშირებულია რიგ რუტინულ მათემატიკურ გამოთვლებთან. აქედან გამომდინარე, მოცემულ პარაგრაფში შემოთავაზებულია დაშიფვრისა და გაშიფვრის პროცედურების მარეალიზებელი ალგორითმების შედარებით გამარტივებული ვარიანტები, რაც იძლევა საშუალებას გამოთვლები შესრულდეს, როგორც პროგრამული, ასევე ემპირიული გზით. აგრეთვე, განხორციელდეს ალგორითმების რეალიზაციის სხვადასხვა ეტაპზე მიღებული შედეგების შემოწმება და ერთმანეთთან შედარება. ამ მიზნით, შებრუნებული მატრიცის მეთოდით ტექსტური ინფორმაციის დაშიფვრისა და გაშიფვრის სადემონსტრაციო მაგალითებში განიხილება მხოლოდ სამი სტრიქონისა და სამი სვეტისაგან შემდგარი DM და GM მატრიცები (DM - დამშიფრავი ანუ საწყისი მატრიცა (1.5), რომელიც გამოიყენება TI-იის დასაშიფრად და GM - გამშიფრავი ანუ DM-ის შებრუნებული მატრიცა (1.6), რომელიც გამოიყენება შიფრ-ტექსტის გასაშიფრად), სადაც:

$$DM = \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a,_{22} \end{pmatrix}, \qquad (1.5)$$

$$GM = \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} \\ b_{1,0} & b_{1,1} & b_{1,2} \\ b_{2,0} & b_{2,1} & b_{2,2} \end{pmatrix}. \qquad (1.6)$$

ერთადერთი მოთხოვნა, რაც უნდა გავითვალისწინოთ DM მატრიცის არჩევისას არის ის, რომ მისი დეტერმინანტი (Det), რომელიც გამოითვლება ფორმულით (1.7), არ უნდა იყოს ნულის ტოლი;

Det= a[0, 0]* a[1, 1]* a[2, 2] + a[2, 0]* a[0, 1]* a[1, 2]+ a[1, 0]* a[2, 1]* a[0, 2] –

-(a[0, 2]* a[1, 1]* a[2,0]+ a[0, 0]* a[2, 1]* a[1, 2]+ a[1, 0]* a[0, 1]* a[2, 2]) . (1.7)

შევნიშნოთ, რომ თუ გამოთვლების შედეგად აღმოჩნდება, რომ Det=0, მაშინ DM მატრიცაში უნდა შეიცვალოს მინიმუმ ერთი მაინც ან მისი

რამდენიმე ელემენტის მნიშვნელობა და ეს ცვლილებები უნდა განხორციელდეს მანამ, სანამ არ დაკმაყოფილდება პირობა Det#0.

ცნობილია, რომ თუ DM მატრიცის დეტერმინანტი Det#0, მაშინ არსებობს მისი შებრუნებული GM მატრიცა, რომლის ელემენტების მნიშვნელობები გამოითვლება შემდეგი წესით:

$$
\begin{aligned}
b[0, 0] &= a[1, 1] * a[2, 2] - a[1, 2] * a[2, 1];\\
b[0, 1] &= a[0, 2] * a[2, 1] - a[0, 1] * a[2, 2];\\
b[0, 2] &= a[0, 1] * a[1, 2] - a[0, 2] * a[1, 1];\\
b[1, 0] &= a[1, 2] * a[2, 0] - a[1, 0] * a[2, 2];\\
b[1, 1] &= a[0, 0] * a[2, 2] - a[0, 2] * a[2, 0]; \qquad (1.8)\\
b[1, 2] &= a[0, 2] * a[1, 0] - a[0, 0] * a[1, 2];\\
b[2, 0] &= a[1, 0] * a[2, 1] - a[1, 1] * a[2, 0];\\
b[2, 1] &= a[0, 1] * a[2, 0] - a[0, 0] * a[2, 1];\\
b[2, 2] &= a[0, 0] * a[1, 1] - a[0, 1] * a[1, 0].
\end{aligned}
$$

ცნობილია, აგრეთვე, რომ თუ DM მატრიცის Det#0, მაშინ

$$E=DM \times GM = \begin{pmatrix} e_{0,0} & 0 & 0 \\ 0 & e_{1,1} & 0 \\ 0 & 0 & e_{2,2} \end{pmatrix}. \qquad (1.9)$$

მიღებულ E მატრიცაში (რომელსაც ერთეულოვან მატრიცას უწოდებენ), მთავარ დიაგონალზე განლაგებული ელემენტების მნიშვნელობები ერთმანეთის ტოლია ანუ $e_{0,0}=e_{1,1}=e_{2,2}$, ავღნიშნოთ მათი მნიშვნელობა K სიმბოლოთი. მაშინ ფორმულა (1.9), შეიძლება ჩაიწეროს შემდეგი სახით:

$$\begin{pmatrix} e_{00} & 0 & 0 \\ 0 & e_{11} & 0 \\ 0 & 0 & e_{22} \end{pmatrix} = K \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = K \times E\,. \qquad (1.10)$$

E მატრიცის ელემენტები გამოითვლება:

$$
\begin{aligned}
e[0, 0] &= a[0,0]*b[0,0]+a[0,1]*b[1,0]+a[0,2]*b[2,0];\\
e[0, 1] &= a[0,0,]*b[0,1]+a[0,1]*b[1,1]+a[0,2]*b[2,1];\\
e[0, 2] &= a[0,0]*b[0,2]+a[0,1]*b[1,2]+a[0,2]*b[2,2];\\
e[1, 0] &= a[1,0]*b[0,0]+a[1,1]*b[1,0]+a[1,2]*b[2,0];
\end{aligned}
$$

e[1, 1] = a[1,0]*b[0,1]+a[1,1]*b[1,1]+a[1,2]*b[2,1]; (1.11)

e[1, 2] = a[1,0]*b[0,2]+a[1,1]*b[1,2]+a[1,2]*b[2,2];

e[2, 0] =a[2,0]*b[0,0]+a[2,1]*b[1,0]+a[2,2]*b[2,0];

e[2, 1] = a[2,0]*b[0,1]+a[2,1]*b[1,1]+a[2,2]*b[2,1];

e[2, 2] = a[2,0]*b[0,2]+a[2,1]*b[1,2]+a[2,2]*b[2,2].

განვიხილოთ დაშიფვრისა და გაშიფვრის შესრულების პროცედურების რეალიზაციის ძირითადი ეტაპები. TI-ის დასაშიფრად საჭიროა:

1. შემოწმდეს დასაშიფრი TI-იის (TI={S_i}, სადაც i=0,1,2,…,Z) სტრიქონის სიგრძე, რომელიც უნდა აკმაყოფილებდეს პირობას;

$$(Z+1)\%3=0\ , \qquad (1.12)$$

წინააღმდეგ შემთხვევაში, თუ (1.12) პირობა არ სრულდება, მაშინ TI-იის სტრიქონს ბოლოში უნდა დაემატოს ერთი ან ორი ნებისმიერი სიმბოლო α ალფაბეტიდან (იხ. ცხრ.1.1), რომ დაკმაყოფილდეს პირობა:

$$(Z_1+1)\%3=0\ , \qquad (1.13)$$

სადაც Z_1 ახლად ფორმირებული TI-იის სტრიქონის სიგრძეა (Z_1 >= Z).

2. ყოველ S_i სიმბოლოს α-ალფაბეტიდან (იხ.ცხრ.1.1) მიენიჭოს მისი შესაბამისი რიგითი ნომერის მნიშვნელობა (კოდი). შედეგად მიიღება TI-იის კოდური სტრიქონი:

$$DasTIn =\{ S_i^n\} \quad \text{სადაც } i=0,1,2,\ldots,Z_1\ . \qquad (1.14)$$

3. DasTIn შემავალი კოდების მნიშვნელობები ჯგუფდება მარცხნიდან (S_0^n კოდის მნიშვნელობიდან) მარჯვნივ ტეტრადებად (ბლოკებად). მიღებულ მიმდევრობაში:

$$T_0, T_1, T_2,\ldots,T_t\ , \qquad (1.15)$$

T_j (j=0,1,2,..,t) -არის ბლოკი (ტეტრადა), t - ამ ბლოკების საერთო რაოდენობა.

თითოეულ ბლოკში გაერთიანებულია ყოველი მომდევნო სამი სიმბოლოს შესაბამისი კოდური მნიშვნელობები. კერძოდ, { S_{3*j}^n , S_{3*j+1}^n, S_{3*j+2}^n}.

4. ყოველ T_j ={ S_{3*j}^n , S_{3*j+1}^n, S_{3*j+2}^n} ბლოკში შემავალი დასაშიფრი TI-იის კოდური მნიშვნელობისათვის ფორმირდება ბლოკი U_j={ D_{3*j}^k, D_{3*j+1}^k, D_{3*j+2}^k},

რომლის ელემენტები შეადგენს შიფრ-ტექსტის კოდის მნიშვნელობებს, მიღებულს შემდეგი გამოთვლების გზით:

$$D_{3*j}^{n} = S_{3*j}^{n} * a[0, 0] + S_{3*j+1}^{n*} a[1, 0] + S_{3*j+2}^{n} * a[2, 0];$$
$$D_{3*j+1}^{n} = S_{3*j}^{n} * a[0, 1] + S_{3*j+1}^{n*} a[1, 1] + S_{3*j+2}^{n} * a[2, 1]; \quad (1.16)$$
$$D_{3*j+2}^{n} = S_{3*j}^{n} * a[0, 2] + S_{3*j+1}^{n*} a[1, 2] + S_{3*j+2}^{k\backslash n} * a[2, 2],$$

სადაც a[*, *] – DM მატრიცის ელემენტებია.

ასეთი წესით გამოთვლილი U_j , j =0,1,..., t ბლოკების მიმდევრობა U_0, U_1 ,..., U_t არის დაშიფრულ TI-ში (შიფრ-ტექსტი) შემავალი გამოთვლილი კოდური მნიშვნელობების ერთობლიობა, ShifTIn= $\{D_i^n\}$.

ShifTI გასაშიფრავად საჭიროა:

5. ყოველი U_j,j=0,1,...,t ბლოკში შემავალი კოდური მნიშვნელობების მიხედვით და GM მატრიცის გამოყენებით. შესაბამისად, გამოვთვალოთ $T_j = \{S_{3*j}^{n}, S_{3*j+1}^{n}, S_{3*j+2}^{n}\}$ ბლოკში შემავალი დასაშიფრი TI-იის კოდური მნიშვნელობა, შემდეგი წესით:

$$S_{3*j}^{n} = (D_{3*j}^{n} * b[0, 0] + D_{3*j+1}^{n*} b[1, 0] + D_{3*j+2}^{n} * b[2, 0]) : K;$$
$$S_{3*j+1}^{n} = (D_{3*j}^{n} * b[0, 1] + D_{3*j+1}^{n*} b[1, 1] + D_{3*j+2}^{n} * b[2, 1]) : K; \quad (1.17)$$
$$S_{3*j+2}^{n} = (D_{3*j}^{n} * b[0, 2] + D_{3*j+1}^{n*} b[1, 2] + D_{3*j+2}^{n} * b[2, 2]) : K,$$

სადაც b[*, *] – GM მატრიცის ელემენტებია.

1.17 ფორმულის შედეგად ჩატარებული გამოთვლებით მიიღება საწყის TI-ში შემავალი სიმბოლოების კოდური მნიშვნელობები

$$DasTIn = \{ S_i^n \}, \text{ სადაც } i=0,1,2,\ldots,Z_1, \quad (1.18)$$

რომლებსაც თავის მხრივ α-ალფაბეტიდან ცალსახად შეესაბამება შესაბამისი სიმბოლოები ანუ მიიღება ახლად ფორმირებული TI, რომლიდანაც პირველი Z სიმბოლო ეკუთვნის საწყის დასაშიფრ TI-ას, რომელზეც განხორციელდა ყველა ზემოთ აღწერილი მანიპულაციები [13,14].

განვიხილოთ კონკრეტულ მაგალითზე ტექსტური ინფორმაციის შებრუნებული მატრიცის მეთოდით დაშიფვრისა და გაშიფვრის აღწერილი პროცედურები. ვთქვათ დასაშიფრია TI - „GEORGIA“ და შერჩეულია დამშიფრავი მატრიცა DM:

$$\mathrm{DM} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 3 & 1 \\ 2 & 1 & 0 \end{pmatrix}.$$

გამოვთვალოთ DM მატრიცის დეტერმინანტი (1.7) ფორმულით:

Det=1*3*0+2*2*1+0*1*3-(2*3*3+1*1*1+0*2*0)=0+4+0-(18+1+0)=4-19= -15 ≠0 .

ვინაიდან DM მატრიცის დეტერმინანტი Det≠0-ს, ე.ი. მას აქვს შებრუნებული (გამშიფრავი) GM მატრიცა, რომლის ელემენტები გამოითვლება (1.8) ფორმულით:

$$\mathrm{GM} = \begin{pmatrix} 3*0-1*1 & 3*1-2*0 & 2*1-3*3 \\ 1*2-0*0 & 1*0-2*3 & 3*0-1*1 \\ 0*1-3*2 & 2*2-1*1 & 1*3-2*0 \end{pmatrix} = \begin{pmatrix} -1 & 3 & -7 \\ 2 & -6 & -1 \\ -6 & 3 & 3 \end{pmatrix}$$

მიღებული შედეგების შემოწმების მიზნით გამოვთვალოთ ერთეულოვანი მატრიცა:

$$E = \mathrm{DM} * \mathrm{GM} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 3 & 1 \\ 2 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} -1 & 3 & -7 \\ 2 & -6 & -1 \\ -6 & 3 & 3 \end{pmatrix} =$$

$$\begin{pmatrix} 1*(-1)+2*2+3*(-6) & 1*3+2*(-6)+3*3 & 1*(-7)+2*(-1)+3*3 \\ 0*(-1)+2*3+1*(-6) & 0*3+3*(-6)+1*3 & 0*(-7)+3*(-1)+1*3 \\ 2*(-1)+1*2+0*(-6) & 2*3+1*(-6)+0*3 & 2*(-7)+1*(-1)+0*3 \end{pmatrix} =$$

$$= \begin{pmatrix} -15 & 0 & 0 \\ 0 & -15 & 0 \\ 0 & 0 & -15 \end{pmatrix} = (-15) \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = K \times E, \quad \text{K= -15.}$$

მოცემული TI-იის - „GEORGIA“-ს დაშიფვრის პროცედურა შედგება შემდეგი ეტაპებისაგან:

1. მოცემულ TI-ას ბოლოში უნდა დაემატოს ორი სიმბოლო α ალფაბეტიდან (ვთქვათ, ეს სიმბოლოებია C და M), ვინაიდან დასაშიფრი საწყისი ტექსტის სტრიქონის სიგრძე უნდა იყოს სამის ჯერადი. ამრიგად, დასაშიფრი იქნება TI: „GEORGIACM“;

2. TI-იის ყოველ S_i სიმბოლოს შეუსაბამოთ მისი რიგითი ნომერი (კოდი) α ალფაბეტიდან. მივიღებთ საწყისი TI კოდურ ფორმას DasTIn: 6, 4, 14, 17, 6, 8, 0, 2, 12, რომელიც უნდა დაიშიფროს;

3. DasTIn შემავალი კოდების მნიშვნელობები დავყოთ სამ-სამი კოდისგან შემდგარ ბლოკებად (ბლოკების რაოდენობა t=3):

$T_0\ (S_{3*0}^n) = 6, 4, 14$; $T_1\ (S_{3*1}^n) = 17, 6, 8$; $T_2\ (S_{3*2}^n) = 0, 2, 12$;

4. T_0, T_1, T_2 ბლოკებში შემავალ კოდებზე (1.16)-ით გამოთვლების შედეგად მიიღება შიფრ-ტექსტში შემავალი რიცხვითი კოდები:

$D_0^n = T_0\ (S_{3*0}^n) \times DM = 6*1 + 4*0 + 14*2 = 34 = U_0^n$;

$D_1^n = T_0\ (S_{3*0}^n) \times DM = 6*2 + 4*3 + 14*1 = 38 = U_1^n$;

$D_2^n = T_0\ (S_{3*0}^n) \times DM = 6*3 + 4*1 + 14*0 = 22 = U_2^n$;

$D_3^n = T_1\ (S_{3*1}^n) \times DM = 17*1 + 6*0 + 8*2 = 33 = U_3^n$;

$D_4^n = T_1\ (S_{3*1}^n) \times DM = 17*2 + 6*3 + 8*1 = 60 = U_4^n$;

$D_5^n = T_1\ (S_{3*1}^n) \times DM = 17*3 + 6*1 + 8*0 = 57 = U_5^n$;

$D_6^n = T_2\ (S_{3*2}^n) \times DM = 0*1 + 2*0 + 12*2 = 24 = U_6^n$;

$D_7^n = T_2\ (S_{3*2}^n) \times DM = 0*2 + 2*3 + 12*1 = 18 = U_7^n$;

$D_8^n = T_2\ (S_{3*2}^n) \times DM = 0*3 + 2*1 + 12*0 = 2 = U_8^n$.

5. მიღებულ ბლოკებში U_0, U_1, U_2 შემავალი შიფრ-ტექსტის რიცხვით კოდებზე ფორმულით (1.17) გამოთვლების შესრულების შედეგად :

$(U_0 \times GM)/K = (34*(-1) + 38*2 + 22*(-6)) : (-15) = (-90) / (-15) = 6 = T_0\ (S_{3*0}^n)$;

$(U_0 \times GM)/K = (34*3 + 38*(-6) + 22*3) : (-15) = (-60) / (-15) = 4 = T_0\ (S_{3*0}^n)$;

$(U_0 \times GM)/K = (34*(-7) + 38*(-1) + 22*3) : (-15) = (-210) / (-15) = 14 = T_0\ (S_{3*0}^n)$;

$(U_1 \times GM)/K = (33*(-1) + 60*2 + 57*(-6)) : (-15) = (-255) / (-15) = 17 = T_1\ (S_{3*1}^n)$;

$(U_1 \times GM)/K = (33*3 + 60*(-6) + 57*3) : (-15) = (-90) / (-15) = 6 = T_1\ (S_{3*1}^n)$;

$(U_1 \times GM)/K = (33*(-7) + 60*(-1) + 57*3) : (-15) = (-120) / (-15) = 8 = T_1\ (S_{3*1}^n)$;

$(U_2 \times GM)/K = (24*(-1) + 18*2 + 2*(-6)) : (-15) = 0 / (-15) = 0 = T_2\ (S_{3*2}^n)$;

$(U_2 \times GM)/K = (24*3 + 18*(-6) + 2*3) : (-15) = (-30) / (-15) = 2 = T_2\ (S_{3*2}^n)$;

$(U_2 \times GM)/K = (24*(-7) + 18*(-1) + 2*3) : (-15) = (-180) / (-15) = 12 = T_2\ (S_{3*2}^n)$.

მიღებული კოდების $S_{3*0}^n(S_0^n, S_1^n, S_2^n)$, $S_{3*1}^n(S_3^n, S_4^n, S_5^n)$, $S_{3*2}^n(S_6^n, S_7^n, S_8^n)$ მნიშვნელობების მიხედვით 1.1 ცხრილით განისაზღვრება საწყისი დასაშიფრი TI.

განვიხილოთ კიდევ ერთი ვარიანტი TI დაშიფვრისა და გაშიფვრის პროცედურების რეალიზაციის შებრუნებული მატრიცის მეთოდით საოფისე პროგრამის MS Excel-ის ფუნქციების გამოყენებით, რომელიც ნაჩვენებია 1.3 ნახაზზე, სადაც:

ა) დეტერმინანტი Det=-15, გამოთვლილია საწყისი მატრიცის DM მონაცემების მიხედვით ფუნქციით: MDETERM ();

ბ) შებრუნებული მატრიცა გამოთვლილია ფუნქციით: MINVERSE ();

გ) ერთეულოვანი მატრიცა გამოთვლილია ფუნქციით: MMULT ().

1.3 ნახაზზე ნაჩვენებია TI="KEKELIA" დაშიფვრისა და გაშიფვრის პროცედურული რეალიზაცია.

საწყისი მატრიცა

DM=	1	2	3
	0	3	1
	2	1	0

შებრუნებული მატრიცა

GM=	0.066667	-0.2	0.466666667
	-0.13333	0.4	0.066666667
	0.4	-0.2	-0.2

ერთეულოვანი მატრიცა

E=	1	-1.11022E-16	-1.11022E-16
	0	1	0
	0	0	1

დეტერმინანტი = -15

TI="KEKELIA" დაშიფვრის/გაშიფვრის მაგალითი

DasTI	K	E	K	E	L	I	A	A	A
DasTIn	10	4	10	4	11	8	0	0	0
ShifTIn	30	42	34	20	49	23	0	0	0
DasTIn	10	4	10	4	11	8	0	0	0
DasTI	K	E	K	E	L	I	A	A	A

ნახ.1.3. TI დაშიფვრა და გაშიფვრა MS Excel-ის ფუნქციების გამოყენებით

1. TI-ის დაშიფვრის პროცედურული რეალიზაცია შედგება შემდეგი ეტაპებისაგან:

1.1. მოცემულ TI ბოლოში ემატება ორი სიმბოლო α ალფაბეტიდან. ვთქვათ A და A, როგორც ეს ნაჩვენებია 1.3 ნახაზზე, ვინაიდან დასაშიფრი საწყისი ტექსტის სტრიქონის სიგრძე უნდა იყოს სამის ჯერადი;

1.2. 1.1 ცხრილის მიხედვით DasTI-იის ყოველ სიმბოლოს S_i ცალსახად შეესაბამება კოდური რიცხვითი მნიშვნელობა, რომლებიც შეიტანება DasTIn სტრიქონში (იხ. ნახ.1.3);

1.3. TI-იის ყოველ სიმბოლოს დაშიფვრის შედეგად შეესაბამება, როგორც შესაბამისი რიცხვითი D_i^n (i=0,2,…,8) მნიშვნელობა, რომელიც გამოითვლება შემდეგი წესით:

$D_0^n = 10*1+4*0 +10*2 = 30;$

$D_1^n = 10*2+4*3 +10*1 = 42;$

$D_2^n = 10*3+4*1 +10*0 = 34;$

$D_3^n = 4*1 +11*0+8*2 = 20;$

$D_4^n = 4*2 +11*3+8*1 = 49;$

$D_5^n = 4*3 +11*1+8*0 = 23;$

$D_6^n = 0*1 +0*0 +0*2 = 0;$

$D_7^n = 0*2 +0*3 +0*1 = 0;$

$D_8^n = 0*3 +0*1 +0*0 = 0.$

მიღებული გამოთვლების შედეგები შეტანილია სტრიქონში, რომელიც აღნიშნულია ShfTIn-ით (იხ. ნახ.1.3), რომლის მიხედვითაც ფორმირდება ბლოკები: U_0 , U_1 , U_t, სადაც U_0={30, 42, 34}, U_0={20, 49, 23}, U_0={0, 0, 0}.

2. შიფრ-ტექსტის გაშიფვრის პროცედურული რეალიზაცია შედგება შემდეგი ეტაპებისაგან:

2.1. შიფრ-ტექსტის ყოველ ელემენტის (ბლოკების U_0, U_1, U_t) გაშიფვრის შედეგად მიიღება დასაშიფრი TI სიმბოლოს შესაბამისი კოდური მნიშვნელობა S_i^n (i=0,2,…,8), რომლიც გამოითვლება შემდეგი წესით:

$S_0^n = (30* 0.066667 +42*(-0.13333) +34* 0.4) = 10 ;$

$S_1^n = (30*(-0.2) +42*0.4 +34*(-0.2)) = 4 ;$

$S_2^n = (30* 0.466666667 +42* 0.066666667 +34*(-.02)) = 10 ;$

$S_3^n = (20* 0.066667 +49*(-0.13333) +23* 0.4) = 4 ;$

$S_4^n = (20*(-0.2) +49*0.4 +23*(-0.2)) = 11 ;$

$S_5^n = (20* 0.466666667 +49* 0.066666667 +23*(-0.2)) = 8 ;$

$S_6^n = (0* 0.066667 +0*(-0.13333) +0* 0.4) = 0 ;$

$S_7^n = (0*(-0.2) +0*0.4 +0*(-0.2)) = 0 ;$

$S_8^n = (0* 0.466666667 +0* 0.066666667 +0*(-0.2)) = 0 .$

მიღებული გამოთვლების შედეგები შეტანილია სტრიქონებში, რომლებიც შესაბამისად აღნიშნულია DasTIn და DasTI -ით (იხ. ნახ.1.3).

თავი 2. სიმეტრიული სისტემის უნივერსალური მოდელი

მეორე თავში TI-იის ფორმირების, დაშიფვრისა და გაშიფვრის პროცედურების შესრულების მიზნით α-ალფაბეტის (იხ. 1.1-1.4) ნაცვლად გამოიყენება გაფართოებული ორი ალფაბეტი - β და η, შედგენილი ინგლისური, ქართული და რუსული ენების სიმბოლოების ნაკრებებისაგან.

β ალფაბეტის ყოველ სიმბოლოს პროგრამულად ენიჭება შიგამანქანური ანუ (კომპიუტერული) კოდის რიცხვითი მნიშვნელობა, ხოლო η ალფაბეტისათვის ანალოგიური მინიჭება სრულდება ემპირულად (ცხრ. 2.11). TI-იის დაშიფვრისა და გაშიფვრის შესასრულებლად β და η ალფაბეტებში შემავალ სიმბოლოებზე ხორციელდება რიგი არითმეტიკული და ლოგიკური ოპერაციები. კერძოდ, არითმეტიკული: შეკრება-გამოკლება, გამრავლება და ლოგიკური: შეკრება ორის მოდულით (mod_2). განხილულია მათემატიკური აპარატი, რომელსაც საფუძვლად უდევს ალგორითმული (მიკროპროგრამული) ალგებრის სისტემის - ოპერატორული ალგებრის და პირობის ალგებრის ცნება, რომელთა ტერმინებშიც შეიძლება იყოს აღწერილი სხვადასხვა სახის ალგორითმული პროცესი [18,19]. კერძოდ, TI-იის დაშიფვრისა და გაშიფვრის ალგორითმები, რომელთა რიცხვს, მაგალითად, მიეკუთვნება შემოთავაზებული უნივერსალური მოდელის მიხედვით დამუშავებული (ცეზარის, ვიჟინერისა და ვერნამის მეთოდების ბაზაზე), შედგენილი ალგორითმი. ეს უკანასკნელი წარმოდგენილია მიკროპროგრამული სახით, დამუშავებულია მისი სქემო-ტექნიკური რეალიზაციის ერთ-ერთი ვარიანტი [19]. შემოთავაზებულია აგრეთვე η ალფაბეტში მოცემული TI-იის დაშიფვრისა და გაშიფვრის პროცედურული რეალიზაცია საოფისე პროგრამის MS Excel-ის ინსტრუმენტულ ინტეგრირებულ გარემოში.

2.1. ტექსტური ინფორმაციის წარმოდგენის კოდური ფორმა

კრიპტოგრაფიაში ცნობილი მეთოდების მარეალიზებელი ალგორითმების უმრავლესობა ორიენტირებულია ერთი მხრივ მხოლოდ ლათინური ალფაბეტის გამოყენებაზე, მეორე მხრივ კი ისეთი მარტივი მანიპულაციების შესრულებაზე, როგორიცაა სიმბოლოს რამდენიმე პოზიციით დაძვრა მარჯვნივ ან მარცხნივ, სიმბოლოების ჩანაცვლება გარკვეული წესით და ა.შ.

თანამედროვე კომპიუტერული ტექნოლოგიების შესაძლებლობები იძლევა საშუალებას საგრძნობლად გაფართოვდეს კრიპტოგრაფიის ამოცანების განსაზღვრის არე. ასე მაგალითად, ქვემოთ აღწერილ მეთოდებში, დაშიფვრა/გაშიფვრის ალგორითმებში, გამოყენებულია არა მარტო ინგლისური ენის - EN(α - ლათინური ალფაბეტი), არამედ ეროვნული (რუსული - RU და ქართული - KA) ენების ფონტის შემცველი სიმბოლოების ნაკრებები და მათი შესაბამისი რიცხვითი კოდების მნიშვნელობები, რომლებიც ფიქსირდება კომპიუტერში პროგრამის Microsoft Visual studio 2010 (Default პრინციპით) ინსტალირების შედეგად. შევნიშნოთ, რომ შესაბამისი ენების {EN, RU და KA} გადამრთველი (ამომრჩევი) ღილაკები განთავსებულია Taskbar-ის ველში, მარჯვენა მხარეს. გამოყენებულია აგრეთვე რიგი არითმეტიკული და ლოგიკური ოპერაციები ამ ნაკრებებზე სხვადასხვა სახის მანიპულაციის ჩასატარებლად.

2.1-2.3 ცხრილებში მოცემულია, ზემოაღნიშნული ენების ფონტებში შემავალი სიმბოლოების ნაკრებები და აგრეთვე, ამ სიმბოლოების შესაბამისი კომპიუტერული (რიცხვითი) კოდები, რომლებიც გამოიყენება ტექსტური ინფორმაციის დასამუშავებლად (დასაშიფვრად და გასაშიფვრად). 2.4 ცხრილში ნაჩვენებია 2.1-2.3 ცხრილებში მოტანილ ფონტებში შემავალი სიმბოლოების ჯამური ნაკრები. მოცემულს ვუწოდოთ აღნიშნული წესით შედგენილი სიმბოლოების ჯამური ნაკრების, ალფაბეტი - β.

ცხრილი 2.1

0. 1. 2. ┐ 3. └ 4. ┘ 5. | 6. – 7. • 8. ◘ 9. 10.
11. ♂ 12. ♀ 13.
14. ♫ 15. ☼ 16. ┼ 17. ◄ 18. ↕ 19. ‼ 20. ¶
21. ┴ 22. ┬ 23. ┤ 24. ↑ 25. ├ 26. → 27. ← 28. 29. 30. 31.

32. 33. ! 34. " 35. # 36. $ 37. % 38. 39. ' 40. (41.) 42. *
43. + 13 44. , 13 46. . 47. / 48. 0 49. 1 50. 2 51. 3 52. 4
53. 5 54. 6 55. 7 56. 8 57. 9 58. : 59. ; 60. < 61. = 62. >
63. ? 64. @ 65. A 66. B 67. C 68. D 69. E 70. F 71. G 72. H
73. I 74. J 75. K 76. L 77. M 78. N 79. O 80. P 81. Q 82. R
83. S 84. T 85. U 86. V 87. W 88. X 89. Y 90. Z 91. [92. \
93.] 94. ^ 95. _ 96. ` 97. a 98. b 99. c 100. d 101. e 102. f
103. g 104. h 105. i 106. j 107. k 108. l 109. m 110. n 111. o 112. p
113. q 114. r 115. s 116. t 117. u 118. v 119. w 120. x 121. y 122. z
123. { 124. | 125. } 126. ~

ინგლისური ენის სიმბოლოების კრებული და მათი კომპიუტერული კოდები

ცხრილი 2.2

4304. ა 4305. ბ 4306. გ 4307. დ 4308. ე 4309. ვ 4310. ზ 4311. თ 4312. ი 4313. კ 4314. ლ
4315. მ 13 4316. ნ 13 4318. ო 4319. პ 4320. რ 4321. ს 4322. ტ 4323. უ 4324. ფ
4325. ქ 4326. ღ 4327. ყ 4328. შ 4329. ჩ 4330. ც 4331. ძ 4332. წ 4333. ჭ 4334. ხ
4335. ჯ 4336. ჰ

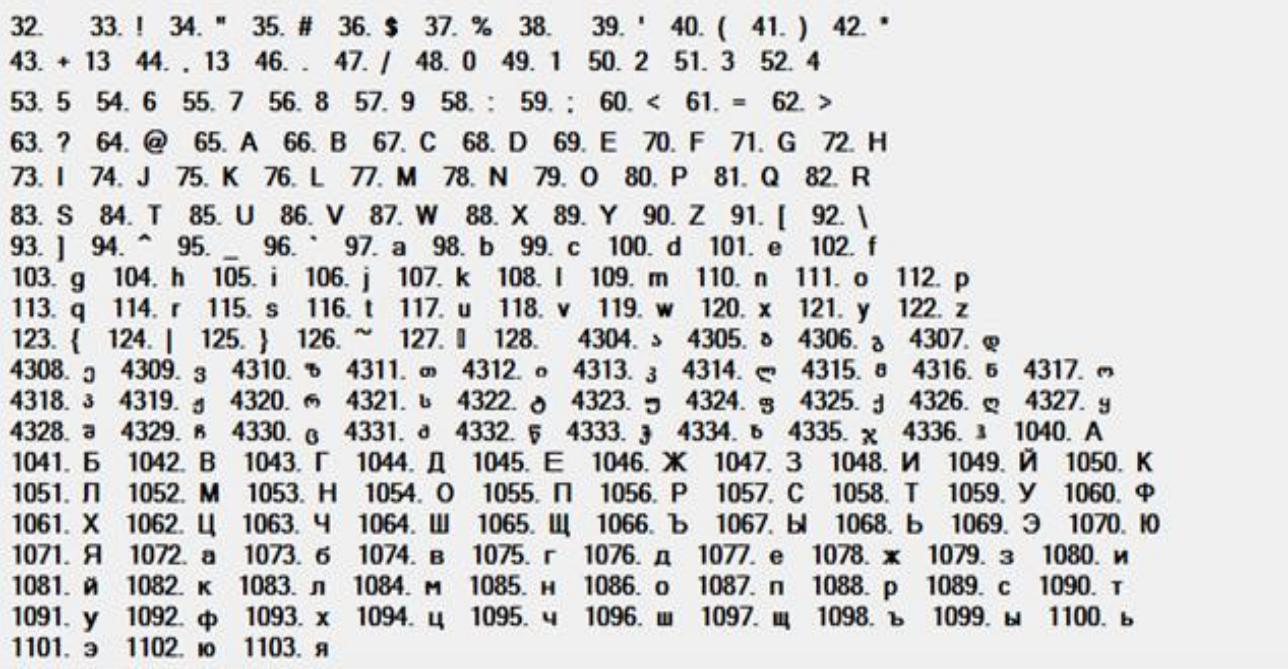
ქართული ენის სიმბოლოების კრებული და მათი კომპიუტერული კოდები

ცხრილი 2.3

1040. А 1041. Б 1042. В 1043. Г 1044. Д 1045. Е 1046. Ж 1047. З 1048. И 1049. Й 1050. К
1051. Л 13 1052. М 13 1054. О 1055. П 1056. Р 1057. С 1058. Т 1059. У 1060. Ф
1061. Х 1062. Ц 1063. Ч 1064. Ш 1065. Щ 1066. Ъ 1067. Ы 1068. Ь 1069. Э 1070. Ю
1071. Я 1072. а 1073. б 1074. в 1075. г 1076. д 1077. е 1078. ж 1079. з 1080. и
1081. й 1082. к 1083. л 1084. м 1085. н 1086. о 1087. п 1088. р 1089. с 1090. т
1091. у 1092. ф 1093. х 1094. ц 1095. ч 1096. ш 1097. щ 1098. ъ 1099. ы 1100. ь
1101. э 1102. ю 1103. я

რუსული ენის სიმბოლოების კრებული და მათი კომპიუტერული კოდები

ცხრილი 2.4



ინგლისური, ქართული და რუსული ენების სიმბოლოების ჯამური კრებული და მათი კომპიუტერული კოდები

2.5-2.10 ცხრილებში ნაჩვენებია, ყველა შესაძლო ვარიანტი სიმეტრიული სისტემის მეთოდებით TI დაშიფვრის შედეგად მიღებული შიფრტექსტში - ShifTI და ShifTIK შემავალი სიმბოლოების D_i ნაკრებები და მათი შესაბამისი კომპიუტერული (რიცხვითი) კოდების D_i^k მნიშვნელობები. აღვნიშნოთ ეს ნაკრებები E_e, R_e, R_r, G_e, G_r, G_g შესაბამისად, ხოლო მათი ჯამური ნაკრებები სიმბოლოთი γ და ვუწოდოთ მას γ ალფაბეტი. ამრიგად, TI დაშიფვრა (გაშიფვრა) ეს არის პროცედურების რეალიზაცია β(γ) ალფაბეტის სიმბოლოების გარდასახვისა γ(β) ალფაბეტის სიმბოლოებში. შევნიშნოთ, რომ გარდასახვების მარეალიზებელ პროცედურებად მოცემულ შემთხვევაში და მომავალში შემოთავაზებულ მასალებში ძირითადად გამოიყენება არითმეტიკული ოპერაცია შეკრება და გამოკლება, ზოგჯერ კი ლოგიკური ოპერაცია შეკრება ორის მოდულით.

ცხრილი 2.5

64. @ 65. A 66. B 67. C 68. D 69. E 70. F 71. G 72. H 73. I 74. J
75. K 13 76. L 13 78. N 79. O 80. P 81. Q 82. R 83. S 84. T
85. U 86. V 87. W 88. X 89. Y 90. Z 91. [92. \ 93.] 94. ^
95. _ 96. ` 97. a 98. b 99. c 100. d 101. e 102. f 103. g 104. h
105. i 106. j 107. k 108. l 109. m 110. n 111. o 112. p 113. q 114. r
115. s 116. t 117. u 118. v 119. w 120. x 121. y 122. z 123. { 124. |
125. } 126. ~ 127. 128. 129. 130. 131. 132. 133. 134.
135. 136. 137. 138. 139. 140. 141. 142. 143. 144.
145. 146. 147. 148. 149. 150. 151. 152. 153. 154.
155. 156. 157. 158. 159. 160. 161. ¡ 162. ¢ 163. £ 164. ¤
165. ¥ 166. ¦ 167. § 168. ¨ 169. © 170. ª 171. « 172. ¬ 173. - 174. ®
175. ¯ 176. ° 177. ± 178. ² 179. ³ 180. ´ 181. µ 182. ¶ 183. · 184. ¸
185. ¹ 186. º 187. » 188. ¼ 189. ½ 190. ¾ 191. ¿ 192. À 193. Á 194. Â
195. Ã 196. Ä 197. Å 198. Æ 199. Ç 200. È 201. É 202. Ê 203. Ë 204. Ì
205. Í 206. Î 207. Ï 208. Ð 209. Ñ 210. Ò 211. Ó 212. Ô 213. Õ 214. Ö
215. × 216. Ø 217. Ù 218. Ú 219. Û 220. Ü 221. Ý 222. Þ 223. ß 224. à
225. á 226. â 227. ã 228. ä 229. å 230. æ 231. ç 232. è 233. é 234. ê
235. ë 236. ì 237. í 238. î 239. ï 240. ð 241. ñ 242. ò 243. ó 244. ô
245. õ 246. ö 247. ÷ 248. ø 249. ù 250. ú 251. û 252. ü 253. ý

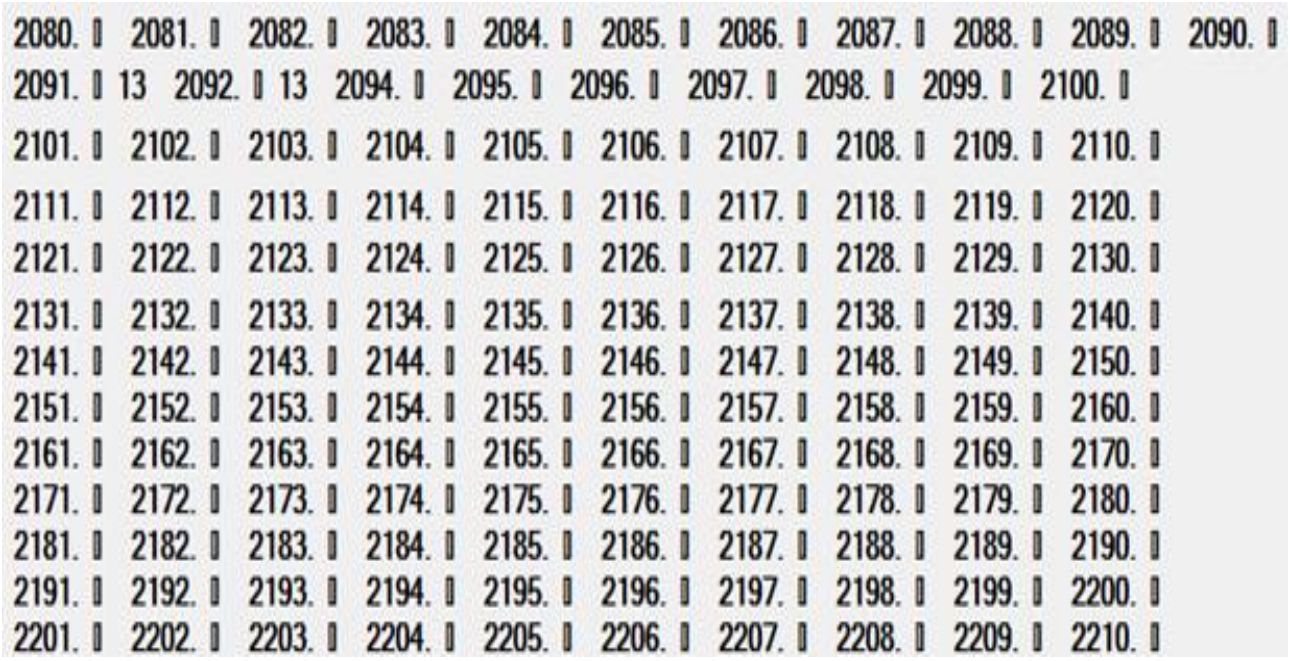
ინგლისური სიმბოლოს დაშიფვრისა (S_i და K_i) და გაშიფვრის (D_i და K_i) ამსახველი სიმბოლოების კრებული (E_E)

ცხრილი 2.6

1072. а 1073. б 1074. в 1075. г 1076. д 1077. е 1078. ж 1079. з 1080. и 1081. й 1082. к
1083. л 13 1084. м 13 1086. о 1087. п 1088. р 1089. с 1090. т 1091. у 1092. ф
1093. х 1094. ц 1095. ч 1096. ш 1097. щ 1098. ъ 1099. ы 1100. ь 1101. э 1102. ю
1103. я 1104. ѐ 1105. ё 1106. ђ 1107. ѓ 1108. є 1109. ѕ 1110. і 1111. ї 1112. ј
1113. љ 1114. њ 1115. ћ 1116. ќ 1117. ѝ 1118. ў 1119. џ 1120. Ѡ 1121. ѡ 1122. Ѣ
1123. ѣ 1124. Ѥ 1125. ѥ 1126. Ѧ 1127. ѧ 1128. Ѩ 1129. ѩ 1130. Ѫ 1131. ѫ 1132. Ѭ
1133. ѭ 1134. Ѯ 1135. ѯ 1136. Ѱ 1137. ѱ 1138. Ѳ 1139. ѳ 1140. Ѵ 1141. ѵ 1142. Ѷ
1143. ѷ 1144. Ꙋ 1145. ꙋ 1146. Ѹ 1147. ѹ 1148. Ѻ 1149. ѻ 1150. Ѽ 1151. ѽ 1152. Ҁ
1153. ҁ 1154. ҂ 1155. ҃ 1156. ҄ 1157. ҅ 1158. ҆ 1159. ҇ 1160. ◌ 1161. ◌ 1162. Ҋ
1163. ҋ 1164. Ҍ 1165. ҍ 1166. Ҏ 1167. ҏ 1168. Ґ 1169. ґ 1170. Ғ 1171. ғ 1172. Ҕ
1173. ҕ 1174. Ж 1175. ж 1176. Ҙ 1177. ҙ 1178. Қ 1179. қ 1180. Ҝ 1181. ҝ 1182. Ҟ
1183. ҟ 1184. Ҡ 1185. ҡ 1186. Ң 1187. ң 1188. Ҥ 1189. ҥ 1190. Ҧ 1191. ҧ 1192. Ҩ
1193. ҩ 1194. Ҫ 1195. ҫ 1196. Ҭ 1197. ҭ 1198. Ү 1199. ү 1200. Ұ 1201. ұ 1202. Ҳ
1203. ҳ 1204. Ҵ 1205. ҵ 1206. Ҷ 1207. ҷ 1208. Ҹ 1209. ҹ 1210. h 1211. h 1212. Ҽ
1213. ҽ 1214. Ҿ 1215. ҿ 1216. I 1217. Ӂ 1218. ӂ 1219. Ӄ 1220. ӄ 1221. Ӆ 1222. ӆ
1223. Ӈ 1224. ӈ 1225. Ӊ 1226. ӊ 1227. Ӌ 1228. ӌ 1229. Ӎ 1230. ӎ 1231. I 1232. Ӑ

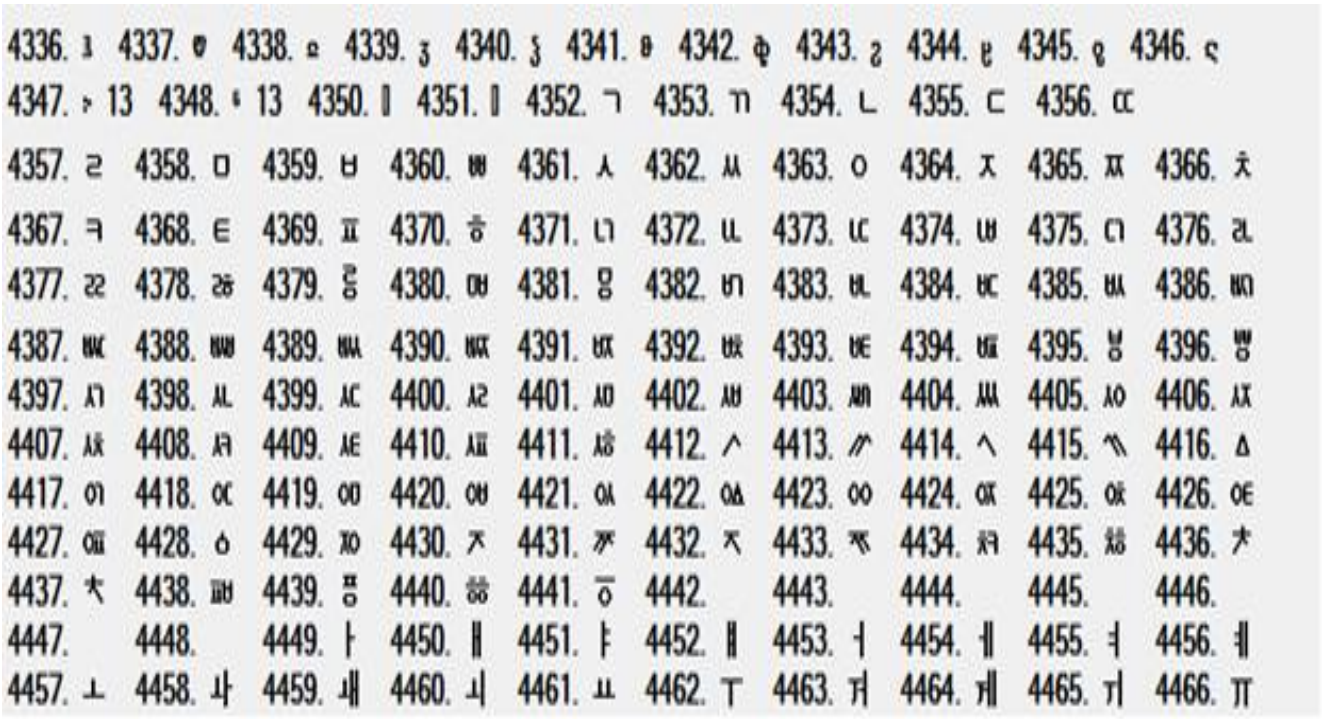
ინგლისური და რუსული სიმბოლოების დაშიფვრისა (S_i და K_i) და გაშიფვრის (D_i და K_i) ამსახველი სიმბოლოების კრებული (E_R ან R_E)

ცხრილი 2.7



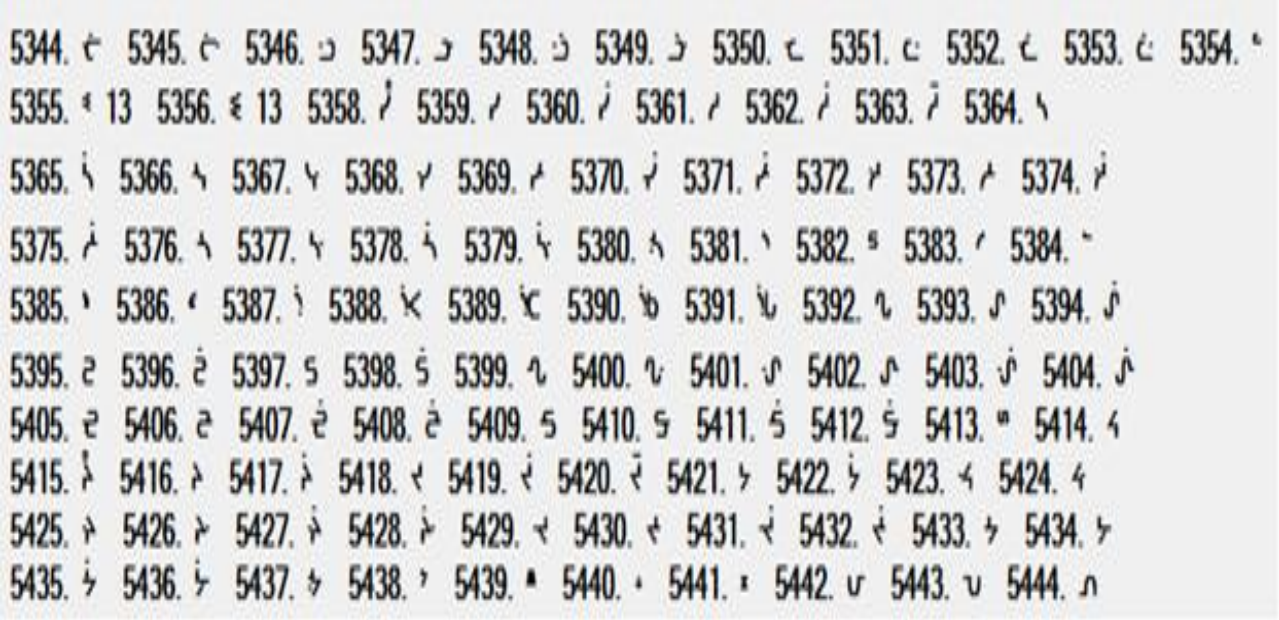
რუსული სიმბოლოს დაშიფვრისა (S_i და K_i) და გაშიფვრის (D_i და K_i) ამსახველი სიმბოლოების კრებული (R_R)

ცხრილი 2.8



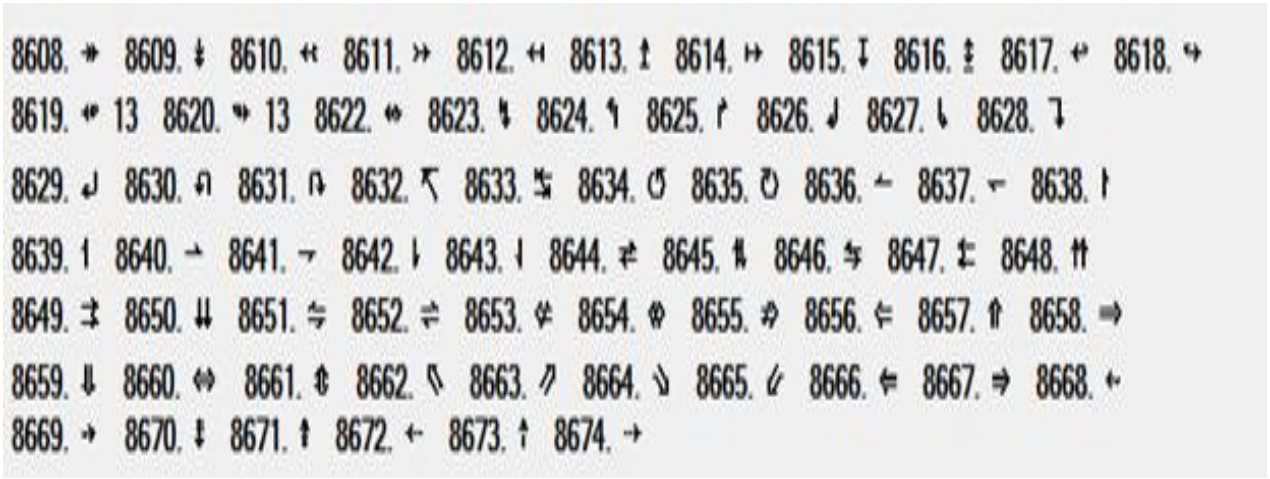
ინგლისური და ქართული სიმბოლოების დაშიფვრისა (S_i და K_i) და გაშიფვრის (D_i და K_i) ამსახველი სიმბოლოების კრებული (E_G ან G_E)

ცხრილი 2.9



ქართული და რუსული სიმბოლოების დაშიფვრისა (S_i და K_i) და გაშიფვრის (D_i და K_i) ამსახველი სიმბოლოების კრებული (R_G ან G_R)

ცხრილი 2.10



ქართული სიმბოლოს დაშიფვრისა (S_i და K_i) და გაშიფვრის (D_i და K_i) ამსახველი სიმბოლოების კრებული (G_G)

2.5-2.10 ცხრილებში ჩანს, რომ ტექსტური ინფორმაციის S_i სიმბოლოების კოდების S_i^k მნიშვნელობების ცვლილების დიაპაზონი (32-4336) საგრძნობ-

ლად დიდია და მათი პრაქტიკული გამოყენება შეიძლება იყოს დაკავშირებული გარკვეულ სირთულეებთან. კერძოდ, შეუძლებელი ხდება დაშიფვრის ან/და გაშიფვრის ალგორითმების პერსონალური კომპიუტერის გამოყენების გარეშე (სასწავლო მაგალითების ემპირული გზით ან გამომთვლელის, თუნდაც საოფისე პროგრამის MS Excel შესაძლებლობების გამოყენებით) შესრულება. აღნიშნული სირთულის თავიდან აცილების მიზნით შემოთავაზებულია სიმბოლოების გაერთიანებული ნუსხა ალფაბეტი-η (იხ. ცხრ.2.11-ის მე-3 სვეტი) და ამ ნუსხაში შემავალი ყოველი სიმბოლოს შესაბამისი კომპიუტერული კოდი (2.11 ცხრილის მე-2 სვეტი), ხოლო 2.11 ცხრილის პირველ სვეტში შეტანილია სიმბოლოების გაერთიანებული ნუსხის ახალი კოდების რიცხვითი მნიშვნელობები. 2.12 ცხრილში მოცემულია ალფაბეტ-η სიმბოლოების გარდასახვის შედეგი ალფაბეტ-λ, სადაც λ არის ალფაბეტი სიმეტრიული სისტემის მეთოდებით TI დაშიფვრის შედეგად მიღებული შიფრ-ტექსტში - ShifTI შემავალი სიმბოლოების D_i ნაკრებები და მათი შესაბამისი კომპიუტერული (რიცხვითი) კოდების D_i^k მნიშვნელობები.

2.11 ცხრილის მიხედვით სიმბოლოების ნაკრებთა რიცხვი 193-ის ტოლია, რომელშიც სიმბოლოს მაქსიმალური კოდი არის 224, რაც იძლევა საშუალებას:

1. ავირჩიოთ ისეთი p მოდული (p მარტივი რიცხვია, გამოიყენება სიმეტრიულ სისტემებში დაშიფვრის დახურული გასაღების გამოსათვლელად), რომელიც უნდა აკმაყოფილებდეს პირობას $p>224$ და არა პირობას $p>4336$ [9];

2. საგრძნობლად გამარტივდეს დაშიფვრა/გაშიფვრის ალგორითმების სქემური რეალიზაცია, რომელსაც საფუძვლად უდევს სიმბოლოების ორობით კოდში წარმოდგენა [19].

ცხრილი 2.11

№	1	2	3
1	32	32	ჰაერი
2	33	33	!
3	34	34	"
4	35	35	#
5	36	36	$
6	37	37	%
7	38	38	&
8	39	39	აპოსტროფი
9	40	40	(
10	41	41	)
11	42	42	*
12	43	43	+
13	44	44	.
14	45	45	-
15	46	46	,
16	47	47	/
17	48	48	0
18	49	49	1
19	50	50	2
20	51	51	3
21	52	52	4
22	53	53	5
23	54	54	6
24	55	55	7

№	1	2	3
25	56	56	8
26	57	57	9
27	58	58	:
28	59	59	;
29	60	60	<
30	61	61	=
31	62	62	>
32	63	63	?
33	64	64	@
34	65	65	A
35	66	66	B
36	67	67	C
37	68	68	D
38	69	69	E
39	70	70	F
40	71	71	G
41	72	72	H
42	73	73	I
43	74	74	J
44	75	75	K
45	76	76	L
46	77	77	M
47	78	78	N
48	79	79	O

№	1	2	3		№	1	2	3
49	80	80	P		**74**	105	105	i
50	81	81	Q		**75**	106	106	j
51	82	82	R		**76**	107	107	k
52	83	83	S		**77**	108	108	l
53	84	84	T		**78**	109	109	m
54	85	85	U		**79**	110	110	n
55	86	86	V		**80**	111	111	o
56	87	87	W		**81**	112	112	p
57	88	88	X		**82**	113	113	q
58	89	89	Y		**83**	114	114	r
59	90	90	Z		**84**	115	115	s
60	91	91	[		**85**	116	116	t
61	92	92	\		**86**	117	117	u
62	93	93	]		**87**	118	118	v
63	94	94	^		**88**	119	119	w
64	95	95	_		**89**	120	120	x
65	96	96	`		**90**	121	121	y
66	97	97	a		**91**	122	122	z
67	98	98	b		**92**	123	123	{
68	99	99	c		**93**	124	124	|
69	100	100	d		**94**	125	125	}
70	101	101	e		**95**	126	126	~
71	102	102	f		**96**	127	127	
72	103	103	g		**97**	128	4304	ა
73	104	104	h		**98**	129	4305	ბ

№	1	2	3
99	130	4306	ⴂ
100	131	4307	ⴃ
101	132	4308	ⴄ
102	133	4309	ⴅ
103	134	4310	ⴆ
104	135	4311	ⴇ
105	136	4312	ⴈ
106	137	4313	ⴉ
107	138	4314	ⴊ
108	139	4315	ⴋ
109	140	4316	ⴌ
110	141	4317	ⴍ
111	142	4318	ⴎ
112	143	4319	ⴏ
113	144	4320	ⴐ
114	145	4321	ⴑ
115	146	4322	ⴒ
116	147	4323	ⴓ
117	148	4324	ⴔ
118	149	4325	ⴕ
119	150	4326	ⴖ
120	151	4327	ⴗ
121	152	4328	ⴘ
122	153	4329	ⴙ
123	154	4330	ⴚ

№	1	2	3
124	155	4331	ⴛ
125	156	4332	ⴜ
126	157	4333	ⴝ
127	158	4334	ⴞ
128	159	4335	ⴟ
129	160	4336	ⴠ
130	161	1040	А
131	162	1041	Б
132	163	1042	В
133	164	1043	Г
134	165	1044	Д
135	166	1045	Е
136	167	1046	Ж
137	168	1047	З
138	169	1048	И
139	170	1049	Й
140	171	1050	К
141	172	1051	Л
142	173	1052	М
143	174	1053	Н
144	175	1054	О
145	176	1055	П
146	177	1056	Р
147	178	1057	С
148	179	1058	Т

№	1	2	3
149	180	1059	У
150	181	1060	Ф
151	182	1061	Х
152	183	1062	Ц
153	184	1063	Ч
154	185	1064	Ш
155	186	1065	Щ
156	187	1066	Ъ
157	188	1067	Ы
158	189	1068	Ь
159	190	1069	Э
160	191	1070	Ю
161	192	1071	Я
162	193	1072	а
163	194	1073	б
164	195	1074	в
165	196	1075	г
166	197	1076	д
167	198	1077	е
168	199	1078	ж
169	200	1079	з
170	201	1080	и
171	202	1081	й

№	1	2	3
172	203	1082	к
173	204	1083	л
174	205	1084	м
175	206	1085	н
176	207	1086	о
177	208	1087	п
178	209	1088	р
179	210	1089	с
180	211	1090	т
181	212	1091	у
182	213	1092	ф
183	214	1093	Х
184	215	1094	ц
185	216	1095	Ч
186	217	1096	ш
187	218	1097	щ
188	219	1198	ъ
189	220	1199	ы
190	221	1100	ь
191	222	1101	э
192	223	1102	ю
193	224	1103	я

ინგლისური, ქართული და რუსული სიმბოლოების კრებული

ცხრილი 2.12

L1 :	64. @	65. A	66. B	67. C	68. D	69. E	70. F	71. G	72. H	73. I	74. J			
L2 :	75. K	76. L	77. M	78. N	79. O	80. P	81. Q	82. R	83. S	84. T				
L3 :	85. U	86. V	87. W	88. X	89. Y	90. Z	91. [	92. \	93.]	94. ^				
L4 :	95. _	96. `	97. a	98. b	99. c	100. d	101. e	102. f	103. g	104. h				
L5 :	105. i	106. j	107. k	108. l	109. m	110. n	111. o	112. p	113. q	114. r				
L6 :	115. s	116. t	117. u	118. v	119. w	120. x	121. y	122. z	123. {	124. |				
L7 :	125. }	126. ~	127. ▯	128.	129.	130.	131.	132.	133.	134.				
L8 :	135.	136.	137.	138.	139.	140.	141.	142.	143.	144.				
L9 :	145.	146.	147.	148.	149.	150.	151.	152.	153.	154.				
L10 :	155.	156.	157.	158.	159.	160.	161. ¡	162. ¢	163. £	164. ¤				
L11 :	165. ¥	166. ¦	167. §	168. ¨	169. ©	170. ª	171. «	172. ¬	173. -	174. ®				
L12 :	175. ¯	176. °	177. ±	178. ²	179. ³	180. ´	181. µ	182. ¶	183. ·	184. ¸				
L13 :	185. ¹	186. º	187. »	188. ¼	189. ½	190. ¾	191. ¿	192. À	193. Á	194. Â				
L14 :	195. Ã	196. Ä	197. Å	198. Æ	199. Ç	200. È	201. É	202. Ê	203. Ë	204. Ì				
L15 :	205. Í	206. Î	207. Ï	208. Ð	209. Ñ	210. Ò	211. Ó	212. Ô	213. Õ	214. Ö				
L16 :	215. ×	216. Ø	217. Ù	218. Ú	219. Û	220. Ü	221. Ý	222. Þ	223. ß	224. à				
L17 :	225. á	226. â	227. ã	228. ä	229. å	230. æ	231. ç	232. è	233. é	234. ê				
L18 :	235. ë	236. ì	237. í	238. î	239. ï	240. ð	241. ñ	242. ò	243. ó	244. ô				
L19 :	245. õ	246. ö	247. ÷	248. ø	249. ù	250. ú	251. û	252. ü	253. ý	254. þ				
L21 :	255. ÿ	256. Ā	257. ā	258. Ă	259. ă	260. Ą	261. ą	262. Ć	263. ć	264. Ĉ				

L1 :	265. ĉ	266. Ċ	267. ċ	268. Č	269. č	270. Ď	271. ď	272. Đ	273. đ	274. Ē				
L2 :	275. ē	276. Ĕ	277. ĕ	278. Ė	279. ė	280. Ę	281. ę	282. Ě	283. ě	284. Ĝ				
L3 :	285. ĝ	286. Ğ	287. ğ	288. Ġ	289. ġ	290. Ģ	291. ģ	292. Ĥ	293. ĥ	294. Ħ				
L4 :	295. ħ	296. Ĩ	297. ĩ	298. Ī	299. ī	300. Ĭ	301. ĭ	302. Į	303. į	304. İ				
L5 :	305. ı	306. IJ	307. ij	308. Ĵ	309. ĵ	310. Ķ	311. ķ	312. ĸ	313. Ĺ	314. ĺ				
L6 :	315. Ļ	316. ļ	317. Ľ	318. ľ	319. Ŀ	320. ŀ	321. Ł	322. ł	323. Ń	324. ń				
L7 :	325. Ņ	326. ņ	327. Ň	328. ň	329. ŉ	330. Ŋ	331. ŋ	332. Ō	333. ō	334. Ŏ				
L8 :	335. ŏ	336. Ő	337. ő	338. Œ	339. œ	340. Ŕ	341. ŕ	342. Ŗ	343. ŗ	344. Ř				
L9 :	345. ř	346. Ś	347. ś	348. Ŝ	349. ŝ	350. Ş	351. ş	352. Š	353. š	354. Ţ				
L10 :	355. ţ	356. Ť	357. ť	358. Ŧ	359. ŧ	360. Ũ	361. ũ	362. Ū	363. ū	364. Ŭ				
L11 :	365. ŭ	366. Ů	367. ů	368. Ű	369. ű	370. Ų	371. ų	372. Ŵ	373. ŵ	374. Ŷ				
L12 :	375. ŷ	376. Ÿ	377. Ź	378. ź	379. Ż	380. ż	381. Ž	382. ž	383. ſ	384. ƀ				
L13 :	385. Ɓ	386. Ƃ	387. ƃ	388. Ƅ	389. ƅ	390. Ɔ	391. Ƈ	392. ƈ	393. Ɖ	394. Ɗ				
L14 :	395. Ƌ	396. ƌ	397. ƍ	398. Ǝ	399. Ə	400. Ɛ	401. Ƒ	402. ƒ	403. Ɠ	404. Ɣ				
L15 :	405. ƕ	406. Ɩ	407. Ɨ	408. Ƙ	409. ƙ	410. ƚ	411. ƛ	412. Ɯ	413. Ɲ	414. ƞ				
L16 :	415. Ɵ	416. Ơ	417. ơ	418. Ƣ	419. ƣ	420. Ƥ	421. ƥ	422. Ʀ	423. Ƨ	424. ƨ				
L17 :	425. Σ	426. ƪ	427. ƫ	428. Ƭ	429. ƭ	430. Ʈ	431. Ư	432. ư	433. Ʊ	434. Ʋ				
L18 :	435. Ƴ	436. ƴ	437. Ƶ	438. ƶ	439. Ʒ	440. Ƹ	441. ƹ	442. ƺ	443. ƻ	444. 5				
L19 :	445. ƽ	446. ƽ	447. ƿ	448. |	449. ǁ	450. ǂ	451. !	452. DŽ	453. Dž	454. dž				

L1 :	455. LJ	456. Lj	457. lj	458. NJ	459. Nj	460. nj	461. Ǎ	462. ǎ	463. Ǐ	464. ǐ				
L2 :	465. Ǒ	466. ǒ	467. Ǔ	468. ǔ	469. Ǖ	470. ǖ	471. Ǘ	472. ǘ	473. Ǚ	474. ǚ				
L3 :	475. Ǜ	476. ǜ	477. ǝ	478. Ā	479. ā	480. Ǡ	481. ǡ	482. Ǣ	483. ǣ	484. G				
L4 :	485. g	486. Ǧ	487. ǧ	488. Ǩ	489. ǩ	490. Ǫ	491. ǫ	492. Ǭ	493. ǭ	494. Ǯ				
L5 :	495. ǯ	496. ǰ	497. DZ	498. Dz	499. dz	500. Ǵ	501. ǵ	502. Ƕ	503. ƿ	504. Ǹ				
L6 :	505. ǹ	506. Ǻ	507. ǻ	508. Ǽ	509. ǽ	510. Ǿ	511. ǿ	512. Ȁ	513. ȁ	514. Ȃ				
L7 :	515. ȃ	516. Ȅ	517. ȅ	518. Ȇ	519. ȇ	520. Ȉ	521. ȉ	522. Ȋ	523. ȋ	524. Ȍ				
L8 :	525. ȍ	526. Ȏ	527. ȏ	528. Ȑ	529. ȑ	530. Ȓ	531. ȓ	532. Ȕ	533. ȕ	534. Ȗ				
L9 :	535. ȗ	536. Ș	537. ș	538. Ț	539. ț	540. Ȝ	541. ȝ	542. Ȟ	543. ȟ	544. Ƞ				
L10 :	545. ȡ	546. Ȣ	547. ȣ	548. Ȥ	549. ȥ	550. Ȧ	551. ȧ	552. Ȩ	553. ȩ	554. Ȫ				
L11 :	555. ȫ	556. Ȭ	557. ȭ	558. Ȯ	559. ȯ	560. Ȱ	561. ȱ	562. Ȳ	563. ȳ	564. ȴ				
L12 :	565. ȵ	566. ȶ	567. ȷ	568. ȸ	569. ȹ	570. Ⱥ	571. Ȼ	572. ȼ	573. Ƚ	574. Ⱦ				

დაშიფვრისა (S_i და K_i) და გაშიფვრის (D_i და K_i) ამსახველი სიმბოლოების კრებული

აღვნიშნოთ, რომ სიმბოლოების აღწერილი ნაკრების და მათი შესაბამისი კოდების გამოყენებით დაპროგრამებულია სიმეტრიული სისტემის უნივერსალური მეთოდი და აგრეთვე შემოთავაზებულია მისი სქემო-ტექნიკური რეალიზაცია.

განვიხილოთ TI დაშიფვრისა და გაშიფვრის პროცედურების რეალიზაცია შებრუნებული მატრიცის მეთოდით (იხ.1.4 ქვეთავი), შესრულებული საოფისე პროგრამის MS Excel-ის ფუნქციებისა და შემოთავაზებული წესით კოდირებული სიმბოლოების ნაკრების გამოყენებით (ცხრ.2.11), რომელიც ნაჩვენებია 2.1 ნახაზის სახით, სადაც:

ა) დეტერმინანტი (Det=-15) გამოთვლილია საწყისი მატრიცის DM მონაცემების მიხედვით ფუნქციით: MDETERM ();

ბ) შებრუნებული მატრიცა გამოთვლილია ფუნქციით: MINVERSE ();

გ) ერთეულოვანი მატრიცა გამოთვლილია ფუნქციით: MMULT ().

2.1 ნახ-ზე ნაჩვენებია TI=“KekELiA” დაშიფვრისა და გაშიფვრის პროცედურული რეალიზაცია.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
2																
3		საწყისი მატრიცა				დეტერმინანტი =	-15									
4		1	2	3												
5	DM=	0	3	1												
6		2	1	0												
7																
8		შებრუნებული მატრიცა														
9		0.066667	-0.2	0.466666667		TI="KekELiA" დაშიფვრის/გაშიფვრის მაგალითი										
10	GM=	-0.13333	0.4	0.066666667												
11		0.4	-0.2	-0.2		DasTI	K	e	k	E	L	i	A	A	A	
12						$DasTI(S_i^k)$	75	101	107	69	76	105	65	65	65	
13		ერთეულოვანი მატრიცა				$ShiTI(D_i^k)$	289	560	326	279	471	283	195	390	260	
14		1	-1.11022E-16	-1.11022E-16		$ShiTI(D_i)$	g	O	0	e	U	e	A	Э	A	
15	E=	0	1	0		$DasTI(S_i^k)$	75	101	107	69	76	105	65	65	65	
16		0	0	1		DasTI	K	e	k	E	L	i	A	A	A	
17																

ნახ. 2.1. TI დაშიფვრა და გაშიფვრა MS Excel-ის ფუნქციების გამოყენებით

1. TI დაშიფვრის პროცედურული რეალიზაცია შედგება შემდეგი ეტაპებისაგან:

1.1. მოცემულ TI ბოლოში ემატება ორი სიმბოლო, ვთქვათ A და A, როგორც ეს ნაჩვენებია 2.1 ნახაზზე, ვინაიდან დასაშიფრი საწყისი ტექსტის სტრიქონის სიგრძე უნდა იყოს სამის ჯერადი;

1.2. 2.11 ცხრილის მიხედვით DasTI-იის ყოველ სიმბოლოს S_i ცალსახად შეესაბამება კოდური რიცხვითი მნიშვნელობა, რომლებიც შეიტანება DasTI(S_i^k) სტრიქონში (ივსება ცხრ. 2.11 მეორე სვეტიდან);

1.3. TI-იის ყოველ სიმბოლოს დაშიფვრის შედეგად შეესაბამება, როგორც სიმბოლო D_i ასევე ამ სიმბოლოს შესაბამისი კოდი D_i^k (i=0,2,…,8), რომელიც გამოითვლება შემდეგი წესით:

$D_0^k = 75*1 + 101*0 + 107*2 = 289$;

$D_1^k = 75*2 + 101*3 + 107*1 = 560$;

$D_2^k = 75*3 + 101*1 + 107*0 = 326$;

$D_3^k = 69*1 + 76*0 + 105*2 = 279$;

$D_4^k = 69*2 + 76*3 + 105*1 = 471$;

$D_5^k = 69*3 + 76*1 + 105*0 = 283$;

$D_6^k = 65*1 + 65*0 + 65*2 = 195$;

$D_7^k = 65*2 + 65*3 + 65*1 = 390$;

$D_8^k = 65*3 + 65*1 + 65*0 = 260$.

მიღებული გამოთვლების შედეგები შეტანილია სტრიქონში, რომელიც აღნიშნულია ShifTI(D_i^k)-ით და ShifTI(D_i)-ით (იხ. ნახ.2.1).

2. შიფრ-ტექსტის გაშიფვრის პროცედურული რეალიზაცია შედგება შემდეგი ეტაპებისაგან:

2.1. შიფრ-ტექსტის ყოველ სიმბოლოს გაშიფვრის შედეგად შეესაბამება, როგორც სიმბოლო S_i ასევე ამ სიმბოლოს შესაბამისი კოდი S_i^k (i=0,2,…,8), რომელიც გამოითვლება შემდეგი წესით:

$S_0^k = (289* 0.066667 + 560*(-0.13333) + 326* 0.4) = 75$;

$S_1^k = (289*(-0.2) + 560*0.4 + 326*(-0.2)) = 101$;

$S_2^k = (289* 0.466666667 + 560* 0.066666667 + 326*(-.0.2)) = 107$;

$S_3^k = (279* 0.066667 + 471*(-0.13333) + 283* 0.4) = 69$;

S_4^k = (279*(-0.2) +471*0.4 +283*(-0.2)) = 76 ;

S_5^k = (279* 0.466666667 +471* 0.066666667 +283*(-.0.2)) = 105 ;

S_6^k = (195* 0.066667 +390*(-0.13333) +260* 0.4) = 65 ;

S_7^k = (195*(-0.2) +390*0.4 +260*(-0.2)) = 65 ;

S_8^k = (195* 0.466666667 +390* 0.066666667 +260*(-0.2)) = 65 .

მიღებული გამოთვლების შედეგები შეტანილია სტრიქონებში, რომლებიც აღნიშნულია TI(S_i^k) და DasTI -იით (იხ. ნახ.2.1).

2.2 სიმეტრიული სისტემის მეთოდების უნივერსალური მოდელი

ზემოაღნიშნულიდან გამომდინარე, დასაშიფრი ტექსტური ინფორმაცია DasTI - ეს არის სიმბოლოების მიმდევრობა - DasTI={ S_i} (სადაც i=0,1,2, ..., Z; Z_1=Z+1 – სიმბოლოების მაქსიმალური რიცხვია DasTI-ში), რომლებიც პერსონალურ კომპიუტერში შეიტანება კლავიატურიდან. ყოველი სიმბოლო S_i ეკუთვნის η ალფაბეტს, რომელსაც თავის მხრივ, ცალსახად შეესაბამება გარკვეული რიცხვითი მნიშვნელობა S_i^k - მისი შესაბამისი კომპიუტერული კოდი (იხ. ცხრ.2.12). ამგვარად, DasTI შეიძლება აგრეთვე წარმოდგენილ იქნეს დადებით მთელ რიცხვთა მიმდევრობით: DasTIK={S_i^k}, სადაც DasTIK - არის DasTI-ში შემავალი სიმბოლოების კოდური რიცხვითი მნიშვნელობები.

DasTIK ელემენტები არის დაშიფვრისა ობიექტებს, მათზე ხორციელდება გარკვეული გარდასახვები, მანიპულაციები (არითმეტიკული და ლოგიკური ოპერაციები), რის შედეგადაც მიიღება რიცხვების ახალი მიმდევრობა - ShiTIK (შიფრ-ტექსტის სიმბოლოების კოდები: ShiTIK= {D_i^k}, i=0,1,2, ..., Z); ნებისმიერი D_i^k შეიძლება ზოგად შემთხვევაში იყოს განსაზღვრული სასრულო რიცხვთა სიმრავლეზე, რომლის თითოეულ ელემენტს ცალსახად შეესაბამება გარკვეული სიმბოლო λ ალფაბეტში (კერძოდ, Microsoft Sans Seris ფონტების იმ კრებულიდან, რომლებიც გენერირებულია PC, იხ. ცხრ.2.1). DasTIK-ზე მანიპულაციების ჩატარების მიზნით განიხილავენ აგრეთვე დამშიფრავი სიმბოლოების DamTI={K_j} შესაბამის კოდურ მნიშვნელობებს DamTIK={K_j^k}, სადაც j = 0,1,2, ..., z<=Z;

- DamTI -დამშიფრავი TI (სიმბოლოების ერთობლიობა η ალფაბეტიდან);

- DamTIK – DamTI შემავალი სიმბოლოების შესაბამისი კოდებია, რომლებიც ეკუთვნის λ ალფაბეტს. იმისდა მიხედვით თუ რას უდრის z_1 ($z_1=z+1$) ცვლადის მნიშვნელობა, განიხილავენ სიმეტრიული სისტემის ამა თუ იმ მეთოდს. კერძოდ, თუ:

ა) $z_1=1$, აღნიშნავენ რომ TI დაშიფვრა/გაშიფვრა ხორციელდება ცეზარის მეთოდით;

ბ) $1<z_1< Z$, აღნიშნავენ რომ TI დაშიფვრა/გაშიფვრა ხორციელდება ვიჟინერის მეთოდით;

გ) $z_1= Z$, აღნიშნავენ რომ TI დაშიფვრა/გაშიფვრა ხორციელდება ვერნამის მეთოდით.

TI დაშიფვრისა და გაშიფვრის პროცედურების არსი მდგომარეობს შემდეგში: DasTI ყოველ S_i სიმბოლოს (მის კომპიუტერული კოდის მნიშვნელობას S_i^K DasTIK მიმდევრობიდან) შეესაბამება DamTI მიმდევრობიდან K_j სიმბოლოს მნიშვნელობა (მის კომპიუტერული კოდის მნიშვნელობას K_j^k DamTIK მიმდევრობიდან) და აწარმოებენ მათზე წინასწარ განსაზღვრულ მანიპულაციებს, სადაც

$$j=(i)\%z_1 . \qquad (2.1)$$

(2.1) ფორმულიდან ჩანს, რომ დამშიფრავი სიმბოლოს ან სიმბოლოების (z) ჯგუფის მონაწილეობა დაშიფვრა-გაშიფვრის პროცესში არ აღემატება Z-ს. აქედან გამომდინარე, დამშიფრავი სტრიქონის ეფექტურად ფორმირების მიზნით განვიხილოთ შეფარდება

$$g=Z_1/z_1 , \qquad (2.2)$$

სადაც g მთელი დადებითი რიცხვია აღებული მეტობით და გვიჩვენებს დამშიფრავ სტრიქონში ამორჩეული საწყისი სიმბოლოების გამეორების ჯერადობას. ცხადია

$$Z_1<=g^*z . \qquad (2.3)$$

იმ შემთხვევაში თუ აღმოჩნდება, რომ ფორმულაში (2.3) ადგილი აქვს უტოლობას, ბოლო ($g^*z - Z_1$) სიმბოლო არ მიიღებს მონაწილეობას დაშიფვრა/გაშიფრის პროცედურების რეალიზაციაში, რაც ნიშნავს იმას, რომ DamTI სტრიქონი შეიძლება წარმოვადგინოთ DasTI სტრიქონის ანალოგიურად ანუ DamTI={K_i} ან/და DamTIK-{K_i^k} სახით. აღნიშნულიდან გამომდინარე, დაშიფვრა/გაშიფვრის პროცედურა (η ალფაბეტის ელემენტების გარდასახვა λ ალფაბეტში) შეიძლება აღიწეროს შემდეგი ფორმულების სახით:

$$F^k(S_i^K, K_i^k) = D_i^K \qquad F(S_i, K_i) = D_i\ , \tag{2.4}$$

$$f^k(D_i^K, K_i^k) = S_i^K \qquad f(D_i, K_i) = S_i\ . \tag{2.5}$$

ფორმულების (2.4) და (2.5) თანახმად:

ა) TI-იის დასაშიფრად ყოველთვის მოიძებნება ერთი მაინც სიმრავლე - η (S_i^k, $K_i^k \in$ η) და ფუნქცია F^k (F), რომლის შესრულების შედეგი - D_i^K (D_i) იქნება განსაზღვრული λ - სასრულო სიმრავლეზე (შევნიშნოთ, რომ η და λ გადაკვეთა შეიძლება არ იყოს ცარიელი (იხ. ცხრ.2.1-2.12);

ბ) თუ არსებობს λ სიმრავლე (D_i^k, $\in$ λ) და სიმრავლე η (K_i^K $\in$η) ყოველთვის მოიძებნება F^k (F) ფუნქციის შებრუნებული ფუნქცია f^k (f), რომლის შესრულების შედეგი - S_i^K (S_i) იქნება განსაზღვრული - η სასრულო სიმრავლეზე (იხ. ცხრ.2.1 და 2.12).

აღვნიშნოთ, რომ ცნობილ სიმეტრიულ სისტემებში F^k ფუნქციის ქვეშ, როგორც წესი, იგულისხმება არითმეტიკული ოპერაცია „შეკრება“ ან ლოგიკური ოპერაცია „შეკრება mod_2“, ხოლო f^k ფუნქციის ქვეშ კი არითმეტიკული ოპერაცია „გამოკლება“ ან ლოგიკური ოპერაცია „შეკრება mod_2“.

უნივერსალური მეთოდის რეალიზაცია.

განვიხილოთ კონკრეტულ მაგალითზე უნივერსალური მეთოდით DasTI დაშიფვრა-გაშიფვრის პროცედურული შესრულების თანამიმდევრობა, რომელიც ნაჩვენებია 2.13-2.15 ცხრილებში (იხ.ნახ.2.2 და 2.3). იგულისხმება, რომ DasTI-ის (ვთქვათ, “VალერიЙ”) და damTI-ის (ვთქვათ, “Aბ”, აქ z=2, g=4) ფორმირება ხორციელდება სიმბოლოების ნაკრებით η ალფაბეტიდან (იხ.ცხრ.2.11. სტრიქონი 3), რომლებიც შეტანილია 2.13 და 2.14

ცხრილებში პირველი სტრიქონის 2-8 და 2-3 სვეტებში, ხოლო მე-2 სტრიქონში მათი კომპიუტერული კოდური მნიშვნელობები აღებული 2.11 ცხრილის მეორე სვეტიდან, შესაბამისად (სიმბოლოები „ს1“, „ს2“ და „ს3“ მიუთითებს 2.11 ცხრილის შესაბამის სვეტების ნომრებს). 2.13 ცხრილის მე-3 სტრიქონში შეტანილია (2.1) და (2.3) ფორმულებით განსაზღვრული წესით ფორმირებული damTI, ხოლო მისი სიმბოლოების კომპიუტერული კოდები აღებული 2.11 ცხრილის მეორე სვეტიდან, შეტანილია მე-4 სტრიქონში. 2.13 ცხრილის მე-5 და მე-6 სტრიქონებში შესაბამისად შეტანილია DasTI და DamTI სიმბოლოების რიცხვითი კოდების მნიშვნელობები აღებული 2.11 ცხრილის პირველი სვეტიდან. მე-7 სტრიქონში დაფიქსირებულია F^k ფუნქციის (მარეალიზებელი ოპერაცია „შეკრება“) შესრულების შედეგი. მე-8 სტრიქონში ნაჩვენებია შიფრ-ტექსტი, რომლის სიმბოლოები განსაზღვრულია მე-7 სტრიქონში გამოთვლილი კომპიუტერული კოდებით 2.12 ცხრილიდან (ჩაწერილია რიცხვითი კოდების შესაბამისი სიმბოლოები მიახლოებითი მიმსგავსებით).

ცხრილი 2.13

№	1	2	3	4	5	6	7	8
1	DasTI (ს3)	V	a	ლ	ე	p	и	Й
2	DasTIK (ს2)	86	97	4314	4308	1088	1080	1049
3	DamTI (ს3)	A	ბ	A	ბ	A	ბ	A
4	DamTIK (ს2)	65	4305	65	4305	65	4305	65
5	DasTIK (ს1)	86	97	138	132	209	201	170
6	DamTIK(ს1)	65	129	65	129	65	129	65
7	ShifTIK (ცხ.2.12)	151	226	203	261	274	330	235
8	ShiTI (ცხ.2.12)		^a	˜E	a|	˜E	N,	`e

ცხრილი 2.14

№	1	2	3
1	DamTI (ს3)	A	ბ
2	DamTIK(ს2)	65	4305
3	DamTI (ს1)	65	129

DamTI სიმბოლოები და მათი კოდების მნიშვნელობები

ცნობილია, რომ შიფრ-ტექსტი (მე-8 სტრიქონში მოცემული სახით) და DamTI (გამშიფრავი გასაღები) გადაეცემა TI გამშიფრავს. შევნიშნოთ, რომ სიმეტრიულ სისტემებში TI დასაშიფრავად და გასაშიფრავად გამოიყენება ერთი და იგივე ტექსტური ინფორმაცია. ShiTI-ის გაშიფვრის პროცედურული შესრულების თანამიმდევრობა ნაჩვენებია 2.3 ნახაზზე და 2.15 ცხრილში. 2.15 ცხრილის:

- მე-9 სტრიქონში ნაჩვენებოია შიფრ-ტექსტი, რომელიც DamTI ერთად გედაეგზავნება გამშიფრავს;

- მე-10 სტრიქონში შეტანილია დაშიფრული ტექსტში შემავალი სიმბოლოების კოდების მნიშვნელობები (იხ. ცხრ.2.13, მე-8 სტრიქონი);

- მე-11 სტრიქონში შეტანილია გამშიფრავი ტექსტში DamTI შემავალი სიმბოლოები (იხ. ცხრ.2.13-ის მე-3 სტრიქონი);

- მე-12 და მე-13 სტრიქონების ფორმირება წარმოებს ანალოგიურად, როგორც ეს იყო შესრულებული მე-4 და მე-6 სტიქონების ფორმირებისას;

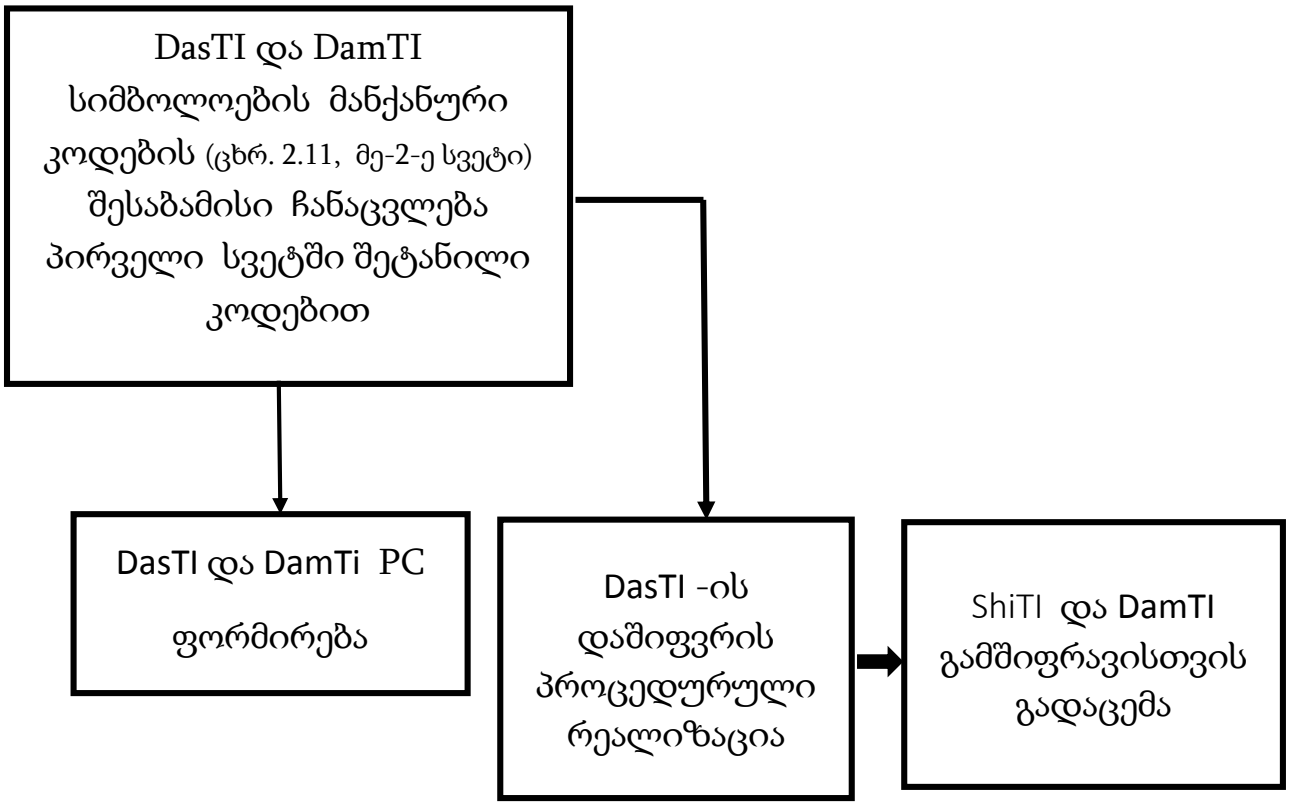
- მე-14 სტრიქონში შეტანილია გაშიფვრის ფუნქციის - f^k, ოპერაცია „გამოკლება“, შედეგი;

- მე-15 სტრიქონი შევსებულია კოდების მნიშვნელობებით, რომლებიც მიიღება მათი შესაბამისი ჩანაცვლებით 2.11 ცხრილის მე-2 სვეტის მნიშვნელობებით;

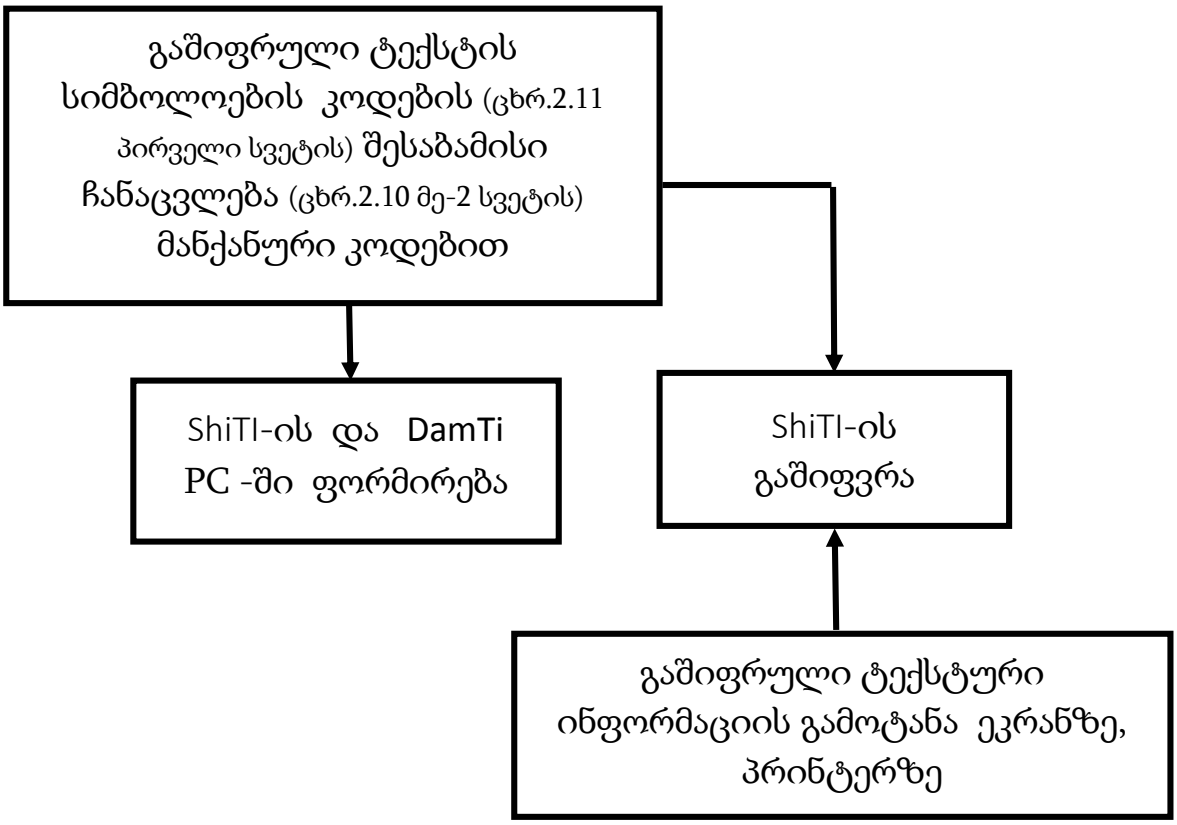
- მე-16 სტრიქონში ასახულია DasTI, რომელშიც შემავალი სიმბოლოების კოდების მნიშვნელობები მე-15 სტრიქონშია ჩაწერილი (იხ. ცხრ.2.10).

ცხრილი 2.15

№	1	2	3	4	5	6	7	8
9	ShiTI		^a	˜E	a|	˜E	N,	`e
10	ShifTIK	151	226	203	261	274	330	235
11	DamTI	A	ბ	A	ბ	A	ბ	A
12	DamTIK	65	4305	65	4305	65	4305	65
13	DamTIK	65	129	65	129	65	129	65
14	DasTIK	86	97	138	132	209	201	170
15	DasTIK	86	97	4314	4308	1088	1080	1049
16	DasTI	V	a	ლ	ე	p	и	Й



ნახ.2.2. საწყისი ტექსტური ინფორმაციის დაშიფვრის პროცედურა



ნახ. 2.3. დაშიფრული ტექსტური ინფორმაციის გაშიფვრის პროცედურა

2.3. უნივერსალური მოდელის მიკროპროგრამული სახით წარმოდგენა

ცეზარის, ვიჯინერისა და ვერნამის მეთოდების ანალიზმა გვიჩვენა, რომ ისინი ძირითადში ერთგვაროვანია პრაქტიკული რეალიზაციის თვალსაზრისით. ეს აადვილებს ამ მეთოდების ერთიან ასპექტში აბსტრაქტული მოდელის სახით წარმოდგენას და მის არა მარტო პროგრამულ რეალიზაციას, არამედ სქემო-ტექნიკურ რეალიზაციასაც. აღნიშნული მეთოდების მარეალიზებელი ალგორითმების ფორმალური ენით აღწერისა და სქემო-ტექნიკური რეალიზაციის მიზნით, განვიხილოთ მათემატიკური აპარატი, რომელსაც საფუძვლად უდევს ალგორითმული (მიკროპროგრამული) ალგებრის სისტემის - ოპერატორული G(G1,G2,…) ალგებრის და პირობის P(α,β,…) ალგებრის ცნება, რომელთა ტერმინებშიც შეიძლება იყოს აღწერილი სხვადასხვა სახის ალგორითმული პროცესები [18].

აღნიშნული ალგებრები წარმოდგენილია ერთრეგისტრიანი ან მრავალრეგისტრიანი პერიოდულად განსაზღვრული გარდასახვების სახით, რომელთა ტერმინებშიც აღიწერება ზოგიერთი ალგორითმული პროცესის მიკროპროგრამა. ცნობილია, რომ ოპერატორული ალგებრის ელემენტებს უწოდებენ ოპერატორებს და განსაზღვრულია ისინი M ინფორმაციულ სიმრავლეზე, სადაც M იმ რეგისტრების მდგომარეობათა საერთო რიცხვია, რომლებიც მონაწილეობას ღებულობს სისტემაში მიმდინარე გამოთვლით პროცესებში. დაუშვათ, რომ $X^R = \{\ldots, x_{-1}^R, x_{-0}^R, x_1^R, \ldots\}$, 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 $E_2 = \{0,1\}$. ოპერატიული ალგებრის (როგორც ბაზურს, ასევე მათგან წარმოებულს) უწოდებენ ოპერატორებს, რომლებიც განისაზღვრება (ერთრეგისტრიანი ან მრავალრეგისტრიანი პერიოდულად განსაზრვრული) გარდასახვებით M სიმრავლისა M სიმრავლეში. პირობის ალგებრის ელემენტები (როგორც ბაზურს, ასევე მათგან წარმოებულს) განისაზღვრება M ინფორმაციულ სიმრავლეზე, როგორც პირობები, რომლებსაც შეუძლიათ

მიიღოს ერთ-ერთი მნიშვნელობა თავისი სამი მნიშვნელობიდან **<T,F,U>,** სადაც T-true, F-false, U-unknown. ოპერატორულ ალგებრაში ძირითად ოპერაციად მიღებულია გამრავლების ოპერაცია ანუ ოპერატორების თანამიმდევრული შესრულება, ხოლო პირობის ალგებრაში - ოპერაციები: კონიუნქცია, დიზიუნქცია, ინვერსია. განვიხილოთ ოპერაციები, რომელთა მეშვეობითაც ხორციელდება G,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, ხოლო თუ α= false შესრულდება $Q=G_2$ ოპერატორი. α=unknowon-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=unknowon-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:

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

$$(G_1 \cup G_2)= (\alpha\ G_2 \cup G_1),$$

$$Q = \{_F\ G\}=e,\ \text{სადაც } e \text{ ცარიელი ოპერატორია.}$$

β=G×α - არის ლოგიკური პირობა, რომელიც ღებულობს იმავე მნიშვნელობას რასაც α, ოღონდ G ოპერატორის შესრულების შემდეგ [3].

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 დადებითი რიცხვის ჯამია:

$$\Sigma^c = \begin{matrix} 0^i & Z^i_{r1} \\ 0^j & Z^j_{r2} \end{matrix} \quad \Sigma_R{}^{ij} \tag{2.6},$$

$$\Sigma_R{}^{i,j} = \left\{ \begin{matrix} \mathrm{mod}_2(X^i_n, X^j_n) & \\ \alpha \quad \&(X^i_n, X^j_n) & L^j_1 \end{matrix} \right\} \tag{2.7},$$

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.

$\mathrm{mod}_2(X^i_n, X^j_n)$ – წარმოებული ლოგიკური ოპერატორია, მარეალიზებელი $f(x^i_n, x^j_n)$ გადამრთველი ფუნქციის $f(x^i_n, x^j_n)=\tilde{}x^i_n\&x^j_n \cup x^i_n\&\tilde{}x^j_n$ (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 - კონიუნქცია).

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

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 (იხ. ცხრ. 2.16).

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 დავუმატოთ 1. აღნიშნული მანიპულაციების შესრულების შედეგად მიიღება მაკლები გადაყვანილი დამატებით კოდში. შედეგად გვექნება: ~(01001110)+00000001=10110010. აღწერილი პროცედურების შესრულების შემდეგ, ვასრულებთ მიკროპროგრამას „შეკრება“- Σ^c (იხ. ცხრ. 2.17).

ცხრილი 2.16

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

ცხრილი 2.17

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

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

აღწერილი პროცედურის (ორი მთელი რიცხვის ოპერაცია „გამოკლება“: r1-r2) მარეალიზებელ მიკროპროგრამას - Σ^s აქვს შემდეგი სახე:

$$\Sigma^s = \begin{matrix} 0^i\ Y^i{}_1 \\ 0^j\ Z^j{}_{r2} \end{matrix} \sim X^j \Sigma_i{}^{i,j}\ 0^j\ Z^j{}_{r1}\ \Sigma_i{}^{i,j} \qquad (2.8),$$

სადაც $Y_1{}^j$ - არის ოპერატორი, რომლის შესრულების შედეგად X^i რეგისტრის ბოლო თანრიგი გადადის ერთის (ანუ - 0000....0001) მდგომარეობაში [3].

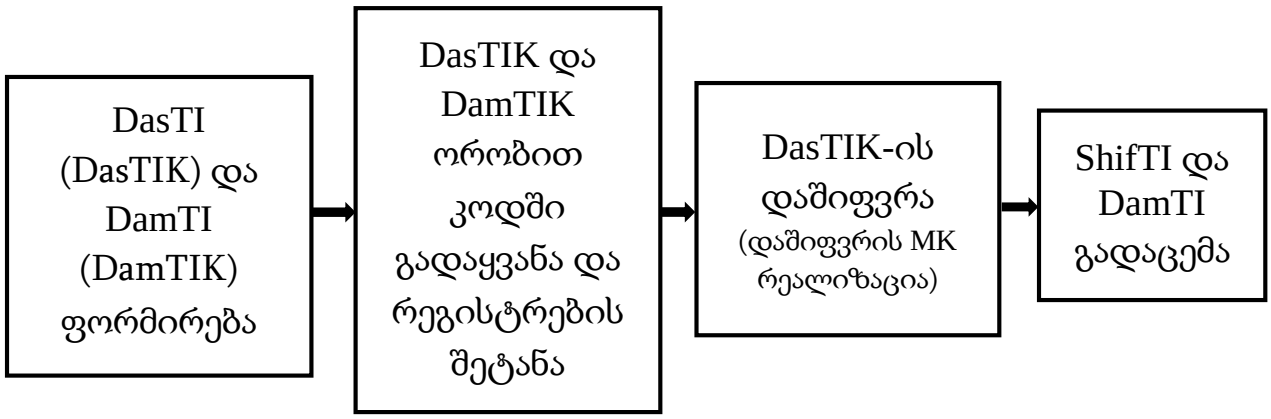
შემოთავაზებული მათემატიკური აპარატის გამოყენებით შედგენილია მიკროპროგრამები და მათი სქემური რეალიზაციის აბსტრაქტული მოდელები (იხ. ნახ.2.6), კრიპტოგრაფიაში ცნობილი სიმეტრიული სისტემის მეთოდებისა, რომელთა რიცხვსაც მიეკუთვნება ცეზარის, ვიჟინერისა და ვერნამის მეთოდები.

2.4. სიმეტრიული სისტემის უნივერსალური მეთოდის სქემო-ტექნიკური რეალიზაცია

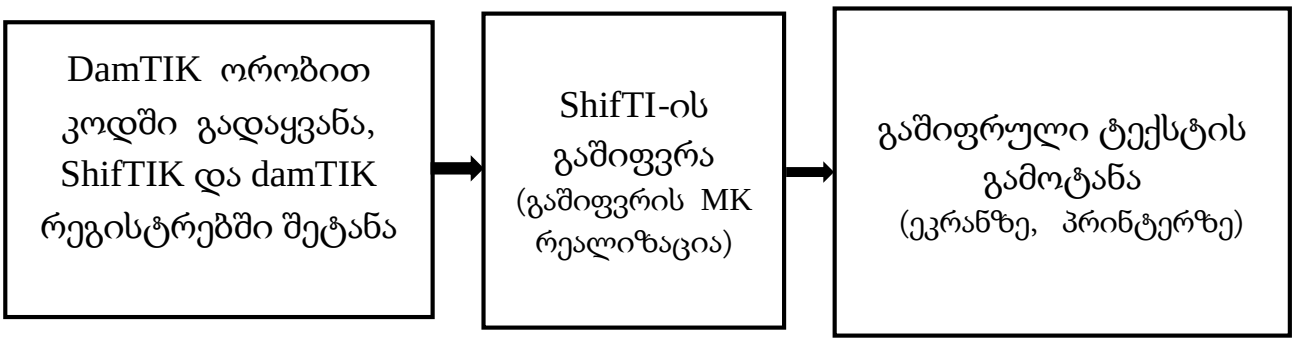
სიმეტრიული სისტემის მეთოდების (დაშიფვრისა და გაშიფვრის ალგორითმების, მარეალიზებელი ძირითადი ეტაპები ნაჩვენებია ნახ.2.4 და ნახ.2.5) მიკროპროგრამების (MK) მრავალრეგისტრიანი პერიოდულად განსაზღვრული გარდასახვების სახით წარმოდგენა და მათი სქემო-ტექნიკური რეალიზაცია მოითხოვს საწყისი ინფორმაციის (DasTIK და DamTIK) ორობით კოდში DasTIKO და DamTIKO გადაყვანას და დამუშავებას, მათზე სხვადასხვა სახის მანიპულაციის ჩატარებას.

როგორც იყო აღნიშნული, DasTI და DamTI ფორმირდება კლავიატურიდან იმ სიმბოლოებით S_i და K_i, რომლებიც განსაზღვრულია η ალფაბეტით (იხ. ცხრ.2.11). ამავე ალფაბეტით განისაზრვრება დასაშიფრ TI შემავალი ნებისმიერი $S_i{}^k$ და $K_i{}^k$ სიმბოლოების კომპიუტერული (რიცხვითი) კოდების მნიშვნელობები (ათობით ათვლის სისტემაში). ცხადია, რომ η ალფაბეტის ნებისმიერი სიმბოლოს რიცხვითი კოდი, რომ წარმოვადგინოთ ორობით

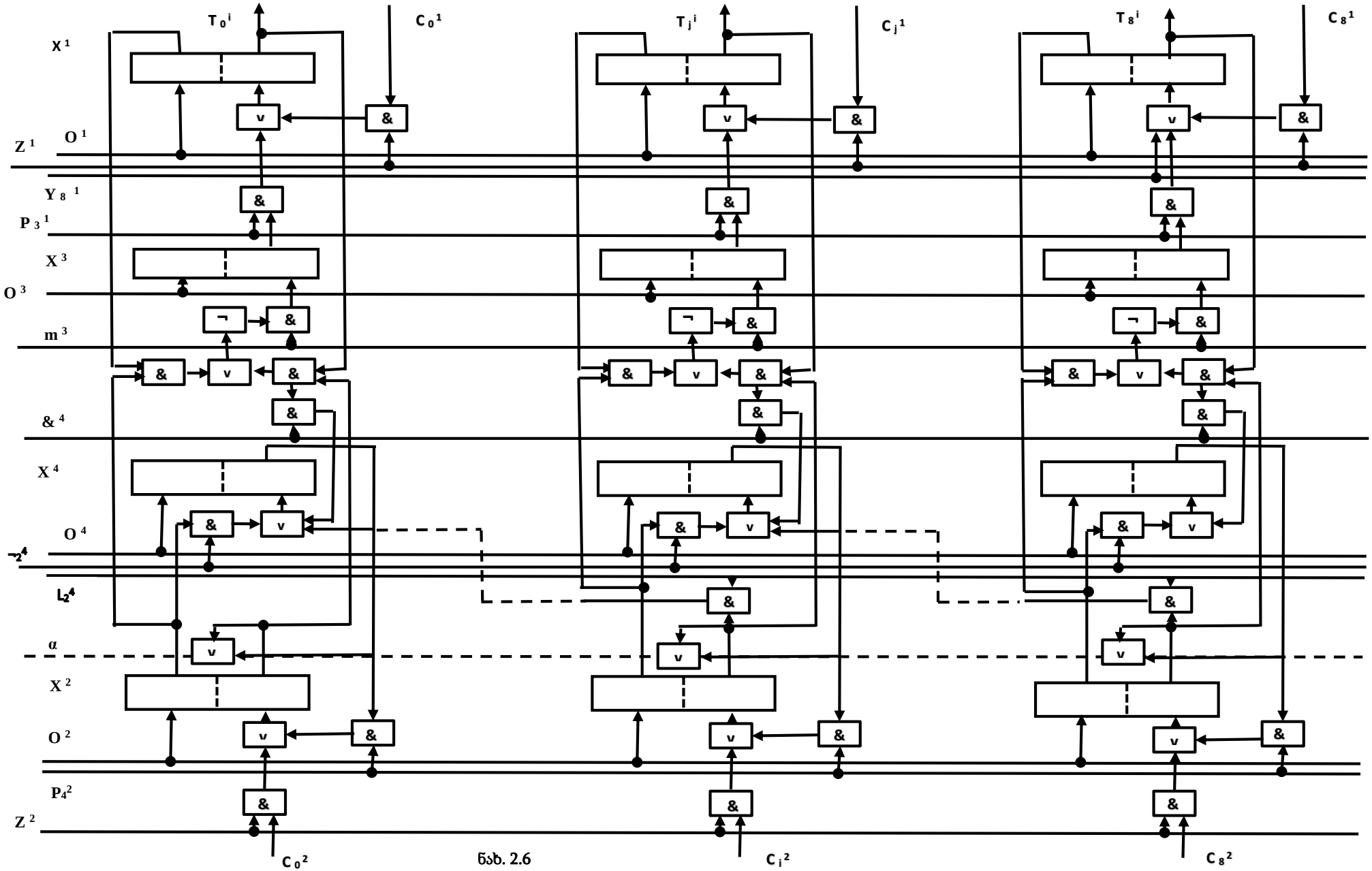
ათვლის სისტემაში, საჭიროა რვა ბიტი. დაშიფვრა-გაშიფვრის პროცედურების განხორციელებისა და მათი მარეალიზებელი ლოგიკური სქემის აღწერის მიზნით, განვიხილოთ ოთხი სასრულო სიგრძის ორობითი რეგისტრი X^1, X^2, X^3 და X^4. დავყოთ ეს რეგისტრები მარცხნიდან მარჯვნივ Z+1 ტოლ ნაწილებად (ბლოკად) ისე, რომ თითოეულ i-ურ (i=0,1,2,…,Z) ნაწილში (ბლოკში) იყოს გაერთიანებული რეგისტრის ცხრა ელემენტი { x_0^i, x_1^i,..., x_7^i, x_8^i} ანუ ცხრა ტრიგერი - T_j^i (j=0, 1, 2, 3, 4, 5, 6, 7, 8). ცხადია, რომ ჯამური რაოდენობა ასეთი ბიტებისა Σ= 9* (Z+1), სადაც b_v (v=`0,1,…Σ-1,Σ). 2.6 ნახაზზე ნაჩვენებია აღწერილ მიკროპროგრამების (მიკროოპერაციების) მარეალიზებელი i ნაწილის ლოგიკური სქემა, ხოლო 2.7 ნახაზზე წარმოდგენილია i ნაწილის T^i-ის ბლოკური სახე.



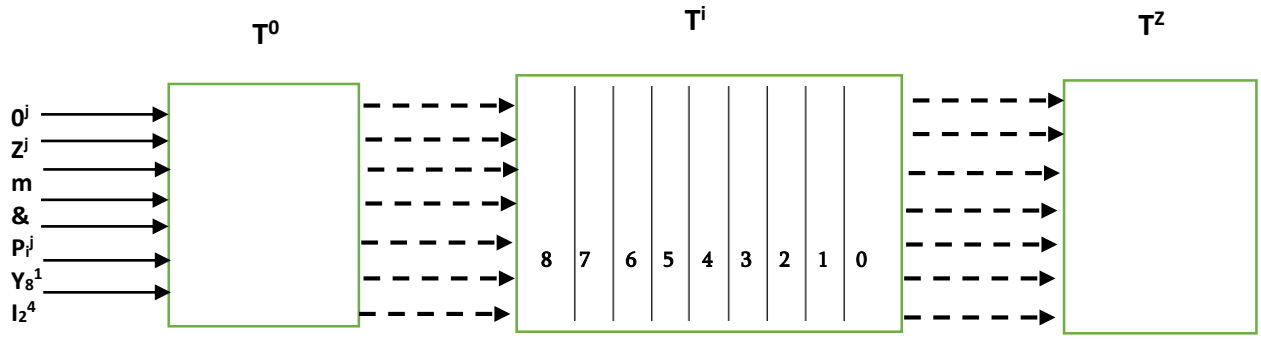
ნახ.2.4. საწყისი ტექსტური ინფორმაციის დაშიფვრის პროცედურა



ნახ.2.5. დაშიფრული ტექსტური ინფორმაციის გაშიფვრის პროცედურა


T0 i
C0 1
Tj i
Cj 1
T8 i
C8 1
X 1
O 1
Z 1
Y 8 1
P 3 1
X 3
O 3
m 3
& 4
X 4
O 4
α
X 2
O 2
P4 2
Z 2
C0 2
Ci 2
C8 2

ნახ. 2.6



ნახ.2.7. მმართველი სალტეები და მასინქრონიზებელი სიგნალები

2.6 და 2.7 ნახაზებზე სალტეების წინ ჩაწერილი აღნიშვნები მიუთითებენ იმ მმართველ სიგნალებზე (იმპულსებზე), რომლებიც გამომუშავდება დროის დისკრეტულ მომენტებში და რომლის სინქრონულადაც ხორციელდება ესა თუ ის (გარკვეული გარდასახვები რეგისტრ(ებ)ზე). ასე მაგალითად, სიგნალები (იმპულსები) მიწოდებული სალტეებზე (0^j, $Z_I{}^j$ და ა.შ.), შესაბამისად იწვევენ:

1. 0^j (j=1,2,3,4) X^j რეგისტრის ნულის მდგომარეობაში გადაყვანას;
2. Z^j (j=1,2) – X^1 რეგისტრში S_i^k ან X^2 რეგისტრში K_i^k,D_i^k კოდების ორობითი მნიშვნელობების შეტანას;
3. m და & – $f(x_i^1,x_i^2)$ - გადამრთველი ფუნქციების $f(x_i^1, x_i^2) = \bar{x}_i^1 x_i^2 \vee x_i^1 \bar{x}_i^2$ და $f(x_i^1,x_i^2)=x_i^1 \& x_i^2$ მნიშვნელობების x^3 და x^4 რეგისტრებში შეტანას, შესაბამისად;
4. P_i^j - X^j რეგისტრის X^i მდგომარეობაში გადაყვანას;
5. Y_8^1 – X^1 რეგისტრის ყოველი i ნაწილის T_8 ტრიგერის ერთიანის მდგომარეობაში გადაყვანას;
6. $\neg_2^4$ - X^2 რეგისტრის მდგომარეობის ინვენტირებული მნიშვნელობის X^4 რეგისტრში შეტანას;
7. L_2^4 – X^2 რეგისტრის მნიშვნელობის ერთი თანრიგით მარცხნივ დაძვრულის X^4 რეგისტრში შეტანას.

შევნიშნოთ:

ა) ლოგიკური სქემა, ნაჩვენები 2.6 ნახ-ზე, აგებულია მასში სიგნალების „რბოლის“ თავიდან აცილებისა და მისი სირთულის ოპტიმიზაციის ძირითადი პრინციპების გათვალისწინებით;

ბ) S_i^k, K_i^k და D_i^k სიმბოლოების შესაბამისი ორობითი კოდების მნიშვნელობები მიეწოდება b_j^c სალტეებზე, რომლებიც სინქრონულად Z^c მმართველი სიგნალისა შეიტანება X^c რეგისტრში (c=1,2);

გ) DasTIK-ზე დაშიფვრისა და ShifTIK-ზე გაშიფვრის მანიპულაციების განხორციელების მიზნით შერჩეულია ერთი ლოგიკური - „შეკრება“ mod_2 და ორი არითმეტიკული: შეკრება და გამოკლება ოპერაციები;

დ) S_i^k და K_i^k სიმბოლოების ორობით კოდში წარმოდგენისათვის, როგორც ეს იყო აღნიშნული, საკმარისია რვა ბიტი, ვინაიდან სიმბოლოს - 'я' კოდური მნიშვნელობა ტოლია 224-ის, რაც უდიდესი მნიშვნელობაა 2.11 ცხრილში შეტანილ სიმბოლოების კოდებს შორის. აქედან ცხადია, რომ D_i^k შეიძლება მიიღოს დაშიფვრის მანიპულაციების შედეგად მაქსიმალური მნიშვნელობა 446, რასაც ცხრა ბიტისაგან შემდგარი (100100000) მნიშვნელობა შეესაბამება. ეს კი თავის მხრივ განსაზღვრავს i ნაწილში (ბლოკში) ტრიგერების რაოდენობას.

(X^1) რეგისტრის i ნაწილში შეტანილ იქნეს DasTIK-ში (DamTIK-ში) შემავალი S_i^k (D_i^k) კოდის ორობითი მნიშვნელობა, ოღონდ X^2 მდგომარეობა უნდა განისაზღვროს წესით, რომელიც იყო აღწერილი 2.2 ქვეთავში [19]:

ცეზარის მეთოდი. X^2 რეგისტრის ყოველ i (i=0,1,2,…Z) ნაწილში შეიტანება დამშიფრავი სიმბოლოს K_i^k კოდის ორობითი მნიშვნელობა. ცეზარის მეთოდით TI დასაშიფრად/გასაშიფრავად, როგორც ცნობილია, გამოიყენება ერთი სიმბოლო.

ვიჯინერის მეთოდი. აღნიშნულ მეთოდში დამშიფრავი სიმბოლოების რაოდენობა z<Z. დაუშვათ შერჩეულია DamTI - სიმბოლოების შემდეგი მიმდევრობა - ABC. ამ სიმბოლოებისგან ფორმირდება (იხ. 2.2 ქვეთავი) დამშიფრავი DamTI სტრიქონი (ABCABCABC .....), რომლის ყოველი სიმბოლოს რიცხვითი კოდი შეიცვლება მისი შესაბამისი ორობითი კოდით და შეიტანება X^2 რეგისტრში.

ვერნამის მეთოდი. ვერნამის მეთოდში დამშიფრავი ტექსტს სიმბოლოების რაოდენობა DasTI-ში შემავალი სიმბოლოების რაოდენობის ტოლია. X^2 რეგისტრში შეიტანება დამშიფრავი ტექსტის სიმბოლოების კოდების ორობითი მნიშვნელობები ანალოგიურად, როგორც ეს იყო შესრულებული DasTIK X^1 რეგისტრში შეტანის შემთხვევაში.

განვიხილოთ ტექსტური ინფორმაციის დაშიფვრა/გაშიფვრის პროცედურების რეალიზაციის ორი ვარიანტი.

ვარიანტი 1.

ა) TI დასაშიფრავად მიმდევრობით სრულდება მიკროპროგრამები M^1 და M^2:

$$M^1 = \begin{matrix} 0^1 Z^1 \\ 0^2 Z^2 \end{matrix} M^2 \qquad (2.9)$$

მიკროპროგრამების რეალიზაციის შედეგად X^1 რეგისტრში დაფიქსირდება შიფრ-ტექსტში ShifTIK შემავალი D_i^k რიცხვითი მნიშვნელობის ორობითი კოდი;

ბ) X^1 რეგისტრში დაფიქსირებული შიფრ-ტექსტის გასაშიფვრად საკმარისია შესრულდეს მხოლოდ M^2 მიკროპროგრამა, შედეგად X^1 რეგისტრში დაფიქსირდება DasTIK შემავალი S_i^k რიცხვითი მნიშვნელობის შესაბამისი ორობითი კოდი.

ძირითად ოპერაციად TI დასაშიფვრად და გასაშიფვრად მოცემულ ვარიანტში შერჩეულია ლოგიკური ოპერაცია „შეკრება“ ორის მოდულით -mod_2, მარეალიზებელი გადამრთველი ფუნქციის: $f(x_i^1, x_i^2) = \bar{x}_i^1 x_i^2 \vee x_i^1 \bar{x}_i^2$, რომლის მნიშვნელობაც m მმართველი სიგნალის სინქრონულად შეიტანება X^3 რეგისტრში.

ვარიანტი 2.

ა) TI დასაშიფრავად მიმდევრობით სრულდება მიკროპროგრამები Σ^1 და Σ^2:

მიკროპროგრამების რეალიზაციის შედეგად X^1 რეგისტრში დაფიქსირდება შიფრ-ტექსტში ShifTIK შემავალი D_i^k რიცხვითი მნიშვნელობის ორობითი კოდი.

მოცემულ ვარიანტში ძირითად ოპერაციად TI დასაშიფვრად შერჩეულია არითმეტიკული ოპერაცია „შეკრება“. შეკრება ხორციელდება კოდების S_i^k და K_i^k (i=0,1,2,…,Z) ორობითი მნიშვნელობების მიკროპროგრამების Σ^1 და Σ^2 თანამიმდევრული შესრულებით (იხ. 2.3 ქვეთავში).

ბ) დაფიქსირებული შიფრ-ტექსტის გასაშიფვრად მიმდევრობით სრულდება მიკროპროგრამები G^1 და G^2:

მიკროპროგრამების რეალიზაციის შედეგად X^1 რეგისტრში დაფიქსირდება DasTIK შემავალი S_i^k რიცხვითი მნიშვნელობის ორობითი კოდი.

ძირითად ოპერაციად შიფრ-ტექსტის გასაშიფვრად მოცემულ ვარიანტში შერჩეულია არითმეტიკული ოპერაცია „გამოკლება“, რომელიც ხორციელდება კოდების S_i^k და K_i^k (i=0,1,2,…,Z) ორობით მნიშვნელობებზე, მიკროპროგრამების G^1 და G^2 თანამიმდევრული შესრულებით (იხ. 2.3 ქვეთავი). კერძოდ, პირველ ეტაპზე G^1 ოპერატორით ხორციელდება ყოველი დამშიფრავი სიმბოლოს შესაბამისი კოდის დამატებით კოდში გადაყვანა და მისი დაფიქსირება X^1 რეგისტრში. მეორე ეტაპზე კი G^2 მიკროპროგრამის შესრულების შედეგად X^1 რეგისტრში დაფიქსირდება DasTIK შესაბამისი ორობითი რიცხვითი მნიშვნელობები. შევნიშნოთ, რომ G^2 მიკროპროგრამაში Z^2 სინქრონული სიგნალით X^2 რეგისტრში შეიტანება 2.11 და 2.12 მიკროპროგრამების შესრულების შედეგად მიღებული შიფრ-ტექსტის ორობითი მნიშვნელობა.

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

2.18-2.21 ცხრილებში მოცემულია ზემოთ განხილული მეთოდით ტექსტური ინფორმაციის დაშიფვრა/გაშიფვრის პროცედურების მარიალიზებელი მიკროპროგრამები კონკრეტულ მაგალითებზე. კერძოდ, განხორციელებულია TI „ხოსრიკა” დაშიფვრა და გაშიფვრა DamTI-იის „ოიი” მეშვეობით.

ვარიანტი 1-ის მიხედვით - 2.18 და 2.19 ცხრილებში ნაჩვენებია TI-იის დაშიფვრისა და გაშიფვრის ეტაპები, ხოლო ვარიანტი 2-ის მიხედვით - 2.20 და 2.21 ცხრილებში.

შევნიშნოთ,რომ გამოყენებული ცხრილების:

ა) პირველ სვეტში მითითებული (ს1), (ს2) აღნიშვნა მიუთითებს იმაზე, რომ სიმბოლოების რიცხვითი კოდების მნიშვნელობები შესაბამისად აღებულია 2.11 ცხრილის პირველი ან მეორე სვეტიდან, აგრეთვე ჩაწერილია ზოგიერთი მიკროოპერაციების და მიკროპროგრამების აღმნიშვნელი სიმბოლოები, რომლებიც იწვევს შესაბამის გარდასახვებს მითითებულ რეგისტრებზე;

ბ) მეცხრე სვეტში შეტანილია ზოგიერთი მიკროოპერაციების და მიკროპროგრამების აღმნიშვნელი სიმბოლოები, რომლებიც იწვევს შესაბამის გარდასახვებს და რეგისტრები, რომლებშიც შეიტანება ამ გარდასახების შედეგები.

ცხრილი 2.18

	1	2	3	4	5	6	7	8	9
1	DasTI	ъ	ѡ	s	p	и	k	A	
2	DasTIK (ს2)	4334	4317	115	1088	1080	75	65	
3	DasTIK (ს1)	158	141	115	209	201	75	65	
4	DasTIKO	010011110	010001101	001110011	011010001	011001001	001001011	001000001	Z^1 , რეგ. X^1
5	DamTI	r	o	o	r	o	o	r	
6	DamTIK (ს2)	1075	4312	111	1075	4312	111	1075	
7	DamTIK (ს1)	196	136	111	196	136	111	196	
8	DamTIKO	011000100	010001000	001101111	011000100	010001000	001101111	011000100	Z^2 , რეგ. X^2
4*	DasTIKO	010011110	010001101	001110011	011010001	011001001	001001011	001000001	Z^1 , რეგ. X^1
9	ShifTIKO (mod_2)	001011010	000000101	000011100	000010101	001000001	000100100	010000101	m, რეგ. X^3
10	ShifTIK	90	5	28	21	65	36	133	
11	ShifTIK (ს2)	90	5	28	21	65	36	4309	
12	ShifTI (D_i)	Z	|		⊥	A	$	3	

ცხრილი 2.19

	1	2	3	4	5	6	7	8	9
1	ShifTI (D_i)	Z	|		⊥	A	$	ȝ	
2	ShifTIK (ს2)	90	5	28	21	65	36	4309	
3	ShifTIK	90	5	28	21	65	36	133	
4	ShifTIKO	001011010	000000101	000011100	000010101	001000001	000100100	010000101	Z^1 , რეგ. X^1
5	DamTI	r	ο	o	r	ο	o	r	
6	DamTIK (ს2)	1075	4312	111	1075	4312	111	1075	
7	DamTIK (ს1)	196	136	111	196	136	111	196	
8	DamTIKO	011000100	010001000	001101111	011000100	010001000	001101111	011000100	Z^2, რეგ. X^2
4*	ShifTIKO	001011010	000000101	000011100	000010101	001000001	000100100	010000101	Z^1 , რეგ. X^1
9	DasTIKO (mod_2)	010011110	010001101	001110011	011010001	011001001	001001011	001000001	m, რეგ. X^3
10	DasTIK (ს1)	158	141	115	209	201	75	65	
11	DasTIK (ს2)	4334	4317	115	1088	1080	75	65	
12	DasTI	ƀ	ơ	s	p	и	k	A	

ცხრილი 2.20

	1	2	3	4	5	6	7	8	9
1	DasTI	ъ	ო	s	p	и	k	A	
2	DasTIK(ს2)	4334	4317	115	1088	1080	75	65	
3	DasTIK(ს1)	158	141	115	209	201	75	65	
4	DasTIKO	010011110	010001101	001110011	011010001	011001001	001001011	001000001	Z^1 , რეგ. X^1
5	DamTI	r	o	o	r	o	o	r	
6	DamTIK(ს2)	1075	4312	111	1075	4312	111	1075	
7	DamTIK(ს1)	196	136	111	196	136	111	196	
8	DamTIKO	011000100	010001000	001101111	011000100	010001000	001101111	011000100	Z^2, რეგ. X^2
4*	DasTIKO	010011110	010001101	001110011	011010001	011001001	001001011	001000001	
9	ShifTIKO (Σ)	101100010	100010101	011100010	110010101	101010001	010111010	100000101	Σ, რეგ. X^3
10	ShifTIK	306	277	178	405	337	186	261	
11	ShifTI(D_i)	lJ	e	2	hu	o	o	a	

შენიშვნა: მე-11 სტრიქონის ShifTI(D_i)-ში შეტანილი სიმბოლოები კლავიატურაზე არ არსებობს

ცხრილი 2.21

	1	2	3	4	5	6	7	8	9
1	DamTI	r	o	o	r	o	o	r	
2	DamTIK(ს2)	1075	4312	111	1075	4312	111	1075	
3	DamTIK(ს1)	196	136	111	196	136	111	196	
4	DamTIKO	011000100	010001000	001101111	011000100	010001000	001101111	011000100	Z^2, რეგ. X^2
5	¬$_2^4$	100111011	101110111	110010000	100111011	101110111	110010000	100111011	
6	Y_8^1	000000001	000000001	000000001	000000001	000000001	000000001	000000001	
7	დამ. კოდი (Σ)	100111100	101111000	110010001	100111100	101111000	110010001	100111100	Σ, X^1
8	ShifTIKO	101100010	100010101	011100010	110010101	101010001	010111010	100000101	Z^2, რეგ. X^2
7*	დამ. კოდი (Σ)	100111100	101111000	110010001	100111100	101111000	110010001	100111100	
9	DasTIKO	010011110	010001101	001110011	011010001	011001001	001001011	001000001	Σ, X^1
10	DasTIK(ს1)	158	141	115	209	201	75	65	
11	DasTIK(ს2)	4334	4317	115	1088	1080	75	65	
13	DasTI	ъ	ѡ	s	p	и	k	A	

თავი 3. სიმეტრიული სისტემის მეთოდების პროგრამული რეალიზაცია

მოცემულ თავში მოკლედ აღწერილია სიმეტრიული სისტემის ცეზარის, ვიჟინერის, ვერნამის, უნივერსალური და შებრუნებული მატრიცის მეთოდების (ალგორითმების) პროგრამული რეალიზაციის საკითხები. ყველა შემოთავაზებული პროგრამა-დანართები (Application), რომელიც შემუშავებულია C# დაპროგრამების ენაზე და ფუნქციონირებს Microsoft Visual Studio.NET გარემოში[20], შეტანილია საქაღალდეში D:\KRipto_VaKe_GuKo\Name, სადაც Name მიუთითებს იმ მეთოდის პროგრამული მოდულის დასახელებაზე (საქაღალდეზე, კერძოდ, NCezari, Cezari_Bute, NVijineri, და ა. შ.) რომელი მეთოდითაც სრულდება ამა თუ იმ TI დაშიფვრა და/ან გაშიფვრა. ყოველ საქაღალდეში განთავსებულია მოდულის (მეთოდის) შესაბამისი .exe ფაილი (კერძოდ, Ncezari.exe,Cezari_Bute.exe,Nvijineri.exe, და ა.შ.), რომელზედაც ორჯერ დაწკაპუნებით ხდება ამორჩეული პროგრამის შესრულებაზე გაშვება, ჩატვირთვა. უნდა აღინიშნოს, რომ ვინაიდან პროგრამული მოდულები სასწავლო მიზნებისათვის არის დამუშავებული, ყოველი მათგანი შედგება ერთი ან ორი ერთმანეთთან დაკავშირებული ქვეპროგრამისაგან. პირველის შემთხვევაში სრულდება როგორც ტექსტის დაშიფვრა ისე მისი გაშიფვრა. მეორე შემთხვევაში ჯერ ხორციელდება საწყისი TI დაშიფვრა, შემდგომ კი მიღებული შიფრ-ტექსტის გაშიფვრა. უნდა აღინიშნოს ისიც, რომ რეალიზებულ ალგორითმებში TI დასაშიფრავად (ანალოგიურად, გასაშიფრავად) ძირითადად გამოიყენება მანიპულაციები, როგორიცაა მაგალითად, ლოგიკური ოპერაცია „შეკრება ორის მოდულით" - mod_2 და არითმეტიკული ოპერაციები: შეკრება, გამოკლება და გამრავლება, ნაცვლად სიმბოლოების ჩანაცვლებისა, გადანაცვლებისა და ა. შ., როგორც ეს იყო შემოთავაზებული ცეზარის, ვიჟინერის და ვერნამის მეთოდებში (1.1-1.4 ქვეთავებში). ამ მეთოდებიდან ქვემოთ აღწერილ და რეალიზებულ ალგორითმებში გამოყენებულია დაშიფვრისა და გაშიფვრის მხოლოდ ის ძირითადი პრინციპები, რომლებითაც განისაზღვრება დამშიფრავი სიმბოლოების რაოდენობის შერჩევა, DamTI ფორმირების და სხვა მსგავსი საკითხები.

შევნიშნოთ, რომ ყოველ პროგრამულ მოდულში რეალიზებულია “HELP” - მეთოდი, რომელზეც დაწკაპუნებით ეკრანზე აისახება ზოგადი სახის ინფორმაცია, რაც შეიძლება გამოვიყენოთ სისტემის მომხმარებლის მიერ.

3.1. ცეზარის მეთოდი

საქაღალდეში D:\KRipto_VaKe_GuKo\NCezari\NCezari.exe იარლიყზე ორჯერ დაწკაპუნების შედეგად პკ ჩაიტვირთება ცეზარის მეთოდით ინფორმაციის დაშიფვრა - გაშიფვრის მარიალიზებელი პროგრამა. ეკრანზე აისახება 3.1 ნახაზზე ნაჩვენები დიალოგური ფანჯარა, რომელშიც მითითება „შეიტანეთ დასაშიფრი ტექსტი“ გულისხმობს იმას, რომ მის ქვევით განთავსებულ გამოყოფილ ზოლში აუცილებლია DasTI-იის შეტანა, რომელიც შეიძლება იყოს ერთი სიმბოლო მაინც ან სიმბოლოების S_i (i=1,2,3,…, Z) მიმდევრობა β ალფაბეტიდან (იხ. ცხრ.2.1-2.3 ან ცხრ.2.4). შევნიშნოთ, რომ მითითებულ ველში დასაშიფრი ტექსტის - DasTI არ შეტანის შემთხვევაში პროგრამა გამოიმუშავებს რეკომენდაციას (შეტყობინებას), რომელიც აუცილებელია გავითვალისწინოთ, რათა შესაძლებელი გახდეს სისტემასთან მუშაობის გაგრძელება.

ცეზარის მეთოდის პროგრამული რეალიზაცია სრულდება ორ ეტაპად, ორი ქვეპროგრამის თანამიმდევრული შესრულებით, ღილაკებზე „ტექსტის დაშიფვრა“ და „ტექსტის გაშიფვრა“ დაწკაპუნებით, შესაბამისად. ღილაკზე „ტექსტის დაშიფვრა“ დაწკაპუნების შედეგად (იხ. ნახ.3.1):

ა) ეკრანზე მიმთითებელი ფრაზის „სიმბოლოს კოდი, რომლითაც უნდა დაიშიფროს ტესტი“ აისახება კოდის რიცხვითი მნიშვნელობა K_1^k, რომელიც შეირჩევა სისტემის მიერ (პროგრამის შესრულების პროცესში) შემთხვევითი რიცხვების გენერირებით C# ენაში განსაზღვრული System.Random კლასის მეთოდით;

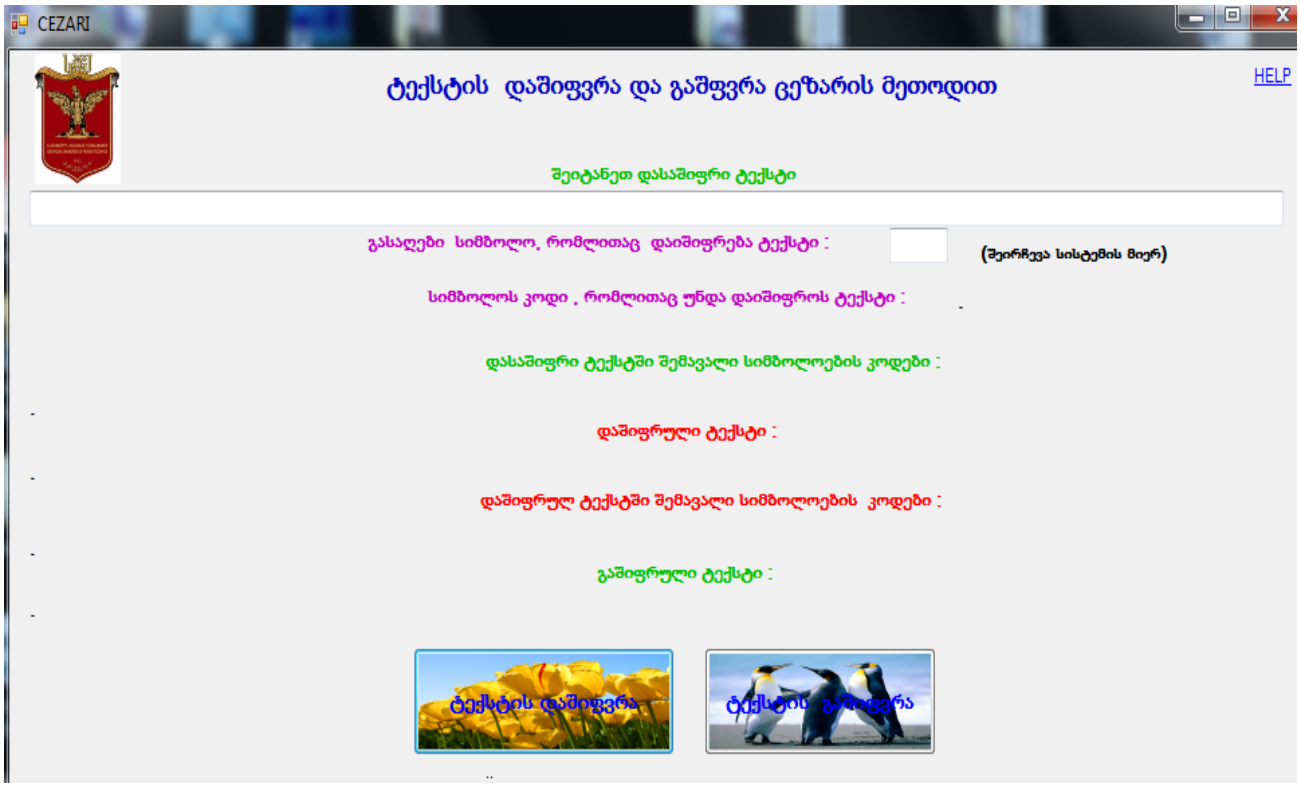
ბ) მიმთითებლის „გასაღები სიმბოლო, რომლითაც დაიშიფრება ტექსტი“-ის გვერდით სპეციალურად გამოყოფილ უჯრაში სისტემის მიერ შეიტანება K_1^k კოდის შესაბამისი დამშიფრავი სიმბოლო K_1;

გ) მიმთითებლის „დასაშიფრ ტექსტში შემავალი სიმბოლოების კოდები“-ის ქვემოთ გამოყოფილ სტრიქონში აისახება DasTI-ში შემავალი სიმბოლოების S_i (i=1,2,3,…, Z) რიცხვითი კოდების მნიშვნელობები S_i^k, განსაზღვრული β ალფაბეტიდან (იხ. ცხრ.2.1-2.3 ან ცხრ.2.4);

დ) დაშიფრული TI - შიფრ-ტექსტი (განსაზღვრული γ ალფაბეტში შემავალი სიმბოლოების სიმრავლეზე იხ. ცხ.2.5-2.10) აისახება მიმთითებლის „დაშიფრული ტექსტი“-ს ქვემოთ გამოყოფილ სტრიქონში;

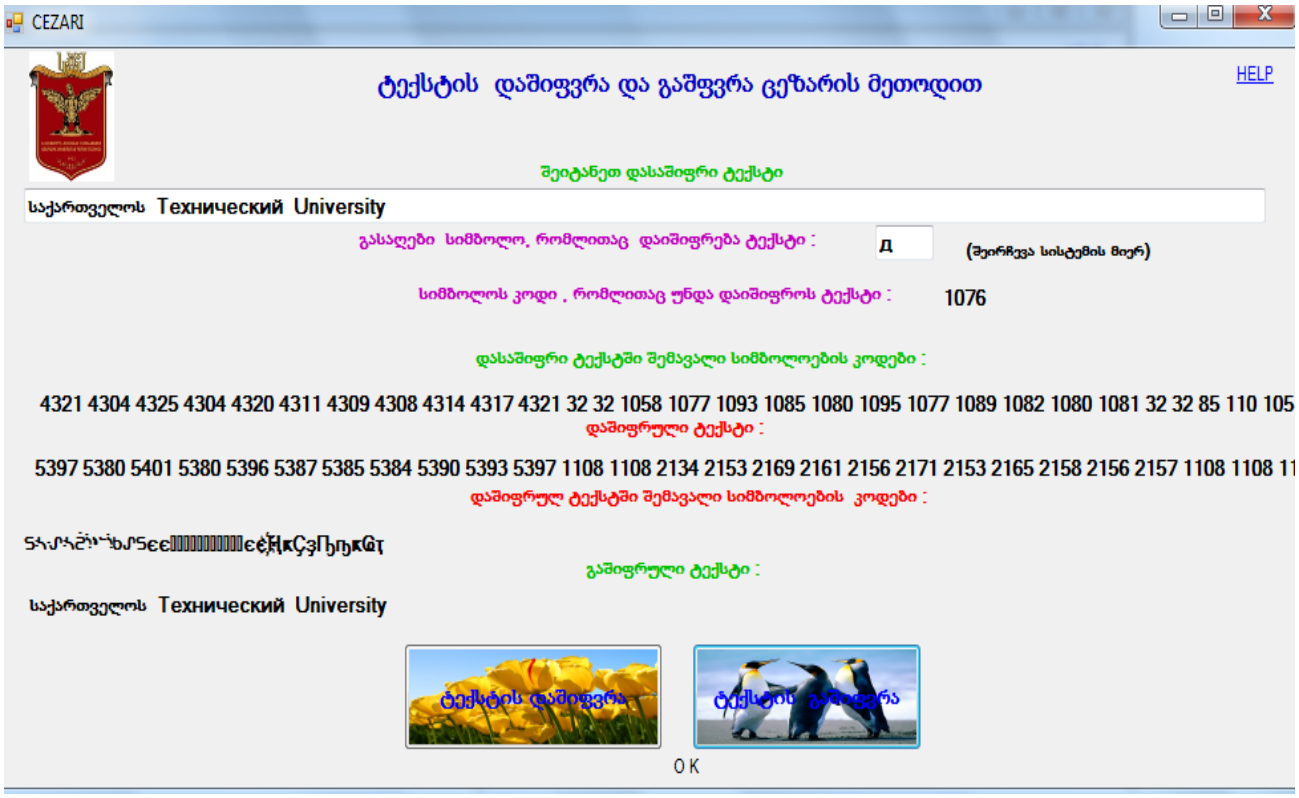
ე) დამხმარე ინფორმაცია, რომელიც შეიძლება გამოყენებულ იქნეს პროგრამული რეალიზაციის სხვადასხვა ეტაპზე მიღებული შედეგის ინდივიდუალური გზით შესამოწმებლად მიმთითებლის „დაშიფრულ ტექსტში შემავალი სიმბოლოების კოდების“ ქვემოთ გამოყოფილ სტრიქონში აისახება შიფრ-ტექსტში შემავალი სიმბოლოების D_i (i=1,2,3,…, Z) რიცხვითი კოდების D_i^k მნიშვნელობები .

მეორე ეტაპზე, ღილაკზე „ტექსტის გაშიფვრა“ დაწკაპუნებით, შესრულდება შიფრ-ტექსტის გაშიფვრა და ეკრანზე აისახება ტექსტი, რომელიც ანალოგიური იქნება DasTI - საწყისი დასაშიფრი ტექსტური ინფორმაციის.



ნახ. 3.1. ცეზარის მეთოდის სარეალიზაციო დიალოგური ფანჯარა

ცეზარის მეთოდით ტექსტური ინფორმაციის დაშიფვრისა და გაშიფვრის პროგრამული რეალიზაციის შესრულების შედეგები რეალურ მაგალითზე ნაჩვენებია 3.2 ნახაზზე.

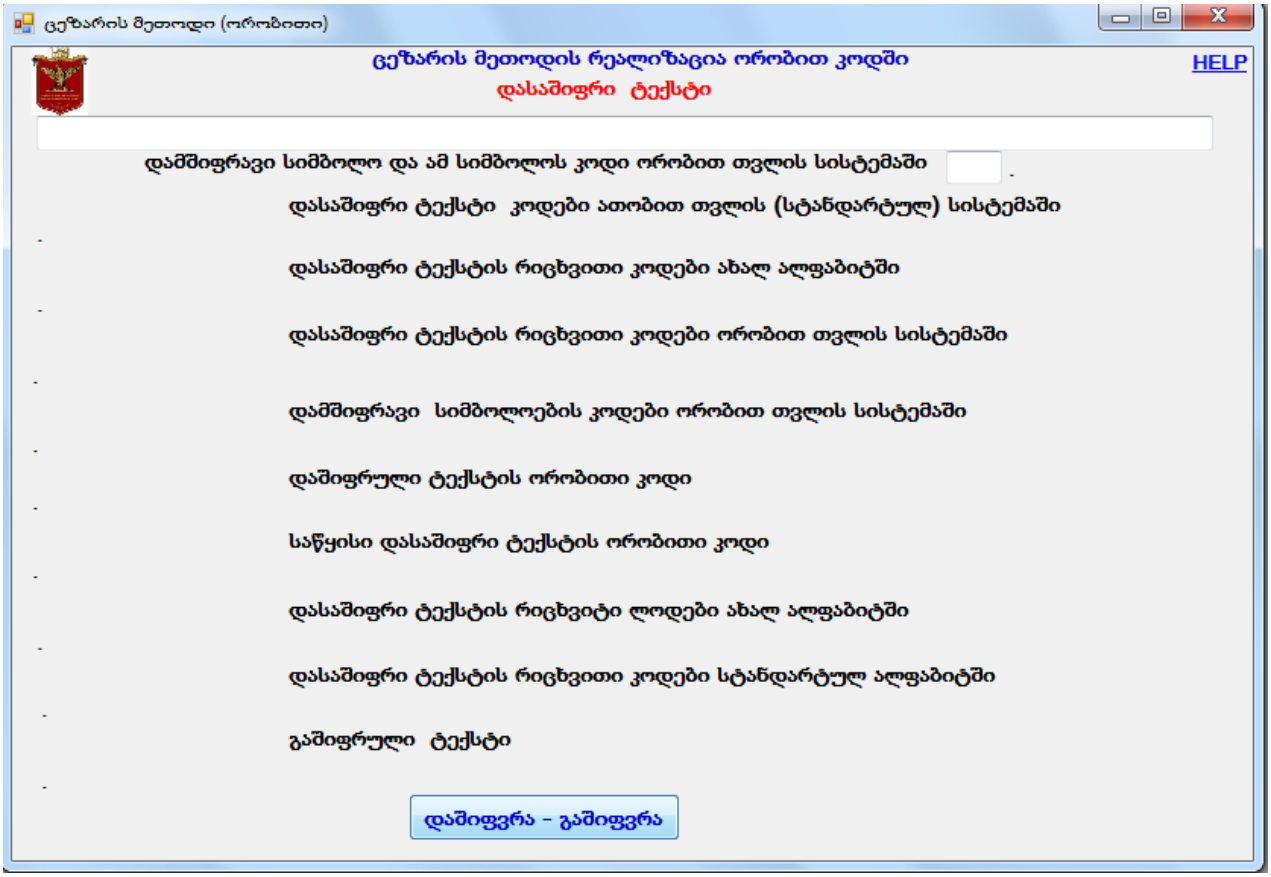


ნახ. 3.2. ცეზარის მეთოდის პროგრამული რეალიზაციის დემონსტრირება რეალურ მაგალითზე

ცეზარის მეთოდი (დაშიფვრა - გაშიფვრა ორობითი კოდით)

საქაღალდეში D:\KRipto_VaKe_GuKo\Cezari_Bute\Cezari_BUTE.exe იარლიყზე ორჯერ დაწკაპუნების შედეგად პკ ჩაიტვირთება ცეზარის მეთოდით ინფორმაციის დაშიფვრა-გაშიფვრის მარიალიზებელი პროგრამა. ეკრანზე აისახება 3.3 ნახაზზე ნაჩვენები დიალოგური ფანჯარა, რომელშიც მითითება „დასაშიფრი ტექსტი“ გულისხმობს, იმას რომ მის ქვევით განთავსებულ გამოყოფილ ზოლში აუცილებლად უნდა იქნეს DasTI-ია შეტანილი, რომელიც შეიძლება იყოს ერთი სიმბოლო მაინც ან სიმბოლოების S_i (i=1,2,3,…, Z) მიმდევრობა ɳ ალფაბეტიდან (იხ. ცხრ.2.11). ɳ ალფაბეტიდან აგრეთვე შეტანილ იქნეს ერთი სიმბოლო მიმთითებლის “დამშიფრავი სიმბოლო და ...“ გვერდით გამოყოფილ უჯრაში. შევნიშნოთ, რომ მითითებულ ველში დასა-

შიფრი ტექსტის (DasTI) ან/და დამშიფრავი სიმბოლოს არშეტანის შემთხვევაში პროგრამა გამოიმუშავებს რეკომენდაციას (შეტყობინებას), რომელიც აუცილებლია გავითვალისწინოთ, რათა შესაძლებელი გახდეს სისტემასთან მუშაობის გაგრძელება.



ნახ. 3.3. ცეზარის მეთოდის პროგრამული რეალიზაცია ორობით კოდში

ცეზარის მეთოდის პროგრამული რეალიზაცია სრულდება ერთ ეტაპად, ღილაკზე „დაშიფვრა-გაშიფვრა“ დაწკაპუნებით. მარეალიზებელი ალგორითმის დაშიფვრა-გაშიფვრის პროცედურული ეტაპები, მოცემულია 3.1 და 3.2 ცხრილების სახით და დეტალურად არის აღწერილი 2.4 ქვეთავში.

შემოთავაზებული ალგორითმის ძირითადი არსი მდგომარეობს იმაში, რომ დასაშიფრი და დამშიფრავი სიმბოლოების ათობით თვლის სისტემაში მოცემული კოდები ჯერ გადაიყვანება ორობით კოდებში და მერე მათზე ხორციელდება (როგორც დაშიფვრის, ისე გაშიფვრის შემთხვევებში) ოპერაცია შეკრება mod_2.

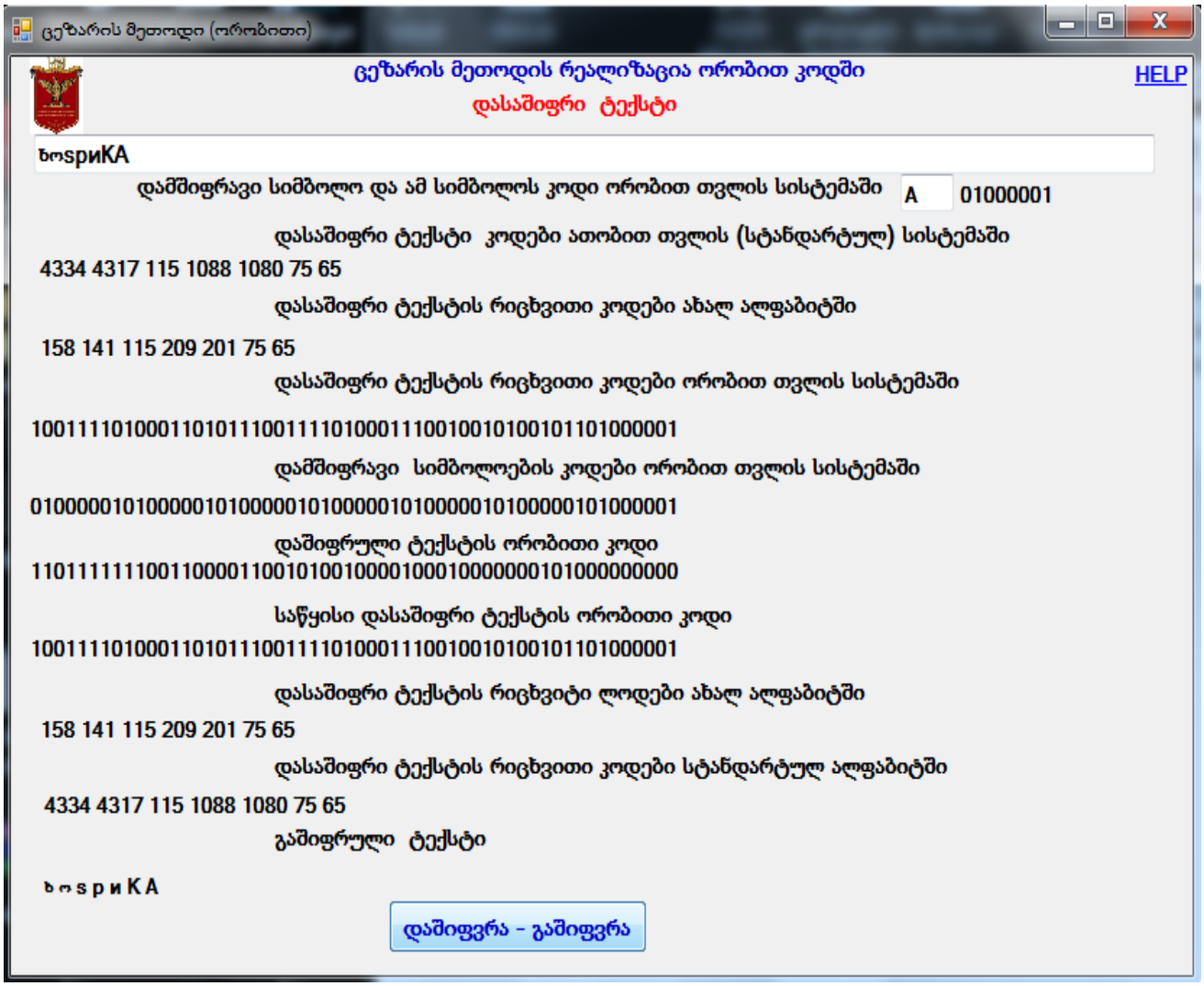
ცხრილი 3.1

	1	2	3	4	5	6	7	8	9
1	DasTI	ъ	ო	s	p	и	k	A	
2	DasTIK (ს2)	4334	4317	115	1088	1080	75	65	
3	DasTIK (ს1)	158	141	115	209	201	75	65	
4	DasTIKO	10011110	10001101	01110011	11010001	11001001	01001011	01000001	Z^1 , რეგ. X^1
5	DamTI	A	A	A	A	A	A	A	
6	DamTIK (ს2)	65	65	65	65	65	65	65	
7	DamTIK (ს1)	65	65	65	65	65	65	65	
8	DamTIKO	01000001	1000001	1000001	1000001	1000001	1000001	1000001	Z^2 , რეგ. X^2
4*	DasTIKO	10011110	10001101	01110011	11010001	11001001	01001011	01000001	Z^1 , რეგ. X^1
9	ShifTIKO (mod_2)	11011111	11001100	00110010	10010000	10001000	00001010	00000000	m, რეგ. X^3
10	ShifTIK	223	204	50	144	136	10	0	
11	ShifTIK (ს2)	1102	1083	50	4320	4312	10	0	
12	ShifTI (D_i)	ю	л	2	რ	o			

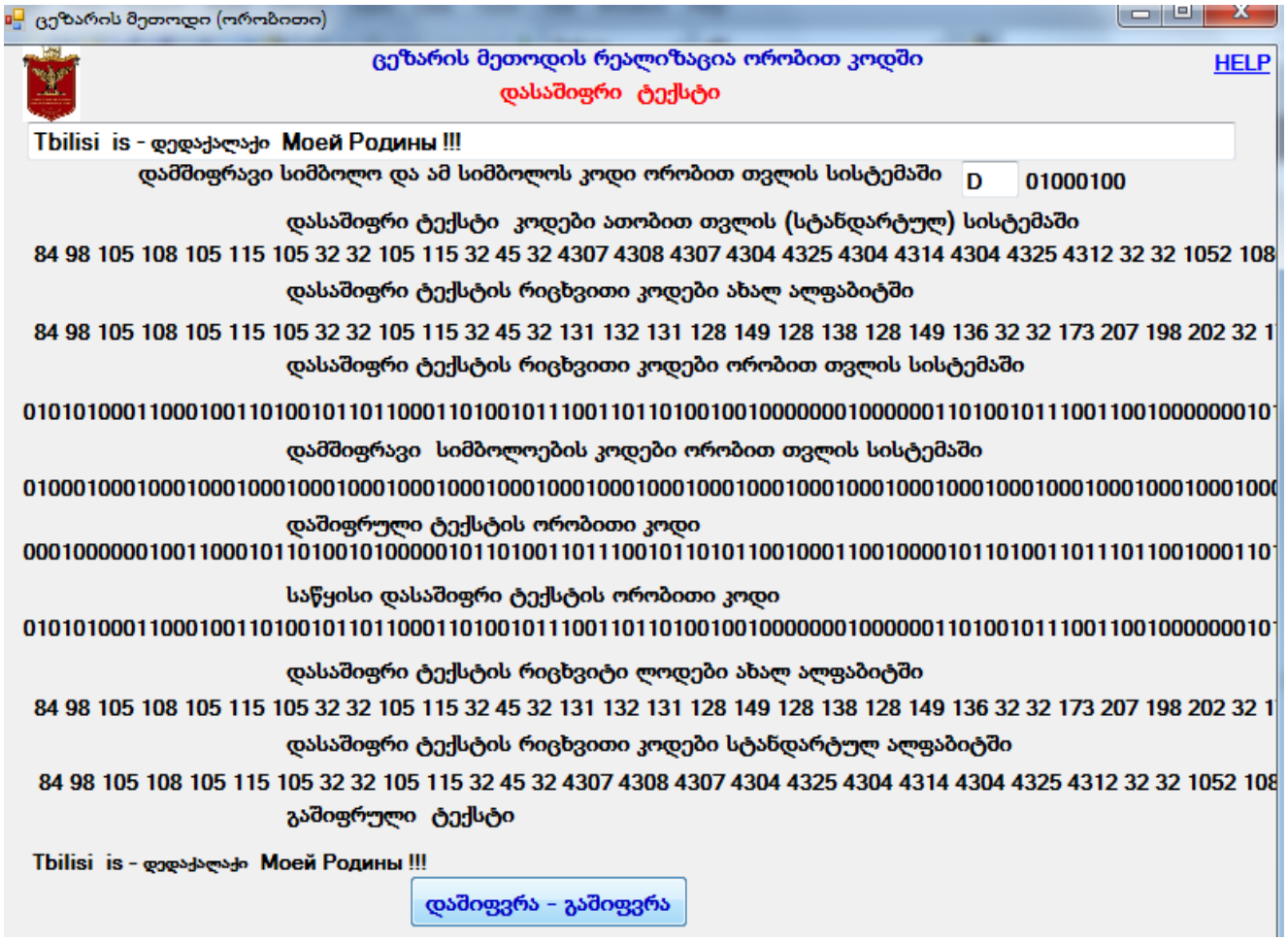
ცხრილი 3.2

	1	2	3	4	5	6	7	8	9
1	ShifTI (D_i)	ю	л	2	р	о			
2	ShifTIK (ს2)	1102	1083	50	4320	4312	10	0	
3	ShifTIK	223	204	50	144	136	10	0	
4	ShifTIKO	11011111	11001100	00110010	10010000	10001000	00001010	00000000	Z^1 , რეგ. X^1
5	DamTI	A	A	A	A	A	A	A	
6	DamTIK (ს2)	65	65	65	65	65	65	65	
7	DamTIK (ს1)	65	65	65	65	65	65	65	
8	DamTIKO	01000001	1000001	1000001	1000001	1000001	1000001	1000001	Z^2, რეგ. X^2
4*	ShifTIKO	11011111	11001100	00110010	10010000	10001000	00001010	00000000	Z^1 , რეგ. X^1
9	DasTIKO (mod_2)	10011110	10001101	01110011	11010001	11001001	01001011	01000001	m, რეგ. X^3
10	DasTIK (ს1)	158	141	115	209	201	75	65	
11	DasTIK (ს2)	4334	4317	115	1088	1080	75	65	
12	DasTI	ъ	ო	s	p	и	k	A	

3.4 და 3.5 ნახაზებზე ნაჩვენებია დამუშავებული პროგრამული მოდულის შესრულების შედეგები კონკრეტულ მაგალითებზე. შევნიშნოთ, რომ 3.4 და 3.5 ნახაზებზე დეტალურადაა ნაჩვენები დაშიფვრის მარეალიზებელი ალგორითმის ამა თუ იმ ეტაპზე გამომუშავებული შუალედური შედეგები და შიფრ-ტექსტი. ხოლო რაც შეეხება გაშიფვრის პროცედურების ზოგიერთ შუალედურ შედეგებს ისინი 3.4 ნახ-ზე ნაჩვენები არ არის, რადგან TI დაშიფვრა და გაშიფვრა ხორციელდება ერთი და იგივე პროცედურების გამოყენებით, თუმცა განსხვავება მდგომარეობს იმაში, რომ გაშიფვრის პროცედურები სრულდება უკუ მიმდევრობით, ვიდრე ეს ხორციელდება დაშიფვრის შემთხვევაში.



ნახ.3.4. ცეზარის მეთოდის პროგრამული რეალიზაცია ორობით კოდში, რეალურ მაგალითზე



ნახ.3.5. ცეზარის მეთოდის პროგრამული რეალიზაცია ორობით კოდში, რეალურ მაგალითზე

3.2. ვიჟინერის მეთოდი

საქაღალდეში D:\KRipto_VaKe_GuKo\NVijineri\NVIJINERI.exe იარლიყ-ზე ორჯერ დაწკაპუნების შედეგად PC ჩაიტვირთება ვიჟინერის მეთოდით ინფორმაციის დაშიფვრა-გაშიფვრის მარეალიზებელი პროგრამა. ეკრანზე აისახება 3.6 ნახაზზე ნაჩვენები დიალოგური ფანჯარა, რომელშიც მითითება „შეიტანეთ დასაშიფრი ტექსტი“ გულისხმობს იმას, რომ მის ქვევით განთავსებულ გამოყოფილ ზოლში აუცილებელია დასაშიფრი ტექსტის (DasTI) შეტანა, რომელიც შეიძლება იყოს ერთი სიმბოლო ან სიმბოლოების S_i (i=1,2,3,…, Z) მიმდევრობა β ალფაბეტიდან (იხ. ცხრ.2.1-2.3). შევნიშნოთ, რომ მითითებულ ველში დასაშიფრი ტექსტის არ შეტანის შემთხვევაში პროგრამა გამოიმუშავებს რეკომენდაციას (შეტყობინებას),

რომელიც აუცილებლია გავითვალისწინოთ, რათა შესაძლებელი გახდეს სისტემასთან შემდგომი მუშაობის გაგრძელება.

ვიჟინერის მეთოდის პროგრამული რეალიზაცია სრულდება ორ ეტაპად, ორი ქვეპროგრამის თანამიმდევრული შესრულებით, შესაბამისად ღილაკებზე „ტექსტის დაშიფვრა“ და „ ტექსტის გაშიფვრა“ დაწკაპუნებით.

ღილაკზე „ ტექსტის დაშიფვრა“ დაწკაპუნების შედეგად (იხ. ნახ.3.6):

ა) ეკრანზე მიმთითებელი ფრაზის „შეტანილი დამშიფრავი სიმბოლოების კოდები“ აისახება დახურული გასაღების სიმბოლოებს კოდების რიცხვითი მნიშვნელობები K_i^k (i=1,2,3), რომლითაც შეირჩევა სისტემის მიერ (პროგრამის შესრულების პროცესში) შემთხვევითი რიცხვების გენერირებით C# ენაში განსაზღვრული System.Random კლასის მეთოდით;

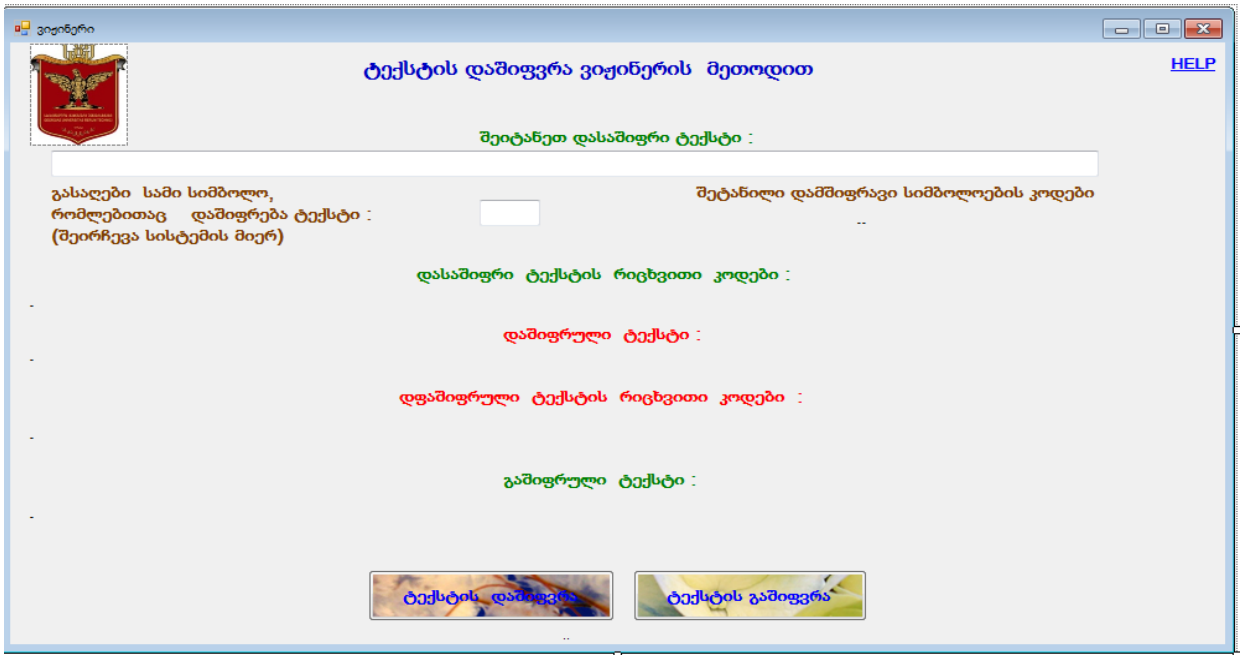
ბ) სისტემის მიერ შეიტანება K_i^k (i=1,2,3) კოდების შესაბამისი დამშიფრავი სიმბოლოები K_1, K_2, K_3 მიმთითებლის „გასაღები სამი სიმბოლო, რომლებითაც დაიშიფრება ტექსტის“ გვერდით სპეციალურად გამოყოფილ უჯრაში (რომლებიც ასევე განისაზღვრება β ალფაბეტიდან);

გ) მიმთითებლის „დასაშიფრი ტექსტის რიცხვითი კოდების“ ქვემოთ გამოყოფილ სტრიქონში აისახება DasTI-ში შემავალი სიმბოლოების S_i (i=1,2,3,…, Z) რიცხვითი კოდების მნიშვნელობები S_i^k, განსაზღვრული β ალფაბეტიდან (იხ. ცხრ.2.1-2.3);

დ) დაშიფრული TI - შიფრ-ტექსტი (განსაზღვრული γ ალფაბეტში შემავალი სიმბოლოების სიმრავლეზე იხ. ცხ.2.5-2.10) აისახება მიმთითებლის „დაშიფრული ტექსტის“ ქვემოთ გამოყოფილ სტრიქონში;

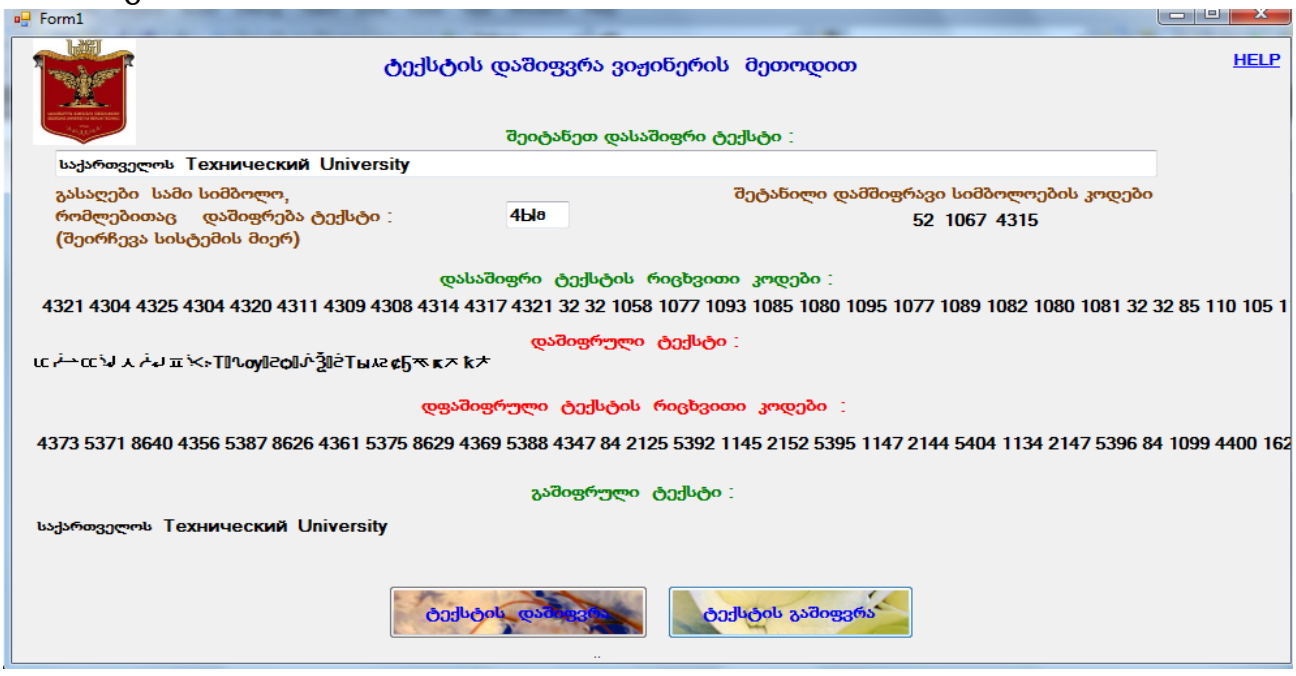
ე) დამხმარე ინფორმაცია, რომელიც შეიძლება გამოვიყენოთ პროგრამული რეალიზაციის სხვადასხვა ეტაპზე მიღებული შედეგის ინდივიდუალური გზით შესამოწმებლად მიმთითებლის „დაშიფრულ ტექსტში შემავალი სიმბოლოების კოდების“ ქვემოთ გამოყოფილ სტრიქონში აისახება შიფრ-ტექსტში შემავალი სიმბოლოების D_i (i=1,2,3,…, Z) რიცხვითი კოდების D_i^k მნიშვნელობები.

მეორე ეტაპზე, ღილაკზე „ტექსტის გაშიფვრა“ დაწკაპუნებით, შესრულდება შიფრ-ტექსტის გაშიფვრა და ეკრანზე აისახება ტექსტი, რომელიც ანალოგიური იქნება DasTI - საწყისი დასაშიფრი ტექსტური ინფორმაციის.



ნახ.3.6. ვიჟინერის მეთოდის სარეალიზაციო დიალოგური ფანჯარა

ვიჟინერის მეთოდით ტექსტური ინფორმაციის დაშიფვრისა და გაშიფვრის პროგრამული რეალიზაციის შესრულების შედეგები ნაჩვენებია 3.7 ნახაზზე.



ნახ.3.7. ვიჟინერის მეთოდის პროგრამული რეალიზაციის დემონსტრირება რეალურ მაგალითზე

3.3. ვერნამის მეთოდით

საქაღალდეში D: \KRipto_VaKe_GuKo \Nvernami \NVERNAMI. exe იარლიყზე ორჯერ დაწკაპუნების შედეგად პკ ჩაიტვირთება ვერნამის მეთოდით ინფორმაციის დაშიფვრა-გაშიფვრის მარეალიზებელი პროგრამა. ეკრანზე აისახება 3.8 ნახაზზე ნაჩვენები დიალოგური ფანჯარა, რომელშიც მითითება „შეიტანეთ დასაშიფრი ტექსტი“ გულისხმობს, იმას რომ მის ქვევით განთავსებულ გამოყოფილ ზოლში აუცილებელია დასაშიფრი ტექსტის (DasTI) შეტანა, რომელიც შეიძლება იყოს ერთი სიმბოლო ან სიმბოლოების S_i (i=1,2,3,…, Z) მიმდევრობა β ალფაბეტიდან (იხ. ცხრ.2.1-2.3). შევნიშნოთ, რომ მითითებულ ველში დასაშიფრი ტექსტის არ შეტანის შემთხვევაში პროგრამა გამოიმუშავებს რეკომენდაციას (შეტყობინებას), რომელიც აუცილებლია გავითვალისწინოთ, რათა შესაძლებელი გახდეს სისტემასთან შემდგომი მუშაობის გაგრძელება.

ვერნამის მეთოდის პროგრამული რეალიზაცია სრულდება ორ ეტაპად, ორი ქვეპროგრამის თანამიმდევრული შესრულებით, შესაბამისად ღილაკებზე „ტექსტის დაშიფვრა“ და „ტექსტის გაშიფვრა“ დაწკაპუნებით.

ღილაკზე „ტექსტის დაშიფვრა“ დაწკაპუნების შედეგად (იხ. ნახ.3.8):

ა) ეკრანზე მიმთითებელი ფრაზის „დასაშიფრ ტექსტში შემავალი სიმბოლოების კოდები“ აისახება დახურული გასაღების სიმბოლოებს კოდების მნიშვნელობები K_i^k(i=1,2,3,..., Z), რომლითაც შეირჩევა სისტემის მიერ (პროგრამის შესრულების პროცესში) შემთხვევითი რიცხვების გენერირებით C# ენაში განსაზღვრული System.Random კლასის მეთოდით;

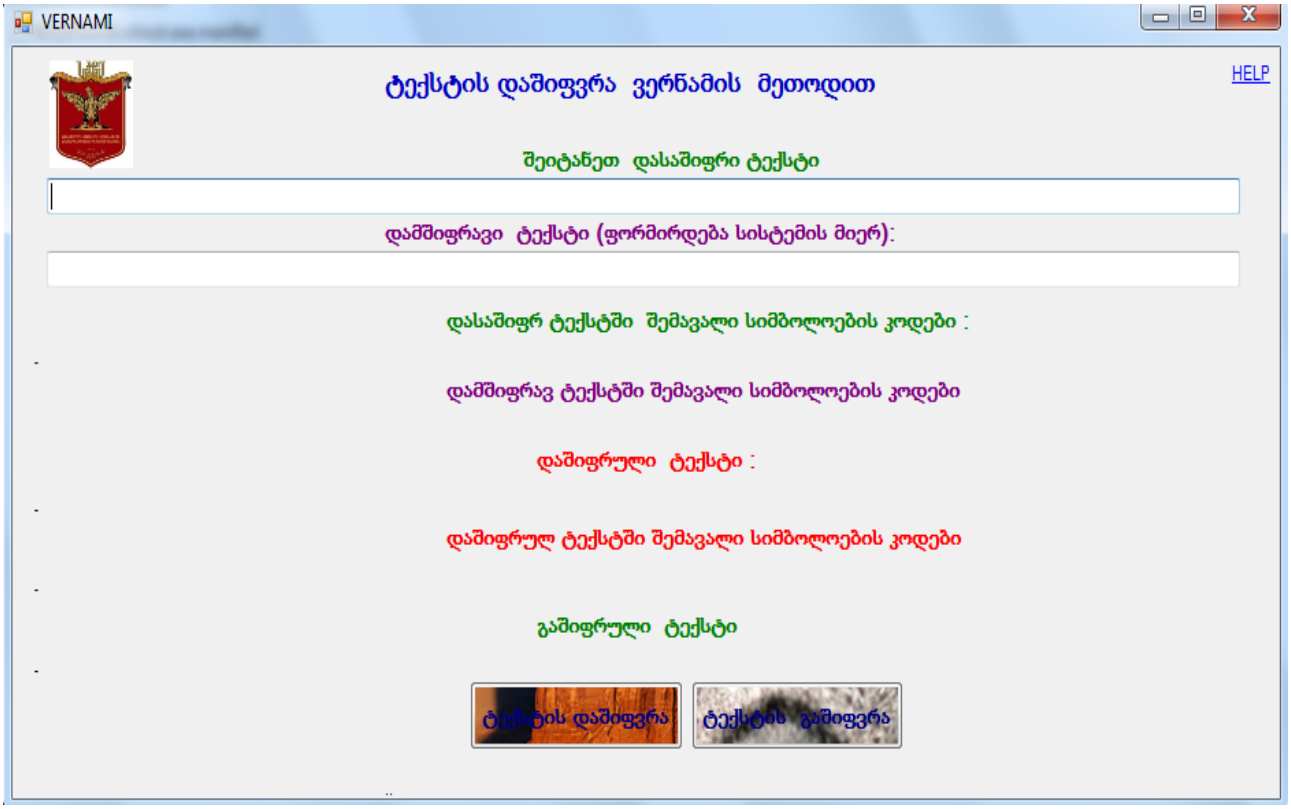
ბ) სისტემის მიერ შეიტანება K_i^k კოდების შესაბამისი დამშიფრავი სიმბოლოები K_1, K_2, …, K_Z, მიმთითებლის „დამშიფრავი ტექსტი (ფორმირდება სისტემის მიერ)” ქვევით სპეციალურად გამოყოფილ უჯრაში (სტრიქონში);

გ) მიმთითებლის „დასაშიფრი ტექსტის რიცხვითი კოდები“-ს ქვემოთ გამოყოფილ სტრიქონში აისახება DasTI-ში შემავალი სიმბოლოების S_i

(i=1,2,3,…, Z) რიცხვითი კოდების მნიშვნელობები S_i^k, განსაზღვრული β ალფაბეტიდან (იხ. ცხრ. 2.1-2.3);

დ) დაშიფრული TI - შიფრ-ტექსტი (განსაზღვრული βB ალფაბეტების შემავალი სიმბოლოების სიმრავლეზე იხ. ცხ.2.5-2.10) აისახება მიმთითებლის „დაშიფრული ტექსტის“ ქვემოთ გამოყოფილ სტრიქონში;

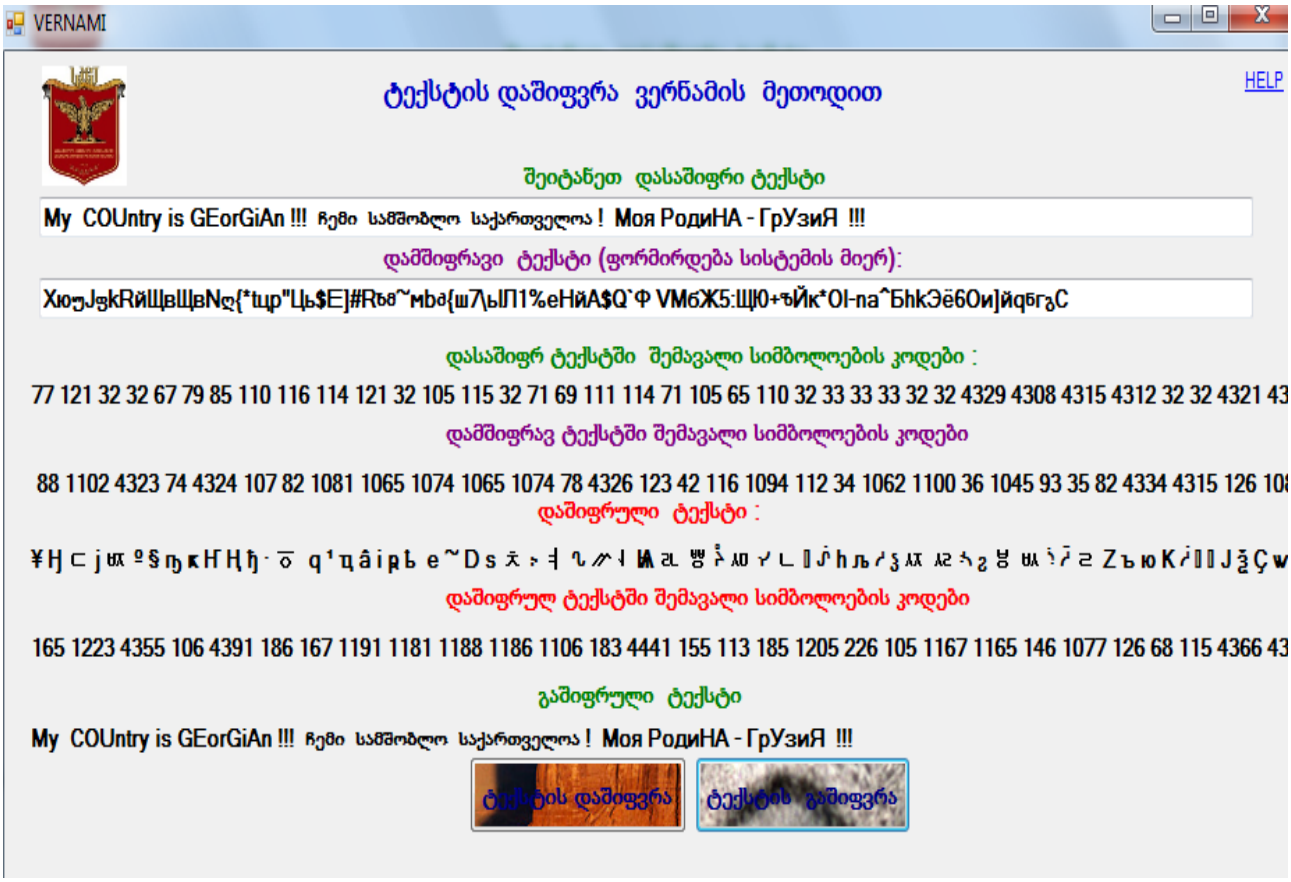
ე) დამხმარე ინფორმაცია, რომელიც შეიძლება გამოვიყენოთ პროგრამული რეალიზაციის სხვადასხვა ეტაპზე მიღებული შედეგის ინდივიდუალური გზით შესამოწმებლად მიმთითებლის „დაშიფრულ ტექსტში შემავალი სიმბოლოების კოდების“ ქვემოთ გამოყოფილ სტრიქონში აისახება შიფრ-ტექსტში შემავალი სიმბოლოების D_i (i=1,2,3,…, Z) რიცხვითი კოდების D_i^k მნიშვნელობები.



ნახ.3.8. ვერნამის მეთოდის სარეალიზაციო დიალოგური ფანჯარა

მეორე ეტაპზე, ღილაკზე „ტექსტის გაშიფვრა“ დაწკაპუნებით, შესრულდება შიფრ-ტექსტის გაშიფვრა და ეკრანზე აისახება ტექსტი, რომელიც ანალოგიური იქნება DasTI - საწყისი დასაშიფრი ტექსტური ინფორმაციის.

ვერნამის მეთოდით ტექსტური ინფორმაციის დაშიფვრისა და გაშიფვრის პროგრამული რეალიზაციის შესრულების შედეგი ნაჩვენებია 3.9 ნახაზზე.



ნახ.3.9. ვერნამის მეთოდის პროგრამული რეალიზაციის დემონსტრირება რეალურ მაგალითზე

3.4. სიმეტრიული სისტემის უნივერსალური მეთოდი

დაშიფვრა/გაშიფვრის შემოთავაზებული მეთოდის და მისი პროგრამული მოდულის (დანართის) ეწოდა „უნივერსალური“ იმიტომ, რომ იგი გვაძლევს საშუალებას დაიშიფროს TI, როგორც ცეზარის, ასევე ვიჟინერის ან ვერნამის ცნობილი სქემების მიხედვით (იხ. 1.1-1.3 ქვეთავები). აგრეთვე, პროგრამული დანართის (რომელიც წინასწარ უნდა შევიტანოთ საქაღალდეში D:\KRipto_VaKe_GuKo\UN_Da_Ga\UN_METH_DaGa.exe) საშუალებით განხორციელდეს ინფორმაციის გაცვლა , ვთქვათ, A და B მომხმარებლებს შორის.

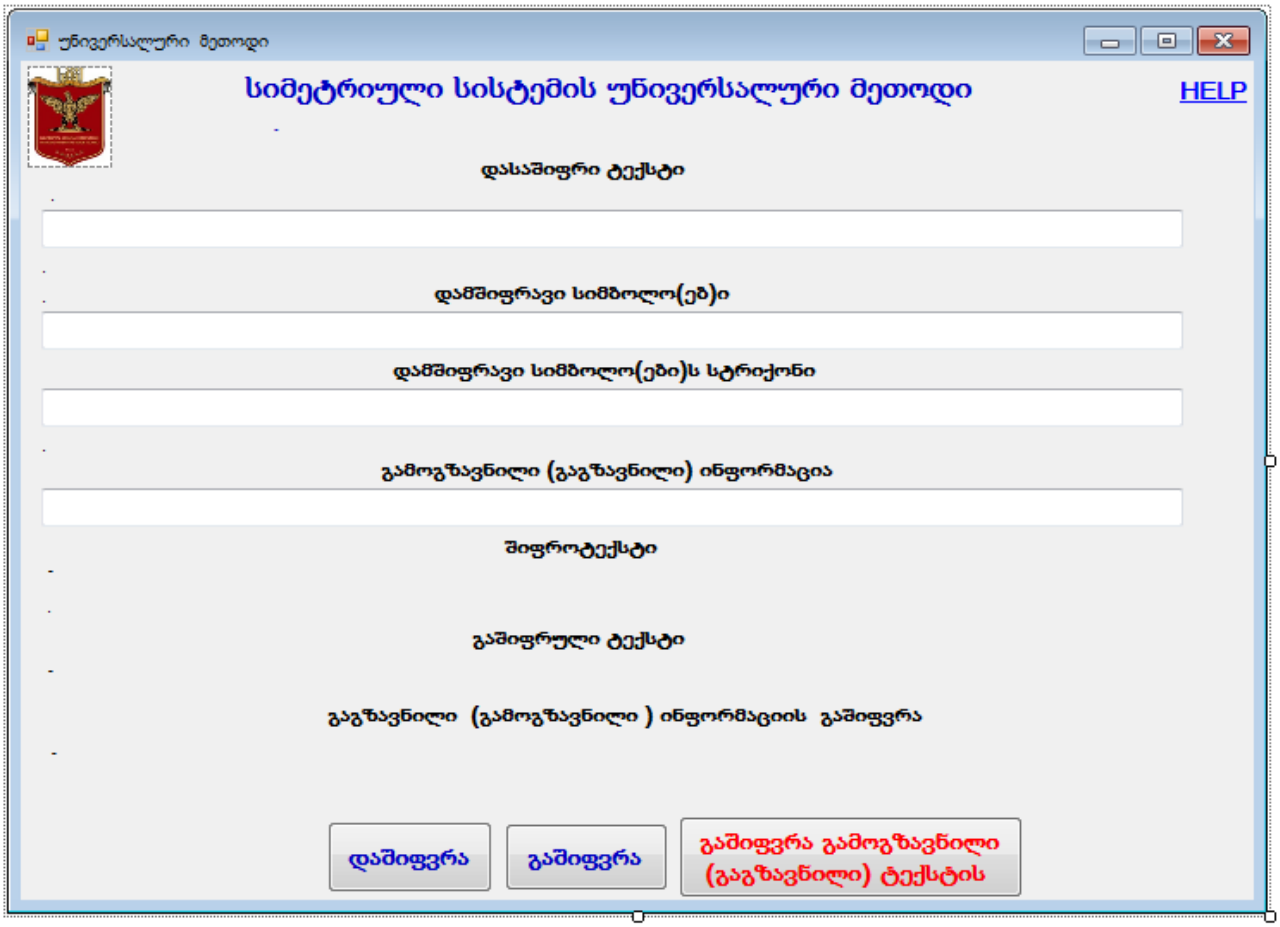
პროგრამის ჩატვირთვამდე (შესრულებაზე გაშვებამდე) მომხმარებლები:

ა) ირჩევენ საიდუმლო გასაღებს;

ბ) ქმნიან საკუთარ პკ-ში საქაღალდეს D:\NewKeKotText_DD.

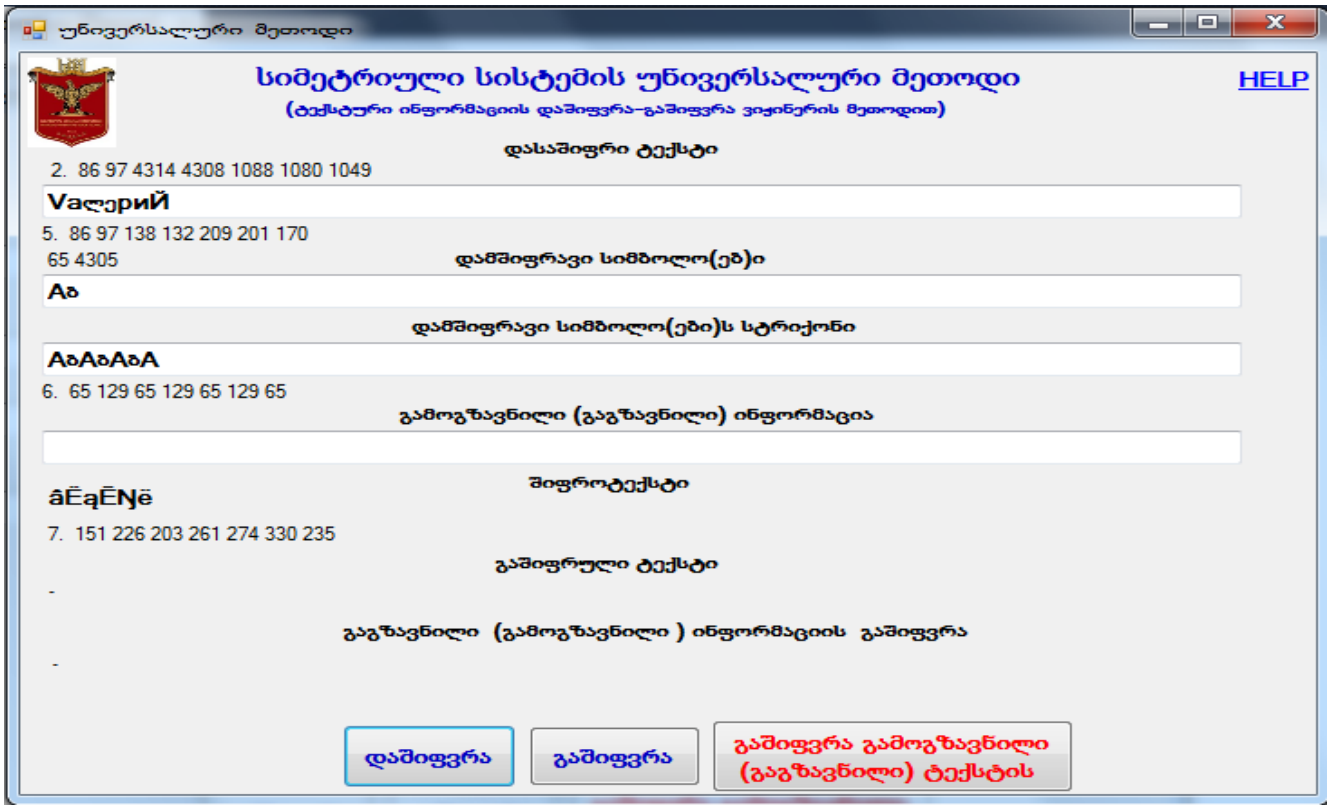
საქაღალდეში D: \KRipto_VaKe_GuKo \UN_Da_Ga \UN_METH_DaGa. exe იარლიყზე ორჯერ დაწკაპუნების შედეგად პკ ჩაიტვირთება უნივერსალური მეთოდით ინფორმაციის დაშიფვრა-გაშიფვრის მარეალიზებელი პროგრამა. ეკრანზე აისახება 3.10 ნახაზზე ნაჩვენები დიალოგური ფანჯარა, რომელშიც მითითება „დასაშიფრი ტექსტი“ გულისხმობს, იმას რომ მის ქვევით განთავსებულ გამოყოფილ ზოლში (აუცილებელია) შევიტანოთ დასაშიფრი ტექსტი DasTI, რომელიც შეიძლება იყოს ერთი სიმბოლო მაინც ან სიმბოლოების S_i (i=1,2,3,…, Z) მიმდევრობა **η** ალფაბეტიდან (იხ. ცხრ.2.11). ამავე ალფაბეტის გამოყენებით მიმთითებლის „დამშიფრავი სიმბოლო(ები)“ ქვევით გამოყოფილ ზოლში (აუცილებელია) შევიტანოთ დამშიფრავი სიმბოლო ან სიმბოლოები (DamTI) K_i (i=1,2,3,…, z, z<Z). შევნიშნოთ, რომ მითითებულ ველებში - DasTI და DamTI სიმბოლოს ან სიმბოლოების არ შეტანის შემთხვევაში პროგრამა გამოიმუშავებს რეკომენდაციას (შეტყობინებას), რომელიც აუცილებელია გავითვალისწინოთ, რათა შესაძლებელი გახდეს სისტემასთან შემდგომი მუშაობის გაგრძელება. შევნიშნოთ აგრეთვე, რომ შეტანილი DamTI გამოყენებით ავტომატურად სრულდება „დამშიფრავი სიმბოლო(ები)ს სტრიქონის ფორმირება, რომელიც მომავალში ასრულებს დაშიფვრის და გაშიფვრის საიდუმლო (დახურული) გასაღების როლს. აღნიშნული სტრიქონის ფორმირების პროცედურა აღწერილია 2.2. ქვეთავში უნივერსალური მეთოდის პროგრამული რეალიზაცია შეიძლება, როგორც ერთ, ასევე ორ ან სამ ეტაპად. პირველი ეტაპის შემთხვევეში ხორციელდება ღილაკებზე „ტექსტის დაშიფვრა“ დაწკაპუნებით, რომლის შესრულების შედეგია შიფრ-ტექსტი (ნახ.3.11) და ორი ტექსტური ტიპის (Notepad რედაქტორის ფორმატში) ფაილი: DSim.txt და

Gmsg.txt, რომლებიც შეიტანება (D:\NewKeKotText_DD) შექმნილ საქაღალდეში.



ნახ.3.10. უნივერსალური მეთოდის სარეალიზაციო დიალოგური ფანჯარა

ვთქვათ, ზემოაღწერილი TI დაშიფვრა (რომლის პროცედურული ეტაპები ნაჩვენებია 3.3 და 3.4 ცხრილებში, ხოლო რეზულტატი ნახ.3.11) განახორციელა A მომხმარებელმა. მეორე ეტაპი, ღილაკზე „ტექსტის გაშიფვრა“ დაწკაპუნებით, რომელიც სრულდება A მომხმარებელს სურვილის მიხედვით, ითვალისწინებს დაშიფრული ფაილის ავტონომიურ გაშიფვრას (ნახ.2.12), შემოწმებას.



ნახ. 3.11. უნივერსალური მეთოდის პროგრამული რეალიზაციის დემონსტრირება რეალურ მაგალითზე

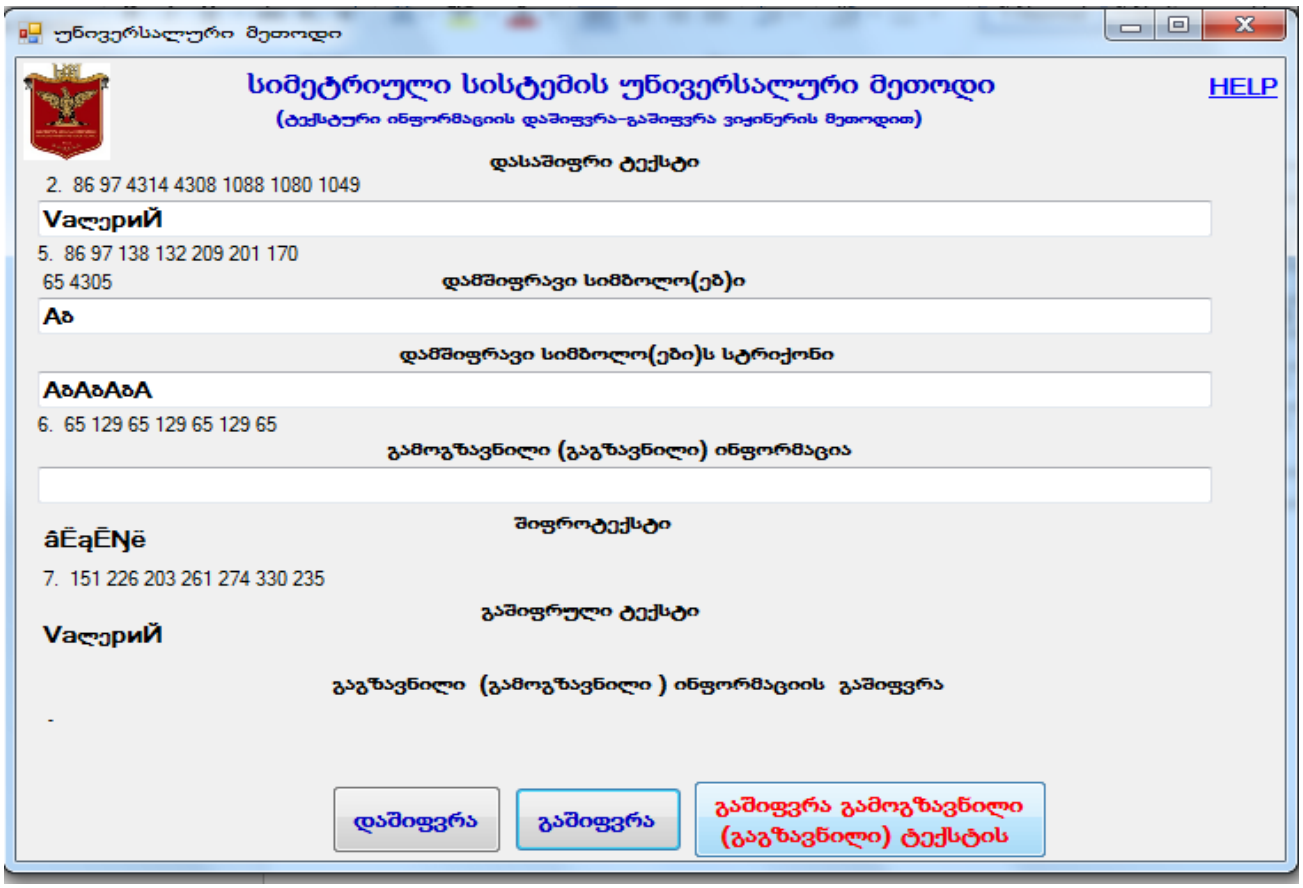
აღვწეროთ თუ რა პროცედურებს ითვალისწინებს გაშიფვრის მესამე ეტაპი. დაუშვათ B მომხმარებელმა თავის პკ-ზე ზემოთ აღწერილი წესით განახორციელა TI ->“ვალერი“ გამოიყენა რა ამისათვის საიდუმლო გასაღები -„gio”. დაშიფვრის პროცედურის შესრულების შედეგად ფორმირებული ფაილები: DSim.txt და Gmsg.txt მან Internet-ით ან ან სხვა საშუალებებით მან გადმოუგზავნა A მომხმარებელს. A მომხმარებელმა მიღებული ფაილები ჩაწერა საქაღალდეში: D:\NewKeKotText_DD და დააწკაპუნა ღილაკზე. აღწერილი მოქმედებების შედეგი ნაჩვენებია 3.13 ნახ-ზე. 3.14 ნახ-ზე კი ნაჩვენებია უნივერსალური მეთოდით კონკრეტული TI დაშიფვრა-გაშიფვრის პროგრამული რეალიზაცია.

ცხრილი 3.3 (2.13)

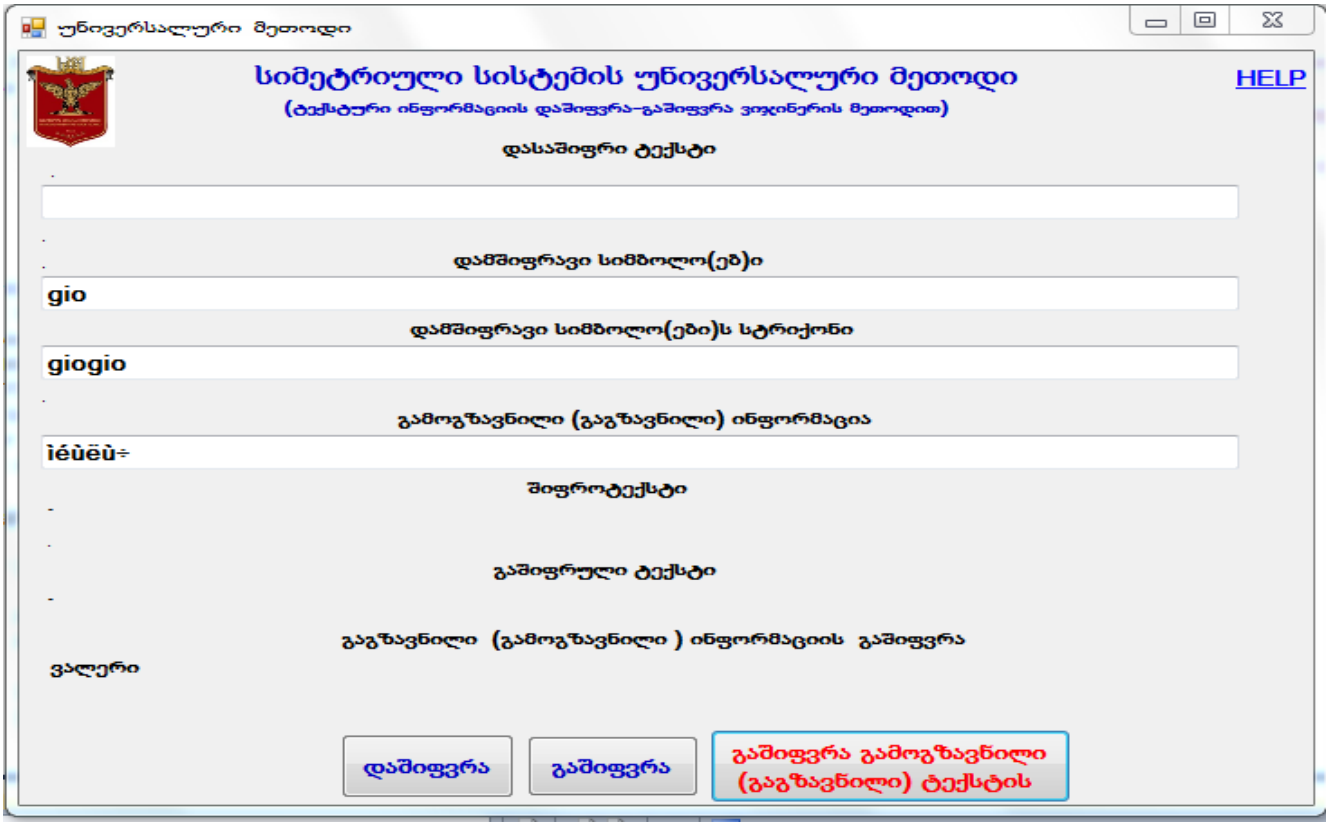
№	1	2	3	4	5	6	7	8
1	DasTI	V	a	ლ	ე	p	и	Й
2	S_i^k(ს2)	86	97	4314	4308	1088	1080	1049
3	DamTI	A	ბ	A	ბ	A	ბ	A
4	D_i^k(ს2)	65	4305	65	4305	65	4305	65
5	S_i^k(ს1)	86	97	138	132	209	201	170
6	D_i^k(ს1)	65	129	65	129	65	129	65
7	$F^k - Q_i^k$ (ცხ. 2.12)	151	226	203	261	274	330	235
8	ShiTI		^a	~E	a|	~E	N,	e

ცხრილი 3.4 (2.14)

№	1	2	3
1	DamTI (ს3)	A	ბ
2	DamTIK(ს2)	65	4305
3	DamTI (ს1)	65	129



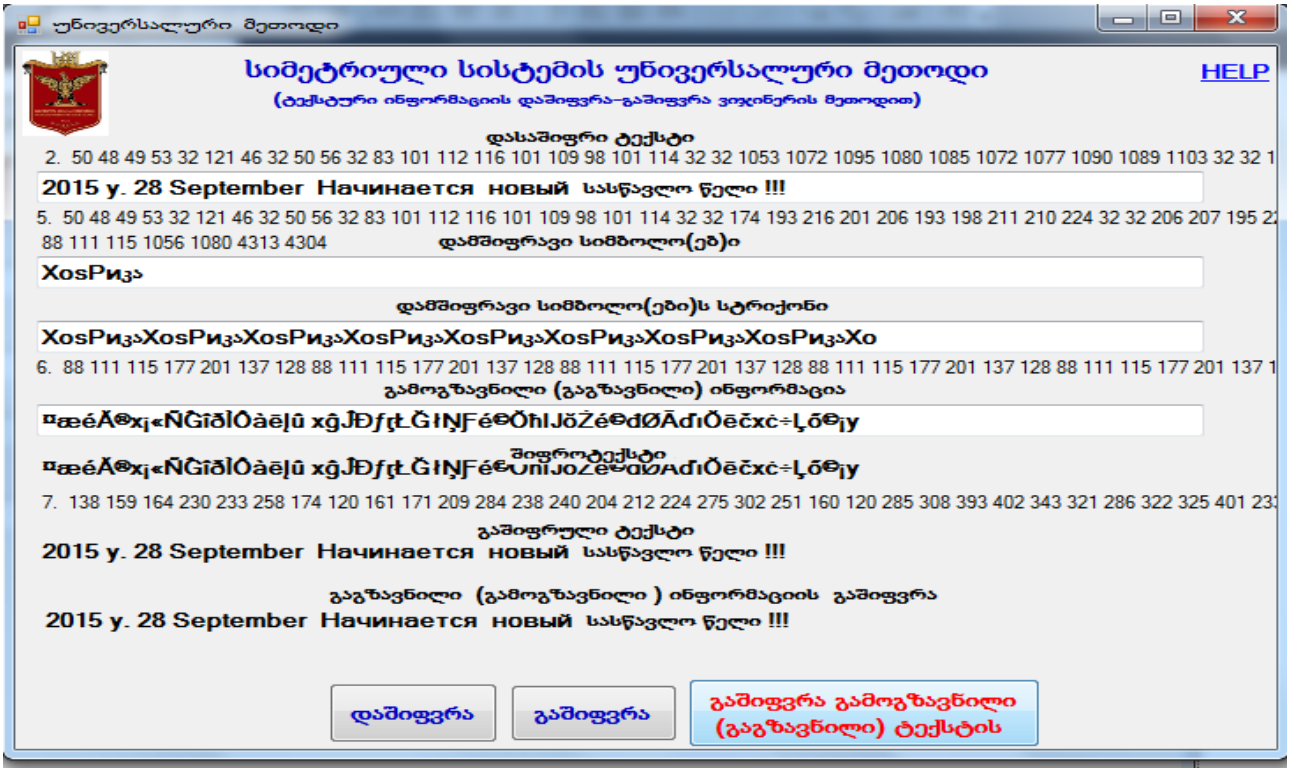
ნახ. 3.12. სიმეტრიული სისტემის უნივერსალური მეთოდის პროგრამული რეალიზაცია



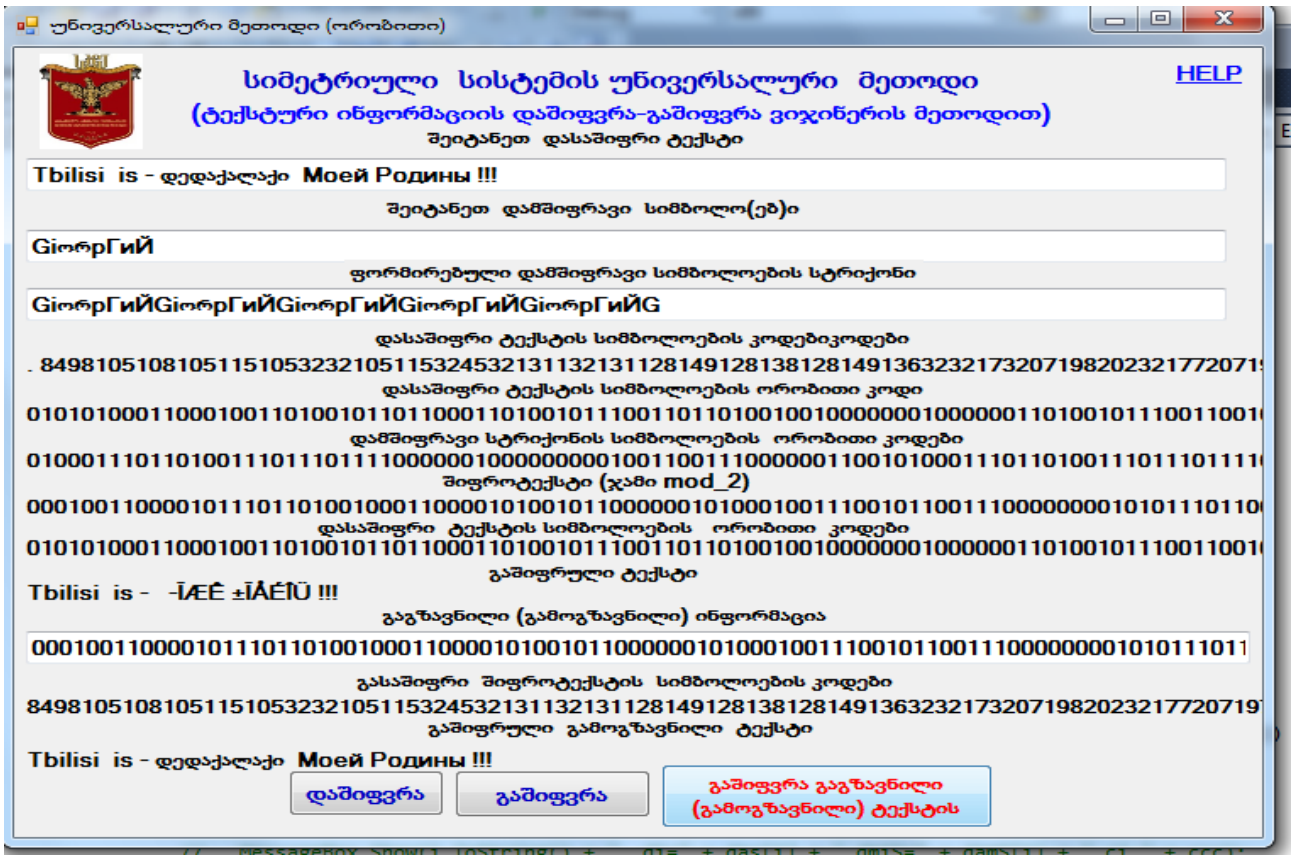
ნახ.3.13. სიმეტრიული სისტემის უნივერსალური მეთოდის პროგრამული რეალიზაცია

სიმეტრიული სისტემის უნივერსალური მეთოდი (დაშიფვრა-გაშიფვრა ორობითი კოდით)

საქაღალდეში D:\KRipto_VaKe_GuKo\UNI_DG_OROBITI\UNI_DG_OROBITI.exe იარლიყზე ორჯერ დაწკაპუნების შედეგად პკ ჩაიტვირთება უნივერსალური მეთოდით ინფორმაციის დაშიფვრა-გაშიფვრის (ორობით კოდში) მარეალიზებელი პროგრამა, რომლის შესრულების შედეგი კონკრეტულ მაგალითზე ნაჩვენებია 3.15 ნახაზზე. შემოთავაზებულ მეთოდის ამ ვარიანში რეალიზებულია ყველა ის პროცედურები, რომლებიც აღწერილია მოცემული პარაგრაფის პირველ ნაწილში, ხოლო დასაშიფრი/გასაშიფრი ინფორმაციის (საწყისი და შუალედური) წარმოდგენის სახეები (ფორმები) დეტალურად არის განხილული 3.1 ქვეთავში.



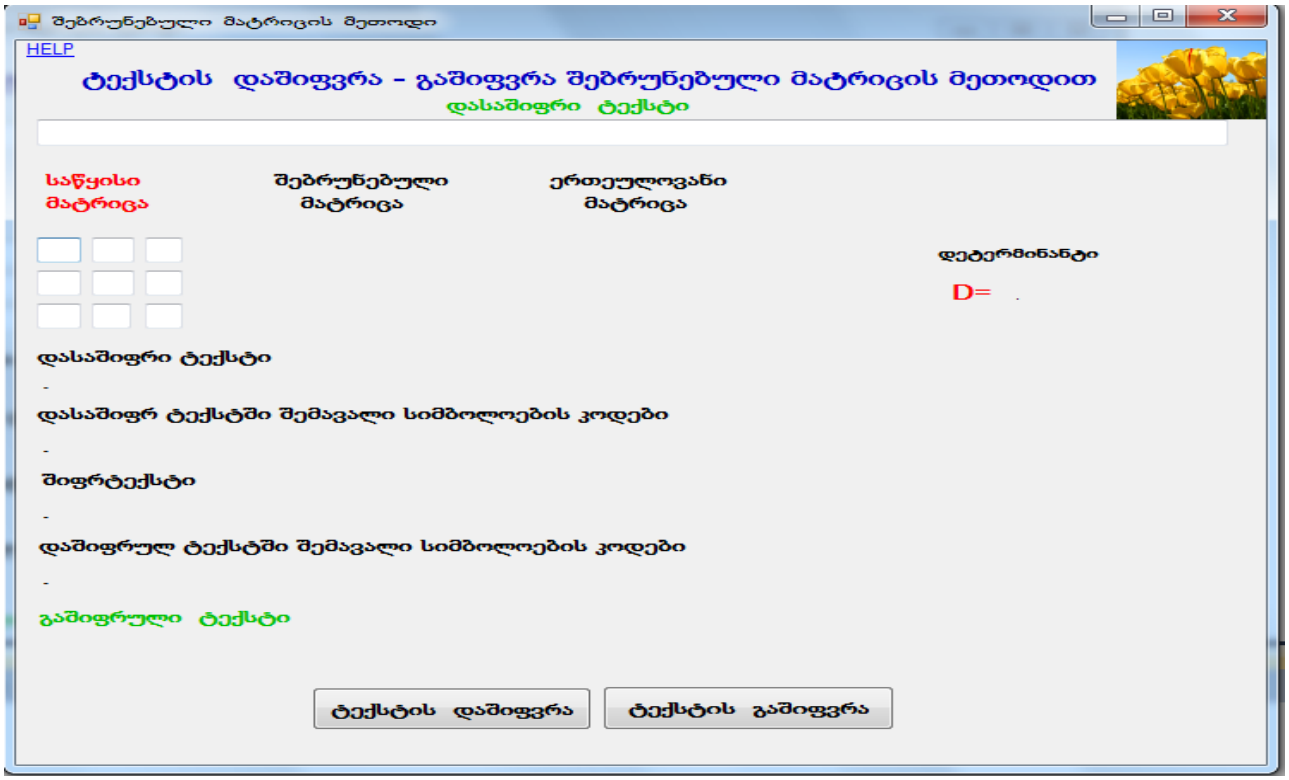
ნახ. 3.14. უნივერსალური მეთოდის პროგრამული რეალიზაციის დემონსტრირება რეალურ მაგალითზე



ნახ. 3.15. უნივერსალური მეთოდის პროგრამული რეალიზაციის დემონსტრირება რეალურ მაგალითზე

3.5. ინფორმაციის დაშიფვრა და გაშიფვრა შებრუნებული მატრიცის მეთოდით

საქაღალდეში D: \KRipto_VaKe_GuKo \Shebrun_Matrica \Shebrun_Matrica. exe იარლიყზე ორჯერ დაწკაპუნების შედეგად პკ ჩაიტვირთება შებრუნებული მატრიცის მეთოდით ინფორმაციის დაშიფვრა-გაშიფვრის მარეალიზებელი პროგრამა.



ნახ.3.16. შებრუნებული მატრიცის მეთოდის პროგრამული რეალიზაცია

ეკრანზე აისახება 3.16 ნახაზზე ნაჩვენები დიალოგური ფანჯარა, რომელშიც მითითება „დასაშიფრი ტექსტი“ გულისხმობს, იმას რომ მის ქვევით განთავსებულ გამოყოფილ ზოლში აუცილებელია დასაშიფრი ტექსტის (DasTI) შეტანა, რომელიც შეიძლება იყოს ერთი სიმბოლო ან სიმბოლოების S_i (i=1,2,3,…, Z) მიმდევრობა β ალფაბეტიდან (იხ. ცხრ.2.1-2.3). შემდგომ ივსება საწყისი მატრიცა რიცხვებით, რომლებიც განსაზღვრულია მთელ დადებით და უარყოფით რიცხვთა სიმრავლეზე. შევნიშნოთ, რომ აღნიშნულ ველებში ინფორმაციის მითითებული სახით არ შეტანის შემთხვევაში პროგ-

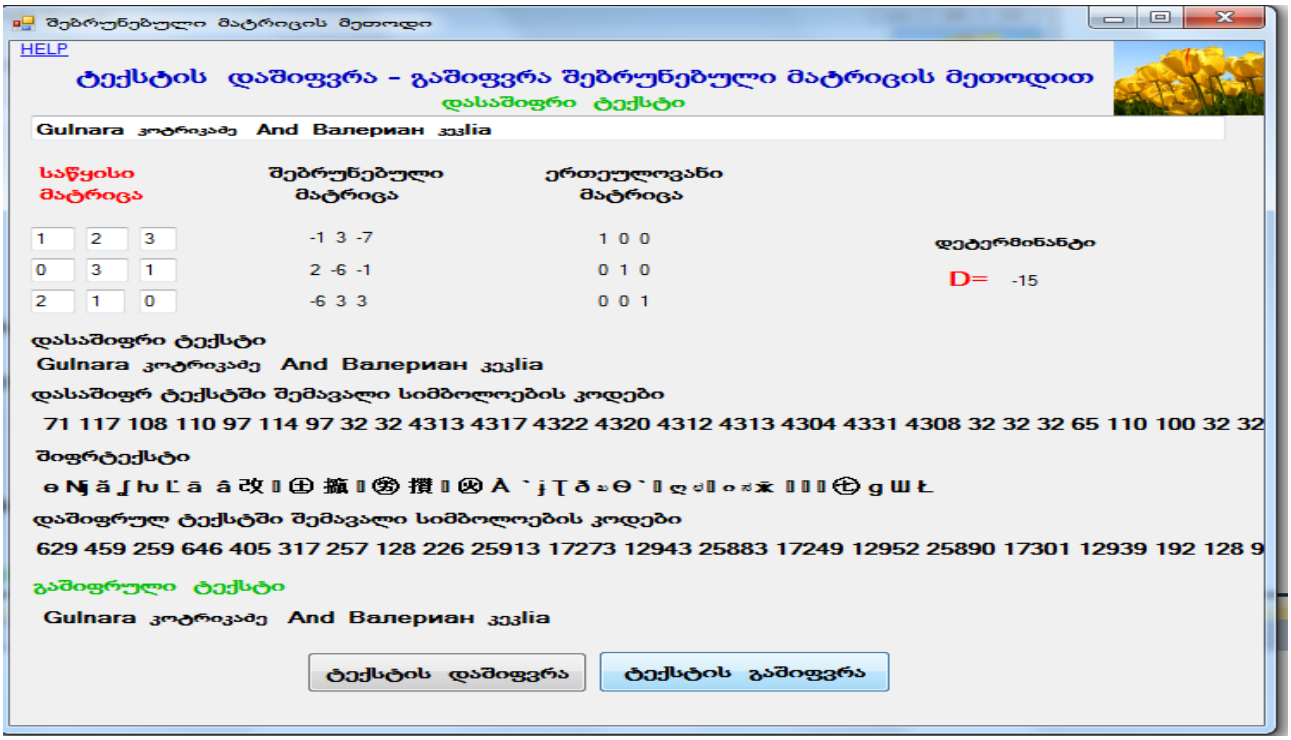
რამა გამოიმუშავებს რეკომენდაციას (შეტყობინებას), რომელიც აუცილებელია გავითვალისწინოთ, რათა შესაძლებელი გახდეს სისტემასთან შემდგომი მუშაობის გაგრძელება.

შებრუნებული მატრიცის მეთოდით დაშიფვრა-გაშიფვრის პროცედურების (იხ. 1.4) პროგრამული რეალიზაცია სრულდება ორ ეტაპად, ორი ქვეპროგრამის თანამიმდევრული შესრულებით, შესაბამისად ღილაკებზე „ტექსტის დაშიფვრა“ და „ ტექსტის გაშიფვრა“ დაწკაპუნებით.

ღილაკზე „ტექსტის დაშიფვრა“ დაწკაპუნების შედეგად (იხ. ნახ.3.14):

ა) გამოითვლება (D) დეტერმინანტი. იმ შემთხვევაში თუ აღმოჩნდება, რომ D=0, პროგრამა შეწყვეტს შესრულებას და გამოიმუშავებს შეტყობინებას „შეცვალეთ საწყისი მატრიცის ელემენტების მნიშვნელობები“. აღნიშნული მითითების შესრულების შემდეგ საჭიროა ღილაკზე „ტექსტის დაშიფვრა“ დაწკაპუნება, შედეგად პროგრამა გააქტიურდება და დაიწყებს ფუნქციონირებას დეტერმინანტის გამოთვლით;

ბ) თუ აღმოჩნდება, რომ D განსხვავდება ნულისაგან , გამოითვლება შებრუნებული და ერთეულოვანი მატრიცები (იხ. ნახ.3.17);



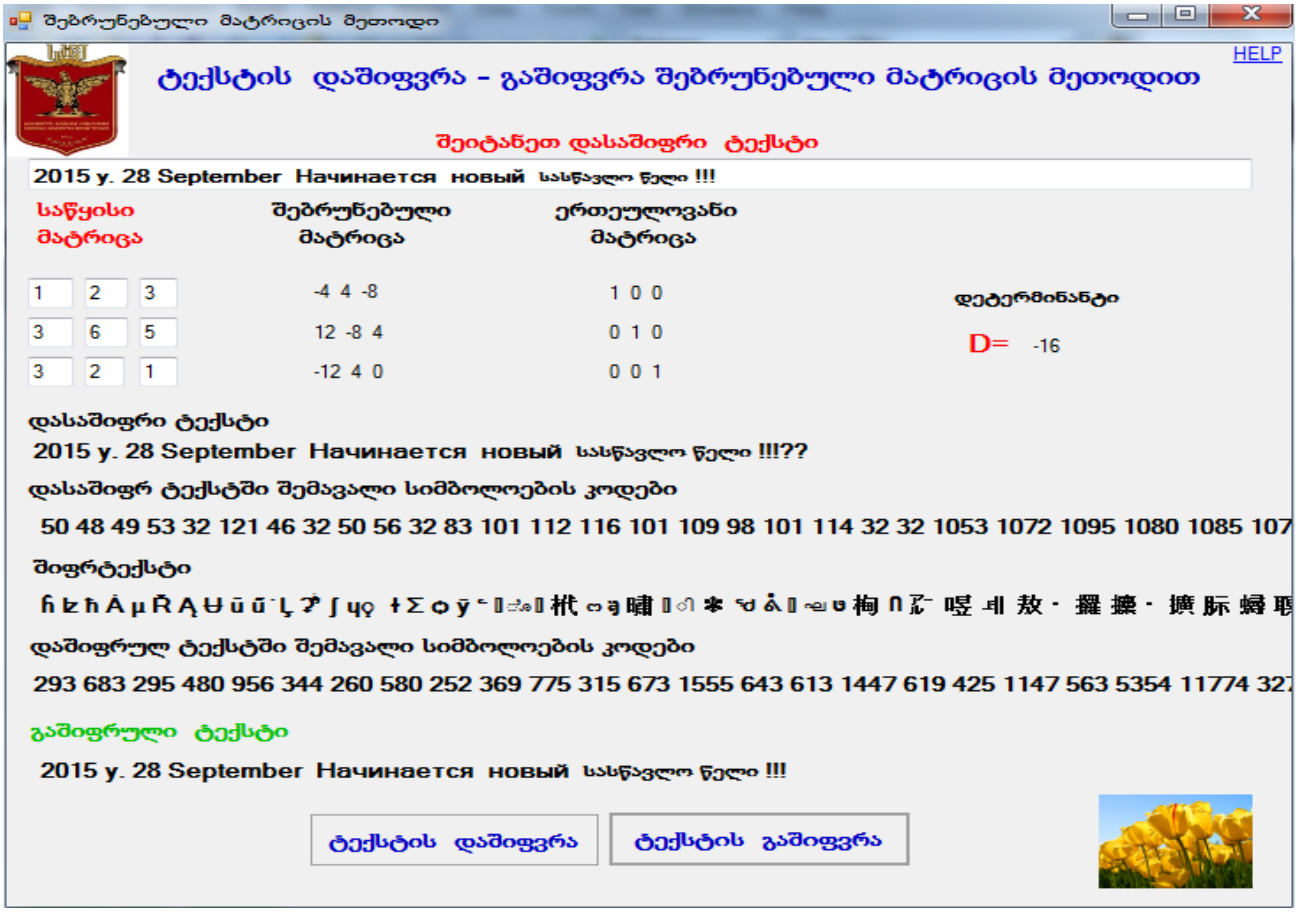
ნახ.3.17. შებრუნებული მატრიცის პროგრამული რეალიზაცია კონკრეტულ მაგალითზე

გ) მიმთითებლების „დასაშიფრი ტექსტი“ და „დასაშიფრ ტექსტში შემავალი სიმბოლოების კოდები“ ქვემოთ გამოყოფილ სტრიქონებში შესაბამისად აისახება DasTI და მასში შემავალი სიმბოლოების კომპიუტერული კოდების მნიშვნელობები;

დ) მიმთითებლების „შიფრ-ტექსტი“ და „დაშიფრულ ტექსტში შემავალი სიმბოლოების კოდები“ ქვემოთ გამოყოფილ სტრიქონებში შესაბამისად აისახება ShifTI-ში შემავალი სიმბოლოები D_i (i=1,2,…,Z) და ამ სიმბოლოების კომპიუტერული კოდების მნიშვნელობები.

მეორე ეტაპზე, ღილაკზე „ტექსტის გაშიფვრა“ დაწკაპუნებით, შესრულდება შიფრ-ტექსტის გაშიფვრა და ეკრანზე აისახება ტექსტი, რომელიც საწყისი დასაშიფრი ტექსტური ინფორმაციის (DasTI) ანალოგიური იქნება.

ტექსტური ინფორმაციის დაშიფვრისა და გაშიფვრის პროგრამული რეალიზაციის შესრულების შედეგები ნაჩვენებია 3.17 და 3.18 ნახაზებზე.



ნახ.3.18. შებრუნებული მატრიცის პროგრამული რეალიზაცია კონკრეტულ მაგალითზე

ლიტერატურა

1. П.В. NewMan and R.L. Pickholtz. Griptograpy in the private sector. IEEE Commen. Mag. Val. 24, 1986y.

2. G.S. Verman. Cipher printing systems for Secret wire and radio telegraphic communications. Inst, Elec, Eng, Vol, 55, 1926y.

3. Шенон К.Э. Теория связи в секретных системах. - В кн: Шенон К. Э. Работы по теории информаций и кибернетике. М., 1963г.

4. Шенон К. Э. Математическая теория связи. - В кн: Шенон К. Э. Работы по теории информаций и кибернетике. М., 1963г.

5. W.Diffie and M.E. Hellman. New directions in Griptography. IEEE, Vol, IT-22, 1986y.

6. R.C. Markle. Secure communication over insequre channels. Comm. ACM, 1978y.

7. Герасименко В.А. Проблемы защиты в системах их обработки. Зарубужная радиоэлектроника, N2, 1989г.

8. А.В. Бабаш. Г.П. Шакин. Криптография. М.: Солон-Пресс, 2007г.

9. Анин Б. Защита компьютерной информации. Москва, 2000г.

10. Молдовян А. А. Молдовян Н. А. Гуц Н.Д. Изотов Б. В. Криптография. Скоростные шифры. Санкт-Петербург, 2002г.

11. Смарт Н. Криптография. М, Технисфера, 2005г.

12. ვ.კუცნავა. გ.კაცაძე. ქ.დიაკონიძე. ინფორმაციის დაცვა. სტუ, 2005წ.

13. რ.მეგრელიშვილი. ინფორმაციის დაცვის სისტემები. 2009წ.

14. გ.კოტრიკაძე. ინფორმაციის დაცვის მოცულობითი მატრიცის მეთოდის დამუშავება და მისი შედარება ასიმეტრიულ მეთოდებთან. //შრომები „მართვის ავტომატიზირებული სისტემები", №1(8), 2010წ. გვ.45-51.

15. Шифр Цезаря, https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%A6%D0%B5%D0%B7%D0%B0%D1%80%D1%8F

16. Шифр Вижинера, http://shifr-online-ru.1gb.ru/shifr-vigenera.htm

17. Шифр Вернама, http://shifr-online-ru.1gb.ru/shifr-vernama.htm

18. ვ.კეკელია. ალგორითმული ალგებრის საშუალებათა გამოყენება მიკროპროგრამირების საკითხებში. ილ.ჭავჭავაძის სახ. თბ. სასწ. უნივერსიტეტი. //სამეცნიერო ძიებანი. ტ.6. თბ. 2010წ.

19. ვ. კეკელია. გ. კოტრიკაძე. კრიპტოგრაფიის სიმეტრიული სისტემის ზოგიერთი მეთოდის რეალიზაციის საკითხების შესახებ. // შრომები „მართვის ავტორიზებული სისტემები“. N 2(20), თბ. საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“. 2015წ.

20. რ. სამხარაძე. V i s u a l C#.N E T. სტუ, თბ. სტუ. 2009წ.