თეიმურაზ შარაშენიძე

ინფორმაციის დაცვა კომპიუტერულ ქსელებში

(სახელმძღვანელო საგანში ISCN08 „ინფორმაციის დაცვა გამოთვლით ქსელებში“)

საქართველოს ტექნიკური უნივერსიტეტი

თბილისი

2016

სარჩევი

შესავალი

§1.1 ინფორმაციის დაცვა , ძირითადი ამოცანები

თავდაპირველად გამოთვლით ქსელები (შემდგომში გქ) გამოიყენებოდა საუნივერსიტეტო მკვლევართათვის (ელექტრონული ფოსტა) და კორპორაციების თანამშრომლების მიერ (ქსელური პრინტერები). შესაბამისად ამ პერიოდისთვის ინფორმაციის დაცვა (იდ) ქსელებში არ იყო პრიორიტეტული ამოცანა. შემდგომში შეუძლებელი გახდა იდ ამოცანის უგულვებელყოფა. მარტო 2015 წ. ინფორმაციის გაჟონვისაგან მიყენებულმა ზარალმა შეადგინა [50] 29 მილიარდი დოლარი. საინტერესო მონაცემები გაავრცელა 2016 წელს Microsoft-მა [1] თავის გამოკვლევაში "2016 Trends in Cybersecurity". ამ გამოკვლევას საფუძვლად დაედო ფირმის მიერ 10 წლის განმავლობაში მთელი მსოფლიოს მაშტაბით 600 მლნ კომპიუტერის ანალიზი. საშიშროებების ჩამონათვალი წამყვან ქვეყნებთან მიბმით გადმოცემულია ცხრ.1.

ცხრ.1

Threat category prevalence worldwide and in the 10 locations with the most computers reporting encounters.

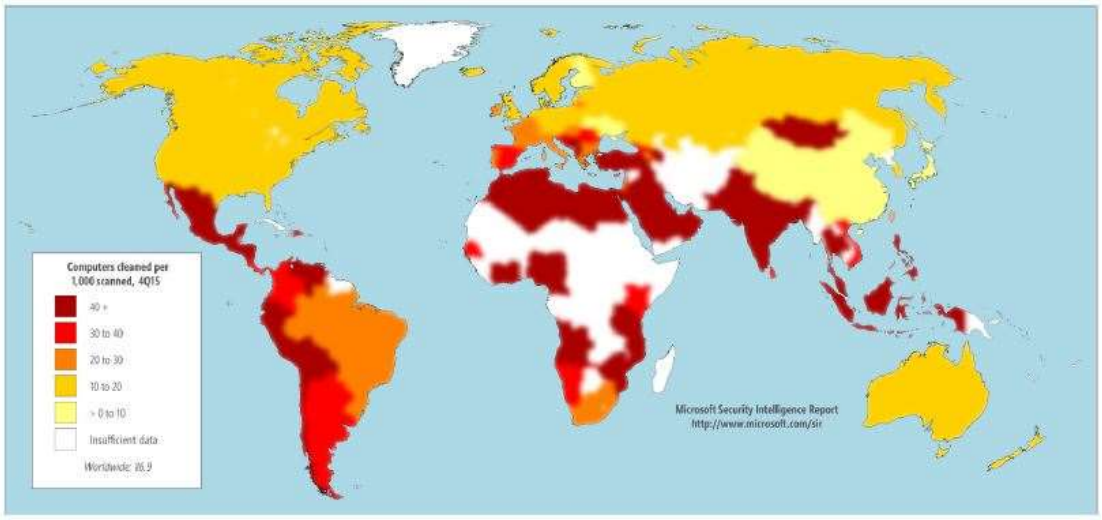
Category	Wordwide	USA	Brazil	China	Russia	France	Germany	UK	Italy	Canada	Japan
Browser Modifiers	7.6%	9.1%	11.8%	0.6%	7.0%	14.3%	8.7%	10.9%	15.3%	11.3%	4.2%
Trojans	7.1%	4.2%	12.7%	10.2%	20.8%	5.7%	4.3%	4.4%	7.0%	5.1%	1.5%
Worms	3.3%	0.6%	8.9%	5.6%	4.6%	1.9%	1.1%	0.8%	3.9%	0.6%	0.7%
Software Bundlers	3.1%	1.7%	1.5%	0.2%	0.5%	2.2%	0.9%	2.3%	2.5%	2.5%	0.5%
Downloaders & Droppers	2.2%	2.3%	6.5%	3.2%	6.6%	2.8%	1.5%	3.2%	3.1%	3.3%	0.4%
Obfuscators & Injectors	1.7%	1.0%	5.3%	5.2%	7.3%	1.9%	1.6%	1.7%	2.8%	1.6%	0.6%
Adware	1.6%	4.5%	7.1%	0.2%	5.2%	7.8%	4.1%	4.7%	7.2%	5.3%	2.0%
Exploits	1.4%	3.4%	2.4%	1.7%	1.3%	2.5%	3.2%	4.4%	4.3%	5.7%	3.2%
Viruses	1.1%	0.4%	2.2%	7.4%	1.5%	0.4%	0.3%	0.3%	0.8%	0.4%	0.2%
Other	0.6%	0.9%	0.3%	1.2%	0.3%	0.5%	0.5%	0.6%	0.7%	1.5%	0.2%
Backdoors	0.5%	0.7%	1.4%	1.8%	2.0%	0.9%	0.6%	0.9%	1.0%	0.7%	0.3%
Randsomware	0.3%	0.6%	0.5%	0.0%	0.6%	0.7%	0.6%	0.4%	1.4%	0.7%	0.4%
Password Stealers & Monitoring Tools	0.2%	0.4%	1.0%	0.5%	0.8%	0.3%	0.4%	0.4%	0.6%	0.6%	0.3%

გამოკვლევაში კლასიფიცირებულია დაახლოებით 6000 საფრთხე და აღნიშნულია, რომ კერძო კომპიუტერები 2-ჯერ მეტი საფრთხის შემცველები არიან, ვიდრე საწარმოო კომპიუტერები.

დკვირვებულ მკითხველს შეიძლება გაუჩნდეს შეკითხვა, თუ რა საშუალებებით და ვისი თანხმობით შეძლო ფირმა Microsoft 600 მლნ. კომპიუტერზე ინფორმაციის შეგროვება მთელი მსოფლიოს მაშტაბით.

საინტერესოა იმავე გამოკვლევის მონაცემები მსოფლიო რეგიონების რეიტინგები საშიშროებების განაწილებების შესახებ ნახ.1.

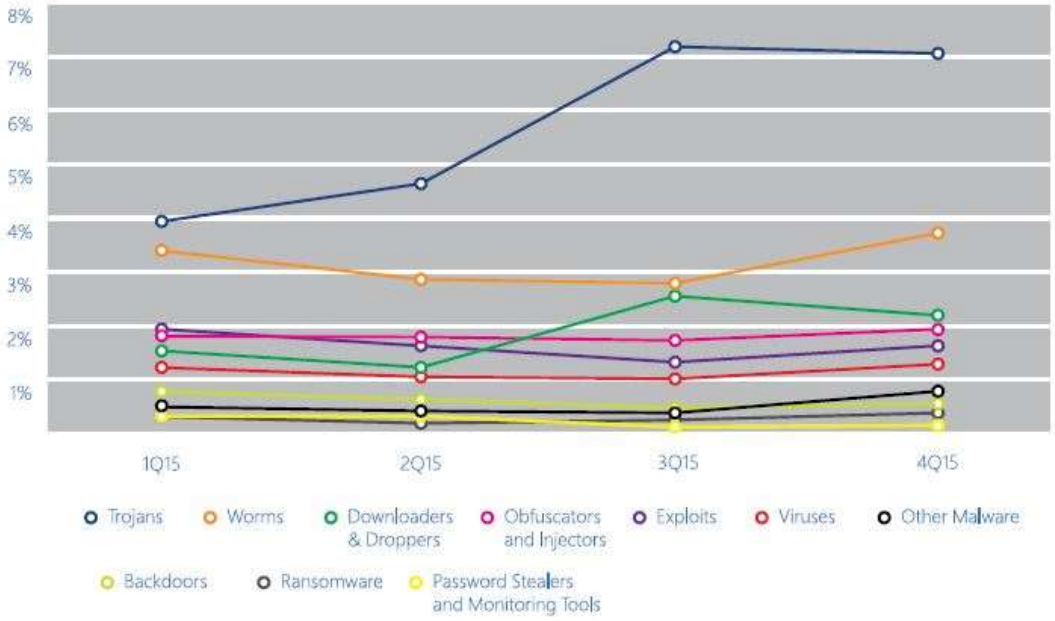
Infection rates by country/region



ნახ. 1

რაგინდ გასაკვირვი არ უნდა იყოს, ამ გამოკვლევით ყველაზე მავნე პროგრამად დადგენილი იქნა Adobe Flash. თვით ვირუსების რეიტინგები, გავრცელების მხრივ, წარმოდგენილია ნახ. 2.

Encounter rates for significant malware categories



ნახ. 2.

იდ ამოცანის გადასაწყვეტად აღარაა საკმარისი ძალიან გათვიცნობიერებული მომხმარებლის არსებობა [43]. დასმული ამოცანის გადასაწყვეტად უნდა შეიქმნას უსაფრთხოების სამსახურები (უს) [2, 3, 38, 39, 40], რომელთა ძირითადი ამოცანებია:

- არაუფლებამოსილმა პიროვნებებმა არ შეძლონ არა მარტო გაეცნონ ინფორმაციას, არამედ მოახდინონ მისი მოდიფიკაცია;
- არაუფლებამოსილმა პიროვნებებმა ვერ მოახდინონ დაცილებულ რესურსებთან წვდომა:
- დაადგინონ ქსელში ინფორმაციის გამგზავნი ავტორის უტყუარობა;
-არ დაუშვან ქსელში გაგზავნილი ინფორმაციის „დაჭერა“ მისი შემდგომი ფალსიფიკაციისათვის;
- დიდი ალბათობით შეძლონ ასეთი პიროვნების არამარტო აღმოჩენა, არამედ მათი ბრალეულობის დამტკიცება. მიღებული იყო ასეთი დამნაშავეების [4, 5] შემდეგი ჩამონათვალი ცხრ.2.

ცხრ.2

	დამნაშავე	მიზანი
1	სტუდენტი	გაეცნოს მისთვის არდაშვებულ ინფორმაციას და მოდიფიცირება გაუკეთოს მას
2	ჰაკერი	მიიღოს ინფორმაცია იდ სისტემის შესახებ და მოიპაროს მონაცემები
3	სავაჭრო აგენტი	წარმოჩინდეს ცნობილი ფირმების (ქვეყნების) წარმომადგენლად, ვიდრე ის არის სინამდვილეში
4	ბიზნესმენი	კონკურენტების სტრატეგიული გეგმების გაცნობა
5	განთავისუფლებული თანამშრომელი	ზიანი მიაყენოს მის ყოფილ ორგანიზაციას
6	ბუღალტერი	მიითვისოს ორგანიზაციის ფული
7	ბირჟის ბროკერი	არ შეასრულოს კლიენტის მიერ ელექტრონული სახით გაცემული დავალება
8	აფერისტი	საკრედიტო რეკვიზიტების მოპარვა მათი შემდგომო არასანქცირებული გამოყენებისთვის
9	ჯაშუში	მოწინააღმდეგის სამხედრო-სამრეწველო პოტენციალიზე მონაცემთა გაცნობა

10	ტერორისტი	შეიცავს ყველა ზემოთ დასახელებულს + მოიპოვოს მასობრივი მოსპობის იარაღებზე ინფორმაცია

პირველი მიახლოებით გქ-ში იდ ამოცანა ოთხ კომპონენტიანია [6,7]:
- საიდულოების დაცვა (კონფიდენციალობა);
- აუტენფიკაცია;
- აღებული ვალდებულებების უპირობო დადასტურება;
- ინფორმაციის სისრულის უზრუნველყოფა.

ცნობილია ინფორმაციის გაჟონვის შესაძლო არხები [4,8,42]:
- მოსმენის სპეციალური მოწყობილობები;
- დისტანციური ფოტოგრაფირება;
- ელექტრომაგნიტური გამოსხივების დაჭერა;
- საწარმოო ნარჩენებში ინფორმაციის მატარებლების დატაცება;
- სხვა მომხმარებლების მასივებიდან მონაცემების წაკითხვა;
- მონაცემთა მატარებლების კლონირება;
- ტერმინალების არასანქცირებული გამოყენება;
- დარეგისტრირებული მომხმარებლების პაროლებისა და შეღწევის რეკვიზიტების მოპარვა;
- პროგრამული ხაფანგების გამოყენება;
- დაცული ინფორმაციის მიღება მრავალჯერადად ნებადართული მოთხოვნების გამოყენებით;
- პროგრამების ენებისა და ოპერაციული სისტემის ნაკლოვანებების გამოყენება;
- პროგრამებში წინასწარ ჩანერგილი „მავნე" პროგრამები;
- არასანქცირებული მიერთება აპარატურასთან ან კავშირის ხაზებთან;
- დაცვის მექანიზმების მწყობრიდან გამოყვანა.

რა თქმა უნდა, საზოგადოების განვითარების ეტაპებზე გამოიყენებოდა სხვა და სხვა დაცვის მექანიზმი: ფელდეგერული კავშირი, რკინის სეიფები, კაბელების იზოლირება, სიგნალიზაცია და სხვა. გაცილებით ადრე საბერძნეთში, კი ისეთი ეგზოტიკური მეთოდი, როგორიც არის სტეგანოგრაფია. ზოგადად, სანამ საკითხი არსებითად არ შეისწავლება, უნდა აღინიშნოს, რომ არ არსებობს ერთიანი მიდგომა (მოთხოვნა) გქ-ის უსაფრთხოების ორგანიზაციაში, ანუ ის დიფერენცირებულია ქსელის დონეების მიხედვით [5, 9]. ასე მაგალითად, ***ფიზიკურ დონეზე*** დაცვა შეიძლება განხორციელდეს საკომუნიკაციო სადენების (კაბელები) წნევის ქვეშ მყოფი არხებში განთავსებით, იქ შეღწევის მცდელობისას ხდება არხში წნევის ვარდნა, რაც იწვევს სიგნალიზაციის ამუშავებას. მონაცემთა **პაკეტების გადაცემის** ქსელურ დონეზე შეიძლება გამოყენებული იქნეს მონაცემთა დაშიფვრა, რომელიც მხოლოდ მიმღებ და გადამცემ მხარეს აძლევს უფლებას ოპერირება გაუკეთონ მონაცემებს. ძალიან ბევრი დადებითი მხარის მიუხედავად ასეთ მიდგომას აქვს საკმაო ნაკლიც, როცა დაშიფრული ინფორმაცია გადის მარშრუტიზატორებში, სადაც ხდება მათი გაშიფვრა-დაშიფვრა. ***ქსელურ დონეზე*** შეიძლება გამოყენებული იქნენ ე.წ.

ქსელთაშორისო ეკრანები ბრანდმაუერები. ისინი უკუაგდებენ საეჭვო პაკეტებს და ახორციელებენ ე.წ. IP დაცვას.

სატრანსპორტო დონეზე შესაძლებელია კავშირის მთლიანი დაშიფვრა კავშირის ერთი ბოლოდან მიმღებამდე. მაქსიმალურ დაცვას უზრუნველყოფს მხოლოდ ასეთი გამჭოლი შიფრაცია.

აუტენფიკაცია და ჭეშმარიტობის (უტყუარობის) დადგენა ხორციელდება ***გამოყენებით დონეზე.***

მიუხედავად იმისა, რომ მიღწეულია უდიდესი წარმატებები გქ-ში ინფორმაციის პროგრამულად და აპარატურულად დაცვაში, უნდა გვახსოვდეს, რომ გადამწყვეტი როლი ენიჭება ადამიანის ფაქტორს. კრიპტოგრაფია და ტექნიკური საშუალებები ვერ დაიცავს ქსელს პერსონალის, თანამშრომლების უპასუხისმგებლო მოქმედებებისაგან, სამსახურეობრივი გადაცდომებისაგან და ა.შ.

აღსანიშნავია, რომ გქ-ში უსაფრთხოებაზე თავდასხმა შესაძლებელია არა მარტო ლოკალურ სისტემაში (სისტემიდან), არამედ ე.წ. დაცილებული ადგილიდან (სისტემიდან) [10], რაც განპირობებულია იმით, რომ გქ ახასიათებს რესურსებისა და ინფორმაციის გადანაწილება. ასეთ დროს შეტევა ხორციელდება არა მარტო კონკრეტულ კომპიუტერზე არამედ, მთელ გქ-ში მოძრავ ინფორმაციაზე (მონაცემებზე). ზოგადად ცხრ.3-ში გადმოცემულია გქ-ში ინფორმაციული უსაფრთხოების რეალიზაციის გზები.

ცხრ.3

	ზემოქმედების ობიექტები	ინფორმაციის კონფიდენციალობის დარღვევა	ინფორმაციის მთლიანობის დარღვევა	სისტემის მუშაობის უნარიანობის დარღვევა
1	აპარატურული საშუალებები	არასანქცირებული შეღწევა-მიერთება, რესურსების გამოყენება, ინფორმაციის მატარებლების დატაცება	არასანქცირებული შეღწევა-მიერთება, რესურსების გამოყენება, მოდიფიცირება, ცვლილება	არასანქცირებული შეღწევა-ცვლილება, მწყობრიდან გამოყვანა, განადგურება
2	პროგრამული უზრუნველ-ყოფა	არასანქცირებული შეღწევა-კოპირება, დატაცება, დაჭერა	არასანქცირებული შეღწევა, „ტროას“ ცხენი, “ჭიაყელების“,	არასანქცირებული შეღწევა-დამახიჯება,

			ვირუსების ჩანერგვა	„გადაგდება“, შეცვლა
3	მონაცემები	არასანქცირებული შეღწევა-კოპირება, დატაცება, დაჭერა	არასანქცირებული შეღწევა-დამახინჯება, მოდიფიკაცია	არასანქცირებული შეღწევა-დამახიჯება, „გადაგდება“, შეცვლა
4	პერსონალი	ინფორმაციის გაცემა მონაცემთა დაცვაზე	პერსონალის მოსყიდვა, მათი გადაბირება, „მასკარადი“	სამუშაო ადგილიდან წასვლა, ფიზიკუ-რი განადგურება

§1.2 ინფორმაციის დაცვის საშუალებები

ინფორმაციის დაცვის უზრუნველსაყოფად გამოიყენება შემდეგი ძირითადი საშუალებები და მეთოდები:

ტექნიკური საშუალებები: ელექტრული, ელექტრომექანიკური, ელექტრონული და სხვა [41]. მათ ახასიათებს საიმედოობა, სუბიექტურ ფაქტორებისგან დამოუკიდებლობა, მოდიფიცირების დიდი შესაძლებლობები. უარყოფითია - ღირებულება, დიდი მასა და გაბარიტები. ტექნიკური საშუალებები თავის მხრივ შედგება აპარატურულ და ფიზიკურ საშუალებებისაგან. აპარატურული უშუალოდ ჩაშენებულია მოწყობილობაში და უერთდება ქსელურ მოწყობილობებს სტანდარტული ინტერფეისით (ლუწობაზე კონტროლის

მოწყობილობა, მახსოვრობის ველების დამცავი მოწყობილობები, სპეციალური რეგისტრები და სხვა). ფიზიკური საშუალებები წარმოდგენილია დამოუკიდებელი მოწყობილობების სახით (დაცვის და თვალთვალის მექანიკური საშუალებები, საკეტები კარზე და ცხაურები ფანჯრებზე);

პროგრამული საშუალებები. ეს არის სპეციალურად ინფორმაციის დასაცავად შექმნილი პროგრამები (მომხმარებელთა იდენტიფიკაციის, შეღწევის კონტროლის, შიფრაციის, ნარჩენი ინფორმაციის წამშლელი ტესტები და სხვ.). დადებითი მხარეა დიდი უნივერსალობა, მოქნილობა, საიმედოება. უარყოფითი მხარეა სისტემური რესურსების ნაწილის გამოყენება, ზოგჯერ ქსელის ფუნქციონირების შეზღუდვა;

შერეული აპარატურულ-პროგრამული საშუალებები;

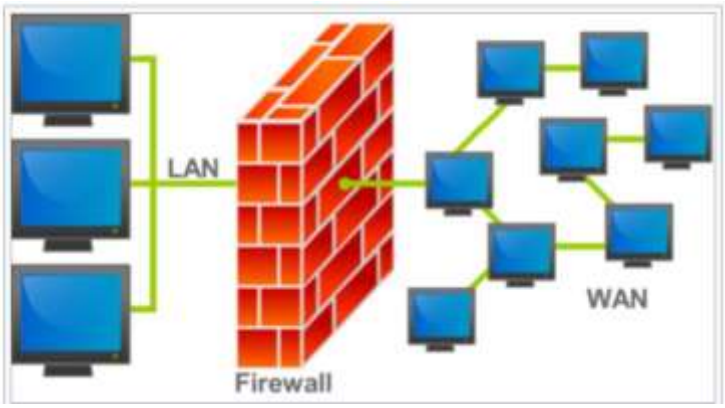
ორგანიზაციული ღონისძიებები. იგი მოიცავს ისეთ ორგანიზაციულ-ტექნიკურ ღონისძიებებს, როგორიცაა კომპიუტერებიანი შენობის მომზადება, კაბელური სისტემების ჩაწყობა-გაყვანა, საკანონმდებლო ბაზის შექმნა, შიგა უწყებრივი დებულებებისა და ინსტრუქციების შექმნა და სხვა. დადებითი მხარეა უნივერსალობა, ხოლო უარყოფითი - სუბიექტური მხარის დიდი გავლენა.

ინფორმაციის დაცვის ორგანიზაციისას არ შეიძლება უგულებელყოთ არცერთი მეთოდი და საშუალება. მაგალითად, იმედი გვქონდეს მხოლოდ აპარატურულ ან პროგრამულ საშუალებებზე და არ ვიცავდეთ ორგანიზაციულს, ან პირიქით. ამოცანა კომპლექსურად უნდა იქნეს გადაწყვეტილი.

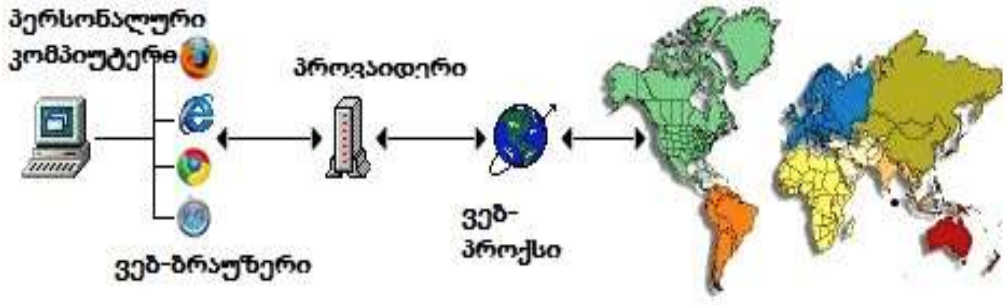
ზემოთ ჩამოთვლილი მეთოდებისაგან შედარებით უფრო გავრცელებულია პროგრამული. იგი შეიცავს შემდეგს:

- მონაცემთა არქივიზაციას;
- ანტივირუსულ პროგრამებს;
- კრიპტოგრაფიულ საშუალებებს;
- იდენტიფიკაციისა და აუტენფიკაციის საშუალებებს;
- შეღწევის მართვას;
- აუდიტს და ოქმებს;
- Firewalls-ს (ბრანდმაუერი გერმანულად ნახ.3.) პეციალური შუალედური სერვერებისათვის, რომლებიც ფილტრავენ ინფორმაციულ ნაკადებს ლოკალურ და გლობალურ ქსელებს შორის. აქ აღსანიშნავია ე.წ. მასკარადის მეთოდი, რომლითაც უხილავი ხდება ლოკალური ქსელი;
- Proxy-sever. აქ მთლიანად იზღუდება ქსელურ-სატრანსპორტო დონეზე ინფორმაციული ნაკადების ურთიერთგაცვლა ლოკალურ და გლობალურ ქსელებს შორის. აქ მარშრუტიზაციის ამოცანა არ დგას. ახორციელებს მონაცემთა ქეშირებას, ახორციელებს ინტერნეტიდან მიღებული მონაცემთა შეკუმშვას, იცავს ლოკალურ ქსელს არასანქცირებული შეღწევებისგან, ზღუდავს ლოკალური ქსელიდან (საჭიროების შემთხვევაში) გარე რესურსებთან წვდომას, ადგენს ტრაფიკებს, ფილტრავს რეკლამებს და ვირუსებს, ჰქმნის ანონიმურ რეჟიმს. ცნობილია მათი შემდეგი სახეობა: „გამჭვირვალე“, „უკუ“ პროქსი, ვებ-პროქსი ნახ.4. ამ უკანასკნელს დამატებით შეუძლია გვერდი აუაროს ლგქ ადმინისტრატორის

შეზღუდვებს, რეალური მისამართების დამალვა და ანონიმური შეღწევა ვებ-საიტებთან და სხვა.



ნახ.3.



ნახ.4.

§ 1.3 მავნე პროგრამები, როგორც ყველაზე აქტიური კომპონენტი ინფორმაციის დაცვის ხელისშემშლელ ფაქტორებს შორის.

ზოგადად გქ-ში ინფორმაციის დაცვის ორგანიზება ძალიან მრავალფეროვანი საკითხია, რომელიც პირდაპირი გაგებით შეიძლება არ შეიცავდეს ისეთ საკითხებს როგორებიცაა ბიომეტრიული დაცვა, პაროლების დაცვა, ანტივირუსულ დაცვას და ა.შ. პირველად ვირუსები

Virus 1.2.3, Elk Cloner დაფიქსირებული იქნენ 1981 წ. Apple II კომპიუტერებზე. მხოლოდ 1984 წელს შეიქმნა პირველი ანტივირუსული პროგრამა CHK4BOMB, BONBSQAD ***ენდი ჰოპკინსის*** მიერ. პირველი კომერციული ანტივირუსული პროგრამა შექმნა Symantec Norton AntiVirus 1990 წ. ცნება „კომპუტერული „ვირუსი“ პირველად შემოთავაზებული იყო სამხრეთ კალიფორნიის თანამშრომლის **ფრედ კოენის** მიერ. იგი ასე განისაზღვრებოდა:

- ***„კომპუტერული ვირუსი”- ეს არის პროგრამა, რომელსაც შეუძლია დააინფექციროს (მოდიფიცირება) სხვა პროგრამები მათში საკუთარი შეცვლილი კოპიოების ჩანერგვით, ისე, რომ მათ შესწევთ უნარი შემდგომი გამრავლებისა“.***

„კომპუტერული ვირუსის“ ცნებაში (განმარტებაში) არსებითია მისი უნარი თვითგამრავლებისა და მის გამოთვლითი პროცესის მოდიფიცირების შესაძლებლობა. მისი შემქნელები ცდილობენ შექმნან ისეთი „კომპუტერული ვირუსი“, რომლის აღმოჩენა ძნელი იქნება. ამისთვის „კომპუტერული ვირუსი“ საწყის პერიოდში „მთვლემარე“ მდგომარეობაში იმყოფება და გააქტიურდება რაღაც პირობის შესრულებისას, მაგალითად, დროის დადგომისას. მათ შეუძლიათ შეცვალონ სისტემური ფაილები. ისინი იკავებენ დისკზე ოპერაციული სისტემისთვის განკუთვნილ არეს, კონკრეტულად ჩამტვირთავ სექტორს და აქტიურდება ყოველი ჩატვირვისას. ზოგი სახის „კომპუტერული ვირუსი“ ინფიცირებას ახდენს ე.წ. შემსრულებელი ფაილებისა.

ფრედ კოენის მიერ 1989 წ დამტკიცებული იქნა, რომ არ არსებობს რაიმე ერთი უნივერსალური ალგორითმი, რომელიც შეძლებს აღმოაჩინოს ყველა სახის ვირუსი.

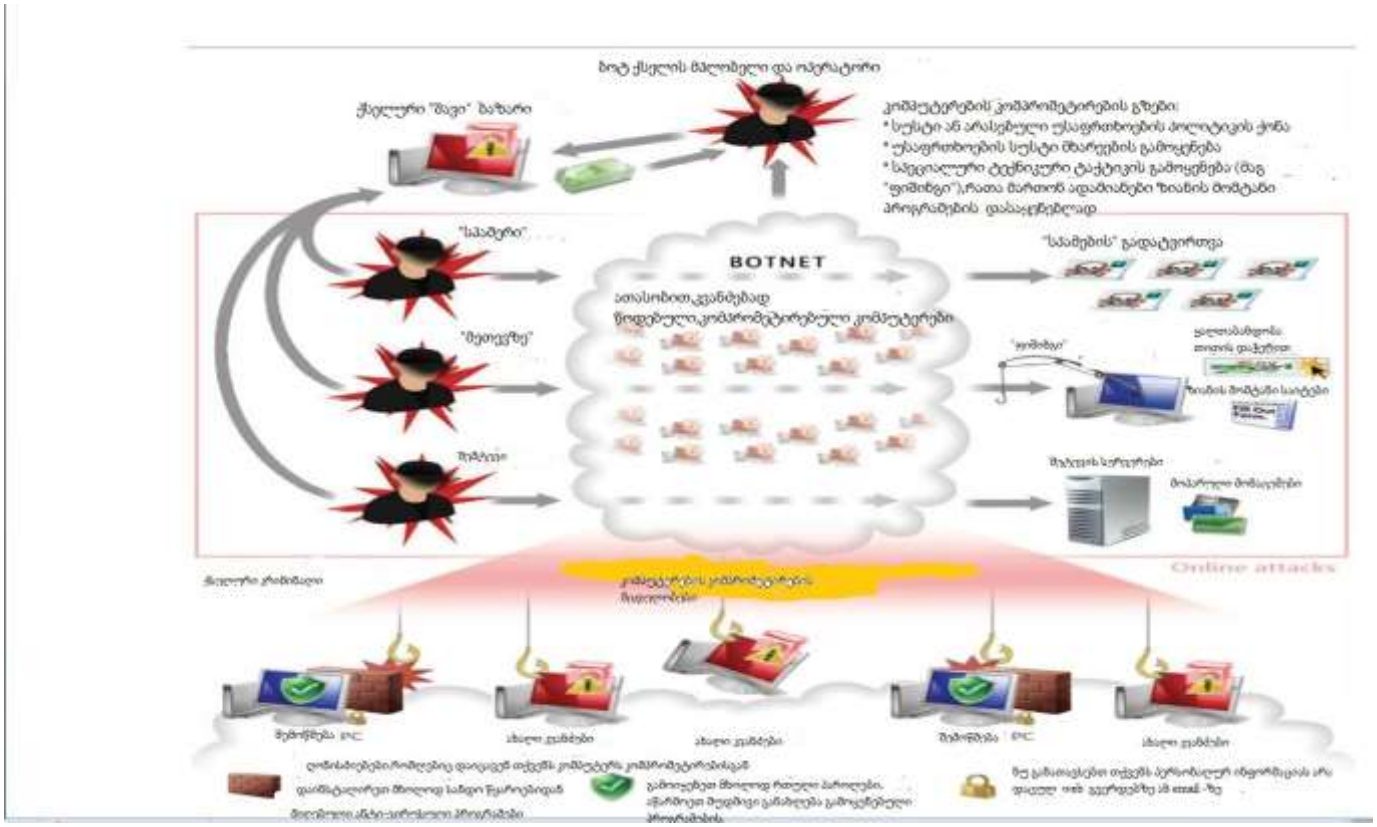
კიბერუსაფრთხოების საკითხებში 2011 წ. თბილისის საერთაშორისო კონფერენციაზე აღინიშნა, რომ დაცვის საკითხების უგუველყოფამ გამოიწვია ირანის სამხედრო მრეწველობის გქ მწყობრიდან გამოსვლა და ასევე სამხრეთ კორეის ეროვნული ბანკის ფუნქციონირების პარალიზება დიდი ხნით.

კომპანიებს, რომლებიც ქმნიან ანტივირუსულ პროგრამებს აქვთ განსხვავებული შეხედულებები მავნე პროგრამების კლასიფიცირებაზე. მაგალითად, შეიძლება მოვიყვანოთ „კასპერსკის ლაბარატორიის“ მავნე პროგრამების ზემოქმედების (გამოვლინების) კლასიფიკატორი:

- კომპიუტერის მუშაობის შეფერხება;
- სხვა მავნე პროგრამების ინსტალაცია, ქსელიდან ჩამტვირთავები (downloader) და ჩაშენებული გამხსნელები (dropper);
- მონაცემების მოპარვა, თაღლითობა, ცრუ ანტივირუსი (scarewaqre), ფულადი სახსრების გამოძალვა (Ransomware) და ა.შ;
- არასამართებლივი წვდომა backdoor, Bitcoin და სხვა;
- არასასურველი პროგრამები, რომლებიც ჭეშმარიტად მავნე პროგრამები არ არიან : რეკლამა, ინფორმაციის უნებართვოდ გამრცელებლები, „მოწამლული დოკუმენტები“, უტილიტები, რომლებიც ხელს უწყობენ მავნე პროგრამების ფუნქციონირებას, IRC-კლიენტი, პროგრამული მარშრუტიზატორები (not-a-virus: NetTool. Win32.Transmit.a — Securelist), კლავიატურის (not-a-virus:Monitor.Win32. KeyPress Hooker — Securelist) და სხვა.

მიღებულია მათი შემდეგი გრადაციები [34] :

- ***ფაილის ვირუსები***. ამ სახის ვირუსების კოპიების გავრცელებისას აუცილლებლად ცვლიან შესრულებად ფაილებს და თვით ფაილები მთლიანად ან ნაწილობრივ კარგავენ მუშაობის უნარს;
- ***ჩამტვირთავი ვირუსები***. ამ სახის ვირუსები ჩაიტვირთებიან მყარი ან დრეკად დისკების საწყის (პირველ) სექტორში და სრულდება კომპუტერის ჩატვირთვისას;
- ***სტელს ვირუსები***. ისინი მთლიანად ან ნაწილობრივ მალულნია და თავს ამჟღავნებს მხოლოდ ოპერაციულ სისტემასთან მიმართვის დროს ინფორმაციის ჩაწერა-წაკითხვისას;
- ***სკრიპტული ვირუსები***. ისინი მოქმედებენ როცა გამოყენებულია სკრიპტული ენა (Javascript, VBScript), ისინი დამოუკიდებლივ შედის ამ ენებზე დაწერილ სკრიპტებში. შედეგად ისინი მთლიანად ცვლიან პროგრამის საწყის ტექსტს ან მის კომპონენტებს OBJ-, LIB, DCU ფაილებს და აგრეთვე VCL და ActiveX კომპონენტებს;
- ***კეილოგერები***. ისინი არიან ვირუსების პროგრამების შემადგენლობაში და წარმართავს კლავიშების ფიქსაციის დაჭერას;
- ***ქსელური ვირუსები***. ისინი ვრცელდება ქსელური ოქმების და სამსახურების საშუალებით. როგორებიცაა ელექტრონული ფოსტის დაგზავნა, FTP (File Trabsfer Protocol) საშუალებით ფაილებთან შეღწევა, ფაილებთან შეღწევა ლოკალური ქსელის სამსახურების საშუალებებით. ცნობილი ვირუსებია "Macro.Word.ShareFun" და "Win.Homer". აქ ცალკე აღნიშვნის საქმეა ე.წ. „ბოტ-ქსელების“ შემქნელი მავნე პროგრამები. ამ შემთხვევაში დამნაშავე ქმნის მავნე პროგრამას, რომელიც გადააქცევს თქვენს კომპიუტერს ბოტად (იგივეა, რაც ზომბი, რობოტი), რაც იმას ნიშნავს, რომ დამნაშავე ქსელის (ინტერნეტი) საშუალებით იყენებს თქვენი კომპიუტერის შესაძლებლობას თქვენს უკითხავად. ე.ი. დამნაშავე ქმნის ბოტქსელს (ვირტუალურად) და იყენებს მას (ბოტქსელში გაერთიანებული კომპუტერების სიმძლავრეებს) თავისი მიზნებისათვის ნახ. 1.2.



ნახ. 1.2.

იგი იქმნება როდესაც ხდება კომპიუტერში შეჭრა მალვეარისაგან, ცნობილს როგორც მავნე პროგრამა. ბოტნეტი უმეტესად იმართება **IRC-დან** (**Internet Relay Chat**), მაგრამ მისი მართვა შესაძლებელია ვებ გვერდიდანაც.

ეს შეიძლება მოხდეს იმის შედეგად, როცა მომხმარებელი ეწვევა არასანდო საიტს და გადმოტვირთავს ამა თუ იმ ინფორმაციას. აგრეთვე შესაძლებელია საზიანო ვირუსი მოვიდეს მიმაგრებული ფაილის სახითაც. სერვერი ცნობილია როგორც (**C&C**) სერვერი. ბოტი ტიპურად გაიშვება მალულად და იყენებს ფარულ არხს (მაგ.**RFC 1459 IRC**), ტვიტერს ან მესიჯს, იმისათვის, რომ დაამყაროს კომუნიკაცია C&C სერვერთან.

მაგალითი გვიჩვენებს, როგორ იქმნება ბოტნეტი და როგორ გამოიყენება იგი სპამ მეილის გასაგზავნად:

1. ბოტნეტის ოპერატორი აგზავნის ვირუსს ან **Worm** და აინფიცირებს ჩვეულებრივი მომხმარებლების კომპიუტერს;
2. ბოტი დაინფიცირებულ კომპიუტერზე შედის კონკრეტულ C&C სერვერზე;
3. სპამერი იძენს ბოტნეტის სერვისებს ოპერატორისაგან ;
4. სპამერი აწვდის შეტყობინებას ოპერატორს, რომელიც სპამ შეტყობინების გაგზავნის განკარგულებას გასცემს.

ბოტნეტი გამოიყენება სხვადასხვა მიზნებისათვის. მაგ: პაროლების მოსაპოვებლად, საკრედიტო ბარათების ნომრების ხელში ჩასაგდებად, აპლიკაციების მოსაპარად და სხვა საზიანო მიზნების მისაღწევად. კიბერდამნაშავეები ბოტნეტს იყენებენ მრავალი

კრიმინალური საქმიანობისათვის, დაწყებული სპამის გაგზავნით და დამთავრებული სახელმწიფო ქსელებზე კიბერშეტევებით ნახ. 1.3.

ბოტ-ქსელის ძირითადი მოქმედებებია:

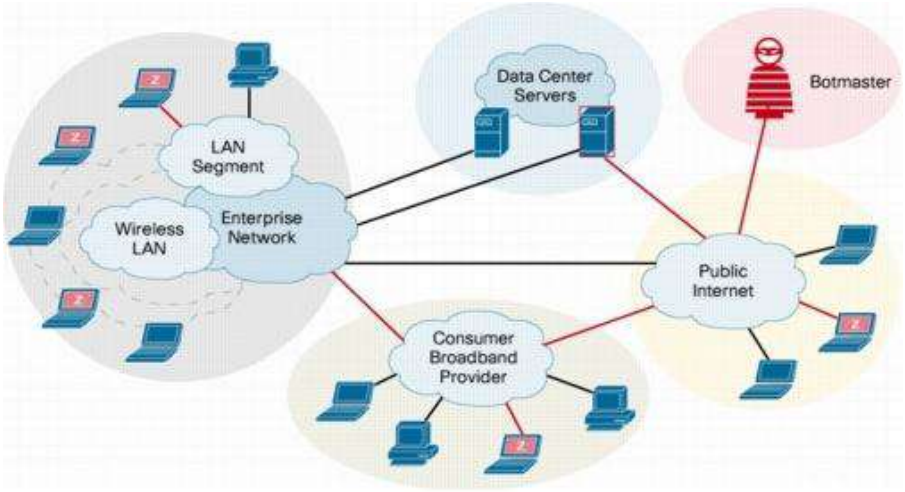
1-სპამის დაგზავნა;

2-კიბერშანტაჟი **DDoS** კიბერშეტევებისთვისაც (**Distributed Denial of Service** - სერვისის შეჩერება);

3-ინტერნეტში ანონიმური წვდომა - ზომბირებული კომპიუტერის მეშვეობით დამნაშავეს შეუძლია განახორციელოს კიბერდანაშაული, მაგალითად, გატეხოს სხვისი ვებ-გვედი ან გადარიცხოს მოპარული ფულადი სახსრები და ამ დროს დარჩეს ანონიმური, რადგან ამ დროს ზომბი კომპიუტერი გამოიყენება როგორც ე.წ. პროქსი სერვერი(**Proxy**) დამნაშავის რეალური მისამართის დასამალად;

4-ბოტნეტის გაყიდვა ან გაქირავება;

5-კონფიდენციალური და პირადი ინფორმაციის მოპარვა.



ნახ. 1.3

ბოტების გავრცელება ხდება ავტომატურად. ცნობილია 2 გზა:

1. სპამ წერილის მეშვეობით, რომელიც შეიცავს სახიფათო ვებ-გვერდის ბმულს ან წერილს მიმაგრებული აქვს ფაილი, რომელიც შეიცავს მავნე პროგრამას;

2. ინფიცირებულ სახიფათო ვებ-გვერდზე შესვლისას, სადაც განთავსებულია ე.წ. i-frame, მალულად ხდება კომპიუტერის დავირუსება მავნე ჩამტვირთავი კოდით, რომელიც შემდგომ ინტერნეტიდან ავტომატურად იწერს სხვადასხვა მავნე პროგრამებს და უკავშირდება ბოტნეტის მართვის ცენტრს.

- ***დოკუმენტების ვირუსები*** *(მაკრო ვირუსები)*. ახდენს საოფისე პროგრამების (Microsoft Office, Open Office, და სხვ.) ინფიცირებას ამ სისტემებში ე.წ. მაკროსების გამოყენებით. მაკროსი ეს არის წინასწარ განსაზღვრული მოქმედებების ნაკრები (მიკროპროგრამა),

რომელიც ჩაშენებულია დოკუმენტში და გამოიძახება დოკუმენტის მოდიფიცირებისას ან სხვა მოქმედებებისათვის.

ამ საშიშროებებიდან განსაკუთრებით გამოსაყოფია ე.წ. „დავირუსების“ საკითხი, რადგან მათთან ყოველდღიური შეხება აქვთ გქ-ის მომხმარებლებს. ვეცადოთ დავაზუსტოთ მათი კლასიფიკაცია.

„ტროას ცხენი“ (Trojan). იგი პირველად შექმნა ამერიკელმა ჰაკერმა (შემდგომში იგი გახდა ამერიკის ნაციონალური უსაფრთხოების სააგენტოს თანამშრომელი) **დანი ედვარდსმა.** სახელი განსაზღვრავს იმ ფუნქციას, რაც ასახულია ძველ ბერძნულ მითოლოგიაში ტროას ომის შესახებ. იგი განთავსებელია რომელიმე სასარგებლო პროგრამაში, რომლის რეკლამირებაც განხორციელებულია და მომხმარებლის მიერ შეძენილია და დაინსტალირებულია. ასეთი პროგრამა ძირითადი სასარგებლო პროგრამასთან ერთად შეიცავს ზიანის მომტან ბრძანებათა ნაკრებს, რომელიც შესრულებას იწყებს რაიმე პირობის შესრულებისას. ეს შეიძლება იყოს თარიღი, შიგა რაიმე მოქმედების დასრულება, წყვეტა და ა.შ.

მისი მავნებლობა მჟღავნდება :

- გასანადგურებელი ობიექტების ინფორმაციის განადგურებით. განადგურების სახე განსაზღვრულია „ტროას ცხენის“ შემქნელის მიერ;
- ინფორმაციის დაჭერა და გადაცემა. მაგალითად, კლავიატურაზე აკრეფილი პაროლები ან სხვა ინფორმაცია;
- მას შეუძლია შეცვალოს დაცვის პროგრამები იმდაგვარად, რომ მათი გატეხა შეძლოს ამ პროგრამის ავტორმა.

„ტროას ცხენი“ გქ-ის ყველაზე საშიში პროგრამაა. (ერთ-ერთი მაგალითია, ***დიმიტრი უვაროვის*** მიერ შექმნილი პროგრამა **Penetrator,** რომლის აქტივირება ხდება 1 იანვრის ღამეს და აზიანებს *avi, .doc, .jpg, .jpeg, .mp3, .mpeg, .mpg, .pdf, .ppt, .rar, .vob, .wma, .wmv, .xls, .zip* ფაილებს).

„ჭიაყელა“ (Worm) (პროგრამა რეპლიკატორი). პირველად ეს ტერმინი ნასესხები იყო ***ჯონ ბრუნერის*** სამეცნიერო-ფანტასტიკური ნაწარმოებიდან. სხვებისგან განსხვავებით, თვით ეს სახე არ შეიცავს რაიმე განსაკუთრებული სახის ზიანის მომტან პროგრამას, მისი დანიშნულებაა „დაანაგვიანოს“ მახსოვრობა (დისკები), გაზარდოს იქ ფუჭი ინფორმაცია და ამით შეაფერხოს ოპერაციული სისტემის მუშაობა. ცნობილია **World Wide Web Worm** ***მორისის*** **„ჭიაყელა“,** რომელიც 1988 წ გაავრცელა ***რობერტ მორის უმცროსმა*** და რომელმაც მხოლოდ 6 სთ დააზიანა 6000 კომპიუტერი ამერიკის მასშტაბით.

„ზომბი“ (Zombie). ეს პროგრამა ვირუსია, რომელიც თუ შეაღწევს ინტერნეტში მიერთებულ კომპიუტერში, იმართება ბოროტმოქმედის მიერ გარედან. ამ დროს შეტევის ობიექტები ხდება სხვა კომპიუტერები, ხდება უზარმაზარი ფუჭი (მავნე, არასასურველი) ინფორმაციების და აგრეთვე მავნე პროგრამებისა და ვირუსების დაგზავნა ელექტრონული ფოსტით.

სპამი (Spam, BulkანJunk) არის ელექტრონული წერილის ტიპი, რომელიც იგზავნება პიროვნების ან კომპანიის მიერ, მიმღების დაუკითხავად და სურვილის გარეშე. მსგავსი

წერილები უმეტეს წილად სარეკლამო ხასიათისაა. პიროვნებას, რომელიც მსგავს წერილებს გზავნის ეწოდება სპამერი. მათი ძირითადი მიზანია თავიანთი პროდუქტის პოპულარიზაცია. სპამერისათვის ყველაზე ძნელია იმ იმეილების ხელში ჩაგდება, რომელთაც უნდა გაუგზავნოს ესა თუ ის იმეილი. რადგან აქ ლაპარაკია არა ერთ და ორ, არამედ ათასობით იმეილზე. ერთ-ერთი გზაა ასეთი იმეილების სხვა პიროვნებისაგან მიღება-ყიდვა. (მაგალითად, ისეთი საიტის მეპატრონისაგან, სადაც აუცილებელია იმეილის მითითება რეგისტრაციის გასავლელად).

მეთოდი პირველი. სპამერისათვის დიდი მნიშვნელობა აქვს ისეთი იმეილ მისამართების მიღებას, რომელსაც ნამდვილად იყენებს და ამოწმებს ვინმე, ამის შესამოწმებლად ისინი აგზავნიან შემდეგი სახის იმეილს:

- *თუ კი აღარ გინდათ რომ მიიღოთ მსგავსი სახის წერილები ჩვენგან "**დააკლიკეთ აქ**".*

თუ კი ასეთი იმეილის მიმღები დააკლიკებს ამ ლინკზე, იგი თავისდა უნებურად, შეატყობინებს სპამერს, რომ ამ იმეილს მართლა ჰყავს პატრონი.

მეორე მეთოდი. სპამერები ზოგჯერ ტექსტთან ერთად აგზავნიან სურათს. შეიძლება იგი იმდენად პატარა იყოს, რომ ვერც კი დავინახოთ. ყველა ამ სურათს განსხვავებული იმეილისთვის განსხვავებული სახელი აქვს. როდესაც იმეილს გავხსნით, ავტომატურად ეგზავნება მოთხოვნა სურათის ჩატვირთვის შესახებ იმ სერვერს სადაც სურათი ატვირთულია. ამის შესახებ კი სპამერები იგებენ და მათთვის ცნობილი ხდება, რომ თქვენ კითხულობთ ამ იმეილს.

„ჯაშუშური პროგრამები“ (Spyware). პროგრამა რომელიც თუ მოხვდა კომპიუტერში ,პატრონის უკითხავად, აქვს შეხწევა კონფიდენციალურ ინფორმაციასთან. ისინი კომპიუტერში შეიძლება მოხდენ ქსელური ჭიაყელების, ტროას ცხენის საშუალებით ან რეკლამებით (adwarw). ერთ ერთი ასეთი შპიონური პროგრამაა **„ფიშინგი“ (Phishing).** ეს არის საფოსტო გზავნილის პროგრამა სადაც მითითებულია გარეგნულად რეალურის მსგავსი საიტი, რომელიც სინამდვილეში არის ფიშ საიტი და სადაც ნაკლებად გამოცდილი მომხმარებელი აგზავნის მის კონფიდენციალურ ინფორმაციას. ამავე სახეს მიეკუთვნება ე.წ. **„ფირმინგი“** ეს უფრო დახვეწილი სახეა „ფიშინგის“. მომხმარებელი ვთქვათ, ინტერნეტ ბანკში შესვლისას მისდამი შეუმჩნევლად გადის მცდარ საიტზე, რომლის გარჩევაც ძალიან ძნელია.

Adware მავნე პროგრამაა, რომელიც აჩვენებს რეკლამას კომპუტერის მუშაობის დროს თქვენი თანხმობის გარეშე. იგი გადაგამისართებთ სარეკლამო ვებ საიტებზე და გასცემს მარკეტინგულ ინფორმაციას თქვენზე. იგი ამას აკეთებს თქვენი თანხმობის გარეშე .

Spyware პროგრამა ჯაშუში. აგროვებს ინფორმაციას კომპიუტერის კონფიგურაციაზე, მომხმარებელზე, მომხმარებლის აქტიურობაზე.

Rootkits ეს არის პროგრამული საშუალებების ნაკრები (სკრიპტები, შემსრულებელი ფაილები, მაკონფიგურებელი ფაილები), *რომელთა საშუალებითაც ხდება ობიექტების მასკირება, მართვა,* მონცემთა შეგროვება. გავრცელდა UNIX-დან.

Backdoors დამპროექტებლის მიერ შეგნებულად ჩანერგილი დეფექტია ალგორითმში, რაც საშუალებას აძლევს მას ფარულად შეაღწიოს თქვენს კომპიუტერში მონაცემებთან ან შეაღწიოს დაცილებულ კომპიუტერში მის სამართავად.

Logic bomb პროგრამაა, რომელიც ეშვება გარკვეული დროითი ან ინფორმაციული პირობების შესრულებისას. ამახინჯებს, ანადგურებს ინფორმაციას და ქმნის წვდომას მონაცემებთან.

Ransomware მავნე პროგრამაა, გამოიყენება შანტაჟისთვის და გამოძალვისთვის.

Polymorphic malware პოლიმორფული მავნე პროგრამაა, რომელიც მუდამ იცვლება და ამიტომ ძნელი აღმოსაჩენია. იგი ცვლის ფაილებს, კუმშავს მათ და შიფრავს სხვადასხვა გასაღებებით.

Armored virus „დაჯავშნული“ ვირუსია. ის სპეციალურად დაკოდირებულია, შეცდომაში შეყავს ანტივირუსული პროგრამა მისი რეალური ადგილმდებარეობის შესახებ.

ზემოთ ჩამოთვლილი მავნე პროგრამებით შესაძლებელია განხორციელდეს შემდეგი სახის შეტევები:

* **Man-in-the-middle** შუამავლის შეტევა ან როგორც მას უწოდებენ „ადამიანი შუაში“ *(MITM).* ამ დროს ხდება არხის კომპრომეტაცია, რომელთანაც მიერთებულია ბოროტგამზრახველი. იგი ხელთ იგდებს შეტყობინებას და ცვლის მას;
* **DDoS** (Distributed Denial of Service) სერვერების მწყობრიდან გამოყვანა მათზე უდიდესი რაოდენობის მცდარი მოთხოვნების მიწოდებით;
* **DoS** მომსახურეობაზე უარი (*Denial of Service*).ჰაკერული შეტევაა. ამ დროს სისტემურ რესურსებთან წვდომა შეზღუდულია ან ბლოკირებულია;
* **Replay ATTACK** ამ დროს ბოროტგამზრახველი მოტყუებით ან სხვა განზრახვით მრავალჯერ ხელმეორედ გადაცემს დამახინჯებულ ინფორმაციას;
* **Smurf attack** ინტერნეტ ოქმის Control Message (ICMP) თვისებების გამოყენებით ქსელში იგზავნება უამრავი ყალბი IP მისამართზე მოთხოვნა შესატევი ობიექტის IP მისამართით. მათგან ობიექტი ღებულობს სტანდარტით გათვალისწინებულ შეტყობინებებს, მათი სიმრავლიდან გამომდინარე ობიექტი ამცირებს წარმადობას, ფუნქციონირების სრულ შეწყვეტამდე;
* **Spoofing** მონაცემთა ფალსიფიკაცია, როცა პროგრამა სხვა პროგრამის სახელით (ინიღბება) იწყებს ფუნქციონირებას;
* **Spim** ეს არის სპამი, რომელიც ელექტრული ფოსტის მაგივრად ვრცელდება IM ოქმით, უსწრაფესად;
* **Vishing** ეს არის ყალბი სატელეფონო შტყობინება ვითომდაც ავტომატური მოპასუხისგან, რითაც კლიენტისგან ითხოვენ მის კონფიდენციალურ მონაცემებს;
* **Spear phishing** იგივეა რაც Phishing, ოღონდ მხოლოდ ერთ მომხმარებელზე;
* **Xmas attack** ცნობილია როგორც საშობაო ნაძვისხის პაკეტი, როგორც „კამიკაძის“ პაკეტი. იგი იგზავნება ქსელში ძალიან დიდი რაოდენობით და შეიცავს გავრცელებული ქსელური ოქმების ოპციებს;

* **Privilege escalation** პრივილეგიების დონის აწევა. იყენებს პროექტირების , კონფიგურაციის ან ოპერაციული სისტემის შეცდომებს;

* **Malicious insider threat** ინსაიდური საშიშროება, რომელსაც იყენებს ორგანიზაციის განაწყენებული თანამშრომელი, დამკვეთები, ყოფილი თანამშრომელი, რომელთაც აქვთ (ჰქონდათ) წვდომა ორგანიზაციის შიგა ინფორმაციაზე უსაფრთხოების კუთხით;

* **DNS poisoning and ARP poisoning.** ARP პაკეტები 2 დონისაა, ქსელში მათი გადაგზავნა არ ხდება და იგი იყენებს კონფიგურაციის შეცდომებს, რის შედეგადაც ორი მოწყობილობა ერთდროულად „ხედავს“ კავშირის არხს. DNS poisoning არ ჭირდება ARP poisoning, იგი მე-3 დონის პაკეტია (UDP/IP, ხანდახან TCP/IP) და მარშრუტიზირებადია. სიტყვა poison ნიშნავს შხამს;

* **Transitive access** ტრანზიტული შეღწევაა. გარდამავალი ნდობაა, როცა თუ A ენდობა B-ს, ხოლო B ენდობა C-ს, მაშინ გარკვეული პირობების შემთხვევაში A ენდობა C-ს;

* **Client-side attacks** შეტევა კლიენტის მხრიდან. ეყრდნობა შემდეგს. კლიენტის სპეციალური (მავნე) მოთხოვნით სერვერი უწევს მას მომსახურებას, რომლის დროსაც ის იყენებს შეცდომებს კლიენტის მომსახურების გამოყენებით პროგრამებში;

* **Password attacks: Brute force; Dictionary attacks; Hybrid; Birthday attacks; Rainbow tables**
პაროლების „გატეხა“, სრული გადარჩევა („უხეში ძალა“), გადარჩევა ლექსიკონის გამოყენებით, ჰიბრიდული (კომბინატორული) შეტევა, „დაბადების დღის“ შეტევა, „ცისარტელას ცხრილები“, რომლითაც ხდება მცდელობა ჰეშ ფუნქციიდან საწყისი ტექსტის აღდგენის;

* **Typo squatting/URL hijacking** მას ხშირად უწოდებენ URL გამტაცებელს, დამგესლავ (sting site) საიტს ან ყალბ საიტს. იგი დამყარებულია მომხმარებლის მიერ დაშვებულ გრამატიკულ (ტიპოგრაფიულ) შცდომებზე როცა მას არასწორად შეყავს ინტერნეტში მისამართები;

* **Watering hole attack** შეტევა „წყლის მსმელებზე“. ბოროტგამზრახველი აკვირდება კლიენტების ჯგუფს (ორგანიზაციები, რეგიონები, მრეწველები) თუ რომელი საიტების მომსახურეობით სარგებლობენ ისინი. ავირუსებს ამ საიტს და მერე აღწევს კლიენტების მონაცემებთან.

რეკომენდაცია შეიძლება გაეწიოს შემდეგ ანტივირუსულ პროგრამებს (***შენიშვნა*** აქ თეორიულად შესაძლებელია, რომ ზოგიერთი ფირმის პროდუქტი თვით იყოს საფრთხის შემცველი !!!) :

Avira AntiVir — TR/Dldr.VB.bnp;
Avast — Win32:Trojan-gen;
AVG — Downloader.VB.AIM;
BitDefender — Trojan.Downloader.VB.VKV;
ClamAV — Trojan.Downloader-15571;
DrWeb — Win32.HLLW.Kati;
Eset NOD32 — Win32/VB.NNJ worm;
F-Secure Anti-Virus — Trojan-Downloader.Win32.VB.bnp;
АнтивирусКасперского — Trojan-Downloader.Win32.VB.bnp;

McAfee — Downloader.gen.a;
Panda Security — W32/Penetrator.A.worm;
VBA32 — Trojan-Downloader.Win32.VB.bnp.

განსაკუთრებით აღსანიშნავია Microsoft მიერ შექმნილი უტილიტა MSRT (Malicious Software Removal Tool) WINDOWS ოპ.ს-ის [33], რომელსაც შეუძლია ნეიტრალიზაცია გაუკეთოს 2005 წლის შემდეგ გამოჩენილ შემდეგ ვირუსებს (malware) ან მავნე პროგრამებს (malicious software) იხ. ცხრ. 3.

ცხრ .3

Afcore	EyeStye	Losksky	Sality
Alcan	Eyeveg	Lolyda	Sasser
Alemon	FakeCog	Lovgate	Sdbot
Allaple	FakeInit	Mabutu	Sinowal
Alureon	FakePAV	Magania	Sienfbot
Anttiny	FakeRean	Magistr	Sober
Atak	FakeScanti	Maslan	Sobig
Badtrans	FakeSecSen	Matcash	spybot
Bagle	FakeSpypro	Mimail	Spyboter
Bagz	FakeSysdef	Mitglieder	Szribi
Bamital	FakeVimes	Mydoom	Storark
Bancos	FakeXPA	Mytob	Stration
Banker	F41Rootkit	Mywife	Stuxnet
Banload	Fizzer	Nachi	Swen
Beenut	Fotomoto	Netsky	Taterf
Berbew	Funner	Newacc	Tibs

Blaster	Gael	Nsag	Tilcun
Bobax	Ganda	Nugel	Torvil
Bofra	Gaobot	Nuwar	Tracur
Bredolab	Gibe	Ocieroor	Valia
Brontok	Gimmiv	Oficia	Virtumoride
Bropia	Goweh	Opaserv	Virut
Bubnix	Hacker Defender	Optix	Vobfus
Bugbear	Hacty	Optixpro	Vundo
Busky	Hamweq	Parite	Waledac
Captua	Hamig	Passalert	WinWebSec
Ceekat	Harnig	Plexus	Wootbot
Chir	Haxdoor	Poison	Wukill
Cissi	Helpud	PrivacyCenter	Yaha
Codbot	Hiloti	Purstiy	Yektel
Conficker	Horst	Pushbot	Yimfoca
Conhook	Hupigon	Ramnit	Zafi
Corripio	InternetAntivirus	Randex	Zbot
Clilnk	IRCBot	Rbot	Zindos
Cutwail	Ispro	Reatle	Ziob
Cycbot	Ieefo	Renocide	Zonebac
Daurso	Kelihos	Renos	Zotob
Domjuice	Kelvir	Rimecud	Zuten
Duaru	Koobface	Rjump	
Dursg	Korgo	Rorpian	
Esbot	Ldpinch	Rustock	

Evaman	Lethic	Ryknos	

§1.4. ინფორმაციის დაცვაში მიღებული ცნებები და განმარტებები

გქ -ის უსაფრთხოება . ეს არის მისი დაცულობის თვისება შემთხვევითი ან წინასწარ განსაზღვრული ზემოქმედებისაგან მის ნორმალურ ფუნქციონირების პროცესზე (იგულისხმება ინფორმაციის კომპონენტების დატაცება, ცვლილება, განადგურება);

- ***სანქცირებული შეღწევა ინფორმაციასთან.*** ამ დროს არ ირღვევა დადგენილი წესები შეღწევის დონეების მიმართ;
- ***არასანქცირებული შეღწევა ინფორმაციასთან.*** ამ დროს ირღვევა დადგენილი წესები შეღწევის დონეების მიმართ;
- ***მონაცემთა კონფიდენციალობა;***
- ***უსაფრთხოების პოლიტიკა*** *(Security Police)* წესების და ნორმების ერთობლიობაა, რომელიც უზრუნველყოფს ინფორმაციის დაცვის სისტემის დაცვას დადგენილი საშიშროებათა სიმრავლიდან;
- ***უსაფრთხოების მოდელი*** *(Sequrity Model)* უსაფრთხოების პოლიტიკის ფორმალური წარმოდგენა;
- ***შეღწევის დისკრეციული მართვა*** *(Discretionaly Acces Control).* შეღწევის მართვა, რომელიც ხორციელდება ადმინისტრატორის მიერ დაშვებულ შეღწევის უფლებათა არეალიდან;
- ***დაშვების მანდატური მართვა*** *(Mandatory acces Control)* შეღწევის მართვა, რომელიც დაფუძნებულია შეღწევის გაცემის წესების ერთობლიობაზე, განსაზღვრულია სუბიექტების და ობიექტების უსაფრთხოების ატრიბუთთა სიმრავლიდან;

- ***უსაფრთხოების ბირთვი*** *(Trusted Computing Base TCB)* გამოთვლითი სისტემის აპარატულ, პროგრამულ, სპეციალურ კომპონენტთა ერთობლიობაა, რომელნიც ასრულებენ დაცვის ფუნქციებს და უზრუნველყოფენ უსაფრთხოებას;

- ***იდენტიფიკაცია*** *(Identifigation)* არსებითი ნიშნების განსაზღვრა მათთვის უნიკალური ნიშნულების მინიჭების გზით;

- ***ადეკვატურობა*** *(Assurande)* რეალურად უზრუნველყოფილი უსაფრთხოების დონის მახასიათებელი, რომელიც გამოხატავს დაცვის საშუალებების ეფექტურობის ხარისხს და საიმედოობას;

- ***ტაქსონომია*** *(Taxonomy)* მეცნიერება, რომელიც შეისწავლის რთულად ორგანიზებული ობიექტებს და მოვლენებს, რომელთაც იერარქიული წყობა ახასიათებს. ის დაფუძნებულია მოვლენათა დეკომპოზიციაზე და მათ ეტაპობრივ დაზუსტებაზე;

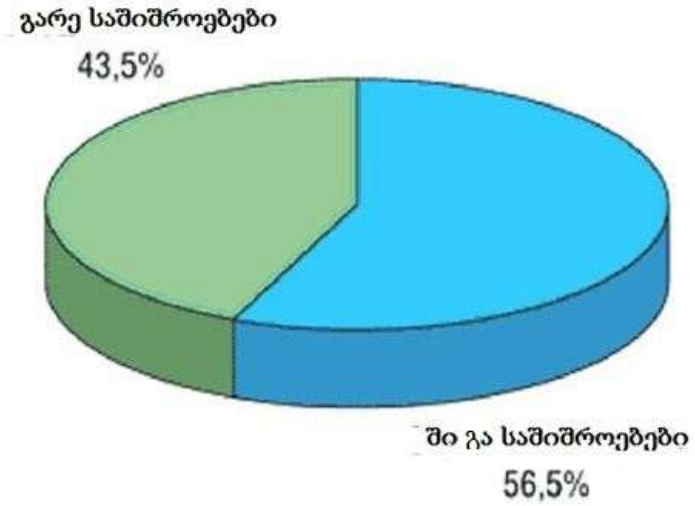
- ***პირდაპირი ურთიერთობა*** *(Trusted Patch)* ინფორმაციული ურთიერთმოქმედების პრინციპები, რომლებიც უზრუნველყფს, რომ გადასაცემი ინფორმაცია არ დამახინჯდება ან არ მოხდება სხვის მიერ ინფორმაციის ხელში ჩაგდება.

საქართველოსკანონში „ინფორმაციული უსაფრთხოების შესახებ" დამატებით განმარტებულია ისეთი ცნებები, როგორიცაა: კიბერსივრცე, კიბერშეტევა, კომპიუტერული ინციდენტი, კრიტიკული ინფორმაციული სისტემა, კრიტიკული ინფორმაციული სისტემის სუბიექტი, შინასამსახურებრივი გამოყენების ინფორმაცია, ინფორმაციული აქტივი.

თავი 2 ფაილები და მონაცემთა ბაზები,როგორც დასაცავი ინფორმაციული ობიექტები

§2.1 დასაცავი ფაილები და მონაცემთა ბაზები

ფაილები განთავსებულია სხვადასხვა ინფორმაციის მატარებლებზე და კუთვნილების მიხედვით ისინი იყოფა პირად, ჯგუფურ და საერთოდ. ფაილების შენახვის უზრუნველსაყოფად გამოიყენება აპარატურული, პროგრამული საშუალებები და აგრეთვე გამოყენებისა და შენახვის (დოკუმენტური აღრიცხვა) ორგანიზაციული ღონისძიებები. საკითხის შესასწავლად საინტერესოა განისაზღვროს საფრთხეთა კლასიფიკაციის ორი სახე: შიგა (თანამშრომელთა უპასუხისმგებლობა, საბოტაჟი და ფინანსური ყალთაბანდობა) და გარე (ვირუსები, ჰაკერები და სპამი). როგორც დიაგრამიდან ნახ.1.4 ჩანს [39] უპირატესი გავლენა აქვს შიგა საფრთხეებს.



ნახ. 1.4

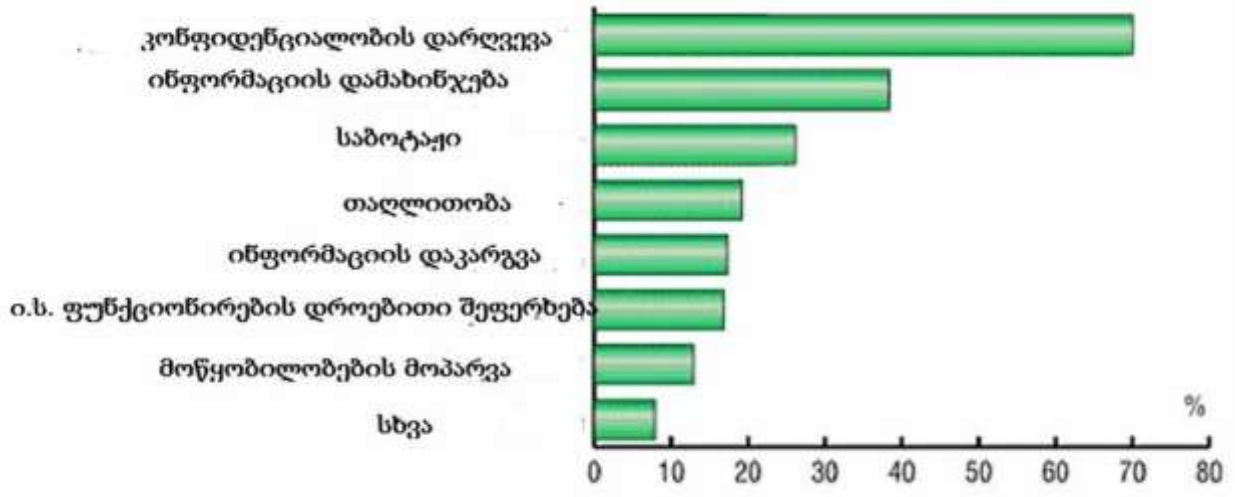
თანამედროვე ***გქ*** გამოყენებისას დომინანტია უდიდესი მოცულობის მონაცემთა შესანახად ე.წ. მონაცემთა ბაზების მართვის სისტემების (შემდგომში **მბმს**) გამოყენება. შესაბამისად დგება საკთხი მრავალ მომხმარებლიან ***მბმს*** ინფორმაციული უსფრთხოების უზრუნველყოფა სამი ფუნდამენტური მიმართულებებით: კონფიდენციალურობა, მთლიანობა და შეღწევადობა. ზოგადად მიღებულია 5 ძირითადი მიმართულება (მოთხოვნა) ***გქ*** ინფორმაციის დაცვის ორგანიზაციისათვის:

1 კონფიდენციალობა;
2 აუტენფიკაცია;
3 მთლიანობა;
4 შეღწევის კონტროლი;
5 თანამონაწილეობა („უტყუარობა“).

განვიხილოთ თითოეული თუმცა ეს ჩამონათვალი ზოგადად მიღებულია, მაგრამ არა ერთადერთია.

§ 2.1.1 კონფიდენციალობა

კონფიდენციალობა ეს არის თვისება, რომელიც გარანტირებულად უზრუნველყოფს იმას, რომ ინფორმაცია ***გქ-ში*** არ იქნება შეღწევადი ან გახსნილი არაავტორიზებული (არაუფლებამოსილი) პიროვნებებისათვის, ობიექტებისთვის ან პროცესებისათვის. როგორც გამოკვლევა [39] აჩვენებს, კონფიდეციალობის დარღვევა უპირველეს საფრთხეს წარმოადგენს ნახ. 1.5.



ნახ. 1.5

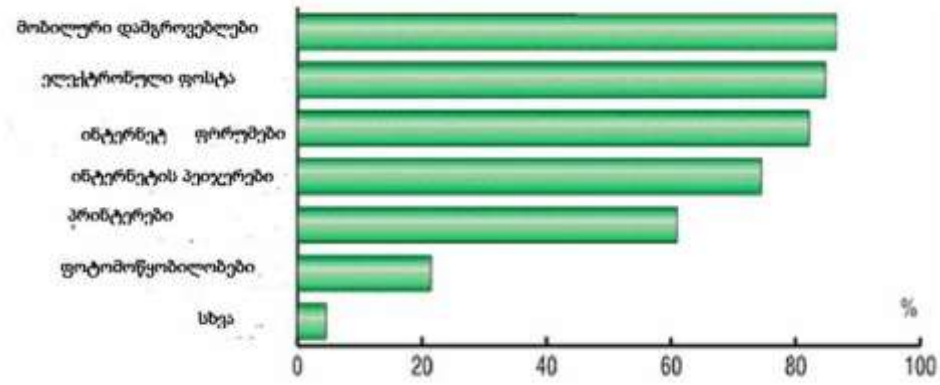
ფუნქციის რეალიზაციისათვის გამოიყენება 4 პროცედურა (სახე, ვერსია):
- სისტემები კავშირის დამყარებთ;
- სისტემები კავშირის დაუმყარებლობით;
- ცალკეული ინფორმაციული ველის დაცვა;
- ტრაფიკის კონტროლის დაცვა.

პრაქტიკული რეალიზაციის მხრივ 1-2 სახე, უზრუნველყოფილია შესაბამისი კავშირის დამყარებისა და კავშირის დაუმყარებელი ოქმებით.

მესამე სახე გამოიყენება ორივე სახის ქსელებისათვის და ახორციელებს არა მთლიანად ინფორმაციის დაცვას, არამედ მხოლოდ ცალკეულ მისი ველების.

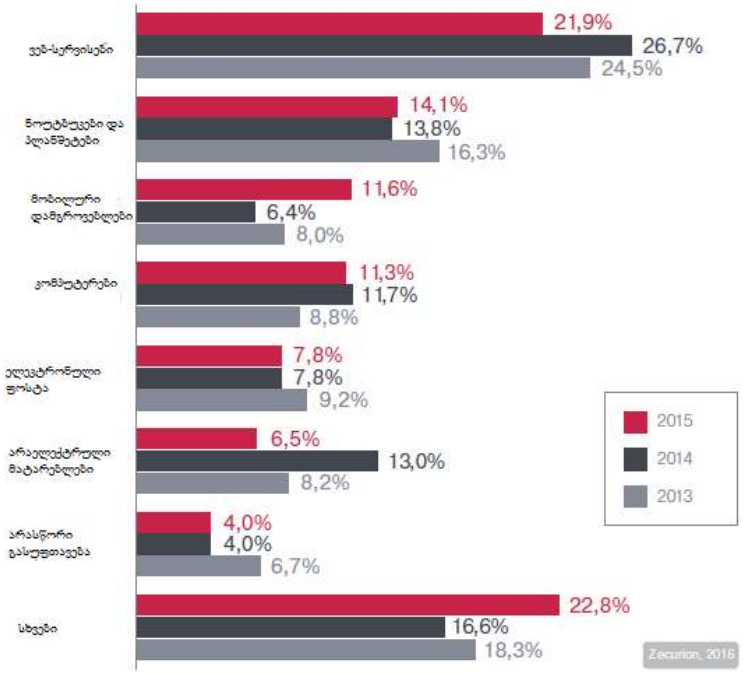
მეოთხე გულისხმობს ინფორმაციის ტრაფიკის კონტროლის შეუძლებლობას არაავტორიზებული (არაუფლებამოსილი) პირის მხრიდან. შედეგი მიიღწევა გამგზავნის, მიმღების მხრიდან, გადასაცემი ინფორმაციის რაოდენობისა და სიხშირის კოდირებით.

საინტერესოა იმ არხების ცოდნა [39], რომელთა საშუალებითაც ხდება კონფიდენციალურ მონაცემთა გაჟონვა ნახ. 1.6.



ნახ. 1.6

საინტერესოა ამ მხრივ მონაცემები [50] 2015 წ. რუსეთის ფედერაციაში. ნახ. 1.7.



ნახ. 1.7.

აქ აღსანიშნავია ე.წ. ინტერნეტ „პეიჯერი“, რომელშიც იგულისხმება Bizex-ICQ „ჭიაყელა“-ს მსგავსი (ცნობილია აგრეთვე Exploit სახელით) ვირუსი. მიმღები თუ გახსნიდა ნაცნობისაგან მიღებულ წერილს მითითებული «http://www.jokeworld.biz/index.html :))LOL» ლინკით, მისი კომპიუტერი დავირუსდებოდა და ამის შემდეგ მისი საკუთარი ინფორმაცია (მათ შორის კონფიდენციალურიც), ICQ სერვერის გავლით ხდებოდა სხვისი საკუთარი.

§ 2.1.2 აუტენფიკაცია

მისი საშუალებით ხდება, რაღაც ინფორმაციის მიხედვით (რიცხვი, სიმბოლოთა ნაკრები, ალგორითმი), ცალსახად ობიექტის დადგენა [10,12]. ასეთ ინფორმაციას უწოდებენ ობიექტის იდენტიფიკატორს და იგი დარეგისტრირებული უნდა იყოს ქსელში. ამ დროს საქმე გვაქვს ლეგალურ ობიექტთან. თუ არა ობიექტი - არალეგალურია. მეორე საფეხურია ობიექტის აუტენფიკაცია, როცა დგინდება ობიექტის ჭეშმარიტობა. ამის შემდეგ, როგორც იტყვიან, ობიექტმა გაიარა ავტორიზაცია და ეძლევა გათვალისწინებული უფლებები. ე.ი. ამ

მიმართულების სარეალიზაციოდ i ობიექტის ინფორმაცია უნდა შეიცავდეს ორ ინფორმაციულ ველს IDi და Ki .

მიღებულია 2 სქემა აუტენფიკაციის სარეალიზაციოდ (იგულისხმება+ იდენტიფიკაციაც). პირველი სქემის დროს (იხ ცხრილი 4.), მას შემდეგ რაც N ობიექტი წარადგენს თავის 1Dn, თუ აღნიშნული ქსელში დარეგისტრირებულია, შემდეგში მოითხოვნება Kn. გამოითვლება ფუნქცია Y=F(1Dn, Kn), ხდება Y გამოთვლილი ფუნქციის და Ei მნიშვნელობის შედარება.

ცხრ. 4

ობიექტის ნომერი	საიდენტიფიკაციო ინფორმაცია	ინფორმაცია აუტენფიკაციისათვის
1	$1D_1$	E_1
2	$1D_2$	E_2
.		
N	$1D_n$	E_n

ტოლობის შემთხვევაში ჩაითვლება, რომ ობიექტმა გაიარა აუტენფიკაცია. მეორე სქემა უფრო უკეთეს შესაძლებლობებს იძლევა: იხ. ცხრ .5.

ცხრ.5

ობიექტის ნომერი	საიდენტიფიკაციო ინფორმაცია	ინფორმაცია აუტენფიკაციისთვის
1	$1D_1, S_1$	E_1
2	$1D_2$, S_2	E_2
.		
N	$1D_n$, S_n	E_n

აქ პირველი სქემიდან განსხვავებით $E_i = F(S_i, K_i)$. თვით S_i შემთხვევითი სიდიდის ვექტორია, რომელიც მიიღება $1D_n$ იდენტიფიკატორის მიღებისას. აუტენფიკაციის შესრულების თანამიმდევრობა შემდეგია: მოწმდება ობიექტის მიერ წარმოდგენილი თავისი $1D_n$ იდენტიფიკატორი. შემოწმების შემდეგ გამოითვლება (გამოიყოფა) ვექტორი S_i . შეკითხვის შემდეგ ობიექტი წარმოადგენს თავის აუტენფიკატორს K_i, გამოითვლება ფუნქცია $Y=F(S_i, K_i)$. Y და E_i ტოლობის შემთხვევაში პროცესი წარმატებით მთავრდება. ეს სქემა გამოიყენება ოპ.ს. UNIX-ში. პრაქტიკულად აქ მოთხოვნა Login-ით სისტემა ითხოვს 1D, ხოლო Password-ით კი K-ს. ფუნქცია F გამოითვლება დაშიფვრის DES ალორითმით. სუსტი მხარეა K პაროლების სისტემა. გაცილებით უკეთეს შედეგს იძლევა ბიომეტრული პარამეტრების გამოყენება, როგორიც არის : თვალის უჯრედოვანი გარსი, თითის ანაბეჭდი, სახის ფორმა და ზომა, ხელის გეომეტრიული ზომა, ხმის ტემბრი, ხელწერა და „კლავიატურული ხელწერა“.

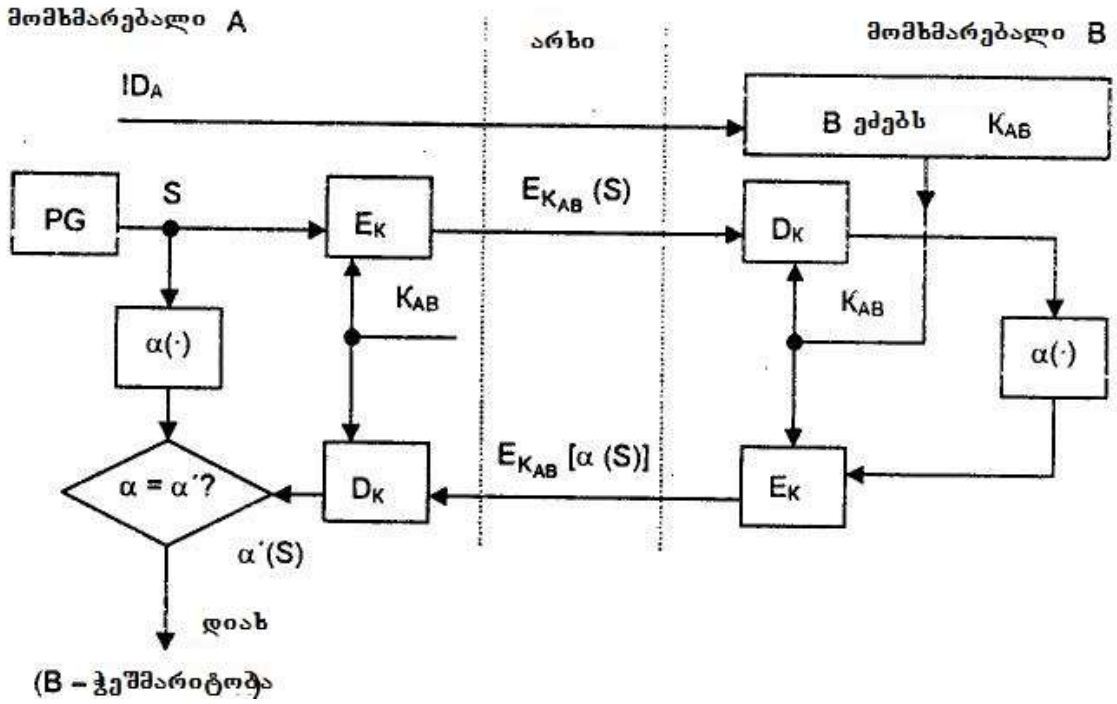
აქ განიხილება აუტენფიკაციის მომსახურების 2 სახე:

- მონაცემთა წარმოშობის უტყუარობა;
- კომუნიკაციის ობიექტების უტყუარობა.

პირველი მოთხოვნის რეალიზაცია განსაკუთრებით მნიშვნელოვანია კომუტაციის გარეშე ფუნქციონირებადი ქსელებისათვის. ამ შემთხვევაში პაკეტები ერთმანეთის მიმართ

დამოუკიდებელი არიან. ამ დროს მთავარია არა მარტო ის, რომ მიმღები უნდა იყოს მხოლოდ ის რომელიც მითითებულია, არამედ ისიც, რომ პაკეტის დასაწყისში მითითებული გადამცემის მისამართიც უტყუარია.

პირველი სახის რეალიზაციის კლასიკურ სახეს წარმოადგენს ე.წ. „ხელის ჩამორთმევის“ პროცედურა: ნახ. 2.1 .



ნახ. 2.1.

ვუშვებთ, რომ ორივე მხარე იყენებს სიმეტრიული შიფრაციის მეთოდს და შესაბამისად ერთსა და იმავე საიდუმლო გასაღებს K_{AB} . თუ პროცესი იწყება A მხარის ინიციატივით, ის უგზავნის B მხარეს თავის იდენტიფიკატორს ID_a. შემდეგში პროცესი მიმდინარეობს შემდეგი თანამიმდევრობით:

- B მხარე მიღებული ID_a იდენტიფიკატორით ბაზაში პოულობს საიდუმლო გასაღებს K_{AB} ;
- A მხარე გენერაციას უკეთებს S შემთხვევით რიცხვების თანამიმდევრობას ფსევდო რიცხვების გენერატორით PG და უგზავნის B მხარეს კრიპტოგრამას $E_{KAB}(S)$;
- B მხარე გაშიფრავს მიღებულ კრიპტოტექსტს და ეცნობა საწყის თანამიმდევრობას S.
- შემდგომში ორივე მხარე გარდაქმნის თანამიმდევრობას S-ს ღია ცალმხრივი ფუნქციით α() :
- B მხარე უგზავნის A მხარეს დაშიფრულ შეტყობინებას Θ(S);
- A მხარე ადარებს მიღებულ Θ(S) დაშიფრულ შეტყობინებას მის მიერ გაგზავნილ α(S). დამთხვევისას ითვლება, რომ აუტენფიკაცია ჩატარდა წარმატებით;

შესაბამისად, B მხარე ანალოგიურად ატარებს A მხარის აუტენფიკაციას.

მეორე მოთხოვნის არსი, ქსელებში კომუნიკაციით, მდგომარეობს იმაში, რომ გარანტირებული უნდა იყოს მოთხოვნილი კომუნიკაციის ობიექტი.

აუტენფიკაციის ორივე სახე გამოიყენება ქსელების (კავშირის დამყარებით და დაუმყარებლად) ***ქსელურ, სატრანსპორტო და გამოყენებით დონეებზე.***

§ 213 მთლიანობა

მთლიანობა არის მონაცემთა ან კომპიუტერული სისტემის ისეთი მდგომარეობა, როდესაც მონაცემები ან პროგრამები გამოიყენება დადგენილი წესით, რომლითაც უზრუნველყოფილია სისტემის მდგრადი ფუნქციონირება. მთლიანობა განიხილება ორივე სახის გქ-ის. ქსელებში კავშირის დამყარებით დამატებითი ფუნქციაა მონაცემთა აღდგენა, რომლის რეალიზაცია ხორციელდება მოდიფიკაციის, ჩაშენების, დაკარგვის ან მონაცემთა ხელმეორედ გადაცემისას პაკეტების თანამიმდევრობაში. ეს მომსახურება გამოიყენება **ქსელურ, სატრანსპორტო და გამოყენებით დონეებზე.** მთლიანობაზე კონტროლი, ქსელებში კავშირის დაუმყარებლად, გულისხმობს მხოლოდ ცალკეულ პაკეტის მთლიანობის კონტროლს დიდი რაოდენობის ინფორმაციის გადაცემისას, რომელიც ახასიათებს სეანსურ გადაცემების ციკლებს, ანალიზის (კონტროლის) გარეშე. შესაბამისად ეს მომსახურება ვერ იცილებს თავიდან ბოროტმზრახველების მიერ მონაცემთა დაკარგვას, ჩართვას ან ხელმეორედ გადაცემას. იგი გამოიყენება აუტენფიკაციასთან ერთად ***ქსელურ, სატრანსპორტო და გამოყენებით დონეებზე.***

ინფორმაციული უსაფრთხოების უზრუნველყოფისათვის, ამ სახის კონტროლისათვის, შემოთავაზებული იქნა 1977 წ. ე.წ. **ბიბას მოდელი** (Biba Model). ***ბელა - ლაპადულას*** მოდელში (***დევიდ ბელომ და ლეონარდო ლაპადულამ***) სუბიექტების და ობიექტების მოდელების დონეებში დამატებული იქნა მთლიანობის დონეები, სხვა და სხვა დონის სუბიექტებსა და ობიექტებს შორის ურთიერთკავშირის აკრძალვა. ამ მოდელში ,ეს იწვევს დამატებით სირთულეს. კერძოდ:

<u>1 თუ სუბიექტი</u> **კითხულობს** დაბალი დონის ობიექტს, მაშინ სუბიექტის მთლიანობის დონე მცირდება ობიექტის მთლიანობის დონემდე;

<u>2 თუ სუბიექტი</u> **ავსებს** ობიექტს მაღალი დონის მთლიანობით, მაშინ ობიექტის მთლიანობა მცირდება სუბიექტის მთლიანობის დონემდე.

მოდელი საკმაოდ რთულია და იწვევს საბოლოოდ მთლიანობის დონეების დაწევას, რაც თავისთავად უარყოფითი მოვლენაა.

ბიბას მოდელთან შედარებით, უკეთესია ***კლარკ-ვილსონის*** მოდელი, რომელიც 1993 წ. იქნა შემოთავაზებული. მისი გამოყენების ძირითადი სფეროებია საბანკო სფერო და კომერცია. იგი ემყარება ორ პრინციპს :

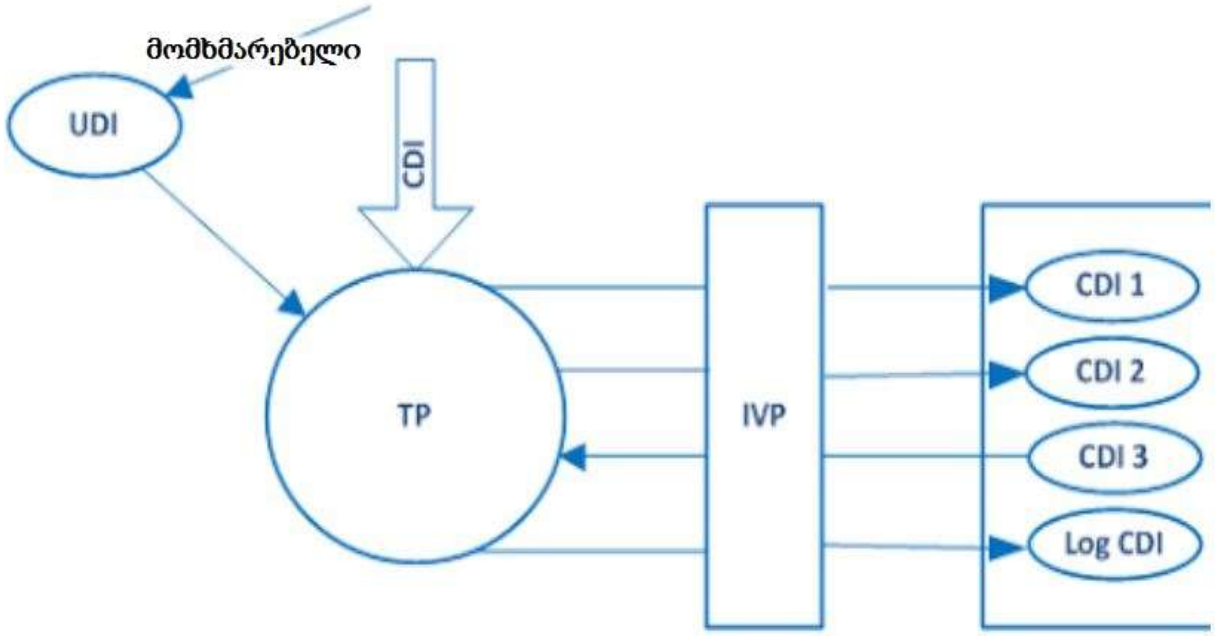
- ***შიგა მთლიანობა,*** რომელიც განსაზღვრავს სისტემის შიგა მდგომარეობას მიღწეულს „სწორი (მართებლივი) შეთანხმებებით“ ;
- ***გარე მთლიანობა***- სისტემის შიგა მდგომარეობის ურთიერთობით გარე სამყაროსთან, რომელიც რეალიზებულია ვალდებულებების დაყოფით.

ეს მოდელი რეალიზდება წესების ერთობლიობით და არა მათემატიკურად ფორმალიზებული მოდელივით (**ჰარისონ-რუზო-ულმანის მოდელი** HRU). გარდა ამისა, სუბიექტსა და ობიექტს შორის არსებობს პროგრამული „შრე“, რომლითაც რეგულირდება ეს ურთიერთკავშირი.

ყველა მონაცემი ამ მოდელში წარმოდგენილია 2 კლასში:

- მონაცემის აუცილებელი ელემენტი CDI ;
- მონაცემის სპონტანური ელემენტი UDI .

შემდგომში დგება წესების ნაკრები, რითაც განისაზღვრება მონაცემთა სხვა და სხვა ტიპთან ურთიერთობა (Certification Rules).



ნახ. 2.2.

ამ მოდელში გამოყენებულია დამატებით შემდეგი განსაზღვრებები:
- გარდაქმნის პროცედურები (**TP transformation procedure**) დაპროგრამებული აბსტრაქტული ოპერაციები წაკითხვა, ჩაწერა, შეცვლა;
- მთლიანობის შემოწმების პროცედურები (**IVP integrity verification procedure**) ეშვება პერიოდულად, რათა შემოწმებული იქნეს CDI მონაცემების არაწინააღმდეგობრიობა გარე ზემოქმედებებთან.

§2.1.4 შეღწევის კონტროლი

შეღწევის კონტროლის დანიშნულებაა, უზრუნველყოს რესურსების მხოლოდ ავტორიზებული გამოყენების შესაძლებლობა. მას ორი ფუნქცია აქვს: უზრუნველყოს ავტორიზებული მომხმარებლებისათვის რესურსების გამოყენება და არაავტორიზებულებისათვის (შიგა და გარე)-არა. გარეგნული ფუნქციონალური დანიშნულებით მსგავსია აუტენფიკაციის და კონფიდენციალობის მოთხოვნებთან (სახეებთან), მაგრამ შეღწევის კონტროლი უფრო ფართო შესაძლებლობებს იძლევა შეღწევის კონტროლის შეზღუდვის ორგანიზებისათვის. შეღწევის კონტროლის ორგანიზაციის პოლიტიკა განსაზღვრულია ორი შემადგენლით:

- შეღწევაზე გადაწყვეტილების მიღების კრიტერიუმების დადგენა;
-კონტროლის საშუალებები.

გადაწყვეტილებები მიიღება მოვლენების ან ობიექტების იდენტურობაზე ან შეღწევის წესებზე, რომლებიც განსაზღვრულია :

- ***მომხმარებლის მიერ;***
- ***ადმინისტრატორის მიერ.***

პირველი, მეორესთან შედარებით, უფრო ხშირად არის მხარდაჭერილი ოპერაციული სისტემების მხრიდან. იგი გამოიყენება ქსელურ, სატრანსპორტო და გამოყენებით დონეზე.

შეღწევადობა შეიძლება განხილული იქნეს, როგორც დამატებითი მომსახურების ფორმა. ის შეიძლება გახდეს შეტევის მიზანი, რათა მომხმარებლებისათვის მიუწვდომელი გახდეს ქსელის რესურსები. მიუხედავად თითქოს და გარეგნული მსგავსებისა, იგი არ განეკუთვნება შეღწევის კონტროლს. მისი რეალიზაცია ხდება სპეციალური მექანიზმებით ქსელური და გამოყენებითი დონეებისათვის.

განვიხილოთ შეღწევის ფორმალური მეთოდები.

დისკრეციული. მივიღოთ, რომ O ეს არის სისტემის ობიექტები, S სისტემის სუბიექტებია და R შეღწევის უფლებებია. მაგალითისთვის მივიღოთ, რომ გვაქვს რაღაც სისტემა, რომლის ევოლუცია დროში წარმოდგენილია მატრიცით M. ნახ.2.3.

ობიექტი/ სუბიექტი	O_1	O_2	O_3	O_4(Printer)
S_1	read			
S_2				Print
S_3		Read	Execute	
S_4	read write		read write	

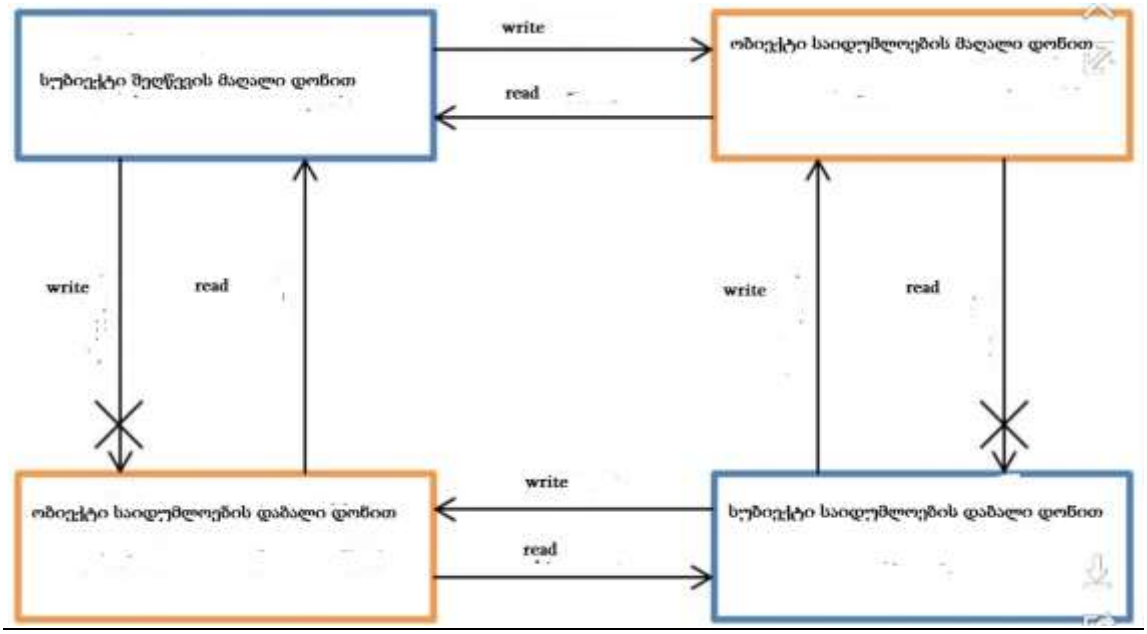
ნახ.2.3

ვადგენთ (ვქმნით) საწყისი ობიექტების და სუბიექტების შეღწევის უფლებებს და ვადგენთ შეღწევის საწყის მატრიცას M და ბრძანებათა ნაკრებს C. ამ სახის მოდელის ნიმუშია ე.წ. **ჰარისონ-რუზო-ულმანის მოდელი HRU**, რომელიც წარმოდგენილი იყო 1971 წ.. საწყის პირობად ვიღებთ, რომ ის არის უსაფრთხო **r** უფლების მიმართ, თუ არ არსებობს ბრძანებათა თანამიმდევრობა, რომელიც M მატრიცის უჯრედში დაამატებს **r** უფლებას, რომელიც საწყის პირობებში არ არსებობდა.

ჰარისონ-რუზო-ულმანის მოდელში დაშვებულია შემდეგი ელემენტარული ოპერაციები:

- **enter r into M[s,o]- S** სუბიექტისთვის r უფლებების დამატება O ობიექტ**ში;**
- **delete r from M[s,o]-** S სუბიექტისთვის r უფლებების წართმევა O ობიექტში;
- **create subject S-** ახალი S სუბიექტის შექმნა;
- **create object O-** ახალი O ობიექტის შექმნა;
- **destroy subject S-** არსებული S სუბიექტის ამოგდება;
- **destroy object O-** არსებული O ობიექტის ამოგდება.

<u>მანდატური მოდელი.</u> ერთ-ერთი პირველი უსაფრთხოების მოდელი, რომელიც შექმნეს დევიდ ბელომ და ლეონარდო ლაპადულამ იყო შეღწევის მართვის მოდელი. ის დაფუძნებულია სახელმწიფო უწყებებში საიდუმლო საქმის წარმოების კანონებზე.



ნახ. 2.4.

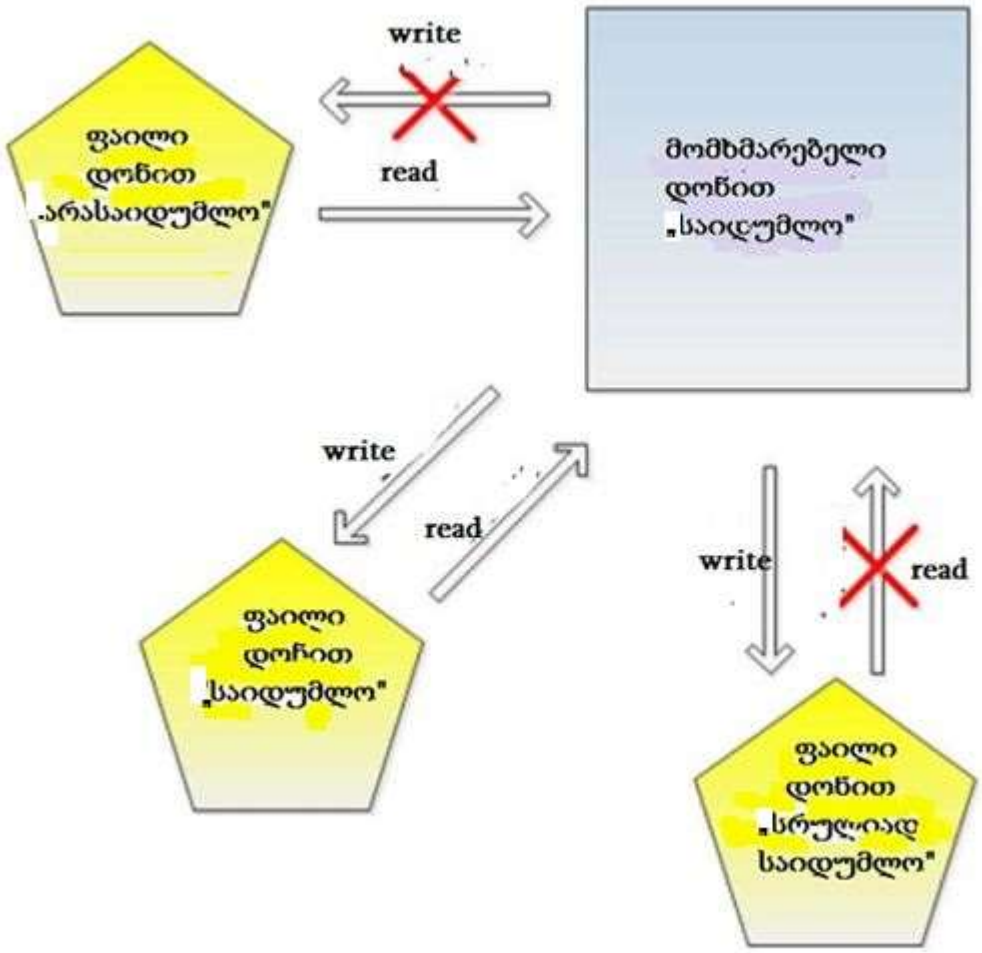
ეს მოდელი დაფუძნებულია საქმის წარმოების პროცესში მონაწილე ყველა მონაწილეთათვის და დოკუმენტებისთვის ე.წ. ნიშნულების მინიჭებაში (მაგალითად; სრულიად საიდუმლო, საიდუმლო და ა.შ.). შესაბამისად შეღწევის კონტროლი ხდება ამ ნიშნულების მიხედვით და იცავს შემდეგ წესებს :

- *<u>სუბიექტს აქვს უფლება გაეცნოს (წაიკითხოს) მხოლოდ იმ დოკუმენტებს, რომელთა უსაფრთხოების დონე (ნიშნული) არ აღემატება მის დონეს;</u>*
- *<u>სუბიექტს აქვს უფება შეიტანოს ინფორმაცია მხოლოდ იმ დოკუმენტებში, რომელთა უსაფრთხოების დონე (ნიშნული) არანაკლებია მის დონეზე.</u>*

პირველი წესით დაცულია ის ინფორმაცია, რომელიც მუშავდება უფრო მაღალი დონის სუბიექტის მიერ- უფრო დაბალი დონის სუბიექტის მიერ შეღწევის მცდელობისაგან.

მეორე წესი თავიდან იცილებს ინფორმაციის გაჟონვას (შეგნებული თუ უნებური) მაღალი დონის სუბიექტების მხრიდან.

მანდატურ მოდელში, დისკრეციული ჰარისონ-რუზო-ულმანის მოდელისაგან განსხვავებით, დაშვებულია შეღწევის ორი სახე **read** წაკითხვა და **write** ჩაწერა. თუმცა აქ შესაძლებელია გარკვეული მოდიფიკაციები ორ ძირითად სახესთან მიმართებით. ასეთი მკაცრი რეგლამენტირება განპირობებულია იმით, რომ ამ მოდელში კონტროლირდება არა ოპერაციები, რომელსაც ახორციელებს სუბიექტი ობიექტზე, არამედ ინფორმაციული ნაკადები. ეს ნაკადები შეიძლება იყოს სუბიექტიდან ობიქტისაკენ (ჩაწერა) ან ობიექტიდან სუბიექტისაკენ (წაკითხვა).

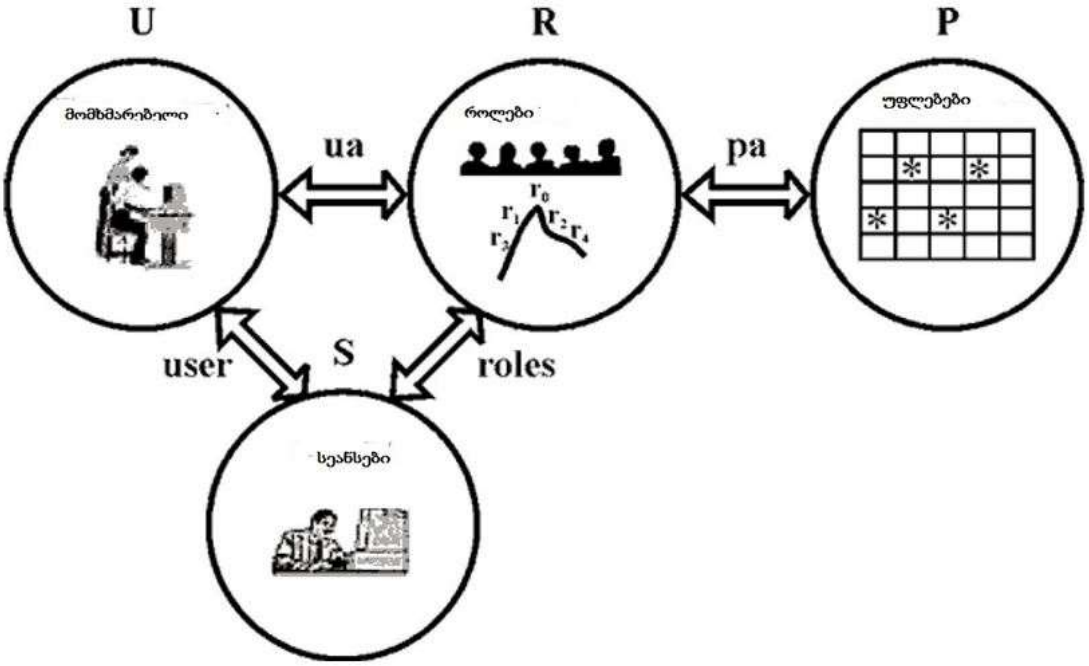


ნახ. 2.5.

სუბიექტის უსაფრთხოების დონეები განისაზღვრება უსაფრთხოების დონის ფუნქციით F:SvO->L. შესაბამისად სისტემის მრავლობითი მდგომარეობა აღიწერება წყვილთა ნაკრებით (F,M), სადაც M შეღწევის მატრიცაა (ჰარისონ-რუზო-ულმანის მოდელი). ამ შემთხვევისათვის მთლიანად სისტემის მოდელია Σ(Vo,R,T). აქ Vo საწყისი მდგომარეობაა, R მიმართვების სიმრავლეა და T:(VxR)->V გადასვლების ფუნქცია.

როლური მოდელი. ამ მოდელში სუბიექტის ცნება შეცვლილია ცნებებით „მომხმარებელი" და „როლი". შეღწევის მართვა ხორციელდება „როლების" ნაკრების დანიშვნით, სადაც ნაგულისხმია „როლების" იერარხია.

მოდელის ტიპები. ურთიერთგამომრიცხავი „როლები", დინამიკური დაყოფა ვალდებულებების (ერთი და იმავე სეანსის დროს შეზღუდვა „როლების" ერთდროულ გამოყენებაზე), შეზღუდვა როლების დანიშვნის რაოდენობაზე.



ნახ. 2.6.

§ 2.1.5 თანამონაწილეობა

თანამონაწილეობა განსაზღვრულია, როგორც პროცედურა, რომელიც გამორიცხავს (იურიდიულადაც) შესაძლებლობას ინფორმაციული ურთიერთგაცვლის რომელიმე მონაწილეს მიერ ამ პროცესში განხორციელებულ საქმიანობაზე უარის თქმას. დაგენილია თანამონაწილეობის ორი ფორმა:

- თანამონაწილეობა შეტყობინების გაგზავნაზე;
- შეტყობინების მიღების დადასტურება (მტკიცებულება).

პირველი ფორმა მიმღებს აძლევს მტკიცებულებას მასზე, რომ შეტყობინება გაგზავნილი აქვს მითითებულ წყაროს და რომ მისი მთლიანობა არ არის დარღვეული მიუხედავად იმისა, თუ გადამცემს (წყაროს) სხვა შეხედულება შეიძლება ჰქონდეს ამ ფაქტზე.

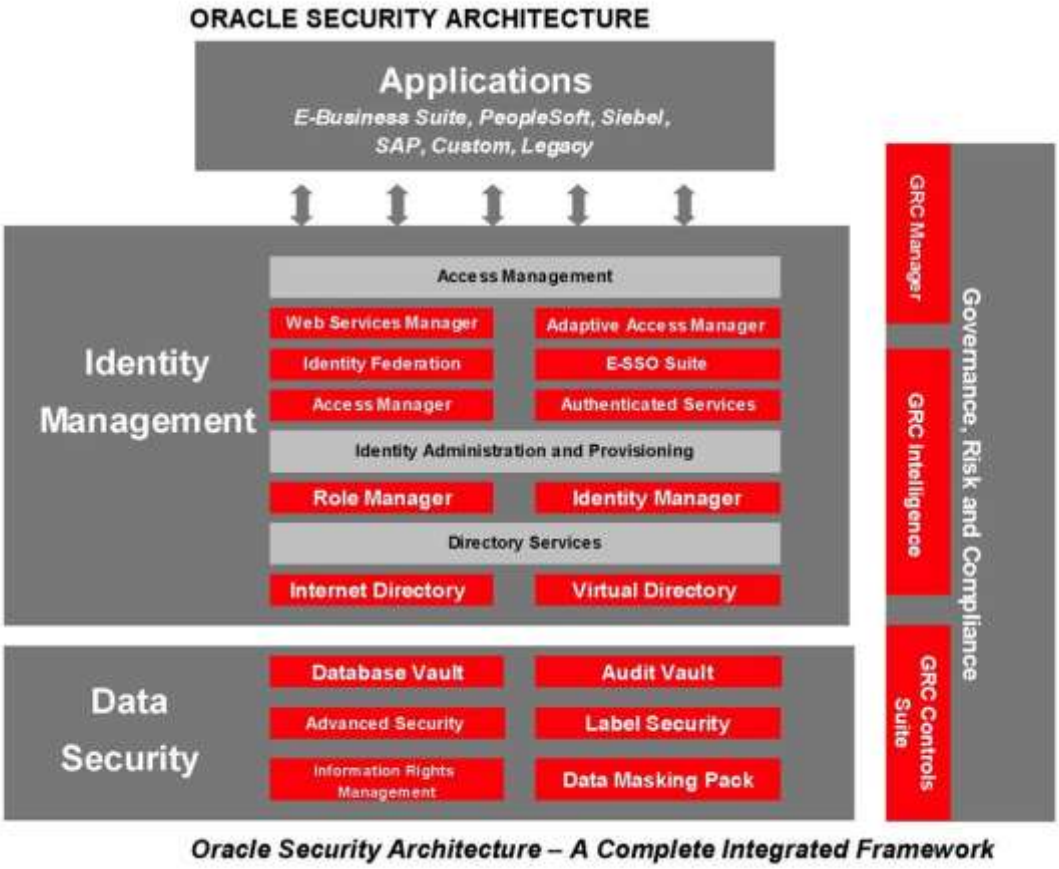
მეორე ფორმა, პირიქით, აძლევს მტკიცებულებას გადამცემს მასზე, რომ მიმღებმა მიიღო შეტყობინება მიუხედავად იმისა დაადასტურებს თუ არა ამ ფაქტს მიმღები .

ეს ფორმები უფრო მძლავრია მონაცემების წარმოშობის აუტენფიკაციასთან შედარებით. განსხვავება ისაა, რომ მიმღებს (გადამცემს) შეუძლია დაუმტკიცოს მესამე მხარეს თავისი სიმართლე.

ეს პროცედურა ძირითადად გამოიიყენება OSI მოდელის მე-7 დონეზე. გამოიყენება „ელექტრონული ხელმოწერის“ და მთლიანობის უზრუნველსაყოფი ტექნოლოგიები და ნოტარიუსი.

§ 2.2 ინფორმაციის დაცვა ORACLE10 მონაცემთა ბაზაში. ძირითადი ამოცანები

ძალიან ზოგადად დაცვის სისტემის არქიტექტურა [13,14,15,8,16,37] შეიძლება წარმოვიდგინოთ შემდეგნაირად ნახ. 2.2.1.



ნახ. 2.2.1.

იგი შედგება შემდეგი ძირითადი მოდულებისაგან:

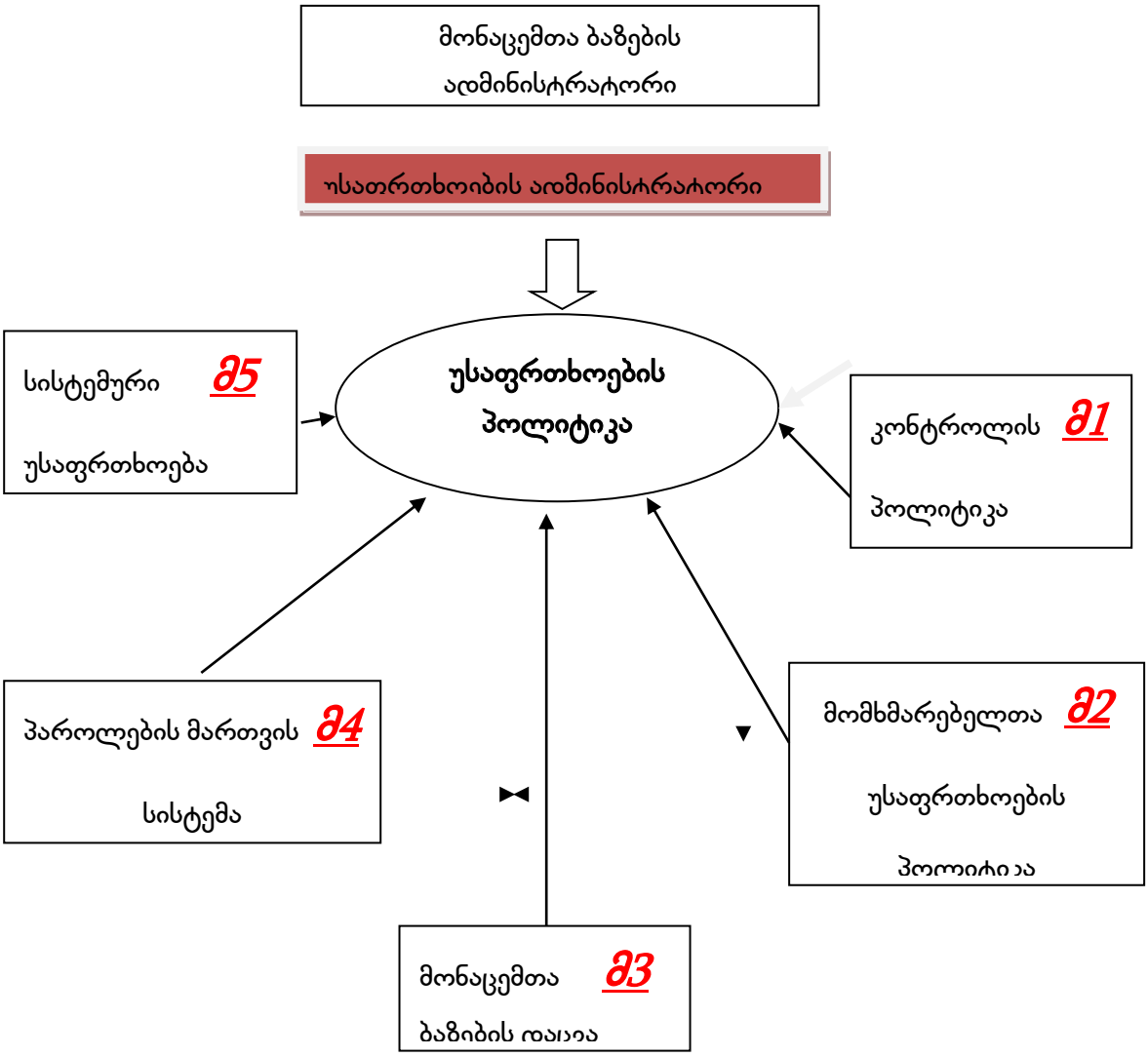
- Oracle Access Manager (OAM) უზრუნველყოფს აუტენთიკაციის ცენტრალიზებულ მართვას, ერთჯერადი რეგისტრაციის (SSO) სერვისებისა და სხვა აუტენთიკაციის მექანიზმებს;
- Oracle Adaptive Access Manager (OAAM) ითვალისწინებს web სერვისებთან და სხვა ქსელურ რესურსებთან შეღწევის კონტროლს;
- Oracle Role Manager როლების მენეჯერი;
- Oracle Identity Manager უზრუნველყოფს მიმართვის საფუძველზე შეთანხმებას გაწყობაზე, თვითმომსახურებას და მონაცემთა იდენტიფიკაციის არაერთგვაროვან სისტემებთან ინტეგრაციაზე კავშირების LDAP, ოპ.ს., მეინფრეიმების, ERP პაკეტების, მონაცემთა ბაზებისა და ა.შ. საშუალებებით;
- Oracle Enterprise SSO, გამჭოლი აუტენთიკაცია, ტერმინურ რეჟიმში (Remote Desktop, VDI, Citrix) მკაცრი და გაძლიერებული აუტენთიკაცია მონაცემებთან შეღწევისას, ადმინისტრატორებისა და მომხმარებლების მოქმედებების დოკუმენტირება ჟურნალებში;

- Oracle Identity Federation. ის აძლევს საშუალებას, კომპანიებმა განახორციელონ (გაწიონ) მომსახურება და გაცვალონ საიდენთიკატორო ინფორმაცია უსაფრთხოების შესაბამის სფეროებში. მომხმარებელი არაა ვალდებული მის ყოველ ახალ მოთხოვნაზე ხელმეორედ შევიდეს სისტემაში, რომ მიიღოს დაშვება იმ ობიექტთან, სადაც იგი აწარმოებს ბიზნესს;
- Oracle Web Services Manager უზრუნველყოფს ვებ-სერვისების უსაფრთხოებას WS-Security ოქმის გამოყენებით;
- Oracle Internet Directory. შეიცავს გამოყენებად პროგრამებს (აპლიკაციებს), რომლითაც ხდება მომსახურებების იდენტიფიკაცია, აგრეთვე მომხმარებელთა პროფილების, მათი შეღწევისა და ავტორიზაციის მონაცემების იდენტიფიკაცია;
- Oracle Virtual Directory იგი არის LDAP (*Lightweight Directory Access Protocol*) ოქმის სერვისი, რომელიც უზრუნველყოფს აბსტრაქტულად, ცალსახად წარმოგვიდგინოს მრავალი მომსახურების გამწევი ორგანიზაციების საწარმოთა და მონაცემთა ბაზების სერვერების კატალოგები;
- Oracle Authentication Services for Operating Systems გამოიყენება Linux/Unix ოპ.ს. საკონფიგურაციოდ.

შედარებით დაწვრილებით (პრაქტიკული გამოყენების თვალსაზრისით) მონაცემთა ბაზის Oracle10 უსაფრთხოების პოლიტიკის არქიტექტურა შეიძლება წარმოვიდგინოთ შემდეგნაირად იხ. ნახ.2.2.2.

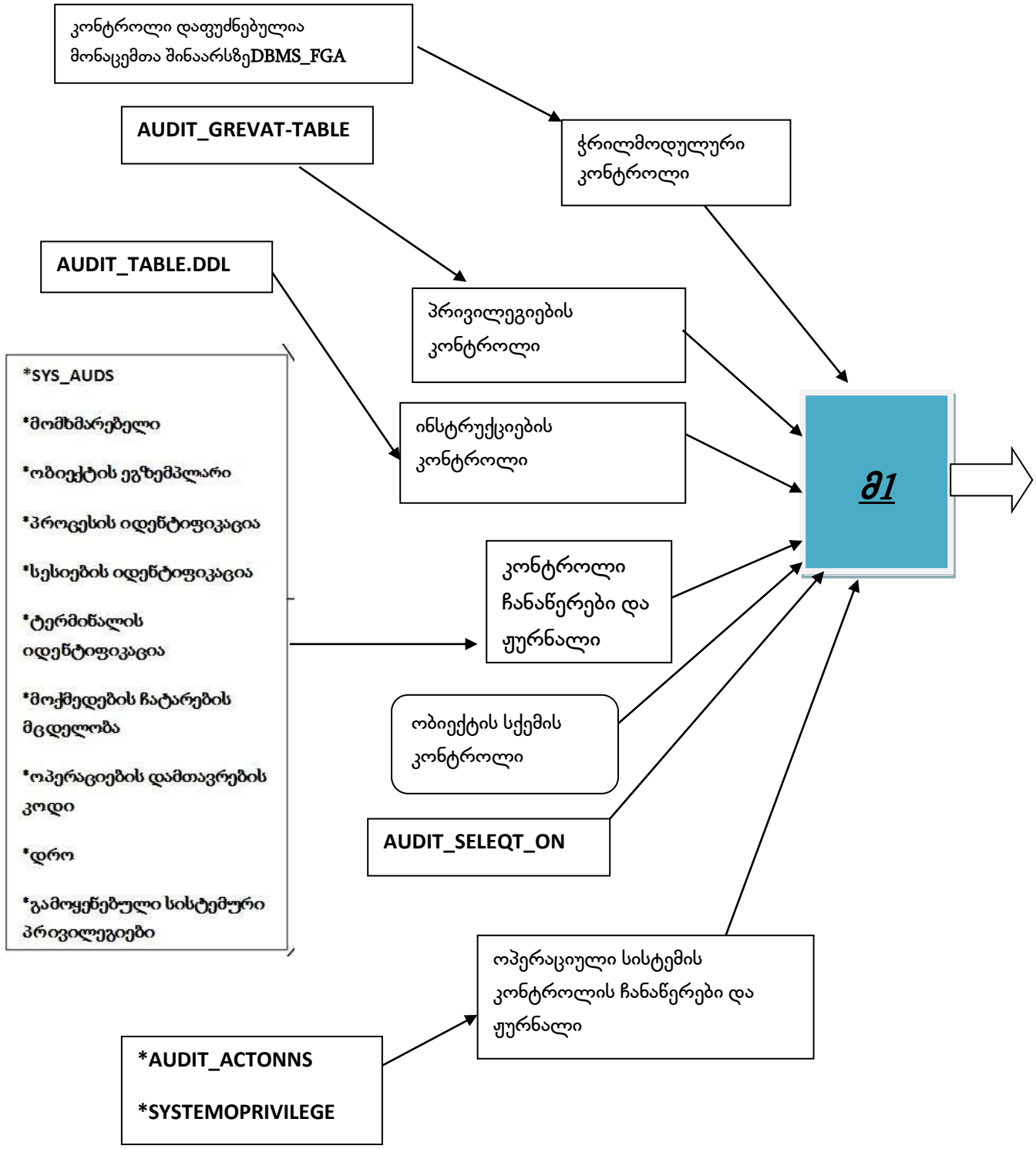
იგი შედგება 5 მოდულისაგან:

- *P1* კონტროლის პოლიტიკა ნახ. 2.2.3;
- *P2* მომხმარებელთა უსაფრთხოების პოლიტიკა ნახ.2.2.4;
- *P3* მონაცემთა ბაზების დაცვა ნახ. 2.2.5;
- *P4* პაროლების მართვის სისტემა ნახ. 2.2.6;
- *P5* სისტემური უსაფრთხოება ნახ. 2.2.7.



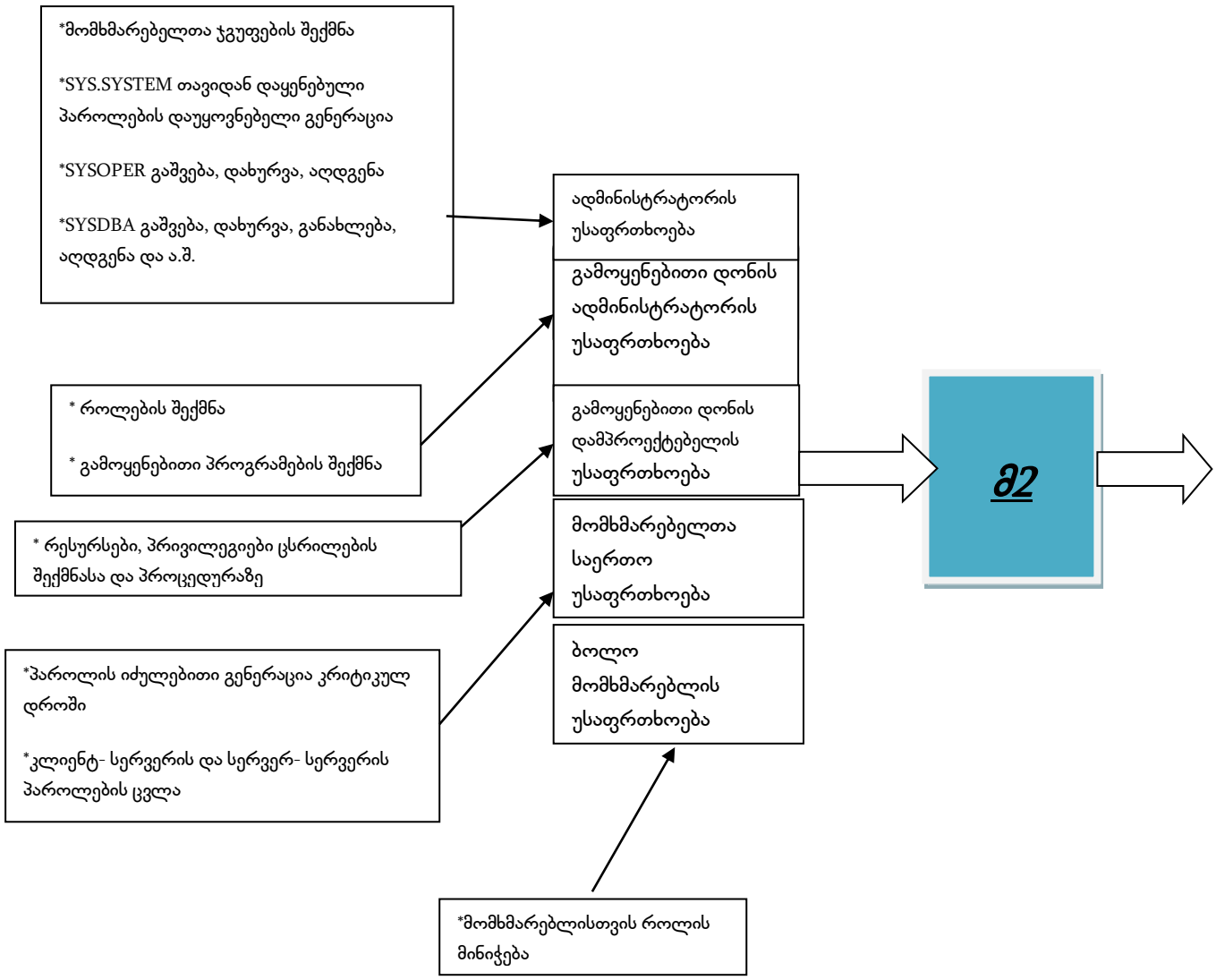
ნახ.2.2.2.

§ 2.2.1 კონტროლის პოლიტიკის მოდული მ1



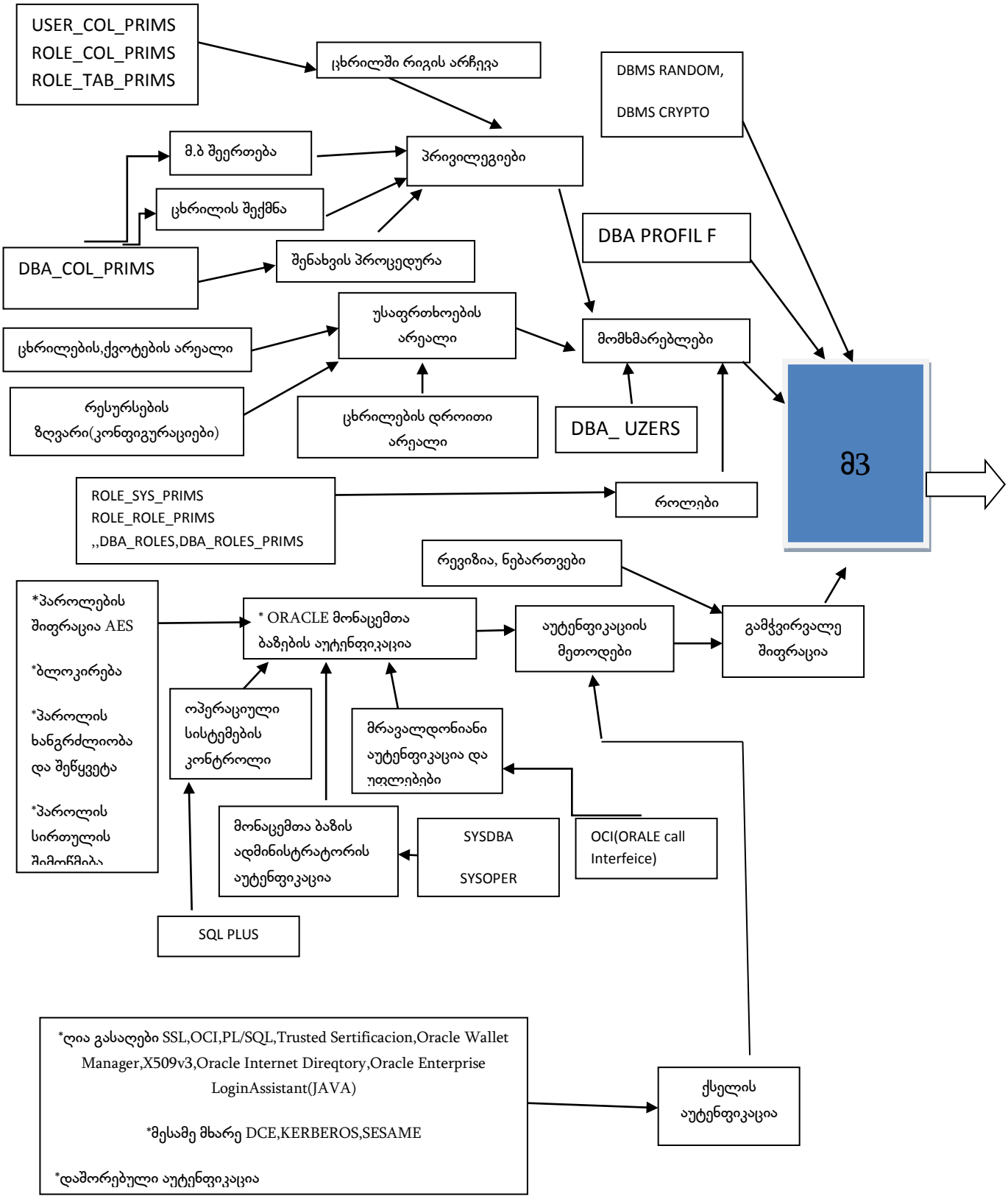
ნახ. 2.2.3.

§ 2.2.2. მომხმარებელთა უსაფრთხოების პოლიტიკა მ2


*მომხმარებელთა ჯგუფების შექმნა
*SYS.SYSTEM თავიდან დაყენებული პაროლების დაუყოვნებელი გენერაცია
*SYSOPER გაშვება, დახურვა, აღდგენა
*SYSDBA გაშვება, დახურვა, განახლება, აღდგენა და ა.შ.
ადმინისტრატორის უსაფრთხოება
გამოყენებითი დონის ადმინისტრატორის უსაფრთხოება
* როლების შექმნა
* გამოყენებითი პროგრამების შექმნა
გამოყენებითი დონის დამპროექტებელის უსაფრთხოება
* რესურსები, პრივილეგიები ცსრილების შექმნასა და პროცედურაზე
მომხმარებელთა საერთო უსაფრთხოება
*პაროლის იძულებითი გენერაცია კრიტიკულ დროში
*კლიენტ- სერვერის და სერვერ- სერვერის პაროლების ცვლა
ბოლო მომხმარებლის უსაფრთხოება
*მომხმარებლისთვის როლის მინიჭება
∂2

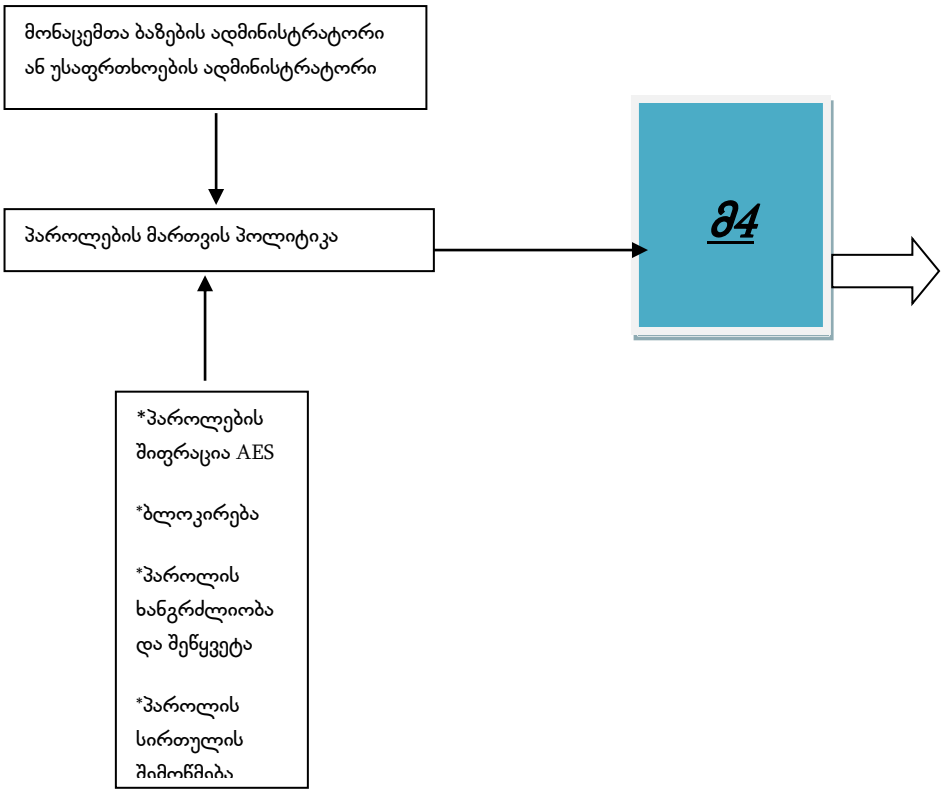
ნახ. 2.2.4.

§ 2.2.3 მონაცემთა ბაზების დაცვა მ3



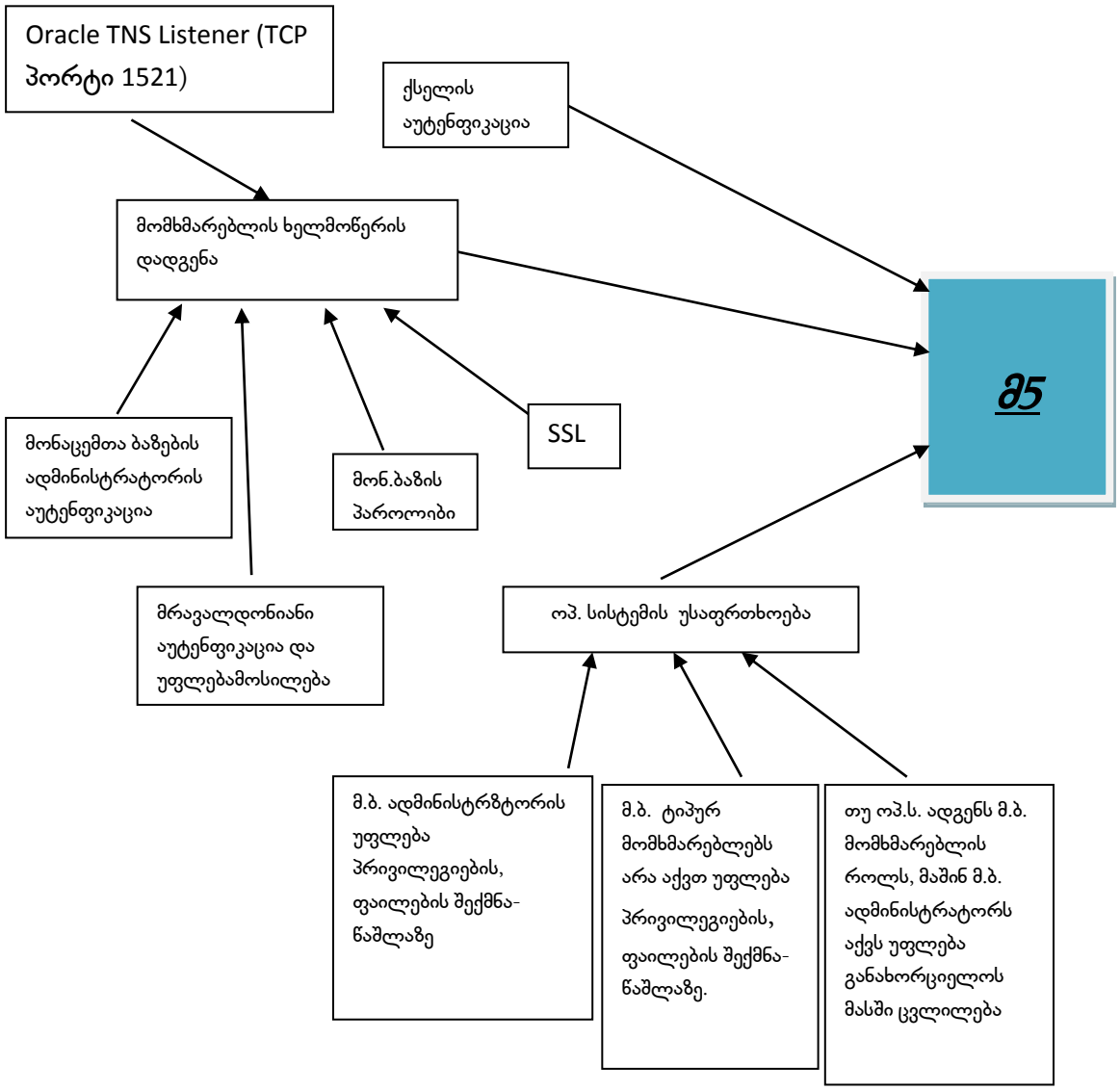
ნახ. 2.2.5.

§ 2.2.4 პაროლების მართვის *Ø4* სისტემა



ნახ. 2.2.6.

§ 2.2.5. სისტემური უსაფრთხოება *მ5*



ნახ .2.2.7.

თავი3. უსაფრთხოების უზრუნველყოფის მექანიზმები

§3.1 სიმეტრიული შიფრაცია

უნდა მივიღოთ, რომ ფიზიკური დონის გარდა, ინფორმაციის დაცვა დაფუძნებულია კრიპტოგრაფიაზე. ეს ტერმინი ბერძნულია და ნიშნავს „ფარულ წერილს“ [32]. მასში ძირითადი განსაზღვრებია შიფრი და კოდი. შიფრი ეს არის სიმბოლოთა ან ბიტთა თანამიმდევრობათა ისეთი გარდასახვა, რომელიც კავშირში არ არის შეტყობინების ლინგვისტურ სტრუქტურასთან. კოდი საპირისპიროდ ცვლის მთლიან სიტყვას სხვა სიტყვაზე ან სიმბოლოზე. კოდები თანამედროვეობაში არ გამოიყენება. თუმცა ისტორიულად ბევრი მაგალითია ცნობილი (მაგ. კოდი „ნავახო“ მეორე მსოფლიო ომის შემთხვევაში). მე-19 საუკუნის ფლამანდრიელმა, სამხედრო დამშიფვრელმა, ***აუგუსტო კერგკოფმა*** ჩამოაყალიბა თავისი პრინციპი, რომელიც საყოველთაოდ მიღებულია დღესაც: ***„შიფრაციის ალგორითმები საყოველთაოდ ხელმისაწვდომია, საიდუმლოა მხოლოდ გასაღებები“.***

კრიპტოგრაფიულ ტერმინოლოგიაში გამოვყოთ შემდეგი ცნებები (განმარტებები):

- საწყისი ტექსტი (plaintext ან cleartext);
- შიფრაციის პროცესი (encryption);
- დეშიფრაცია (decryption);
- დაშიფრული ტექსტი (chipertext).

განარჩევენ შიფრაციის ორ სახეს: სიმეტრიულს და ასიმეტრიულს. სიმეტრიულში დაშიფვრა-გაშიფვრა ხდება ერთი და იმავე გასაღებით, მაშინ იმავე ასიმეტრიულში დაშიფვრა ხორციელდება ერთი წყვილი „ღია“ და გაიშიფვრება „საიდუმლო“ გასაღებით. პირველი გახსნილია ყველა მომხმარებლისათვის, ხოლო მეორე დახურულია. პირველის ცოდნა არ იძლევა არავითარ საშუალებას დადგენილი (ნაპოვნი, გამოთვლილი) იქნეს მეორე.

სიმეტრიული შიფრაციაში მიღებულია მონაცემთა დაშიფვრის სამი სახე :

- ნაკადური;
- ბლოკური;
- ბლოკური უკუკავშირით.

საწყისი ტექსტის პირდაპირი დაშიფვრის სახე პრაქტიკაში მიღებული არ არის. არსებითი განსხვავებები, სიმეტრიულ შიფრაციისას, მდგომარეობს შემდეგში (ცხრ. 6):

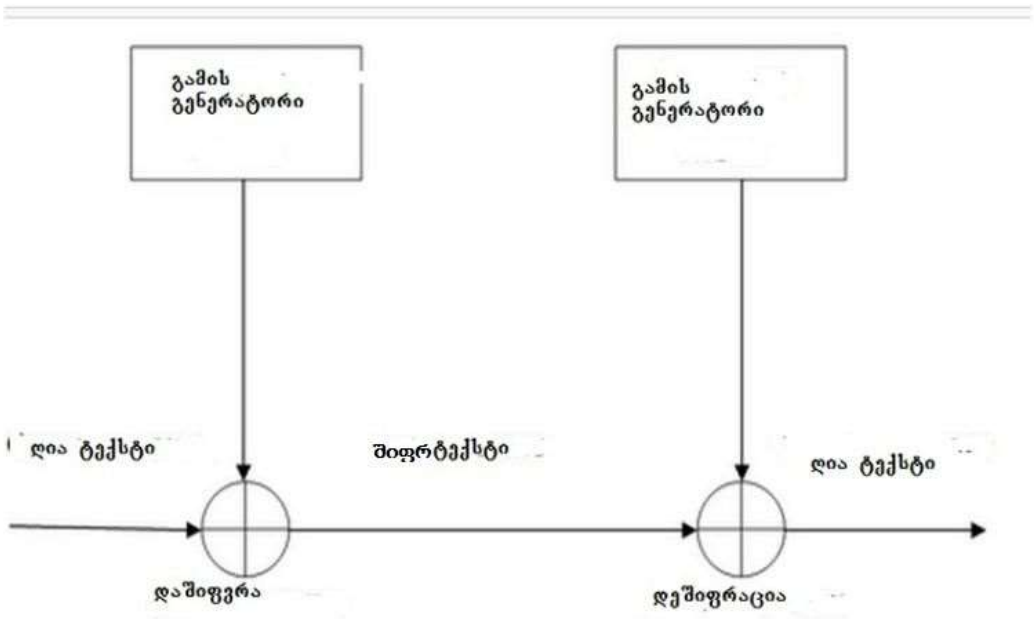
- ცხრ. 6.

კრიპტოსისტემის სახე	ოპერაციები ბიტებზე ან ბლოკებზე	წინამდებარე ნიშნებზე დამოკიდებუ-ლება	პოზიციური დამოკიდებულება	შიფრაციის სახე
ნაკადური შიფრაცია	ბიტი	არაა დამოკიდებული	დამოკიდებულია	სიმეტრიული
ბლოკური შიფრაცია	ბლოკი	არაა დამოკიდებული	არაა დამოკიდებული	სიმეტრიული ან არასიმეტრიული
ბლოკური შიფრაცია უკუ კავშირით	ბიტი ან ბლოკი	დამოკიდებული	არაა დამოკიდებული	სიმეტრიული

- დაშიფვრის ოპერაციები ხორციელდება ან ცალკეულ ინფორმაციულ ბიტებზე ან გარკვეული რაოდენობის ბიტებზე ანუ ბიტების ბლოკებზე;
- მიმდინარე დაშიფვრის პროცესის დამოკიდებულება ან არდამოკიდებულება წინა დაშიფვრულ ინფორმაციის შედეგზე;
- ცალკეული ნიშნების შიფრაციის დამოკიდებულება ტექსტში მათ ადგილმდებარეობაზე. მაგალითად ნაკადური შიფრაციის სახეში გათვალისწინებულია ტექსტის ნიშნის მდგომარეობა, სხვებში არა (ზოგადად ამას ეწოდება პოზიციური დამოკიდებულება ან შიფრის დამოუკიდებლობა);
- სიმეტრიული (ერთ გასაღებიანი) და ასიმეტრიული (ორ გასაღებიანი) შიფრაციის სახეების ძირითადი განსხვავება ისაა, რომ ასიმეტრიული შიფრაციის დროს ე.წ. ღია გასაღების ცოდნა არაა საკმარისი შიფრის გასატეხად ანუ ე.წ. საიდუმლო გასაღების (მეორე გასაღები) დასადგენად და შესაბამისად დაშიფრული ტექსტის გასახსნელად.

§ 3.1.1 ნაკადური შიფრაცია

ნაკადური შიფრაცია. 1965 წ. ნორვეგიელმა კრიპტოგრაფმა *ერნსტ სილმერმა* ჩამოაყალიბა თეორია რეგისტრების თანამიმდევრობითი წანაცვლების შესახებ, რომელიც შემდეგ განავრცო ამერიკის ნაციონალური სააგენტოს მათემატიკოსმა *სოლომონ გოლომბამ.* აქ გასაღებს (მას ხშირად უწოდებენ ვერნამის გასაღებს) აქვს თვით საწყისი ტექსტის სიგრძე და მას იყენებენ როგორც გამას (ნაკადს) ნახ. 3.1.



ნახ. 3.1.

ნაკადური შიფრაციის არსი მდგომარეობს შემდეგში. საწყისი, ღია, ტექსტის თვითოეული ბიტი იკრიბება ორის მოდულით რაღაც ფსევდო შემთხვევით თანამიმდევრობის ბიტებთან, რომელიც გამომუშავდება გამა გენერატორში.

თუ m1,m2,…..mi ღია ტექსტის ბიტების ნაკადია, ხოლო k1,k2….ki გასაღების (გამის) ბიტების ნაკადია, მაშინ შიფრ ტექსტი c1,c2…ci მიიღება ამ ორი ნაკადის ბიტებზე „ან გამორიცხვის“ ოპერაციით XOR, ხოლო დაშიფრაცია უკუ პროცესით.

ეს მეთოდი ძალიან სწრაფია, მარტივად რეალიზებადია და არ ახასიათებს შეცდომების ტირაჟირების თვისება. უარყოფითი მხარეა სინქრონიზაციის საკითხი, რომელიც თხოულობს შეტყობინების თავსართის გადაცემამდე სინქრონიზაციის ინფორმაციის გადაცემას. ხშირად სხვა და სხვა ინფორმაციის ერთდროულად გადაცემისას, როცა ვიყენებთ ერთი და იმავე ფსევდო შემთხვევით თანამიმდევრობებს, კრიპტო მდგრადობის შესანარჩუნებლად, ვიყენებთ დამატებით, შემთხვევით შერჩეულ შეტყობინების გასაღებს, რომელიც გადაიცემა შეტყობინების დასაწყისში და რომელიც გამოიყენება შიფრაციის გასაღების მოდიფიკაციისთვის. ეს იმას ნიშნავს, რომ ერთდროული შეტყობინებები იშიფრება სხვა და სხვა თანამიმდევრობებით.

ნაკადური შიფრაცია გამოიყენება ციფრულ სახეში გარდაქმნილი ხმოვანი სიგნალების და ციფრული მონაცემების დასაშიფრად. ყველაზე ცნობილი ნაკადური შიფრებია **A3, A5, A8, MUGI, PIKE, RC4, SEAL,ORION.**

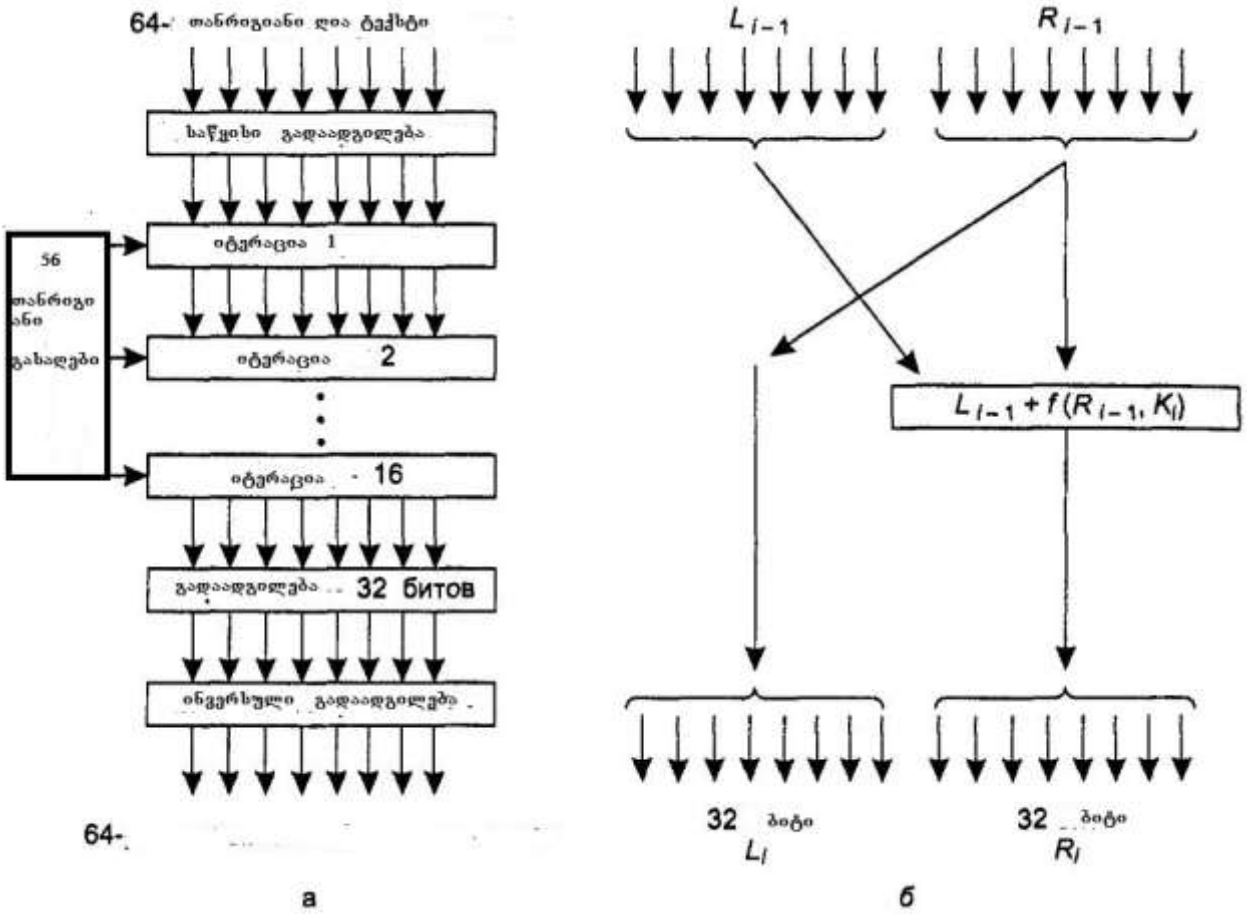
მაგალითი. თუ შემავალი ინფორმაციაა A = 1011 0011 და გამა ფუნქციაა B =0110 1101, მაშინ შიფრაციის შემდეგ ვიღებთ C = 11011110, რომელიც მიიღება C = A XOR B ლოგიკური ოპერაციით. ეს დაშიფრული ტექსტი C მიმღებ მხარეს გადის იმავე ლოგიკური ოპერაციით XOR უკუ გარდაქმნას, რის შედეგადაც ვიღებთ საწყის ტექსტს A.

C XOR B = (11011110) xor (0110 1101) =1011 0011 =A.

§ 3.1.2 ბლოკური შიფრაცია

ბლოკური შიფრაციის დროს საწყისი ტექსტი წინდაწინ იყოფა ფიქსირებული სიგრძის ბლოკებად, ხოლო შემდეგ ხდება მათი დაშიფვრა იმავე სიგრძის დაშიფრულ ტექსტად შერჩეული გასაღების მიხედვით [17,18,19]. ძირითადია, რომ ტექსტის ნებისმიერი ბიტი დამოკიდებულია საწყისი ტექსტის მთლიან ბლოკზე. გამორიცხულია დაშიფრული ბლოკების ერთმანეთთან დამთხვევა.

ბლოკური შიფრაციის კლასიკური მაგალითია მონაცემთა შიფრაციის სტანდარტი DES (Date Encryption Standart). ამ სტანდარტის საწყისი ვერსია დამუშავებული იყო IBM მიერ და იწოდებოდა „***ლუციფერი***“. განვიხილოთ ამ სტანდარტის ფუნქციონირების ბლოკ-სქემა ნახ. 3.2 ა) და ბ). ეს ალგორითმი შედგება 19 ეტაპისგან და იყენებს 64-თანრიგიან გასაღებს. აქედან 56 თანრიგი ნიშნადია, ხოლო 8 ბიტი ლუწობაზე კონტროლს ახორციელებს. პირველ ეტაპზე ხდება საწყისი 64-ბიტიანი ტექსტის დამოუკიდებელი გადადგილებები. იგივე ხორციელდება, ოღონდ საპირისპიროდ, მე-19 ეტაპზე. მე-18 ეტაპზე ხდება მარცხენა და მარჯვენა 32 ბიტების ადგილების ურთიერთ შეცვლა. რაც შეეხება 2 – 17 ეტაპებს, მათი რეალიზაცია გადმოცემულია ნახ. 3.2. ბ)-ზე.



ნახ. 3.2.

პროცესის უფრო დაწვრილებით აღსაწერად გამოვიყენოთ ნახ.3.2. ა) და ცხრ.7, სადაც გადმოცემულია გადაადგილების მატრიცები. საწყის ეტაპზე შემავალი 64-ბიტიანი ბლოკი ,გარდაიქმნება ახალ ბლოკად ცხრ. 7 (საწყისი გადაადგილების მატრიცა) მიხედვით. ე.ი. შემავალი ბლოკის 58-ე ბიტი ხდება 1, 50-ე ბიტი 2, 42-ე ბიტი 3 და ა.შ. მიღებული ბლოკი იყოფა ორად, მარცხენად L ანუ უფროს თანრიგებად და მარჯვენა R უმცროს თანრიგებიან ბლოკებად (თითოეული 32-ბიტიანია). შემდგომში ხდება 16-ბიჯიანი ერთნაირი იტერაციების ციკლი, სადაც შემდეგი იტერაციისათვის მარჯვენა ბლოკი ხდება მარცხენა, ხოლო მარჯვენა ბლოკი მიიღება გამოთვლით

$$L_i = R_{i-1} \quad i=1,2,3\ldots16$$

$$R_0 =(L_0 \oplus f (R_{i-1} K_i)) \; i=1,2,\ldots16$$

აქ f შიფრაციის ფუნქციაა, რომელიც მიიღება წინა იტერაციაზე მიღებული მარჯვენა ბლოკის და 48-ბიტიანი გასაღებით (R_{i-1} K_i). ბოლო იტერაციაზე მიღებული ბლოკების გადაადგილება ხდება მე-7 ცხრილის მეორე ნაწილის მიხედვით. რის შედეგადაც ვიღებთ შიფრტექსტს.

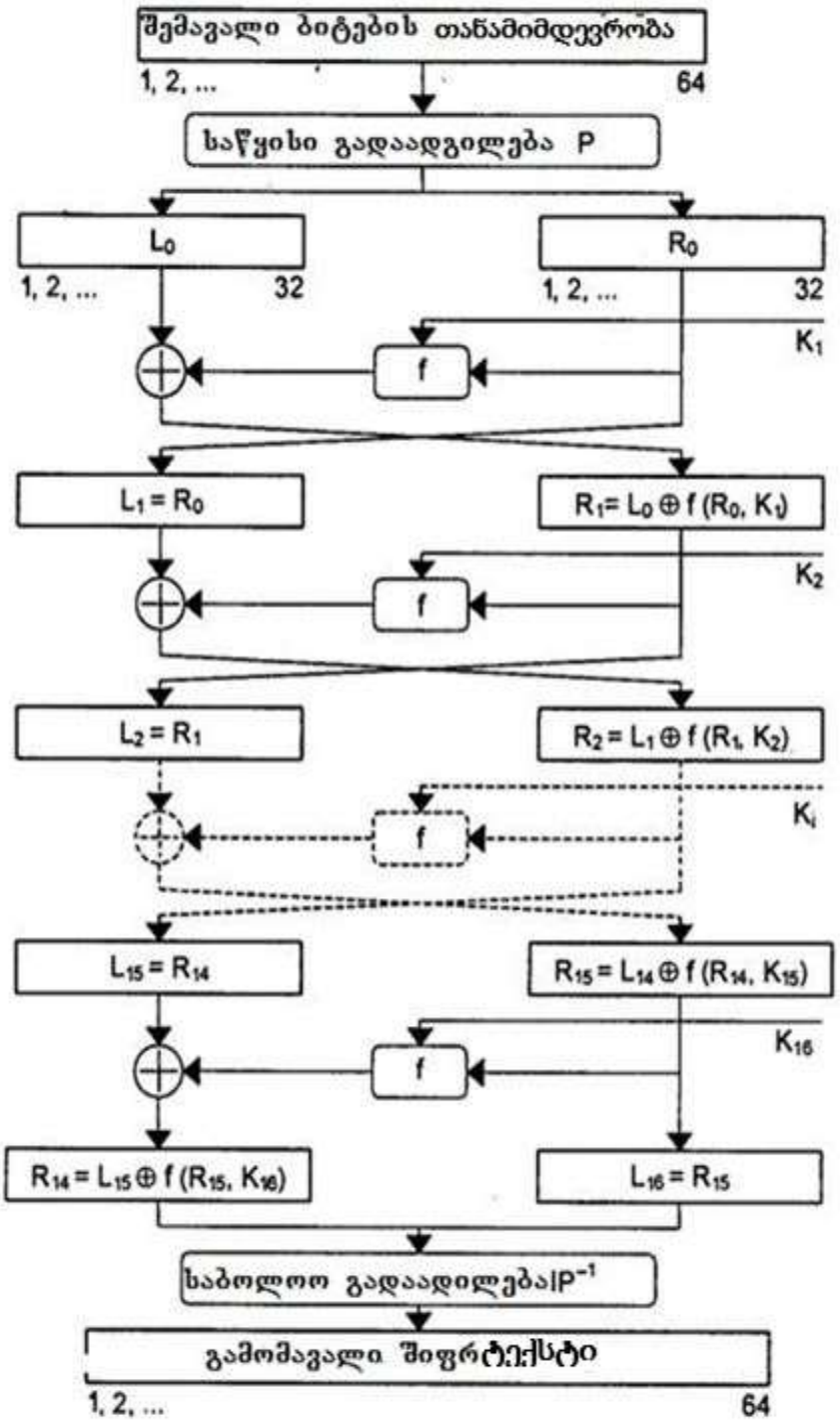
ცხრ.7.

საწყისი გადაადგილების მატრიცა IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

შპშ გადაადგილების მატრიცა IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25



ნახ. 3.2 ა)

შიფრტექსტის მიღების შემდეგ ბიტების პოზიციების აღდგენა ხდება ცხრ.8 გადმოცემული თანამიმდევრობით:

ცხრ. 8

მატრიცის ელემენტებს შორის კავშირი

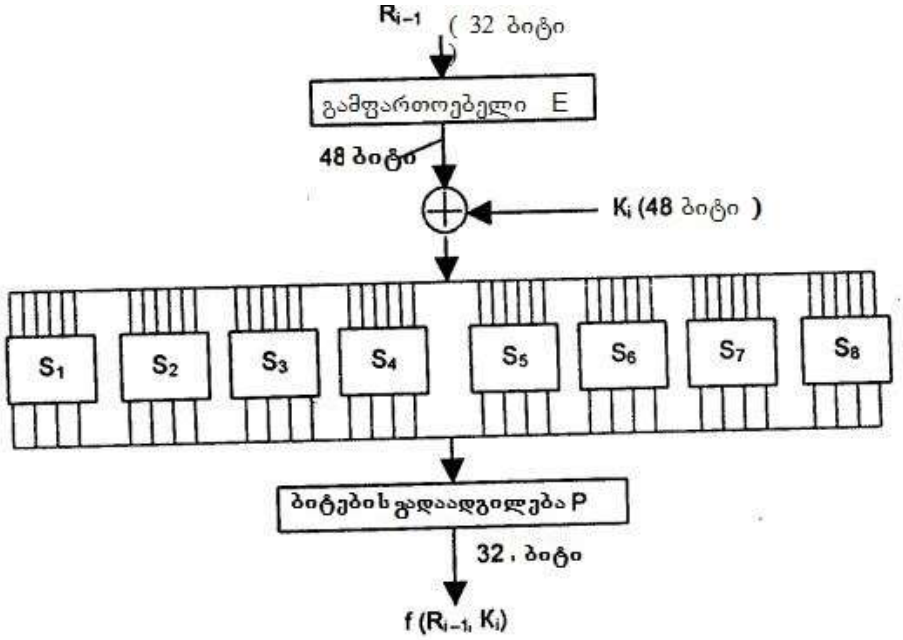
მატრიცის ელემენტი IP^{-1}	მატრიცის ელემენტი IP
40	01
8	02
48	03
16	04
56	05
...	...

გაშიფვრა ხორციელდება უკუ თანამიმდევრობით, პროცესი იწყება IP^{-1} მატრიცის მიხედვით, ბიტების გადაადგილებებით, ხოლო შემდეგ იტერაციების ციკლით უკუ თანამიმდევრობით, ვიდრე შიფრაციის დროს. ე.ი. იწყება R_{16} L_{16}-დან. გამოითვლება ფუნქცია

$$R_{i-1} = L_i \quad i=16,15,...11$$

$$L_{i-1}=(R_i \oplus f(L_i K_i)) \quad i=16,15,...,1$$

შესაბამისად დეკოდირების პირველ იტერაციაზე გამოიყენება მე-16 გასაღები, მე-2 იტერაციაზე მე-15 და ა.შ. პროცესში გამოყენებული ფუნქციები განისაზღვრება შენდეგნაირად: **f შიფრაციის** ფუნქციის მუშაობის ალგორითმი წარმოდგენილია ნახ. 3.2. ბ).



ნახ. 3.2. ბ)

f შიფრაციის ფუნქციის გამოსათვლელად საჭიროა:

ვიანგარიშოთ **E ფუნქცია**, რომელიც მიიღება საწყისი 32-ბიტიანი R_{i-1} ფუნქციის გაფართოებით 48 ბიტამდე. გარდავქმნით მიღებულ 48-ბიტიან E ფუნქციას, 8 S_1 , S_2….S_8 ფუნქციებად, თითოეულს 6-ბიტიანს. ვიღებთ მიღებული 32-ბიტიანი თანამიმდევრობის უკუგადაადგილებით **P ფუნქციას.**

E ფუნქციის შექმნის ალგორითმი რეალიზებულია ცხრ.9 მიხედვით:

ცხრ. 9

გაფართოების ფუნქცია					**E**
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

$$E(R_{i-1}) \oplus K_i = B_1B_2 \ldots B_8.$$

აქ მიღებული E (R_{i-1}) ფუნქცია იჯამება 2-ის მოდულით გასაღების მიმდინარე მნიშვნელობასთან და მიღებული 48 ბიტი იყოფა 6-ბიტიან ბლოკებად B_1 , B_2 …. B_8. შემდგომში თითოეული B_1 , B_2 …. B_8 ბლოკი გამოიყენება S_1 , S_2….S_8 ფუნქცია-მატრიცების ელემენტების ნომრების მისაღებად (ალგორითმის სირთულის გამო ტექსტში იგი გადმოცემული არ არის), სადაც თვითოეული 4-ბიტიანია. მიღების ალგორითმი გადმოცემულია ცხრ.10-ში.

ცხრ. 10

გარდაქმნის ფუნქცია $S_1, S_2, ..., S_8$

		სვეტის ნომერი																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
სტრიქონის ნომერი	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
	2	4	1	4	8	13	6	2	11	15	12	9	7	3	10	5	0	
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6	
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

ვიღებთ 8 **S** ფუნქციას $S_1(B_1)$,$S_2(B_2)$ ….$S_8(B_8)$ სულ 32-ბიტიანს, თავის მხრივ მიღებული თანამიმდევრობა გარდაიქმნება ცხრ.11-ში მითითებული წესით.

ცხრ. 11

ბიტების გადაადგილების P ფუნქცია

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

საბოლოოდ ვიღებთ დაშიფვრის ფუნქციის მნიშვნელობას

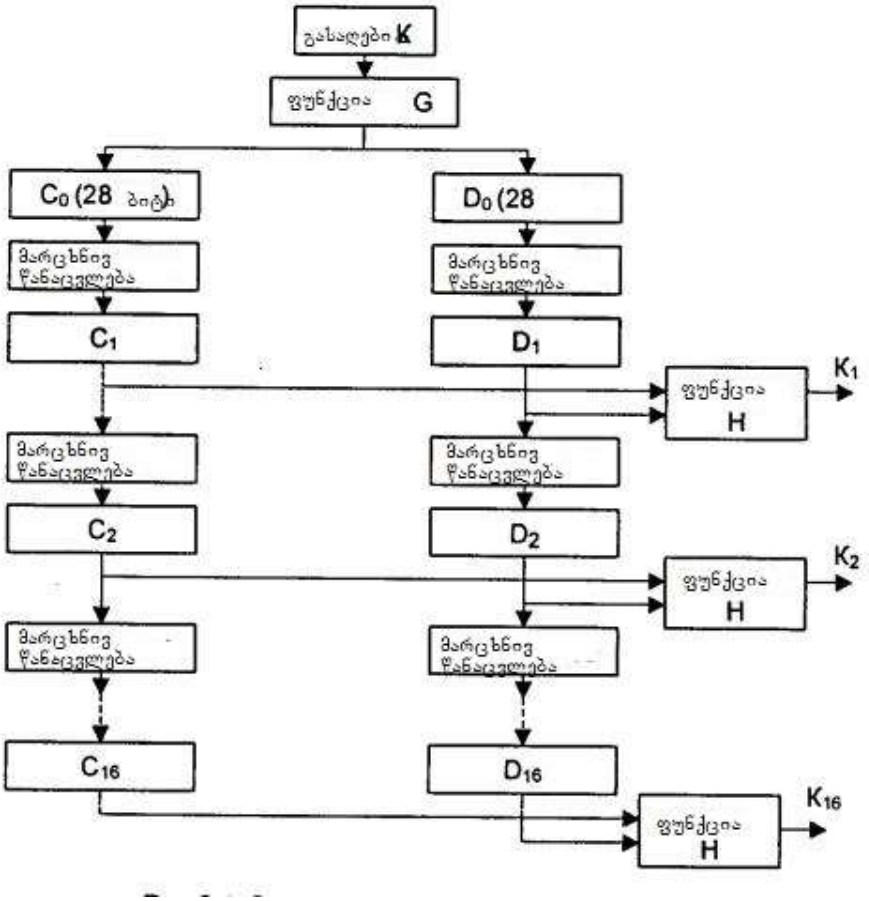
$$f(R_{i-1}K_i) = P(S_1(B_1), S_2(B_2) \ldots S_8(B_8))$$

ამ იტერაციების დროს, როგორც ავღნიშნეთ, გამოიყენება გასაღების ახალ-ახალი მნიშვნელობები. ისინი მიიღება ცხრ.12 და ნახ. 3.2.გ) მითითებული ალგორითმის მიხედვით.

ცხრ .12

გასაღების პირველადი მომზადების G ფუნქცია
(1 გადაადგილების ფუნქცია)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4



ნახ. 3.2. გ)

აქ ნახ.3.2. გ), გასაღების ახალი მნიშვნელობა K_i(სიგრძით 48 ბიტი) მიიღება საწყისი K (64 ბიტი ცხრ.6) მნიშვნელობიდან, სადაც ლუწობაზე კონტროლი განთავსებულია 8; 16, 24, 32, 40,

48, 56 და 64 ბიტებში. ამისათვის გამოიყენება ფუნქცია G, რომელიც ფუნქციონირებს ცხრ.9-ს მიხედვით. ცხრ.9 გაყოფილია ორ ნაწილად C_0 და D_0, თვითოეული 28 ბიტი. ამ ცხრილის პირველი ნაწილი განსაზღვრავს როგორი თანამიმდევრობით იქმნება C_0: პირველი ბიტია გასაღების 57 ბიტი, მეორე 49 და ა.შ. ხოლო ბოლო 44 და 36 ბიტი. ამ ცხრილის მეორე ნაწილი განსაზღვრავს როგორი თანამიმდევრობით იქმნება D_0 : პირველი ბიტია გასაღების 63 ბიტი, მეორე 55 და ა.შ. ხოლო ბოლო 12 და 4. ყურადღება მისაქცევია, რომ პროცესში არ მონაწილეობს საკონტროლო თანრიგები.

C_0 და D_0 გამოთვლის შემდეგ, ანალოგიურად გამოითვლება C_1 და D_1, რომლებიც მიიღება მათი ციკლურად გადაადგილებით მარცხნივ 1 ან 2 პოზიციით ცხრ.13.

ცხრ. 13

იტერაციის ნომერი	მარცხნივ წანაცვლების S_i ბიტი	იტერაციის ნომერი	*მარცხნივ წანაცვლების S_i ბიტი*
1	1	9	*1*
2	1	10	*2*
3	2	11	*2*
4	2	12	*2*
5	2	13	*2*
6	2	14	*2*
7	2	15	*2*
8	2	16	*1*

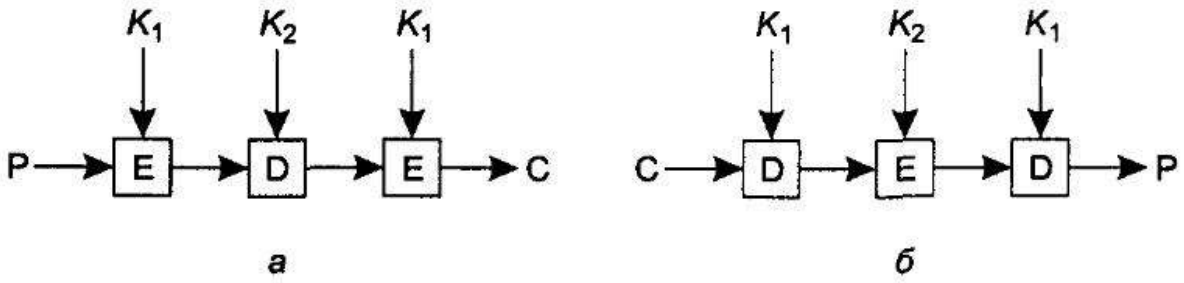
მიღებული გასაღებიდან, იტერაციების ციკლებისათვის, ცხრ.14 საშუალებით მიიღება K_i გასაღების (i ციკლი) ბიტები შემდეგი თანამიმდევრობით : პირველი ბიტი -14, მეორე ბიტი-17, მესამე ბიტი-11 და ა.შ.

ცხრ .14

გასაღების საბოლოო დამუშავების H ფუნქცია
მე-2 გადანაცვლებული ამორჩევა

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

მიუხედავად წარმატებისა, ამ სტანდარტმა 1979 წ-ს ამოწურა თავისი შესაძლებლობა და IBM დააანონსა ახალი ე. წ. სამმაგი **3 DES** შიფრაციის სტანდარტი **8732**.იხ ნახ. 3.3.



ნახ .3.3.

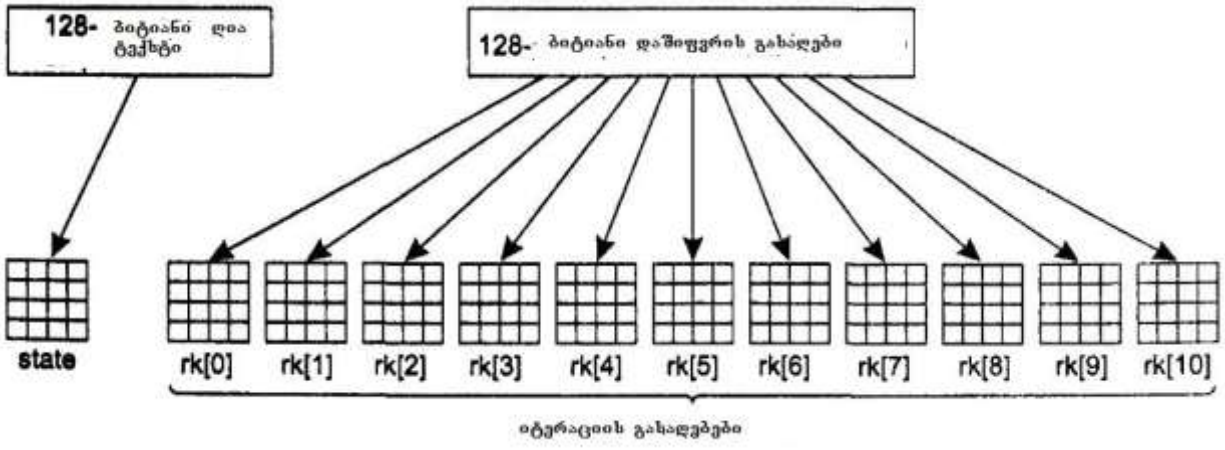
აქ შიფრაცია ხორციელდება (ნახ.3.3 ა) სამ ეტაპად. პირველზე ხდება საწყისი ტექსტის შიფრაცია k1 გასაღებით, მეორე ეტაპზე მიღებული დაშიფრული ტექსტის დეშიფრაცია k2 გასაღებით, ხოლო მესამე ეტაპზე ისევ შიფრაცია k1 გასაღებით. ნახ.3.3. ბ)-ზე ნაჩვენებია დეშიფრაციის პროცესი. წარმოიშობა 2 კითხვა:

- რატომ ვიყენებთ ორ გასაღებს 112-ბიტიანს და არა 158-ბიტიანს (3 გასაღები);
- რატომ არ ვიყენებთ EDE რეჟიმს და არა EEE.

პირველზე ჩაითვალა, რომ 112-ბიტიანის გამოყენება უფრო ეკონომიურია ვიდრე 158 კრიპტომდგრადობის შესამჩნევი გაუარესების გარეშე.

მეორე გადაწყვეტა საშუალებას აძლევს მომხმარებლებს, ზედმეტი დანახარჯების გარეშე გამოიყენოს არსებული DES და 3DES.

ამ სტანდარტის შემდეგი გაუმჯობესებული სტადარტი იყო AES (Advanced Encryption Standard) გასაღებთა სიგრძით 128, 192 და 256 ბიტი. 2000 წ. იგი **Rijndael** (ავტორები ჯონ დაიმენი John Daemen და ვინსენტ რიჯმენი Vincent Rijmen) სახელით გახდა ცნობილი და მიღებული იქნა სახელმწიფო სტანდარტად **FIPS 197** ნახ. 3.4 .



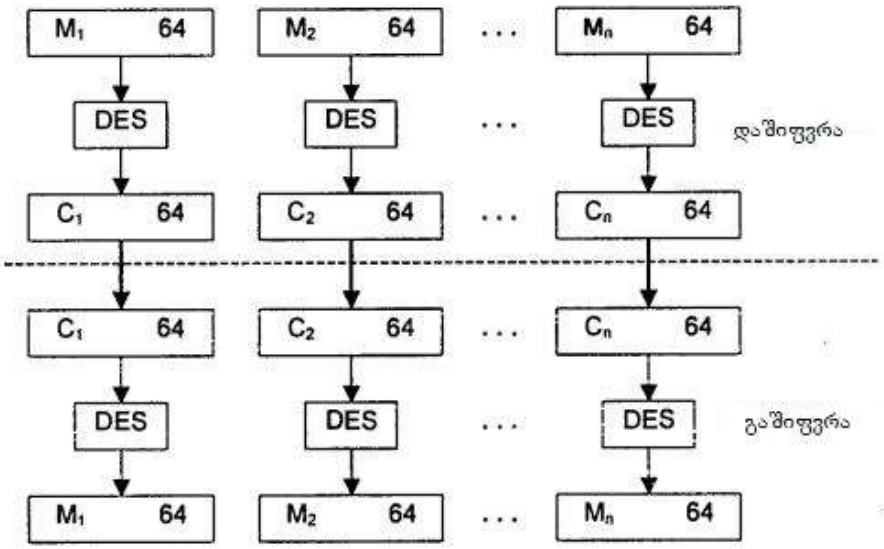
ნახ. 3.4

აქ საწყისი ტექსტი წარმოდგენილია 16-ბაიტიან მასივში State, ხოლო rk(i) მასივებია დაშიფვრის პროცესის იტერაციებისათვის. პროცესი იწყება rk(0) მასივის შექმნით. ამისთვის State მასივის პირველი 4 ბაიტი ჩაიწერება rk(0) მასივის 0 სვეტში, შემდეგი 1 სვეტში და ა.შ. შემდეგი 10 იტერაციისთვის ხდება ინდივიდუალური გასაღებთა გენერირება ძალიან რთული ალგორითმით. ე.ი. ყველა i იტერაციაზე იქმნება rk(i) მასივის მისაღებად ინდივიდუალური გასაღები. პროცესის დაწყებამდე ხდება State მასივისა და rk(0) მასივის თანრიგობრივი აჯამვა ორის მოდულით. შედეგი ხვდება State მასივიში. შემდეგ იტერაციებზე ხდება მონაცემების „გაბნევა“ რთული ალგორითმებით. დაშიფვრის სწრაფმოქმედებაა 700 მბ/წმ, პროცესორის 2 გგჰ სიხშირისას.

ბლოკური შიფრაციისას, განარჩევენ შიფრაციის შემდეგ რეჟიმს:

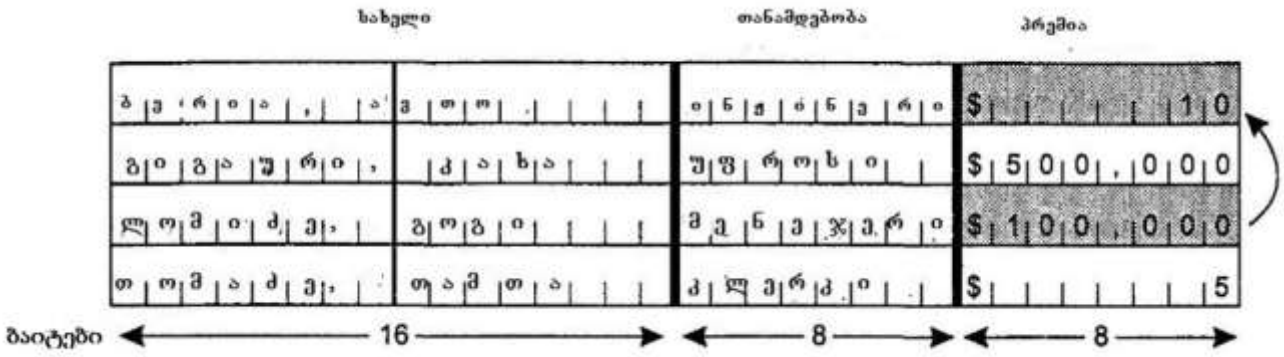
- **პირდაპირი შიფრაციის რეჟიმი** (შიფრაცია კოდების ელექტრონული წიგნის გამოყენებით) ECB (Electronic Code Book);
- **შიფრაცია, დაშიფრული ბლოკების ურთიერთ ჩავლებით** (შეჭიდებით) CBC (Cipher Block Chaining);
- **შიფრაცია შიფრტექსტით უკუკავშირით** CFB (Cipher Feeback);
- **შიფრაცია გამოსავალთან უკუკავშირით** OFB (Output Feedback).

პირდაპირი შიფრაციის რეჟიმი ECB ნახ.3.5 ხასიათთება შემდეგი დადებითი თვისებით. ნებისმიერი, თუნდაც უმნიშვნელო, ცვლილება დაშიფრულ ტექსტში იწვევს მნიშვნელოვან დაუპროგნოზებელ ცვლილებებს გაშიფრისას (ე.ი. გაშიფრული და საწყისი ტექსტები მთლიანად არ დაემთხვევა ერთმანეთს, იგი ხდება წაუკითხავი) და პირიქით. მაგრამ მას ახასიათებს მნიშვნელოვანი ნაკლოვანებები, რის გამოც მისი გამოყენება არაა რეკომენდებულია განსაკუთრებით დიდი სიდიდის საწყისი ტექსტის შემთხვევაში. პირველი მდგომარეობს იმაში, რომ იმის გამო, რომ საწყისი ტექსტის ერთნაირი საწყისი ბლოკები იშიფრება ერთნაირად, კრიპტოანალიტიკოსს შესაბამისი „ლექსიკონის“ არსებობის შემთხვევაში - არ გაუჭირდება საწყის ტექსტზე შეხედულების ჩამოყალიბება. ***მეორე*** მდგომარეობს შეცდომების ტირაჟირებაში. აქ ნაგულისხმებია, რომ მიღებულ შიფრტექსტში თუნდაც ერთი ბიტის დამახინჯება იწვევს მთლიანად საწყისი ტექსტის დაკარგვას. მიუხედავად ამ ნაკლოვანებისა, ასეთი ბლოკურ შიფრაციის სახე ხშირად გამოიყენება ფინანსურ ორგანიზაციებში, სადაც საწყისი ტექსტი ხშირად არ აღემატება ორ ბლოკს. ამ დროს მოთხოვნაა შიფრაციის გასაღებების ხშირი ცვლა.



ნახ. 3.5.

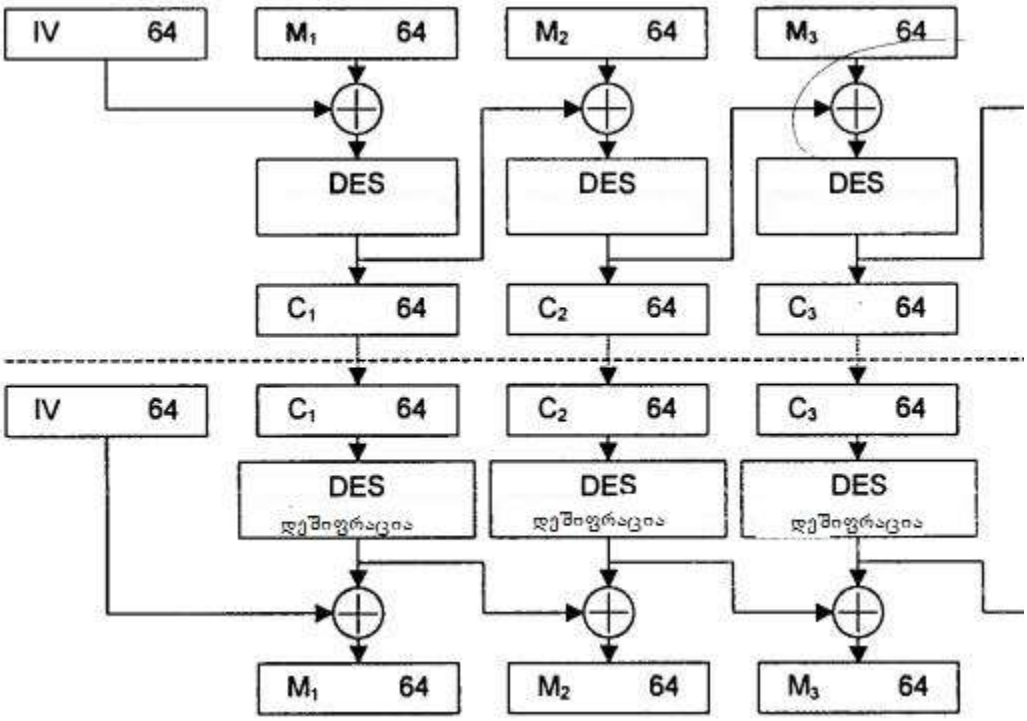
განვიხილოთ ამ შიფრაციის კომპრომეტირების მაგალითი ნახ. 3.6.



ნახ. 3.6.

მაგალითად, ავთო ბერიამ იცის, რომ მას უფროსი უპირებს, სხვებთან შედარებით, ძალიან მცირე თანხით წახალისებას, არადა მას ძალიან სჭირდება ფული. თუ მას ხელი მიუწვდა უკვე დაშიფრულ, ბანკში ჯერ არ გაგზავნილ ტექსტთან, იგი გადააწერს თავის დაშიფრულ მეოთხე ბლოკს (რიგითი ნომერი ცნობილია, რადგან ის ძალიან ხშირად ემთხვევა საშტატოს) ვთქვათ, მესამე პოზიციაში კოპიით აღებულ მონაცემებს მეოთხე ბლოკიდან. გარეგნულად დაშიფრულ ტექსტს არაფერი შეეტყობა, რის შედეგად ავთო მიიღებს მომატებულ პრემიას.

შიფრაცია, დაშიფრული ბლოკების ურთიერთ ჩავლებით (შეჭიდებით) ნახ. 3.7.



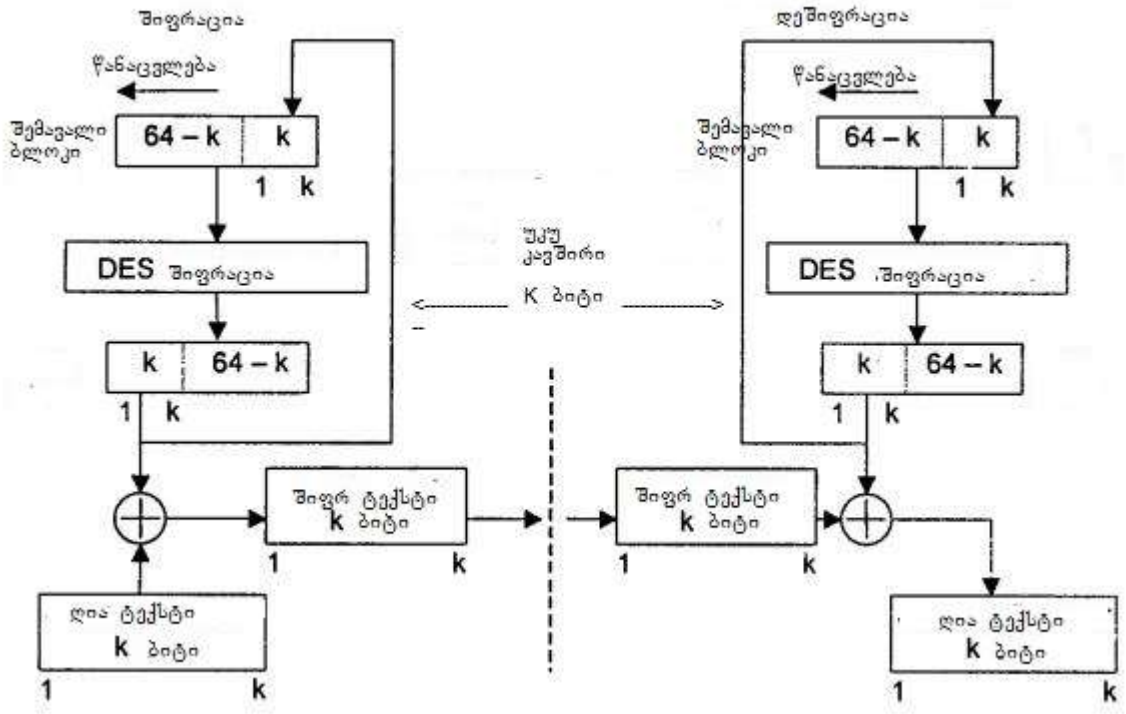
ნახ. 3.7.

საწყისი M ტექსტი იყოფა 64-ბიტიან ბლოკებად ისე, რომ $M=M_1 + M_2 +\ldots+ M_n$. პირველ ეტაპზე, საინიციალიზაციო ვექტორით IV (საიდუმლო ინფორმაციაა), ხდება მოდ 2-ით შეკრება M_1 და IV ბლოკების. შემდგომში ხდება შედეგის დაშიფვრა ცნობილი ალგორითმით. მიღებულ შედეგზე, ანალოგიურად ადრე განხილულისა, ხორციელდება ტექსტის შემდეგ ბლოკთან 2 მოდულით შეკრება და შიფრაცია. პროცესი გრძელდება მანამდე, სანამ საწყისი ტექსტის ყველა ბლოკი არ იქნება დაშიფრული. დეშიფრაცია ხორციელდება ანალოგიურად.

§ 3.1.3 ბლოკური შიფრაცია უკუკავშირით

უკუკავშირიან დაშიფვრის ბლოკური შიფრაციის სახის დადებითი მხარეებია მათი შესაძლებლობა დაშიფრონ მონაცემთა დიდი მასივები, აღმოაჩინონ შეღწევის მცდელობები და გამოყენებული იქნენ აუტენფიკაციისათვის.

განვიხილოთ ბლოკური შიფრაციის სახე ***„შიფრით უკუკავშირიანი“*** (ნახ. 3.8.).



ნახ. 3.8.

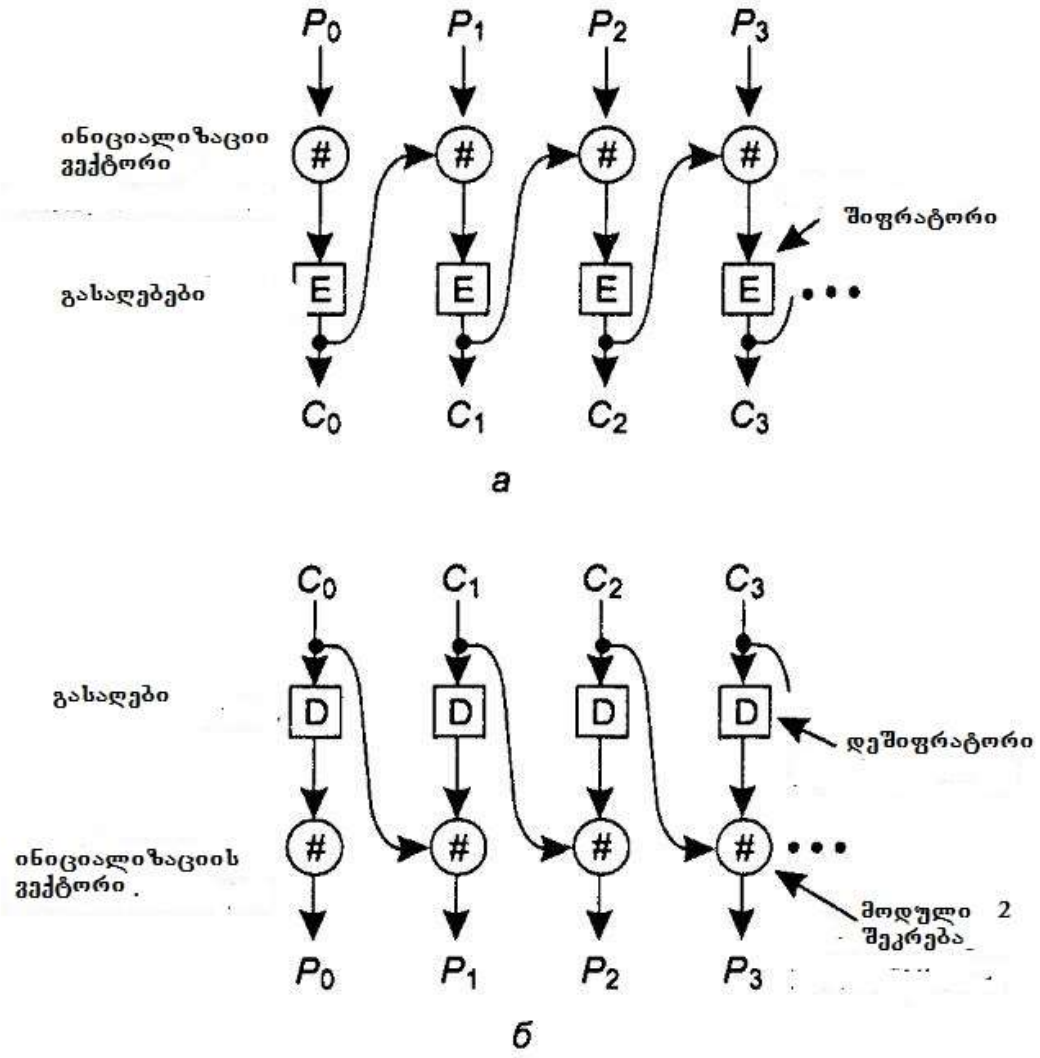
ამ რეჟიმის დროს ბლოკის სიგრძე შეიძლება განსხვავდებოდეს 64 ბიტისაგან. საწყისი დასაშიფრი (გასაშიფრი) ტექსტი იკითხება 1-64-ბიტიანი ბლოკების სახით. შემავალი ბლოკი (64 ბიტიანი წანაცვლების რეგისტრი) საწყის ეტაპზე შეიცავს მხოლოდ IV ვექტორს. საწყისი ტექსტი იყოფა n თანაბარი სიგრძის ბლოკებად (თუ საჭირო გახდა ბოლო ბლოკი ივსება ნულებით). მაშინ თითოეული M_i ბლოკი იშიფრება შემდეგნაირად:

$$C_I = M_I \bmod 2\ P_{i-1}$$

აქ P_{i-1} აღნიშნავს წინა დაშიფრული ბლოკის k უფროსს თანრიგებს. წანაცვლების რეგისტრის განახლება ხდება მასში k უფროსი თანრიგების უკუგდებით და მათ მაგივრად რეგისტრში C_I ჩაწერით. გაშიფვრა ანალოგიურია:

$$M_I = C_I \bmod 2\ P_{i-1}$$

უარყოფით მხარეებად შეიძლება ჩაითვალოს შეცდომის ტირაჟირების უნარი და ხშირად მისი დამუშავებისა და რეალიზაციის სირთულე. სამაგიეროდ საკითხის ამდაგვარად გადაწყვეტა გამორიცხავს წინა მაგალითში კომპრომეტირების განხილულ შემთხვევას ნახ. 3.9.

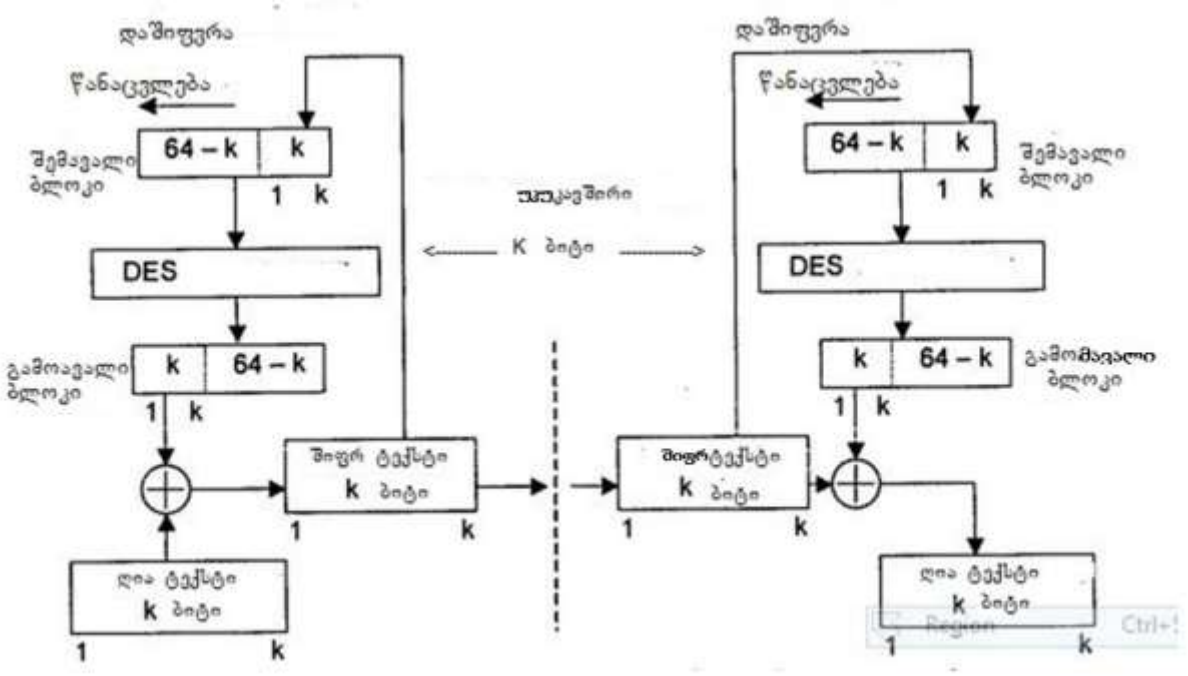


ნახ. 3.9.

ალგორითმი ფუნქციონირებს შემდეგნაირად. საწყისი ტექსტის პირველი ბლოკი იჯამება 2-ის მოდულით ე.წ. ინიციალიზაციის ვექტორთან IV, რომელიც შემდგომში გადაეცემა დაშიფრულ ტექსტთან ერთად. ხდება C0=E(P0 XOR IV) ფუნქციის გამოთვლა. შემდეგ ტაქტზე C1=E(P1 XOR IV) და ა.შ. აღსანიშნავია, რომ ნებისმიერი i ბლოკი არის ყველა მის წინა მდებარე ბლოკების შიფრაციის შედეგი. შედეგად ისინი განუმეორებელია. მაგრამ მას ახასიათებს მნიშვნელოვანი ნაკლიც. საქმე ისაა, რომ შიფრაციის-დეშიფრაციის პროცესის დაწებამდე საჭიროა წინასწარ 64-ბიტიანი ინფორმაციული ნაკადის მიღება, რაც მიუღებელია ტერმინალებისათვის, რომლებიც 8 სიმბოლოიანზე ნაკლებ ინფორმაციით ქმნის სტრიქონებს და მუშაობს შეკითხვა-პასუხის რეჟიმში.

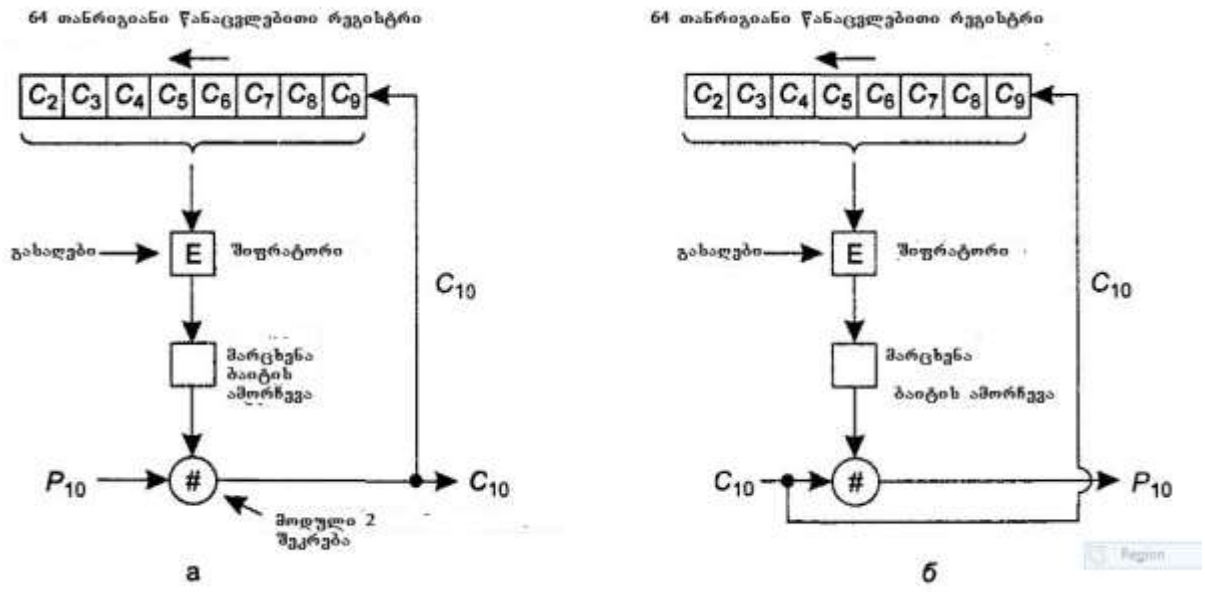
<u>შიფრაცია შიფრტექსტით უკუკავშირით</u> **.** აქ, ნახ.3.10. ისევე, როგორც ზემოთ, ***„შიფრით*** უკუკავშირიანი“ რეჟიმის შემთხვევაში, ტექსტი დაყოფილია ცვლადი სიგრძის ბლოკებად. ბლოკების დაშიფვრა ხდება შიფრაციის ფუნქციით, რომელიც დამოკიდებულია არა მარტო

გასაღებზე, არამედ აგრეთვე წინა ერთ ან რამდენიმე დაშიფრულ ბლოკებზე. ყოველი სეანსისთვის გამოიყენება წანაცვლების რეგისტრის ახალი მნისვნელობები, რომლებიც მიიღება არხით ღია ტექსტის სახით.



ნახ. 3.10.

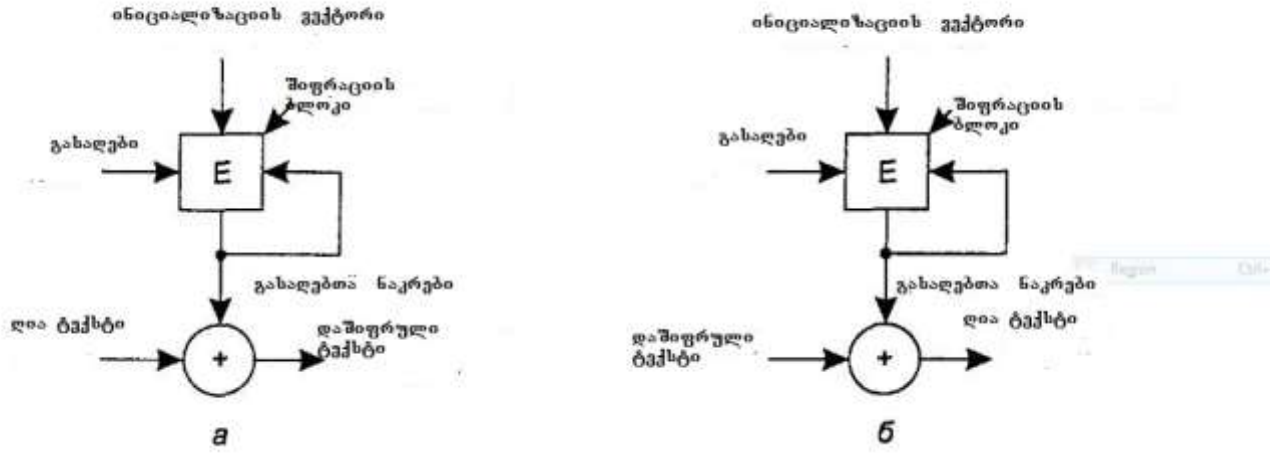
თუ M=M1*M2*M3…..*Mn , მაშინ ყველა i= 1….n გვექნება: C I = MI mod2 Pi . აქ Pi უფროსი k ბიტებია DES (Ci-1) ოპერაციის. წინა სახისაგან განსხვავებით, აქ წანაცვლების რეგისტრის განახლება ხდება უფროსი k ბიტების უკუგდებით და მათი შევსებით, მარჯვნიდან Pi -თი. ამ რეჟიმს (CFB) არ ახასიათებს ზემოთ განხილული ნაკლოვანება, რადგან მუშაობს ბაიტებთან ნახ. 3.11.



ნახ. 3.11.

ნახ,3.11ა) ნაჩვენებია DES (3DES) შიფრაციის სტანდარტის გამოყენების შემთხვევა. ფუნქციონირების ალგორითმი იმავე იქმნება AES სტანდარტის შემთხვევისათვისაც. ნახ-ზე გადმოცემულია შემთხვევა, როცა საწყისი ტექსტის 0-8 ბაიტი უკვე დაშიფრულია და იწყება მე-10 ბაიტის P10–ის დაშიფვრა. DES (3DES) ალგორითმის თანახმად 64-თანრიგიანი რეგისტრის დამუშავება შემდეგნაირად ხდება: იღება ამ დამუშავებული ტექსტის განაპირა მარცხენა ბაიტი, რომელიც იკრიბება 2-ის მოდულით P10-ან. მიღებული C10 იგზავნება არხში და მიეწოდება რეგისტრს. წინასწარ რეგისტრიდან გამოდის C2 ბაიტი, ყველა ბაიტი ინაცვლებს მარცხნივ. განთავისუფლებულ განაპირა მარჯვენა ბაიტში იწერება C10 ბაიტი. შემდეგი პროცესი მეორდება ანალოგიურად. აქაც მიიღება ***განმეორებელი*** დაშიფრული ბაიტები (რადგან რეგისტრის შემადგონლობა განუწყვეტლივ იცვლება). ეს რეჟიმი წინას ანალოგიურად თხოულობს საწყის ინიციალიზაციის ვექტორს. უნდა გვახსოვდეს, რომ ამ რეჟიმში თუნდაც 1 ბიტის სახეცვლილება იწვევს პრაქტიკულად მთელი 8 ბაიტის გაფუჭებას. რეჟიმის ფუქციონირება შეიძლება აღდგეს 8 ტაქტის შემდეგ (სანამ დამახინჯებული ბაიტი არ გავა რეგისტრიდან).

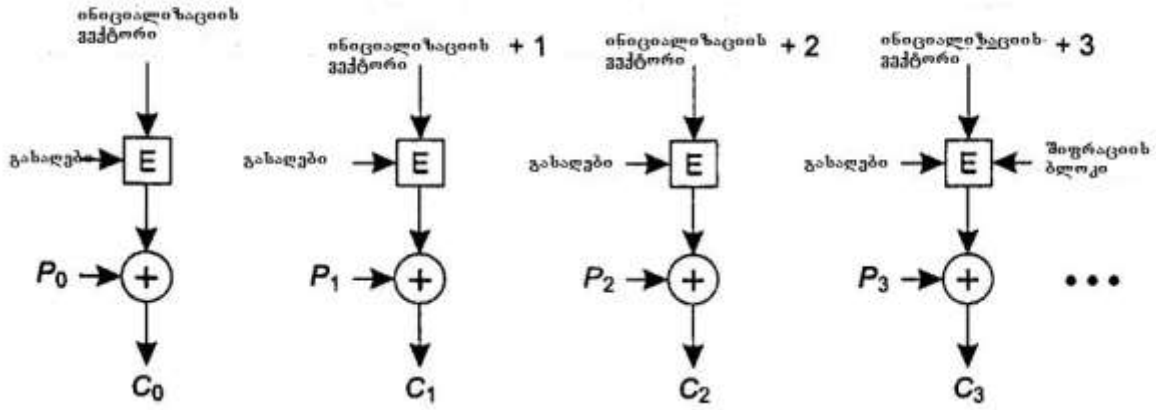
ჯგუფური შიფრის რეჟიმი. ხშირად, ზემოთ განხილული მეთოდების ნაკლი (ერთი ბიტის გაფუჭება იწვევს 64 ბიტის გაფუჭებას) მიუღებელია. ამ ნაკლს ასწორებს ჯგუფური შიფრის რეჟიმი ნახ .3.12.



ნახ. 3.12.

აქ შიფრაციის პროცესი სამეტაპოვანია (ნახ. 3.12 ა). პირველზე ხდება ინიციალიზაციის ვექტორის შიფრაცია გასაღების საშუალებით. შემდეგში ხდება ამ ნაკადის შიფრაცია გასაღებით. ბოლოს ეს ბლოკი აიჯამება ორის მოდულით ღია ტექსტთან და ვიღებთ დაშიფრულ ტექსტს. დეშიფრაციის დროს საჭიროა ისეთივე გასაღებთა ნაკადი, როგორც შიფრაციისას. ამ მეთოდის დროს ბიტის დეშიფრაცია არაა დამოკიდებული მთლიან ტექსტთან. ამიტომაც ერთ ბიტში შეცდომა იწვევს დეშიფრაციის დროს მხოლოდ ამ ერთ ბიტში შეცდომას და არა ყველაში. ამ დროს მთავარია არ იქნეს გამოყენებული ერთი და იმავე ინიციალიზაციის ვექტორი და გასაღებთა წყვილები.

<u>შიფრაცია მრიცხველის რეჟიმით.</u> ზემოთ განხილულ რეჟიმებს ახასიათებთ ერთი საერთო ნაკლი .ჩვეულებრივად შეუძლებელია დაშიფრული ტექსტის ნებისმიერ ნაწილთან შეღწევის შეუძლებლობა. ამის განსახორციელებლად საჭირო ხდება მთელი ტექსტის დეშიფრაცია, რაც გარკვეულ უხერხულებას იწვევს (თუმცა დაცულობის თვალსაზრისით კარგი გადაწყვეტილებაა). ამის თავიდან ასაცილებლად შემოღებულია ე.წ. შიფრაცია მრიცხველის რეჟიმით ნახ. 3.13.



ნახ. 3.13.

აქ დაშიფრული ბლოკი, რომელიც მიიღება ინიციალიზაციის ვექტორისა და გასაღების დაშიფვრით, შემდგომში იჯამება ორის მოდულით ღია ტექსტთან და იგზავნება ქსელში. პარალელურად ხდება ინიციალიზაციის ვექტორის წანაცვლება ერთით. შემდეგ ტაქტზე კიდევ ერთით და ა.შ. შესაბამისად დეშიფრაციის დროს შესაძლებელია ნებისმიერი ბლოკის დეშიფრაცია, რადგან ცნობილია ინიციალიზაციის ვექტორის წანაცვლების ბიჯების რაოდენობა. ნაკლი მდგომარეობს იმაში, რომ არ შეიძლება განმეორდეს გასაღების და ინიციალიზაციის ვექტორის გამოყენება ორჯერ ან მეტჯერ. ამ დროს კრიპტოანალიტიკოს თუ აქვს ორი დაშიფრული შეტყობინება, მათი ორის მოდულის შეკრებით შეუძლია მიიღოს ორი ღია ტექსტის ჯამი ორის მოდულით. შემდეგ ეტაპზე ის ეცდება იოლად გახსნას ტექსტები.

მიახლოებით შეიძლება წარმოვადგინოთ გავრცელებული შიფრების ჩამონათვალი ცხრ . 15.

ცხრ. 15.

1	Blowish	1-448 ბიტი	ძველი და ნელი;
2	DES	56 ბიტი	ძალიან სუსტი;
3	IDEA	128 ბიტი	კარგია, მაგრამ დაპატენტებულია;
4	RC4	1-2048 ბიტი	ზოგიერთ გასაღებთა კომბინაცია სუსტია;
5	RC5	128-256 ბიტი	კარგია, მაგრამ დაპატენტებულია;
6	Rijdael	128-256 ბიტი	საუკეთესოა პირველი ადგილი;
7	Serpent	128-256 ბიტი	ძალიან ძლიერი;
8	3DES	168 ბიტი	საუკეთესოა მე-2 ადგილი;
9	Twofish	128-256 ბიტი	ძალიან ძლიერი, ფართოდ გავრცელებულია.

წარმოდგენილი ცხრილიდან ცხრ.15, განსაკუთრებული აღნიშვნის ღირსია შიფრაციის ალგორითმი **IDEA**, რომლის პირველი ვარიანტები წარმოდგენილი იყო 1990 წ. შვეიცარიის ინსტიტუტის თანამშრომლების **ლაი სიუძეის (Xuejia Lai) და ჯეიმს მესსის (James Massey)** მიერ. იგი იყენებს 128-ბიტიან გასაღებს და 64-ბიტიან ბლოკებს. ყველა ბლოკი იშიფრება შიფრაციის სხვა და სხვა რეჟიმით. საწყისი 64-ბიტიანი ბლოკი იყოფა ოთხ 16-ბიტიან ბლოკებად. შიფრაცია და დეშიფრაცია ხორციელდება ერთი და იმავე ალგორითმით. ეს სიახლე მდგომარეობს იმაში, რომ გამოიყენება:

- მოდ 2^{16} შეკრება;
- გამრავლება მოდ $2^{16} + 1$;
- ბიტური ოპერაციები XOR.

აღსანიშნავია, რომ ამ ოპერაციებს არ ახასიათებთ დისტრიბუციალობა და ასოციაციურობა.

$$a^* (b+c) \neq (a^*b)+(a^*c) \quad a+(b \oplus c) \neq (a+b) \oplus c$$

ასეთი გადაწყვეტა DES-თან შედარებით ართულებს კრიპტოანალიტიკოსის საქმეს, რაც უთუოდ დადებით მხარედ უნდა ჩაითვალოს. როგორც აღინიშნა შიფრაციის პროცესი 8 იტერაციისაგან შედგება და თითოეულისათვის გასაღებების გენერაციის ციკლი შემდეგია:
128-ბიტიანი გასაღები იყოფა 8, 16-ბიტიან ქვეგასაღებებად

$$(K_1^{(2)}\ K_2^{(2)}\ K_3^{(2)}\ K_4^{(2)}\ K_5^{(2)}\ K_6^{(2)}\ K_7^{(2)}\ K_8^{(2)})$$

შემდეგში ეს 128-ბიტიანი გასაღები მარცხნივ წანაცვლდება 35 ბიტით და ვიღებთ ახალ გასაღებს

$$(K_3^{(2)}\ K_4^{(2)}\ K_5^{(2)}\ K_6^{(2)}\ K_1^{(2)}\ K_2^{(2)}\ K_3^{(2)}\ K_4^{(2)})$$

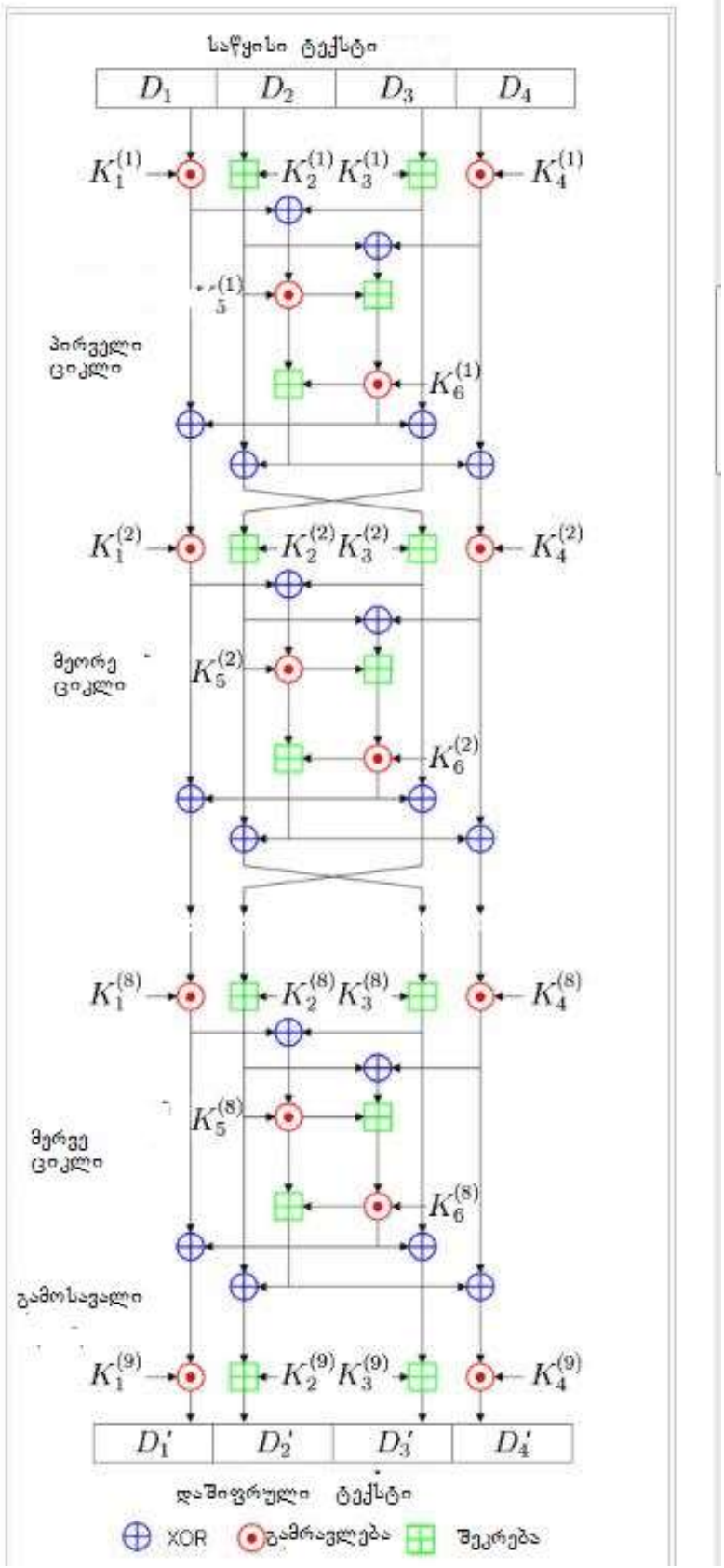
პროცესი გრძელდება ანალოგიურად, სანამ არ შევქმნით 8 ციკლისათვის ქვე გასაღებს. ცხრ.16.

ციკლის ნომერი	ქვე გასაღები
1	$K_1^{(1)}\ K_2^{(1)}\ K_3^{(1)}\ K_4^{(1)}\ K_5^{(1)}\ K_6^{(1)}$
2	$K_1^{(2)}\ K_2^{(2)}\ K_3^{(2)}\ K_4^{(2)}\ K_5^{(2)}\ K_6^{(2)}$
3	$K_1^{(3)}\ K_2^{(3)}\ K_3^{(3)}\ K_4^{(3)}\ K_5^{(3)}\ K_6^{(3)}$
4	$K_1^{(4)}\ K_2^{(4)}\ K_3^{(4)}\ K_4^{(4)}\ K_5^{(4)}\ K_6^{(4)}$
5	$K_1^{(5)}\ K_2^{(5)}\ K_3^{(5)}\ K_4^{(5)}\ K_5^{(5)}\ K_6^{(5)}$
6	$K_1^{(6)}\ K_2^{(6)}\ K_3^{(6)}\ K_4^{(6)}\ K_5^{(6)}\ K_6^{(6)}$
7	$K_1^{(7)}\ K_2^{(7)}\ K_3^{(7)}\ K_4^{(7)}\ K_5^{(7)}\ K_6^{(7)}$
8	$K_1^{(8)}\ K_2^{(8)}\ K_3^{(8)}\ K_4^{(8)}\ K_5^{(8)}\ K_6^{(8)}$
გარდაქმნის შედეგი	$K_1^{(9)}\ K_2^{(9)}\ K_3^{(9)}\ K_4^{(9)}$

ცხრ. 16.

შიფრაცია (ნახ. 3.14), შედგება ერთნაირი, შიფრაციის 8 ციკლისაგან და ერთი გამოსასვლელი გარდაქმნისაგან. საწყისი ტექსტი იყოფა 64-ბიტიან ბლოკებად, ისინი კი თავის მხრივ ოთხ 16-ბიტიან D_1 , D_2 , D_3 , D_4 ქვე ბლოკებად. თითოეულ იტერაციაზე გამოიყენება თავისი ქვეგასაღები. გამოსავალ ციკლზე, მიღებულ ინფორმაციაზე ტარდება შემდეგი ოპერაციები :

- გამრავლება მოდ $2^{16} + 1$;
- მოდ 2^{16} შეკრება.


საწყისი ტექსტი
D_1 D_2 D_3 D_4
$K_1^{(1)}$ $K_2^{(1)}$ $K_3^{(1)}$ $K_4^{(1)}$
$K_5^{(1)}$
პირველი ციკლი
$K_6^{(1)}$
$K_1^{(2)}$ $K_2^{(2)}$ $K_3^{(2)}$ $K_4^{(2)}$
მეორე ციკლი
$K_5^{(2)}$
$K_6^{(2)}$
$K_1^{(8)}$ $K_2^{(8)}$ $K_3^{(8)}$ $K_4^{(8)}$
მერვე ციკლი
$K_5^{(8)}$
$K_6^{(8)}$
გამოსავალი
$K_1^{(9)}$ $K_2^{(9)}$ $K_3^{(9)}$ $K_4^{(9)}$
D_1' D_2' D_3' D_4'
დაშიფრული ტექსტი
XOR
გამრავლება
შეკრება

ნახ. 3.14.

საბოლოოდ ხდება მიღებულის „კონკატენაცია“ (გადაჯაჭვა, შერწყმა). შემდეგ 64-ბიტიან ბლოკზე ტარდება იგივე. მათემატიკურად ეს ციკლები შემდეგნაირად აღიწერება ცხრ.17.

ცხრ. 17.

$$
\begin{aligned}
A^{(i)} &= D_1^{(i-1)} * K_1^{(i)} \\
B^{(i)} &= D_2^{(i-1)} + K_2^{(i)} \\
C^{(i)} &= D_3^{(i-1)} + K_3^{(i)} \\
D^{(i)} &= D_4^{(i-1)} * K_4^{(i)} \\
E^{(i)} &= A^{(i)} \oplus C^{(i)} \\
F^{(i)} &= B^{(i)} \oplus D^{(i)} \\
D_1^{(i)} &= A^{(i)} \oplus ((F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \\
D_2^{(i)} &= C^{(i)} \oplus ((F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \\
D_3^{(i)} &= B^{(i)} \oplus (E^{(i)} * K_5^{(i)} + (F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \\
D_4^{(i)} &= D^{(i)} \oplus (E^{(i)} * K_5^{(i)} + (F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)})
\end{aligned}
$$

შედეგად ვიღებთ 4 ქვებლოკს

D_1^8, D_2^8, D_3^8, D_4^8

მიღებულზე ვასრულებთ საფინალო გარდაქმნას. ცხრ. 18.

ცხრ. 18

$$
\begin{aligned}
D_1^{(9)} &= D_1^{(8)} * K_1^{(9)} \\
D_2^{(9)} &= D_3^{(8)} + K_2^{(9)} \\
D_3^{(9)} &= D_2^{(8)} + K_3^{(9)} \\
D_4^{(9)} &= D_4^{(8)} * K_4^{(9)}
\end{aligned}
$$

ამ საბოლოო გარდაქმნით ვიღებთ დაშიფრულ ტექსტს

D_1^9, D_2^9, D_3^9, D_4^9

დეშიფრაციის პროცესი ანალოგიურია დაშიფვრისა იმ განსხვავებით, რომ ქვეგასაღებები მისთვის ფორმირდება ცხრ.19-ის მიხედვით, მაგალითში, განხილულია შემთხვევა, როცა საწყისი ტექსტია (0000,0001,0002,0003) და გასაღები (0001,0002,0003, 0004,0005,0006,0007,0008). დაშიფვრის შედეგად ვიღებთ შიფრტექსტს (11fb, ed2b,0198,6de5).

ცხრ.19

	იტერაციის გასაღებები						მონაცემთა ბლოკები			
იტერაცია	$K_1^{(i)}$	$K_2^{(i)}$	$K_3^{(i)}$	$K_4^{(i)}$	$K_5^{(i)}$	$K_6^{(i)}$	$D_1^{(i)}$	$D_2^{(i)}$	$D_3^{(i)}$	$D_4^{(i)}$
—	—	—	—	—	—	-	0000	0001	0002	0003
1	0001	0002	0003	0004	0005	0006	00f0	00f5	010a	0105
2	0007	0008	0400	0600	0800	0a00	222f	21b5	f45e	e959
3	0c00	0e00	1000	0200	0010	0014	0f86	39be	8ee8	1173
4	0018	001c	0020	0004	0008	000c	57df	ac58	c65b	ba4d
5	2800	3000	3800	4000	0800	1000	8e81	ba9c	f77f	3a4a
6	1800	2000	0070	0080	0010	0020	6942	9409	e21b	1c64
7	0030	0040	0050	0060	0000	2000	99d0	c7f6	5331	620e
8	4000	6000	8000	a000	c000	e001	0a24	0098	ec6b	4925
9	0080	00c0	0100	0140	-	-	11fb	ed6b	0198	6de5

გასაშიფრად ვიყენებთ იმავე შიფრტექსტს და ცხრ.20 განსაზღვრულ ქვეგასაღებებს და მონაცემთა ბლოკებს.

ციკლები	ციკლების გასაღებები						მონაცემთა ბლოკები			
	$K_1^{\prime(i)}$	$K_2^{\prime(i)}$	$K_3^{\prime(i)}$	$K_4^{\prime(i)}$	$K_5^{\prime(i)}$	$K_6^{\prime(i)}$	$D_1^{(i)}$	$D_2^{(i)}$	$D_3^{(i)}$	$D_4^{(i)}$
1	fe01	ff40	ff00	659a	c000	e001	d98d	d331	27f6	82b8
2	fffd	8000	a000	cccc	0000	2000	bc4d	e26b	9449	a576
3	a556	ffb0	ffc0	52ab	0010	0020	0aa4	f7ef	da9c	24e3
4	554b	ff90	e000	fe01	0800	1000	ca46	fe5b	dc58	116d
5	332d	c800	d000	fffd	0008	000c	748f	8f08	39da	45cc
6	4aab	ffe0	ffe4	c001	0010	0014	3266	045e	2fb5	b02e
7	aa96	f000	f200	ff81	0800	0a00	0690	050a	00fd	1dfa
8	4925	fc00	fff8	552b	0005	0006	0000	0005	0003	000c
9	0001	fffe	fffd	c001	-	-	0000	0001	0002	0003

თავი 4 **ასიმეტრიული შიფრაცია**

როგორც ცნობილია, ასიმეტრიული შიფრაციის ძირითადი განმასხვავებელი ნიშანი მდგომარეობს ცალ–ცალკე შიფრაციისათვის და დეშიფრაციისათვის გასაღებების არსეობაში. მიღებულია, რომ გასაღებთა წყვილს, რომელიც გამოიყენება ინფორმაციის გასაშიფრად, ეწოდება საიდუმლო, ხოლო გასაღებთა წყვილს, რომლითაც იშიფრება ინფორმაცია ღია. პირველად ეს იდეა 1976 წელს წამოაყენეს სტენფორდის უნივერსიტეტის მეცნიერებმა **დიფმა** (Diffie) და **ჰელმანმა** (Hellman). თანამედროებაში საკმაოდ გავრცელებულია შემდეგი ასიმეტრიული შიფრაციის სისტემები:

- RSA;
- EDH;
- DSA,ECDSA;
- ელ–გამალეის (El-Gamal);
- ГОСТ Р 34.10-2012;
- დიფი–ჰელმანის სქემა;
- ДСТУ 4145-2002;
- СТБ 1176.2-99;
- შნორის (Schnorr) სქემა;
- სქემა BLS (Boneh-Lynn-Shacham);
– სქემა GMR (Goldwasser-Micali-Rivest) და სხვა.

§4.1 ასიმეტრიული შიფრაცია. სისტემა RSA

ასიმეტრიული შიფრაციის შესასწავლად განვიხილოთ ყველაზე უფრო გავრცელებული სისტემა RSA [17,32], რომლის სახელიც ემთხვევა სისტემის შემქმნელს მასაჩუტესტის უნივერსიტეტის მეცნიერების აბრევიატურას (Rivest, Shamir, Adleman). სისტემის ანონსირება მოხდა 1978 წელს. სისტემის არსი იმდენად მარტივია, რომ მას ხშირად არითმეტიკულსაც კი უწოდებენ. სისტემის ფუნქციონირება აგებულია ერთ მარტივ იდეაზე: რამდენაც მარტივია ორი მარტივი რიცხვის არითმეტიკული გამრავლება, იმდენად თეორიულადაც შეუძლებელია ამ ნამრავლიდან მამრავლების (ორი მარტივი რიცხვის) პოვნა. სამასი წელია, რაც მსოფლიოს მათემატიკოსები ცდილობენ იპოვონ ასეთი დაშლის თეორიული შესაძლებელობა, მაგრამ უშედეგოდ. არსებობს მხოლოდ ერთი გზა, ნამრავლის თანამიმდევრობითი დაშლა–ფაქტორიზაცია. ეს პროცესი 700 და მეტი ათეული თანრიგის მარტივი რიცხვების შემთხვევაში ათეულ და მეტ წლებს თხოულობს, რაც პრაქტიკულად შეუძლებელს ქმნის შიფრის გატეხვის შესაძლებელობას. პრაქტიკაში აღწერილია შემთხვევა, როცა 700 თანრიგიანი ნამრავლის დაშლა ორ მარტივი რიცხვების მამრავლებად შეძლეს 100 თანამედროვე კომპუტერით შექმნილ ქსელში, ორი წლის განმავლობაში.

გავეცნოთ RSA სისტემის ფუნქციონირების ალგორითმს. გვაქვს ორი მარტივი რიცხვი p და q რიცხვები. მათ საფუძველზე იანგარიშება n=p.q და ე.წ „ეილერის ფუნქცია“.

$$\varphi(n)=(p-1).(q-1)$$

n რიცხვის ათობითი თანრიგების სიდიდე განსაზღვრავს ასიმეტრიული შიფრაციის ალგორითმის მთავარ პარამეტრს-გასაღების სიგრძეს. ვირჩევთ შემთხვევთ დიდ რიცხვს $d>1$, რომლისათვისაც სრულდება პირობა $\mathrm{mod}\,(d, \varphi(n))=1$ და ვანგარიშობთ e რიცხვს, შემდეგი პირობების დაცვით $1 < e < \varphi(n)$ და $(e.d, \mathrm{mod}\ \varphi(n)) = 1$.

ამ რიცხვებიდან e და n ქმნიან ღია გასაღებთა წყვილს, რომლებიც გამოიყენება ინფორმაციის დასაშიფრად, ხოლო რიცხვები ρ, q, d და φ(n) საიდუმლო გასაღების წყვილს. ეს წყვილი გამოიყენება დაშიფრული ინფორმაციის გასაშიფრად. ცხადია, რომ თუნდაც ერთ–ერთი ρ ან q რიცხვის ცოდნა იწვევს საიდუმლო გასაღების სხვა შემადგენლების განსაზღვრას და მთლიანობაში სისტემის კომპრომეტირებას.

შიფრაციის თანამიმდევრობაა:

- ხდება საწყისი ინფორმაციის თითოეული სიმბოლოს კოდირება ათობითი ათვლის სისტემაში (შესაძლებელია ორობითშიც);
- ხდება მიღებული ინფორმაციის დაყოფა კრიპტოტექსტის ბლოკებად. ბლოკის სიგრძე i იანგარიშება შემდეგი უტოლობიდან.

$$10^{i-1} \le n < 10^{i}$$

რეკომენდებულია ბლოკის სიგრძედ აღებული იქნეს i –ს მნიშვნელობა და არა (i–1).

შემდეგში ხდება თითოეული კრიპტოტექსტის ბლოკების w ინდივიდუალური დაშიფვრა შემდეგი ფორმულით:

$$c=(w^e \text{ , mod } n)$$

აქ c წარმოადგენს საწყისი ტექსტის ω ბლოკის დაშიფრულ ბლოკს.

– წარმოებს მიღებული დაშიფრული c ბლოკების შემდეგი ფორმულით გაშიფვრა

$$w=(c^d \text{ , mod } n)$$

თუ შევადარებთ სიმეტრიული და ასიმეტრიული შიფრაციის ალგორითმებს, ცალსახად ჩანს ასიმეტრიული შიფრაციის ალგორითმის სიმარტივე და მისი მომხიბვლელობა. თუმცა ეს ρ და q სიდიდის დაბალი მნიშვნელობებისათვისაა (< 100 ათობით თანრიგისათვის) განმარტებული. ამ მეთოდს ახასიათებს ორი მნიშვნელოვანი სირთულე:

პირველი - მძლავრი გამოთვლითი რესურსების მიმართ მოთხოვნის არსებობა და მნიშვნელოვანი დროითი დანახარჯების გამო (n> 1000 ათობით თანრიგისთვის) ρ და q მარტივი რიცხვების გამოთვლისათვის. მაგალითად, ორ ბირთვიან პერსონალურ კომპუტერზე ≈ 5 სთ საჭირო 1500- თანრიგიანი (ათობითი) მარტივი რიცხვის მოსაძებნად;

მეორე-დეშიფრაციის პროცესში პრაქტიკულად შეუძლებელია, მისაღებ პრაქტიკულ დროში, w=(c^d , mod n) გამოთვლის ჩატარება სპეციალური ალგორითმების გამოყენების გარეშე. ეს ალგორთმი ცნობილია „ხარისხში მოდულური აყვანის“ სახელით. მისი არსი მდგომარეობს შემდეგში. N რიცხვი ორობით სახეში შეიძლება წარმოვიდგინოთ შემდეგნაირად:

$$N=\Sigma^{k}_{j=0},x_j*2^j \text{ , სადაც } x_j=0,1 \text{ ,k= (} \log_2 N \text{) + 1 და } 0\le j\le k$$

ეს ნიშნავს იმას, რომ ნებისმიერი მნიშვნელობებისთვის e^d , შესრულდება მხოლოდ (k-1) რაოდენობის ნამრავლთა შესრულება და თითოეულთა ნამრავლისათვის mod(n)–ის გამოთვლა. მაგალითისთვის გამოსათვლელია 7^{23} mod 61 . ცხადია, რომ 23 = 10111 (ორობით სახეში). შესაბამისად ვიღებთ ცხრილს

j	0	1	2	3	4
$(7^2)^j$	*7*	*49*	*22*	*54*	*16*

მეორე სტრიქონის ელემენტების და რიცხვის 23 ორობით ფორმის გათვალისწინებით ვიღებთ (7^{23} mod 61) გამოსათვლელად გამოსახულებას (16*22*49*7) mod 61 = 17. ცხადია, შემოთავაზებული ალგორითმის უპირატესობა. თუ ჩვეულებრივად 7^{23} გამოსათვლელად საჭირო იყო 23 გამრავლების ჩატარება - შემოთავაზებულში მხოლოდ 4. მიახლოებით ცნობილია, რომ 100 ბიტის სიდიდის სიმეტრიული შიფრაციის გასაღების სიგრძე, დაცულობის თვალსაზრისით, ტოლფასია ასიმეტრიული შიფრაციის 700 თანრიგიანი გასაღების. შესაბამისად დიდი მოცულობის საწყისი ინფორმაციების დასაშიფრად რეკომენდებულია (დროითი რესურსების შიფრაცია – დეშიფრაციაზე დანახარჯების გათვალისწინებით) გამოყენებული იქნას სიმეტრიული შიფრაციის ალგორითმები. თუმცა ასიმეტრიული შიფრაციის RSA სისტემა შეუცვლელია ელექტრონული ხელმოწერის პროცედურების პროგრამული უზრუნველყოფის შესაქმნელად.

განვიხილოთ მაგალითი მოცემული ალგორითმისთვის. თუ ნაპოვნია გასაღებების შემდეგი მნიშვნელობები :

- საიდუმლო p=47, q=59, φ(n)=(p-1). (q-1)=2668, d=157 ;
- ღია n=p*q=2773, e=17.

ვთქვათ, შექმნილი გვაქვს ქართული ანბანისთვის შემდეგი ცხრილი. ცხრ.4.1.1 (მოცემულია მხოლოდ საწყისი ალფავიტისთვის). ავიღოთ ბლოკის სიგრძე i=4.

ცხრ. 4.1.1

ა	01	ლ	11
ბ	02	მ	12
გ	03	ნ	13
დ	04	ო	14
ე	05	პ	15
ვ	06	რ	16
ზ	07	ს	17
თ	08	ტ	18
ი	09	და ა.შ	...
კ	10	Space	25

ვთქვათ, საწყისი ტექსტია „ აი მზე“. ცხრილის მიხედვით ვიღებთ:
ω=0109 2512 0705

- ვაწარმოოთ თითოეული ბლოკის დაშიფვრა :
- $C_1=109^7 \bmod n=170$;
- $C_1=2512^7 \bmod n=2362$;
- $C_1=0705^7 \bmod n=1692$.

შესაბამისი დაშიფრული ტექსტი იქნება:

c= 0170 2362 1692.

გასაშიფრად ვიყენებთ საიდუმლო გასაღებს d=157.

- $\omega_1=c_1^{157} \bmod n=0109$;
- $\omega_2=c_2^{157} \bmod n=2512$;
- $\omega_3=c_3^{157} \bmod n=0705$.

საბოლოოდ ვიღებთ ω=0109 2512 0705. შესაბამისი ალფავიტური გამოსახულებაა „აი მზე“.

გავეცნოთ ასიმეტრიული შიფრაციის შემადგენელ კომპონენტებს.

§ 4.2 ასიმეტრიული შიფრაცია, მარტივი რიცხვების გენერაცია

ერთ-ერთი ყველაზე ეფექტური გზა მარტივი რიცხვების მოსაძებნად არის ***ფერმას თეორემა***. მისი არსი შემდეგშია: თუ გვაქვს N და S კენტი ნატურალური რიცხვები და N–1= S.R, თანაც S რიცხვის ყველა მარტივი გამყოფისათვის q არსებობს ისეთი a მთელი რიცხვი, რომლისთვისაც $n^{N-1} = 1 \pmod N$, უსგ $(\alpha^{(N-1)/q} -1,N)=1$ (აქ უსგ უდიდესი საერთო გამყოფია). მაშინ N რიცხვის თითოეული მარტივი გამყოფი p აკმაყოფილებს პირობას $p = 1 \pmod{2S}$. ფერმას თეორემიდან გამოდის ძალიან პრაქტიკული შედეგი:

$$R <= 4*S +2$$

პრაქტიკულად მარტივი რიცხვის მოძებნა ხორციელდება შემდეგნაირად. ვიღებთ მარტივ დიდ რიცხვს S, ვირჩევთ შემთხვევით კენტ რიცხვს R, რიცხვების დიაპაზონიდან R <= 4.S+2 და ვანგარიშობთ სავარაუდო მარტივ რიცხვს N=SR+1. მიღებული რიცხვის შემოწმება სიმარტივეზე ხდება ელემენტარული ტესტით ცხრილებში მოყვანილი პატარა მარტივ რიცხვებზე გაყოფით. ამ ეტაპზე ასეთი სახის შემოწმება დიდი ალბათობით ვერ ადგენს მიღებული N რიცხვის სიმარტივეს და ამიტომაც რეკომენდებულია მარტივი რიცხვების დიაპაზონი განვსაზღვროთ ≈30000. თუ ეს მინიმალური პირობა არ შესრულდა, ვიღებთ R რიცხვის სხვა მნიშვნელობას და შემოწმების ციკლს ვიმეორებთ, სანამ არ მივიღებთ სასურველ შედეგს. დამატებით ამ შემთხვევაში, მიღებულ შედეგს (კერძოდ N და R რიცხვებს) ვამოწმებთ ევკლიდეს გაფართოებული ალგორითმით და თუ უსგ=1, რაღაც ალბათობით ვთვლით, რომ შედეგი დამაკმაყოფილებელია. თუ მიღებული N რიცხვის სიდიდე არ დაგვაკმაყოფილებს, ძებნის ციკლს ვიმეორებთ, ოღონდ საწყის სიდიდედ ვიღებთ N და ა.შ სანამ არ მივიღებთ სასურველ შედეგს. შემოთავაზებული ალგორითმის ნაწილის რეალიზაცია PYTHON პროგრამულ ენაზე გადმოცემულია ნახ.4.1.1.

```
pp=sn1
while jj!=nnn-1 :
    m=4*pp+2
    q=pp*m+1
    while j < nn :
       pp=q
       m=4*pp+2
       q=pp*m+1
       j=j+1
       continue
    j=0
    #psearch[jj]=pp
    qsearch[jj]=q

    while j<len(pn) :
        fff=qsearch[jj]//pn[j]
        k=fff*pn[j]
        if k==qsearch[jj] :
            qsearch[jj]=qsearch[jj]-qkor
            if qsearch[jj]<3 :
                print ('783 dont find q for this p',p,'nn=',nn,
                break
            j=0
        j=j+1
        qsearch[jj]=qsearch[jj]
        continue

    jj=jj+1
    j=0
    #print '1metka p=',psearch[jj],'q=',qsearch[jj]
    gcd=0
    p=sn1
    q=q
    f=(p-1)*(q-1)
    n=p*q
```

ნახ. 4.1.1

უნდა გავითვალისწინოთ, რომ შემოთავაზებული კონტროლის მეთოდიკა ყველაზე მარტივია და არ იძლევა დიდი ალბათობით დასტურს მიღებული შედეგის ჭეშმარიტებაზე. იმის გათვალისწინებით, რომ დაშიფრული ინფორმაციის მედეგობაში (კრიპტო ანალიტიკური გადარჩევა-გატეხვის თავიდან ასაცილებლად) გადამწყვეტი მნიშვნელობა აქვს ძალიან დიდი ალბათობით დადასტურებულ გამოყენებულ მარტივ რიცხვებს. განვიხილოთ უფრო სრულყოფილი ალგორითმი ე.წ. მილერ-რაბინის მეთოდი.

§ 4.3 ასიმეტრიული შიფრაცია,მარტივი რიცხვების კონტროლი. მილერ–რაბინის თეორემა.

მილერ-რაბინის ტესტი [20]. ეს არის პოლინომიალური ალბათური ტესტი რიცხვის სიმარტივის დასადგენად. ის ძალიან დიდი (მოთხოვნილი) ალბათობით ადგენს შესამოწმებელი რიცხვის სიმარტივეს. მაგრამ მისი საშუალებით არ შეიძლება (ვიმსჯელოთ) მკაცრად დავამტკიცოთ დიდი რიცხვების მარტივობა. პირველად ეს ალგორითმი წამოაყენა გარი მილერმა 1976 წ, რომლის მოდიფიცირება შეძლო მაიკლ რაბინმა 1980 წ. ამ ალგორითმის არსი შემდეგშია. თუ გვაქვს $R>1$ შესამოწმებელი კენტი რიცხვი, მაშინ ცალსახად განისაზღვრება $R-1 = 2^s * t$, სადაც t კენტი რიცხვია, ხოლო **s** დადებითი მთელი რიცხვია. **a** მთელ რიცხვს $1 \le a \le R$ ეწოდება R რიცხვის „სიმარტივის მოწმე" თუ სრულდება ორი პირობა :

– R არ იყოფა a–ზე;

- $a^t = (\text{mod } R)$ ან თუ არსებობს ისეთი k რიცხვი, რომლისთვისაც $0 \le k < s$ და

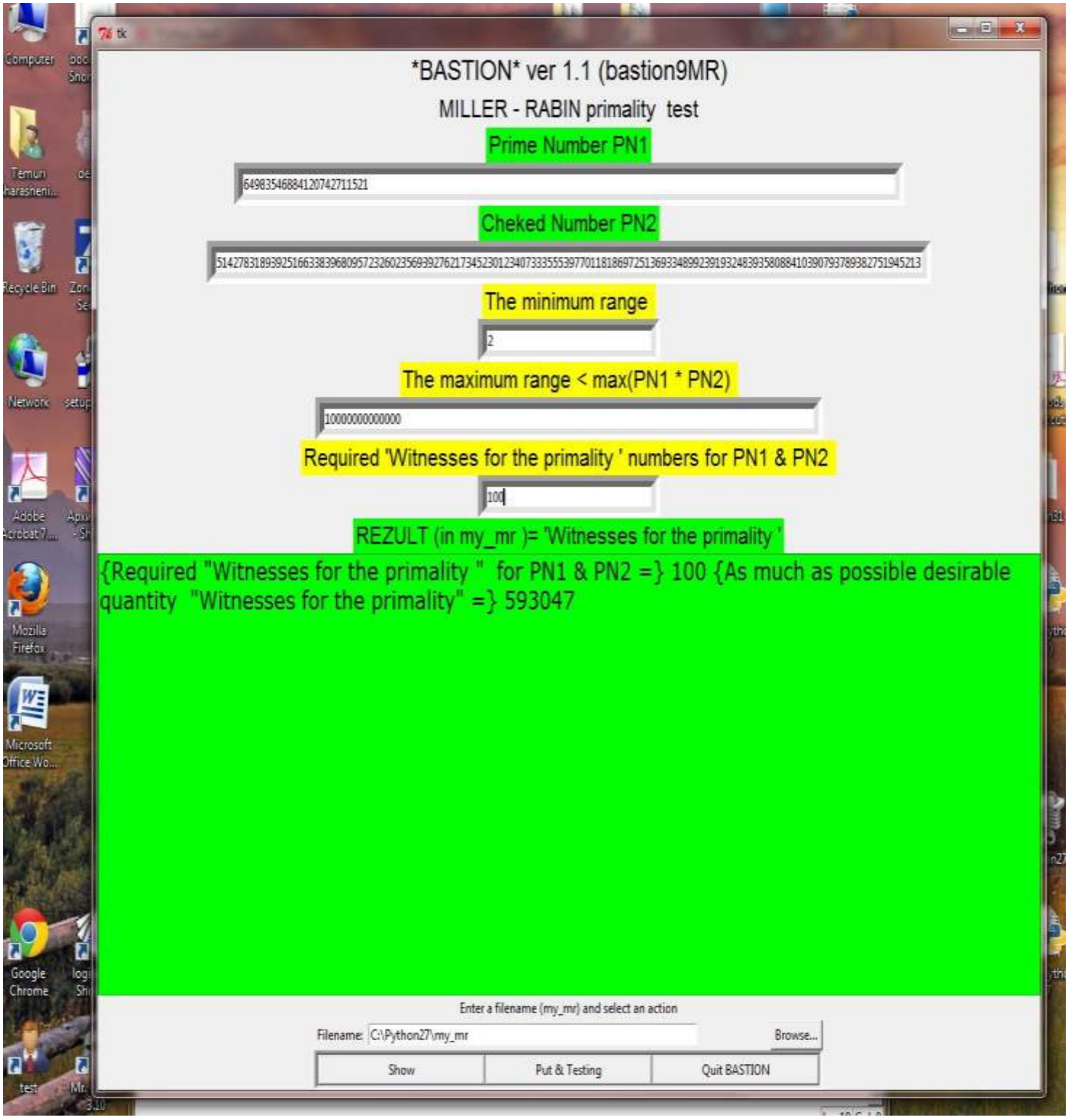
$a^{\mu \cdot t} = (\text{mod } R)$, სადაც $\mu = 2^k$.

რეკომენდებული მოსაძებნი „სიმარტივის მოწმე"–თა რაოდენობა განისაზღვრება ფორმულით $r = \log_2(R)$. ამ შემთხვევისათვის ალბათობა იმისა, რომ რიცხვი მარტივი არ არის, ტოლია 4^{-r}. ყურადღება მისაქცევია, რომ R რიცხვის ძალიან დიდი მნიშვნელობებისთვის **r** რეკომენდებული მნიშვნელობების გამოთვლა ძალიან მნიშვნელოვან დროს მოითხოვს. მილერ – რაბინის ტესტის პირველი მოთხოვნის შემოწმების ტესტის რელიზაცია PYTHON პროგრამულ ენაზე მოცემულია ნახ.4.1.2.

```
i=len(pn)
j=0
rezult='ERROR for First Test'
  print '1739  rezult',rezult
g=0
gg=0
a=p
b=q
while gg!=2 :
    while j<i :
        c=a/pn[j]
        if c*pn[j]==a :
            print 'rezut=',rezult,'p=',p,'pn[j]=',pn[j]
              print 'p=',a
              print 'pn[j]=',pn[j]
            j=i
            g=g+1
        j=j+1
    j=0
    while j<i :
        c=b/pn[j]
        if c*pn[j]==b :
            print 'rezut=',rezult,'q=',q,'pn[j]=',pn[j]
              print 'q=',b
              print 'pn[j]=',pn[j]
            j=i
            g=g+1
        j=j+1
    gg=2
if g==0 :
     rezult= 'Rezult for First Test ----> GOOD Prime Numbers  p & q'
     FTGOOD=1
print '1765 ',rezult
print #2222
```

ნახ. 4.1.2

საქართველოში შექმნილი ასიმეტრიული შიფრაციის კრიპტოგრაფიული სისტემა „ბასტიონის“ [21] შესაბამისი პროგრამული მოდულის ეკრანული ფორმა მოცემულია ნახ. 4.1.3 .


tk
BASTION ver 1.1 (bastion9MR)
MILLER - RABIN primality test
Prime Number PN1
649835468841207427115 21
Cheked Number PN2
5142783189392516633839680957232602356839276217345230123407333555397701181869725136933489923919324839358088410390793789382751945213
The minimum range
2
The maximum range < max(PN1 * PN2)
10000000000000
Required 'Witnesses for the primality ' numbers for PN1 & PN2
100
REZULT (in my_mr)= 'Witnesses for the primality '
{Required "Witnesses for the primality " for PN1 & PN2 =} 100 {As much as possible desirable quantity "Witnesses for the primality" =} 593047
Enter a filename (my_mr) and select an action
Filename: C:\Python27\my_mr
Browse...
Show
Put & Testing
Quit BASTION

бзб. 4.1.3

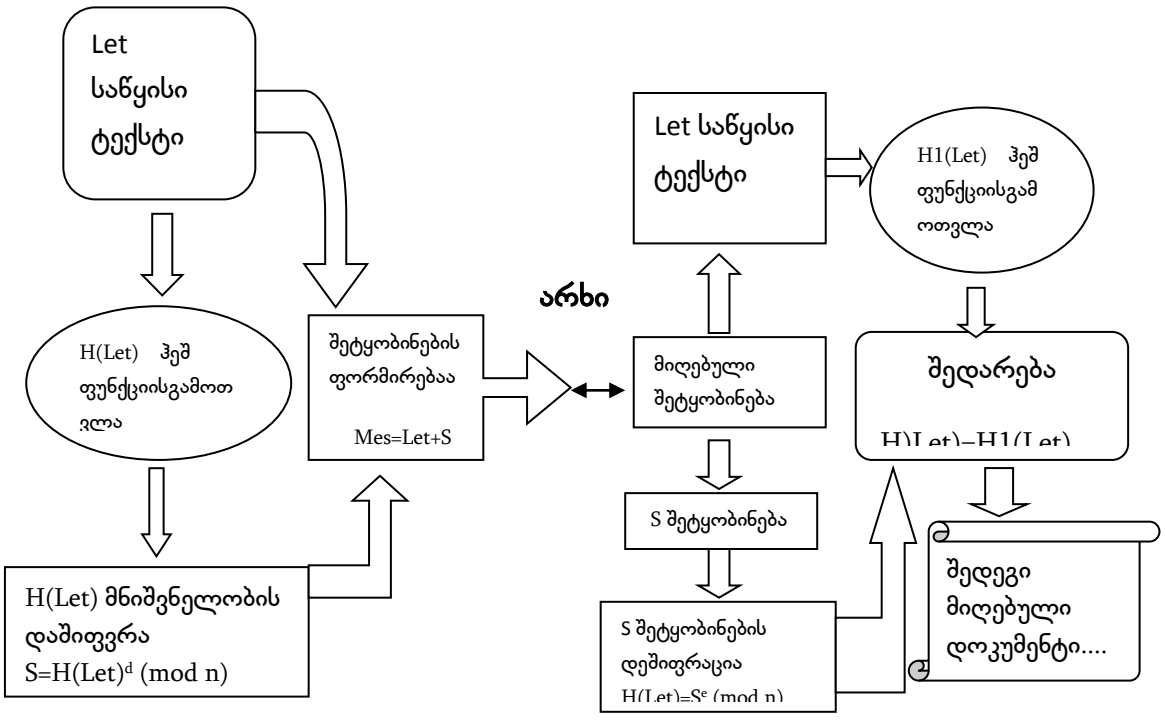
§ 4.4 ასიმეტრიული შიფრაცია. მისი გამოყენება „ელექტრონული ხელმოწერის“ ორგანიზაციაში.

„ელექტრონული ხელმოწერა“ (ხშირად იხმარება ტერმინიც „ელექტრონული ციფრული ხელმოწერა“)- ეს არის ელექტრონული დოკუმენტის რეკვიზიტი, რომელიც მიიღება ხელმომწერის საიდუმლო გასაღების გამოყენებით, დოკუმენტის კრიპტოგრაფიული გარდაქმნის შედეგად და რომლის საშუალებით დგინდება, რომ დოკუმენტს არ განუცდია მოდიფიცირება, სახეცვლილება დოკუმენტის შექმნის მომენტიდან მის მიღებამდე [22,17,]. ამგვარად მტკიცდება გამომგზავნის აუტენფიკაცია.

„ ელექტრონული ხელმოწერა “ უზრუნველყოფს :

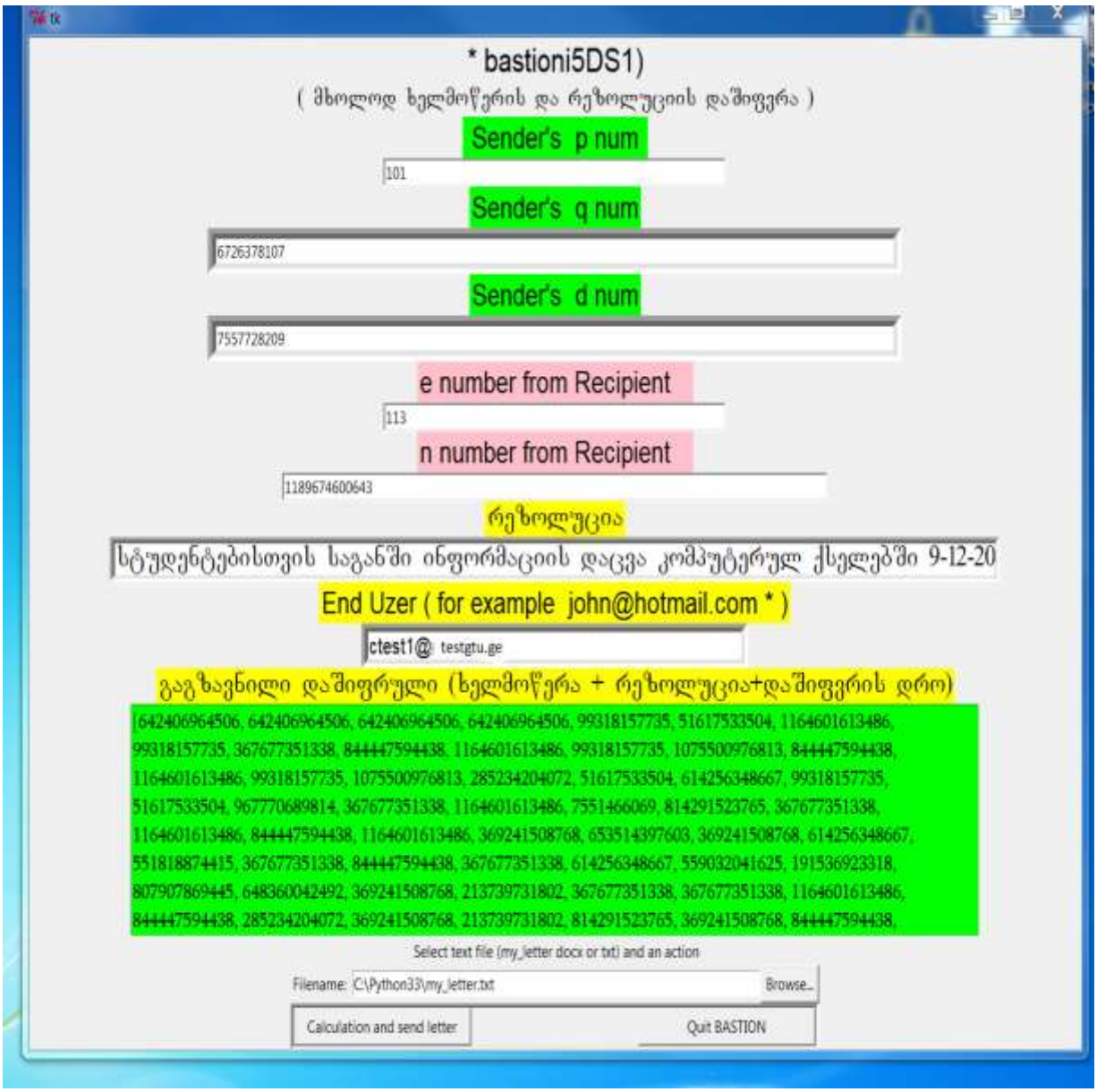
- კონტროლს გადასაცემი ინფორმაციის მთლიანობაზე ;
- გადასაცემი ინფორმაციის დამახინჯებისაგან დაცვას ;
- გადამცემის პასუხისმგებლობას;
- ანალოგიურად ცალსახად განსაზღვრავს დოკუმენტის ავტორს.

ყველა ეს საკითხი დაცულია შესაბამისი საკანონმედებლო ბაზით მრავალ ქვეყანაში და მათ შორის საქართველოშიც. ჩამოთვლილი საკითხების პრაქტიკაში რეალიზაციის უკეთ აღსათქმელად ,განვიხილოთ ასიმეტრიული შიფრაციის კრიპტოგრაფიული დაცვის სისტემა „ბასტიონი“ ნახ.4.1.4. მასში დამატებით რეალიზებულია მოდიფიცირებული ალგორითმი ე.წ. „სტეგანოგრაფიული“ ეფექტით [21]. აქ გათვალისწინებულია მიმღებსა და გადამცემ მხარეებს შორის წინასწარ „ღია გასაღებთა“ წყვლების ურთიერთგაცვლა. გადამცემი მხარე საწყის წერილთან ერთად მიმღებს უგზავნის საიდუმლო, დაფარულ (დაშიფრული სახით), მოკლე ტექსტურ ინფორმაციას რეზოლუციის სახით, რომელიც უხილავია არხში არასანქციონირებულად შეჭრილ ბოროტგამზრახველისათვის.

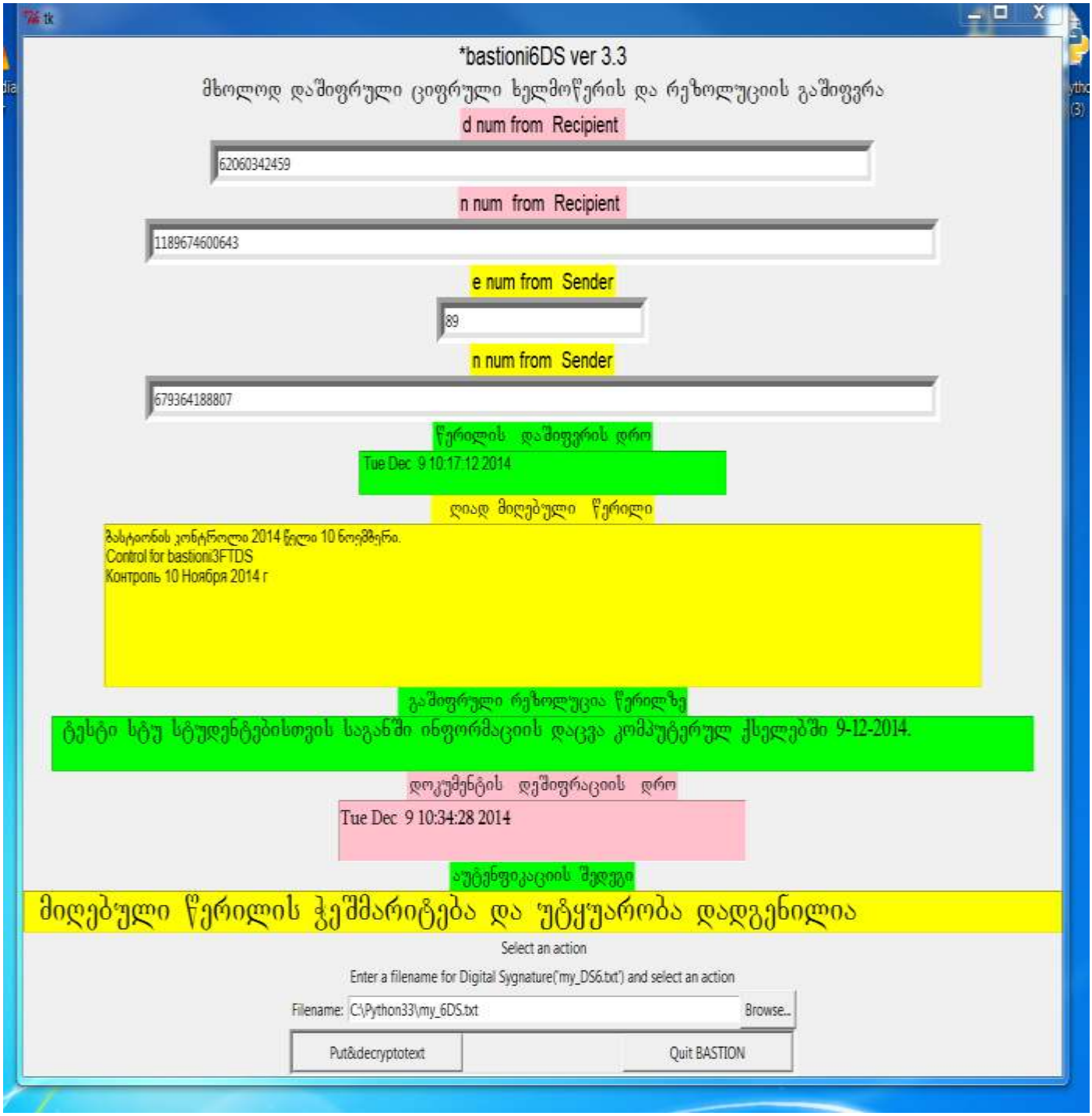


ნახ. 4.1.4

ბოროტგამზრახველი ვერ ამჩნევს ფარულ რეზოლუციას, რადგან კითხულობს მხოლოდ ღია წერილს, რომელიც შეიძლება არც იყოს მთავარი. მიმღები მხარე ადარებს მიღებული წერილისა და გაშიფრულის ჰეშ ფუნქციათა მნიშვნელობებს და აწარმოებს გადამცემის მიერ მისი ღია გასაღებთა წყვილით დაშირული რეზოლუციის გაშიფვრას და იღებს გადაწყვეტილებას მიღებული წერილისა და რეზოლუციის ჭეშმარიტებაზე. „ბასტიონის“ შესაბამისი პროგრამული მოდულების ეკრანული ფორმები მოცემულია ნახ .4.1.5 და ნახ. 4.1.6.


* bastioni5DS1)
(მხოლოდ ხელმოწერის და რეზოლუციის დაშიფვრა)
Sender's p num
101
Sender's q num
6726378107
Sender's d num
7557728209
e number from Recipient
113
n number from Recipient
1189674600643
რეზოლუცია
სტუდენტებისთვის საგანში ინფორმაციის დაცვა კომპუტერულ ქსელებში 9-12-20
End Uzer (for example john@hotmail.com *)
ctest1@ testgtu.ge
გაგზავნილი დაშიფრული (ხელმოწერა + რეზოლუცია+დაშიფვრის დრო)
642406964506, 642406964506, 642406964506, 642406964506, 99318157735, 51617533504, 1164601613486, 99318157735, 367677351338, 844447594438, 1164601613486, 99318157735, 1075500976813, 844447594438, 1164601613486, 99318157735, 1075500976813, 285234204072, 51617533504, 614256348667, 99318157735, 51617533504, 967770689814, 367677351338, 1164601613486, 7551466069, 814291523765, 367677351338, 1164601613486, 844447594438, 1164601613486, 369241508768, 653514397603, 369241508768, 614256348667, 551818874415, 367677351338, 844447594438, 367677351338, 614256348667, 559032041625, 191536923318, 807907869445, 648360042492, 369241508768, 213739731802, 367677351338, 367677351338, 1164601613486, 844447594438, 285234204072, 369241508768, 213739731802, 814291523765, 369241508768, 844447594438,
Select text file (my_letter docx or txt) and an action
Filename: C:\Python33\my_letter.txt
Browse...
Calculation and send letter
Quit BASTION

ნახ. 4.1.5


tk
*bastioni6DS ver 3.3
მხოლოდ დაშიფრული ციფრული ხელმოწერის და რეზოლუციის გაშიფვრა
d num from Recipient
62060342459
n num from Recipient
1189674600643
e num from Sender
89
n num from Sender
679364188807
წერილის დაშიფვრის დრო
Tue Dec 9 10:17:12 2014
ღიად მიღებული წერილი
Basatonis kontroli 2014 wli 10 noemberi.
Control for bastioni3FTDS
Контроль 10 Ноября 2014 г
გაშიფრული რეზოლუცია წერილზე
ტესტი სტუ სტუდენტებისთვის საგანში ინფორმაციის დაცვა კომპიუტერულ ქსელებში 9-12-2014.
დოკუმენტის დეშიფრაციის დრო
Tue Dec 9 10:34:28 2014
აუტენფიკაციის შედეგი
მიღებული წერილის ჭეშმარიტება და უტყუარობა დადგენილია
Select an action
Enter a filename for Digital Sygnature('my_DS6.txt') and select an action
Filename: C:\Python33\my_6DS.txt
Browse...
Put&decryptotext
Quit BASTION

ნახ. 4.1.6

§ 4.5 ასიმეტრიული შიფრაციის გამოყენების პრაქტიკული რეალიზაციის მაგალითი

განვიხილოთ ასიმეტრიული შიფრაციის პრაქტიკული რეალიზაცია ძალიან, მარტივ მაგალითზე, როცა წარმოებს საწყისი ტექსტის გადაცემა თანხმლები „ელექტრონული ხელმოწერით“. განხილვისა და რეალიზაციის თანამიმდევრობა გადმოცემულია ცხრ.4.5. საწყის ეტაპზე გამოყენებული ენის (ენების) საფუძველზე, დგინდება დამოკიდებულება სიმბოლოებსა და მათ რიცხვით მნიშვნელობებს შორის. ჩვენი მაგალითისათვის, გამოყენებული გვაქვს მხოლოდ ინგლისური ენა. ვიღებთ შესაბამის ცხრილს 4.5. ქვემოთმოყვანილი მონაცემები მიღებულია Python პროგრამული ენის არეში. მრავალენოვანი ცხრილების შედგენა შედარებით რთულია და ითხოვს რიცხვითი მნიშვნელობების გაზრდას და გამოყენებული პროგრამული ენების თავისებურებების ცოდნას.

ჩვენი მაგალითისათვის საწყის ტექსტად აღებულია ‘1234567890 bastion’. თავიდანვე ვსაზღვრავთ, რომ შეტყობინება message უნდა შეიცავდეს დამატებით ინფორმაციას რეზოლუციის სახით და შეტყობინების შექმნის დროს და თარიღს. ეს უკანასკნელი ამაღლებს სისტემის დაცულობის ხარისხს.

წინასწარ გამოთვლილი, იხ. ნახ.4.1.7, გასაღებთა წყვილის (n , e) თანახმად განისაზღვრება **e**–ს ორობითი ფორმა oroe=1001101 (ათობითი 89) და შიფრაციის ბლოკების სიგრძე blsize=12. ცხრ.4.5-დან Tab და message გათვალისწინებით ვქმნით digtext და მის საფუძველზე 12 თანრიგა ბლოკების მასივს block.

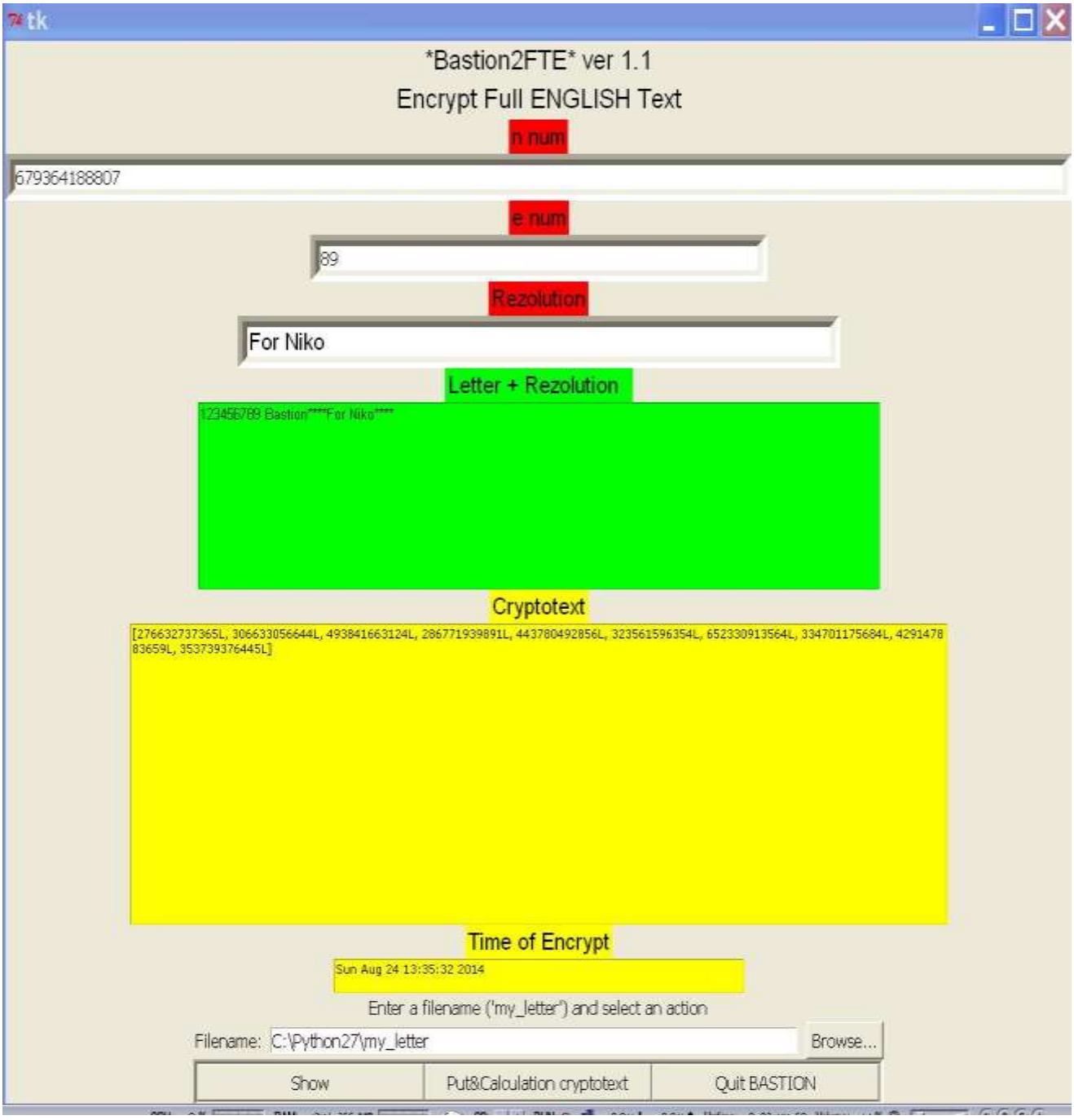
ასიმეტრიული შიფრაციის ცნობილი დაშიფვრის ფორმულით ხდება თითოეული ბლოკის დაშიფვრა და cryptotext კრიპტოტექსტის მიღება, რომელიც გადაეგზავნება მიმღებს.

.

ცხრ. 4.5

```
Tab= {' ': 45, 'S': 90, '(': 62, ',': 61, '0': 38, '4': 32, '8': 36, '<': 98, '@': 50, 'D': 76, 'H': 79, 'L': 82, 'P': 74, '
T': 39, 'X': 83, 'd': 13, 'h': 16, 'l': 19, 'p': 10, 't': 55, 'x': 21, '|': 97, '#': 89, '+': 46, '/': 49, '3': 31, '7': 35,
'C': 42, 'G': 78, 'K': 81, 'O': 73, 'S': 40, 'W': 43, '_': 27, 'c': 22, 'g': 15, 'k': 18, 'o': 59, 's': 12, 'w': 52, '{': 93
, '\n': 66, '"': 95, '&': 92, '*': 48, '.': 28, '2': 30, '6': 34, ':': 67, '>': 99, 'B': 85, 'F': 77, 'J': 80, 'N': 86, 'R':
41, 'V': 84, 'Z': 44, '^': 63, 'b': 24, 'f': 14, 'j': 17, 'n': 25, 'r': 54, 'v': 23, 'z': 20, '\r': 65, '!': 88, '%': 91, '}
': 64, '-': 47, '1': 29, '5': 33, '9': 37, '=': 60, 'A': 75, 'E': 69, 'I': 72, 'M': 87, 'Q': 68, 'U': 71, 'Y': 70, 'a': 11,
'e': 53, 'i': 58, 'm': 26, 'q': 51, 'u': 57, 'y': 56, '}': 94}

letter= 123456789 Bastion

n= 679364188807 e= 89

digtext= [29, 30, 31, 32, 33, 34, 35, 36, 37, 45, 85, 11, 12, 55, 58, 59, 25, 48, 48, 48, 48, 77, 59, 54, 45, 86, 58, 18, 59
, 48, 48, 48, 48, 40, 57, 25, 45, 75, 57, 15, 45, 30, 32, 45, 29, 31, 67, 31, 33, 67, 31, 30, 45, 30, 38, 29, 32, 63, 63, 63
]

oroe= [1, 0, 0, 1, 1, 0, 1]

Block size= 12

Block= [293031323334L, 353637458511L, 125558592548L, 484848775954L, 458658185948L, 484848405725L, 457557154530L, 32452931673
1L, 336731304530L, 382932636363L]


cryptotext message= [276632737365L, 306633056644L, 493841663124L, 286771939891L, 443780492856L, 323561596354L, 652330913564L
, 334701175684L, 429147883659L, 353739376445L]

message= 123456789 Bastion****For Niko****Sun Aug 24 13:35:32 2014
```

მაგალითის პრაქტიკული რეალიზაციის ეკრანული ფორმა სისტემა Bastion არეში მოცემულია ნახ.4.1.7.

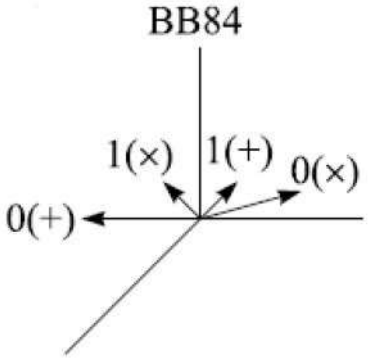

tk
Bastion2FTE ver 1.1
Encrypt Full ENGLISH Text
n num
679364188807
e num
89
Rezolution
For Niko
Letter + Rezolution
123456789 Bastion****For Niko****
Cryptotext
[276632737365L, 306633056644L, 493841663124L, 286771939891L, 44378049285 6L, 323561596354L, 652330913564L, 334701175684L, 429147883659L, 353739376445L]
Time of Encrypt
Sun Aug 24 13:35:32 2014
Enter a filename ('my_letter') and select an action
Filename: C:\Python27\my_letter
Browse...
Show
Put&Calculation cryptotext
Quit BASTION

ნახ. 4.1.7

თავი5 კვანტური კრიპტოგრაფია, სტეგანოგრაფია

კვანტური კრიპტოგრაფია [23,24] (ხშირად ხმარობენ ტერმინს ინფორმაციის კვანტური თეორია) წარმოიშვა წინა საუკუნის 80-ან წლებში. თუმცა, ჯერ კიდევ 1970 წელს კოლუმბიის უნივერსიტეტის სტუდენტმა სტივენ ვიზნერმა (Stephen Wiesner) ჟურნალ IEEE Information Theory წარადგინა სტატია კოდირების თეორიაში მონაცემების დაცვის შესახებ, რისთვსაც იგი იყენებდა ნივთიერებების კვანტურ მდგომარეობებს. სტატია იმდენად ფანტასტიკური მოეჩვენა ჟურნალის რედაქციას, რომ იგი არ დაუბეჭდეს. შემდგომში მონრეალის უნივერსიტეტის მეცნიერმა ჯილ ბრასარდმა (Gilles Brassard) და IBM ფირმის თანამშრომელმა ჩარლზ ბენეტმა (Charles Bennett) დაამუშავეს კოდირებისა და შეტყობინების გადაცემის მეთოდი. პირველი პრაქტიკული ექსპერიმენტი ჩატარდა 1989 წ. გასაღებების კვანტური გადანაწილებაზე 30 სმ მანძილზე ღია ეთერში. თანამედროვეობაში მიღწეულია ოპტიკური სადენებით გადაცემა 184 კმ და მეტ მანძილზე. პირველი საწარმოო ეგზემპლარი შეიქმნა ფირმა GAP-Optique შვეიცარიელმა ინჟინრებმა. ფოტონების წყაროდ აქ გამოიყენებოდა ინფრაწითელი ლაზერი. მიღებულია გასაღებების კვანტური განაწილების ოქმი BB84, თუმცა ცნობილია დამხმარე ЭПР, Ло-Чу, B92 ოქმები.

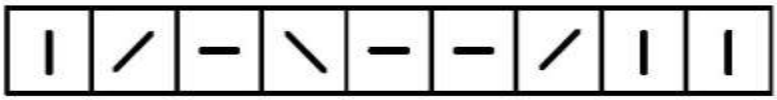
ზოგადად, კვანტური კრიპტოგრაფიის გასაცნობად, განვიხილოთ BB84 ოქმის ფუნქციონირება. აქ გამოიყენება ფოტონების 4 კვანტური მდგომარეობა, რომელიც ამოირჩევა გადამცემის მიერ გადასაცემი ბიტის მიხედვით ნახ. 5.1.



ნახ. 5.1

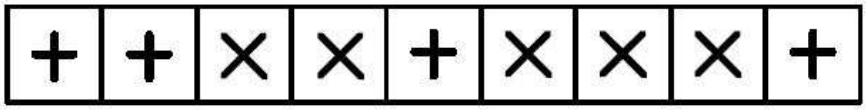
ვნახოთ, როგორ ხდება გასაღებების ფორმირება:

- გადამცემი ირჩევს შემთხვევით ერთ-ერთ ბაზისს, და ამ ბაზისში შემთხვევით ირჩევს ერთ-ერთ მდგომარეობას, რომელიც შეესაბამება 0 ან 1 და აგზავნის ფოტონებს ნახ .5.2 ;



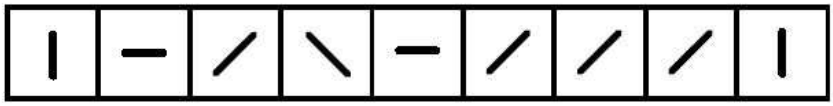
ნახ. 5.2

- მიმღები ,გადამცემისგან დამოუკიდებლად, ირჩევს ყოველი მიღებული ფოტონისთვის სწორხაზოვან (+) ან დიაგონალურ (x) ბაზის ნახ. 5.3 ;



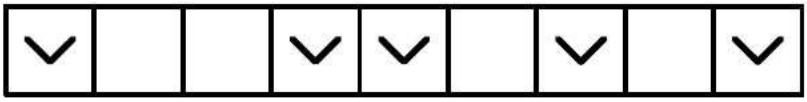
ნახ. 5.3

- მიმღები ინახავს გაზომვის შედეგებს ნახ. 5.4 ;



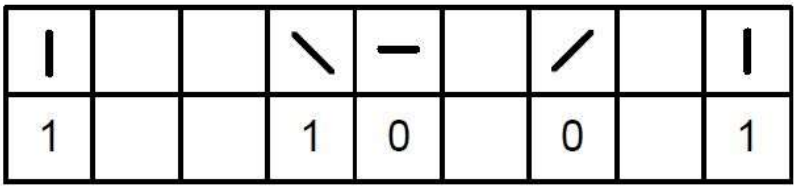
ნახ. 5.4

- მიმღები ღია არხით ატყობინებს გადამცემს, რა სახის გაზომვები იქნა ჩატარებული თითოეული ფოტონისათვის (ანუ თუ რა სახის ბაზისი ამოირჩა) , თუმცა გაზომვის შედეგებს საიდუმლოდ ინახავს ;
- გადამცემი ატყობინებს მიმღებს ღია არხით, თუ რა გაზომვები შეირჩა გადამცემის საწყისი ბაზისის მიხედვით ნახ. 5.5 ;



ნახ. 5.5

- მიმღები ტოვებს მხოლოდ იმ მდგომარეობებს, რომელშიც ამორჩეული ბაზისები ემთხვევა ერთმანეთს ნახ. 5.6.



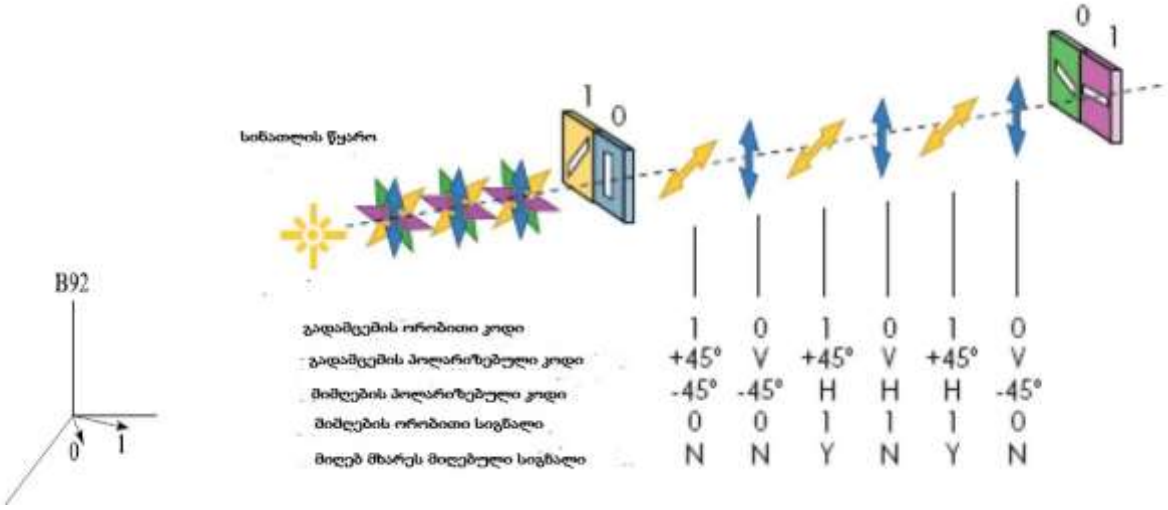
ნახ. 5.6

რაოდენობა, როცა ამორჩეული ბაზისები ერთმანეთს ემთხვევა, საშუალოდ ტოლია საწყისი თანამიმდევრობის ნახევრის. ე.ი. დამახინჯებებისა და ხელისშემშლელი ფაქტორების არ არსებობისას სწორად დარეგისტრირდება საშუალოდ 50% ფოტონი ნახ. 5.7.

გადამცემის ორობითი სიგნალი	0	1	0	1
გადამცემის პოლარიზებული სიგნალი	↔	↕	↘	↗
მიმღებში დატექტირებული სიგნალი	✣	✣	✣	✣
ორობითი სიგნალი მიმღებში	0	1	?	?

ნახ. 5.7.

ამ ნაკლის აღმოსაფხვრელად გამოიყენება სხვა და სხვა მეთოდი და ოქმი. მეტი ინფორმაციისათვის ნახ.5.8 წარმოდგენილია სხვა B92 ოქმის ფოტონების პოლარიზაცია და ოქმის მუშაობის ალგორითმი.



ნახ. 5.8.

<u>სტეგანოგრაფია</u>, ბერძნულად στεγανός ნიშნავს ფარულს, დამალულს. ეს ტერმინი პირველად იქნა ნახმარი ***იოგან ტრიტემიის*** მიერ 1499 წ. მის ნაშრომში *Steganographia*. არსებითი განსხვავება კრიპტოგრაფიასთან შედარებით მდგომარეობს შემდეგში [25,26,27]: თუ კრიპტოგრაფია მალავს შეტყობინებას, სტეგანოგრაფია ღიად გავრცელებულ შეტყობინებაში მალავს საიდუმლო შეტყობინებას, რომელიც უხილავია არადაშვებულ პირთათვის. ამ ტექნოლოგიის მეთოდის, გამოყენებისას, შეტყობინება გარეშე პირთა

ყურადღების ქვეშ არ ექცევა, რაც უთუოდ ზრდის დაცულობის ხარისხს. ამ მეთოდის გამოყენების პირველი შემთხვევები აღწერილია საბერძნეთში ძვ.წ. 440 წელს.

ცნობილია სტეგანოგრაფიის ოთხი სახე: კლასიკური, კომპიუტერული, ციფრული და ქსელური.

კლასიკური ძირითადად წარმოდგენილია სხვა და სხვა სახის მელნით.

კომპუტერული სტეგანოგრაფიისას ინფორმაცია იმალება გამოუყენებელი ფაილების ფორმატების სივრცეებში მაგ. StegFS ოპ.ს. *Linux*, აქ გამოყენებულია შემდეგი ტექნოლოგიები:

- ფაილების ფორმატში დარეზერვირებული (დაჯავშნული) ველების გამოყენება, ან დროს გაფართოების გამოუყენებელი ველი ივსება ნულებით, რომელიც საჭიროებისამებრ ივსება დასამალი ინფორმაციით. ნაკლია დამალულობის დაბალი დონე და დამალული ინფორმაციის მცირე სიდიდე. ინფორმაციის დამალვა დრეკადი დისკების გამოუყენებელ ნაწილებში. ეკრანზე გამოსახულებების ფორმატების სპეციალური თვისებების გამოყნება, მაგ. შავ ფონზე შავი შრიფტი.
- მყარ დისკზე ფაილური სისტემის თავისებურებების გამოყენება. მაგ. ფაილურ სისტემაში FAT32 ფაილს უჭირავს სტანდარტულად 4 კბტ. მოცულობა, რომელიც მას შეიძლება არც სჭირდებოდეს. ზედმეტი ივსება მალული ინფორმაციით. ნაკლია მისი აღმოჩენის სიადვილე.

ციფრული სტეგანოგრაფიის დროს მალული ინფორმაცია ინერგება ზოგადად ინფორმაციის ციფრულ ნაწილში. ასეთი ობიექტებია, ხშირად, გამოსახულებები, ვიდეო, აუდიო, ობიექტების ტექსტურა. ეს ჩანერგვები იწვევს საწყისი ობიექტების გარკვეულ სახეცვლილებებს, რომლებიც ჩვეულებრივი ადამიანების აღთქმის ორგანოებისათვის შეუმჩნეველია.

ნახ. 5.9

ამ მაგალითში, იხ. ნახ.5.9, პირველ სურათზე გადმოცემულ ხეების სურათში ჩამალულია კატის სურათი. ანალოგიური ტექნოლოგიის გამოყენებით საქართველოში შექმნილი იყო სტეგანოგრაფიული პროგრამა „რიფები“. აღნიშნულ მეთოდს იყენებდნენ რუსეთის საელჩოს თანამშრომლები აშშ-დან ქსელით კონფიდენციალური ინფორმაციის გადაცემისათვის.

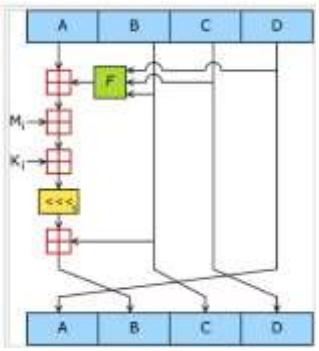
ამ ბოლო პერიოდისათვის გავრცელება პოვა ე.წ. ***ქსელურმა*** სტეგანოგრაფიამ. იგი პირველად შემოგვთავაზა ***კრიშტოპ შიპერსკიმ*** (*Krzysztof Szczypiorski)* 2003 წ. ამ დროს იცვლება ქსელში გამოყენებული ოქმების თვისებები. ცნობილია ასეთი პროდუქტები WLAN ქსელისთვის HICCUPS (Hidden Communication System for Corrupted Networks) და LACK (Lost Audio Packets Steganography) IP ტელეფონებში აუდიო ინფორმაციის გადაცემისას. სტეგანოგრაფიაში გამოყენებული ჩასაშენებელი მეთოდებია:

- **LSB** (Least Significant Bit) ნიშნადი უმცირესი ბიტი. ამ მთოდის არსი მდგომარეობს საწყისი ინფორმაციის (გამოსახულების, აუდიო ან ვიდეო) ბოლო ნიშნადი ბიტების შეცვლაში დამალული შეტყობინების ბიტებზე. ეს ცვლილება ადამიანის აღთქმის ორგანოებისათვის შეუმჩნეველი უნდა იყოს. მაგალითად, თუ გვაქვს ერთ ბაიტიანი შეტყობინება 01101011 და თუ ბოლო 2 ბიტს შევცვლით (დავუშვათ შავი ფერით) 00000001, 00000010, 00000010, 00000011, მაშინ ტონალობა საწყისი გამოსახულების პირველი ბიტისა შეიცვლება 1/255, მეორე და მესამესი 2/255 და მეოთხესი 3/255. კარგად მუშაობს JPEG ფორმატისათვის და ძლიერ არის დამოკიდებული არხის „ხმაურზე“ .
- ***ექო მეთოდი*** გამოიყენება ციფრულ აუდიო სტეგანოგრაფიაში. ექოს მახასიათებლებია საწყისი ამპლიტუტის სიდიდე, „ჩაქრობის“ ხარისხი, დაყოვნება. რაღაც დონის მიღწევისას ხდება აღრევა სიაგნალისა და ექოსი, რომელიც ადამიანის სმენით არ აღითქმება. ხშირად ტექნიკურად ამ დაყოვნების სიდიდეა 1/1000. ლოგიკური ერთიანისა და ნოლის აღსანიშნავად გამოიყენება ორი სხვა და სხვა დაყოვნება. ისინი უნდა იყვნენ მცირენი, რომ არ იქნენ აღთქმული ადამიანის მიერ.
- ***ფაზური კოდირება*** (phase coding). ამ დროს საწყისი ხმოვანი ელემენტი იცვლება ფარდობით ფაზაზე, რომელიც არის საიდუმლო \შეტყობინების ნაწილი.
- ***გაფართოებული სპექტრის*** მეთოდი. არსი მდგომარეობს შემდეგში. სპეციალური შემთხვევითი თანამიმდევრობა ჩაშენდება შეტყობინების კონტეინერში, ხოლო მიმღებ მხარს სპეციალური ფილტრით ხდება ჩაშენებული შეტყობინების აღდგენა.

§ 6 ჰეშ ფუნქცია

ასიმეტრიული შიფრაციის სისტემებში გადამწყვეტი სიტყვა ეკუთვნის ე.წ. ჰეშ ფუნქციას, რომელსაც ხშირად უწოდებენ კრიპტოგრაფიულ „ცალმხრივ ფუნქციას“. მისი არსი მდგომარეობს შემდეგში. ცნობილი არგუმენტის x-ის მიხედვით მარტივად გამოითვლება ფუნქციის მნიშვნელობა f(x), მაშინ როცა ფუნქციის ამ მნიშვნელობიდან შეძლებელია (ალბათურად ძალიან ძნელად გამოსათვლელია) x არგუმენტის მნიშვნელობის განსაზღვრა. გავეცნოთ თანამედროვე სისტემებში ყველაზე გავრცელებულ კლასიკურ სისტემებს MD5 და SHA-512 [27,28] (თუმცა ცნობილია სხვა სისტემებიც Whiripool, RIPEMD-160, HAVAL და სხვ).

MD5 (*Message Digest 5*). იგი არის 128-ბიტიანი ჰეშირების ალგორითმი, რომელიც შექმნა მასაჩუსესტის უნივერსიტეტის პროფესორმა *რონალდ რივესტრმა* 1991 წ. მისი ფუნქციონირების ალგორითმი შემდეგია ნახ. 6.1. იგი შედგება 5 ბიჯისაგან:



ნახ. 6.1

- **<u>ნაკადის გათანაბრება</u>**. ნაკადის ახალი ზომა ტოლია L^i =512[N+448];
- **<u>შეტყობინების სიგრძის დამატება</u>**. შეტყობინების ბოლოში ემატება 64 ბიტამდე (მონაცემთა სიგრძის მიხედვით) გათანაბრებამდე. პირველად დაემატება უმცროსი 4 ბაიტი, შემდეგ უფროსი. თუ სიგრძე ნაკადის მეტია 2^{64} -1, მაშინ დაემატება მხოლოდ უმცროსი ბიტები. ამის შემდეგ ნაკადის სიგრძე გახდება 512 -ის ჯერადი. ე.ი. გამოთვლები იწარმოებს 512 ბიტიან მასივებზე;
- **<u>ბუფერის ინციალიზაცია</u>**. ხდება ინიციალიზაცია 4 32-ბიტიანი ცვლადის :

A = 01 23 45 67; // 67452301h;

B = 89 AB CD EF; // EFCDAB89h;

C = FE DC BA 98; // 98BADCFEh;

D = 76 54 32 10. // 10325476h.

ხდება 4 რაუნდიანი გამოთვლები :

FunF(XYZ)=(X ^ Y) OR (notX^ Z);

FunG(X,Y,Z)=(X ^Y) OR(not Z^ Y);

FunH(X,Y,Z)=XxorYxorZ;

FunG(X,Y,Z)=(Y) xor((notZORX) .

- **ვსაზღვრავთ 64-ელემენტიან მონაცემთა ცხრილებს** T(n)=int(2^{32}. |sinn|). თითოეული 512-ბიტიანი ბლოკი გადის 4 ეტაპიან გამოთვლას თითოეული 16-რაუნდიანია. ბლოკი წარმოდგენილია 16 32-ბიტიანი სიტყვიანი მასივებით a=b+((a+Fun(b,c,d)+X(k)+T(i)<<<s). აქ k 32 ბიტიანი სიტყვის (მიმდინარე 512 ბიტიანი ბლოკიდან) ნომერია, ხოლო <<< ციკლური წანაცვლებაა მარცხნივ s ბიტით. ციფრი s თვითეული რაუნდისთვის განისაზღვრება ინდივიდუალურად.
- **გამოთვლები ციკლში**. ვღებულობთ AA=A BB=B, CC=C, DD=D. სრულდება 4 ეტაპი:

/* [abcd k s i] a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */;
[ABCD 0 7 1][DABC 1 12 2][CDAB 2 17 3][BCDA 3 22 4];
[ABCD 4 7 5][DABC 5 12 6][CDAB 6 17 7][BCDA 7 22 8];
[ABCD 8 7 9][DABC 9 12 10][CDAB 10 17 11][BCDA 11 22 12];
[ABCD 12 7 13][DABC 13 12 14][CDAB 14 17 15][BCDA 15 22 16];

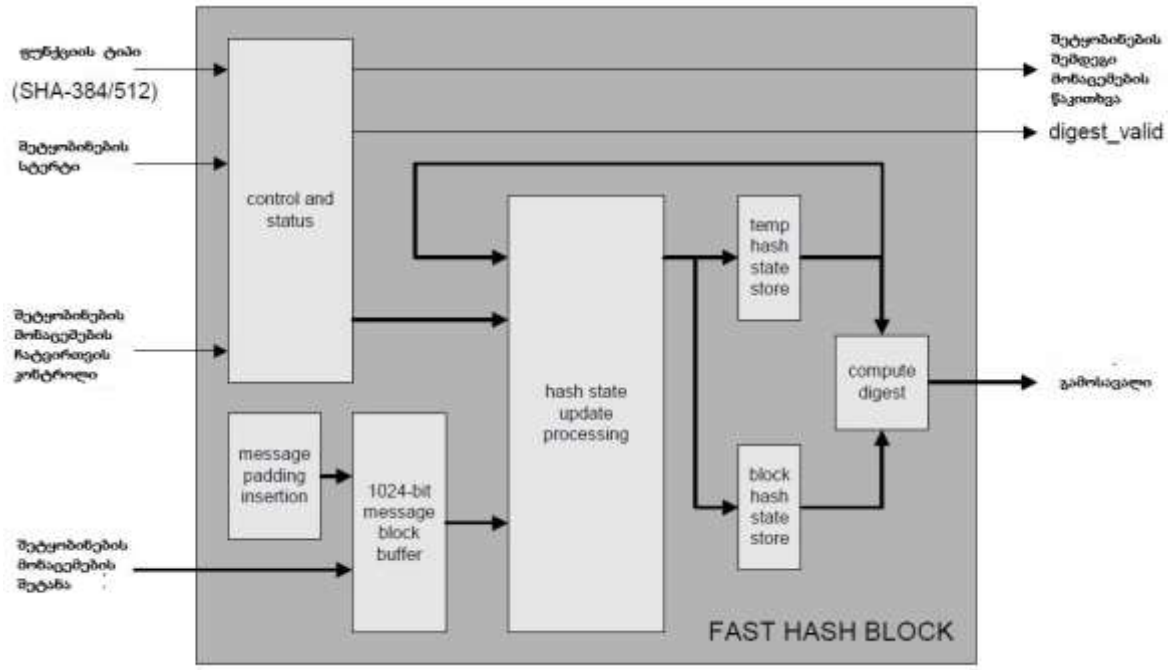
/* [abcd k s i] a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */;
[ABCD 1 5 17][DABC 6 9 18][CDAB 11 14 19][BCDA 0 20 20];
[ABCD 5 5 21][DABC 10 9 22][CDAB 15 14 23][BCDA 4 20 24];
[ABCD 9 5 25][DABC 14 9 26][CDAB 3 14 27][BCDA 8 20 28];
[ABCD 13 5 29][DABC 2 9 30][CDAB 7 14 31][BCDA 12 20 32];

/* [abcd k s i] a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */;
[ABCD 5 4 33][DABC 8 11 34][CDAB 11 16 35][BCDA 14 23 36];
[ABCD 1 4 37][DABC 4 11 38][CDAB 7 16 39][BCDA 10 23 40];
[ABCD 13 4 41][DABC 0 11 42][CDAB 3 16 43][BCDA 6 23 44];
[ABCD 9 4 45][DABC 12 11 46][CDAB 15 16 47][BCDA 2 23 48];

/* [abcd k s i] a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */;
[ABCD 0 6 49][DABC 7 10 50][CDAB 14 15 51][BCDA 5 21 52];
[ABCD 12 6 53][DABC 3 10 54][CDAB 10 15 55][BCDA 1 21 56];
[ABCD 8 6 57][DABC 15 10 58][CDAB 6 15 59][BCDA 13 21 60];
[ABCD 4 6 61][DABC 11 10 62][CDAB 2 15 63][BCDA 9 21 64].

ავჯამავთ მიღებულ შედეგებს A=AA+A, B=BB+B, C=CC+C, D=DD+D. მიღებული შედეგი ანუ ჰეშ ფუნქცია ინახება ABCD ბუფერში.

ცნობილია SHA ალგორითმის სხვადასხვა ვარიანტი SHA-0, SHA-1, SHA-2, SHA-3, რომლებიც განსხვავდება ერთმანეთისაგან პარამეტრებით. განვიხილოთ SHA 512 (SHA-3), ფუნქციონირების ალგორითმი, რომელიც გაცილებით რთულია ვიდრე MD5. გამოთვლის შედეგად მიიღება 512-ბიტიანი ჰეშ ფუნქცია (8 x 64), ბლოკის სიგრძე 1024 ბიტი, მაქსიმალური შეტყობინების სიგრძე (2^{64} – 1) ბიტი, რაუნდების რაოდენობა 64, გამოყენებულია ოპერატორები Add (mod 2^{32}), Or, Shr, დაცვა 256 ბიტი, ნახ.6.2 მოცემულია **Fast SHA-384/512** ჰეშირების ბირთვის ბლოკ-დიაგრამა.



ნახ. 6.2.

ქვემოთ ნაჩვენებ მაგალითში ჩანს, როგორ იცვლება 100%-ით SHA512 ალგორითმით მიღებული ჰეშ ფუნქცია, თუნდაც ერთი სიმბოლოს რეგისტრის ცვლილებით საწყის ინფორმაციაში.

Dato Adamia

c988ebe575415cefcc0c6354d4d0133962a4c105c5948c612af31440558ce3d95925904f099b7d8

650412f237eaff31a09c6e32f07f429a23679dc51353b1d8f

Dato adamia

666144f4d0223a99320ba65265f1907a75e66f33b0d8fd42688dbb03e464eeb41629ef9e7276751

0998afeb88a058b7d14a9341b33497a752005caa0aca42256

თავი 7 ზოგი გადაწყვეტილებები თანამედროვე გამომთვლელ სისტემებსა და საინფორმაციო ტექნოლოგიებში ინფორმაციული უსაფრთხოების პრაქტიკული უზრუნველყოფის თვალსაზრისით

§ 7.1. ინფორმაციული უსაფრთხოების საკითხის გადაწყვეტა *cloud computing* „ღრუბლოვანი გამოთვლების“ *არქიტექტურაში*

განმარტებით „ღრუბლოვანი გამოთვლები“ არის მოთხოვნის საყოველთაო და მოხერხებული ქსელური შეღწევის უზრუნველყოფის მოდელი, რომლითაც უზრუნველყოფილია საერთო კონფიგურებად გამოთვლით რესურსებთან წვდომა, რომლებიც ოპერატიულად მიეწოდება შესაბამისი პროვაიდერების მიერ. აღნიშნული არქიტექტურა პირველად შემოთავაზებული იყო გამოჩენილი ამერიკელი ინფორმატიკოსის *ჯონ მაკარტის* (John McCarthy) მიერ 1960 წ. ეს ურთულესი არქიტექტურაა, მაგრამ მისი ძირითადი კომპონენტებისაგან, (ინფორმაციული უსაფრთხოების საკითხთან მიმართებით), გამოვყოთ მომსახურების მოდელები *SaaS (Software-as-a-Service), PaaS* (*Platform-as-a-Service), IaaS (IaaS or Infrastructure-as-a-Service), DaaS* (Desktop as a Service), *WaaS* (Workspace as a Service), *EaaS (*Everything as a service).

მიუხედავად უდიდესი უპირატესობებისა და მოხერხებულობებისა, „ღრუბლოვანი გამოთვლებში“ არქიტექტურის მიმართ არსებობს გარკვეული კითხვები ინფორმაციული უსაფრთხოების მიმართ:

- მიზანშეწონილი არაა ვებ სერვისების გამოყენება საკუთარი გამოთვლითი რესურსების საწარმოებლად, რადგან თქვენ კარგავთ მასზე კონტროლს. აწარმოეთ გამოთვლები თქვენს კომპუტერზე ლიცენზირებული პროგრამების საშუალებით. საწინააღმდეგო შემთხვევაში შეიძლება ქსელური პროგრამების შემქნელის ხელში აღმოჩნდეთ;
- არის საშიშროება, რომ ამ ტექნოლოგის საყოველთაო გავრცელების შემთხვევისათვის, ჩნდება საშიშროება უკონტროლო მონაცემთა შექმნისა, როცა მომხმარებლის მიერ დატოვებული ინფორმაცია ინახება წლობით მისი ნებართვის გარეშე ან ისე, რომ მას არ შეუძლია მისი სახეცვლილება;
- ჯერჯერობით არაა შექმნილი ისეთი საშუალებები, რომლებიც საშუალებას მისცემენ ინფორმაციის მფლობელებმა წაშალონ (ამოიღონ) საკუთარი ინფორმაცია სერვერებიდან.

- ანალიტიკები ვარაუდობენ, რომ მომხმარებელთა ზრდასთან ერთად გაიზრდება შეცდომები და ინფორმაციის დაკარგვის ალბათობებიც.

ზემოთქმულის დასადასტურებლად 2015 წ. აგვისტოში გავრცელებულია ინფორმაცია იმის შესახებ, რომ აშშ-ის სასამართლომ დაავალდებულა Microsoft ფირმა რათა მას გადაეცა მთავრობისთვის ინფორმაცია მათი კლიენტების შესახებ. Amazon Webservices ცნობით 2011 წ. სისტემურმა ადმინისტრატორმა თვითნებური მოქმედებით ერთ–ერთ მომხმარებელს მიაყენა ზარალი 300000 დოლარის ოდენობით.

მიღებული იქნა მთელი რიგი დირექტივა აღნიშნული საკითხების გადასაწყვეტად:

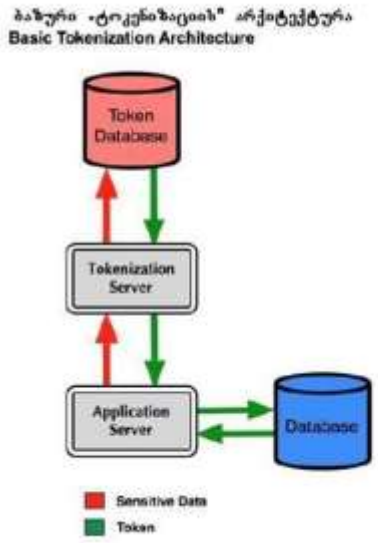
- დირექტივა 95/46/EC მონაცემთა დაცვაზე;
- დირექტივა კონფიდენციალობაზე 2002/58/EC (ახალი რედაქცია 2009/136/EC) Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector;
- დირექტივა EU სივრცეში გამოყენებულ გამოთვლით ტექნიკაზე Article 29 Working Party Working document on determining the international application of EU data protection law to personal data processing on the Internet by non-EU based web sites (WP56 of 30 May 2002)
- დირექტივა EU სივრცის გარეთ გამოყენებულ გამოთვლითი ტექნიკაზე, რომელიც იყენებს EU სივრცეში განთავსებულ გამოთვლით ტექნიკას .

http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2010/wp179_en.pdf.

ძირითადი საკვანძო გადასაწყვეტი ამოცანებია:

- ვირტუალიზაციის გარემოში შეღწევის მართვა;
- ორ ფაქტორიანი მომხმარებლის აუტენფიკაცია;
- ადმინისტრატორების მოქმედებების მონიტორინგი;
- კონფიგურაციების ცვლილებების კონტროლი;
-ვირტუალური მანქანების ბაზური სერვისების დაცვა ისეთი საშუალებებით (ანტივირუსული პროგრამები), როგორიცაა ქსელთაშორისი ეკრანები, შეჭრის აღმოჩენა, მთლიანობაზე კონტროლი და სხვა.

ზოგადად ფირმა RISSPA (www.riispa.ru) მიერ კლასიფიცირებული იქნა ინფორმაციული უსაფრთხოების მხრივ „ღრუბლოვანი გამოთვლებში“ არსებული პრობლემები და შემოთავაზებული იქნა ამოცანების გადაწყვეტის შესაბამისი წინადადებები [49,50]. იხ. ნახ. 7.1.1.



გადაწყვეტილებები

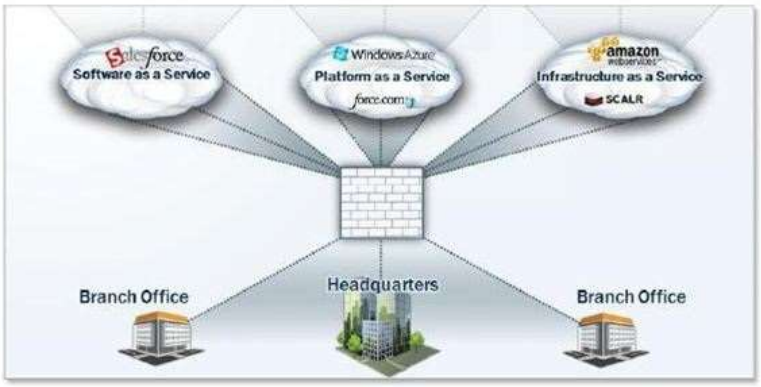
1) Open Tokenization Server (http://opentokenization.codeplex.com/)
2) RSA Tokenization Server (www.rsa.com/services/pdfs/10160_TOKEN_DS_0410.pdf)
3) Paymetric's tokenization solution, XiSecure™ (http://www.paymetric.com/solutions/xisecure-on-demand/)
4) Safenet (http://www.safenet-inc.com/solutions/data-protection/tokenization/)

ნახ 7.1.1

შენიშვნა: „ტოკენიზაცია" ნიშნავს კონფიდენციალურ მონაცემთა მნიშვნელობების შეცვლას სხვა მონაცემებზე (ტოკენებზე).

RISSPA
be professional

„ღრუბელში" გაჟონვის კონტროლი

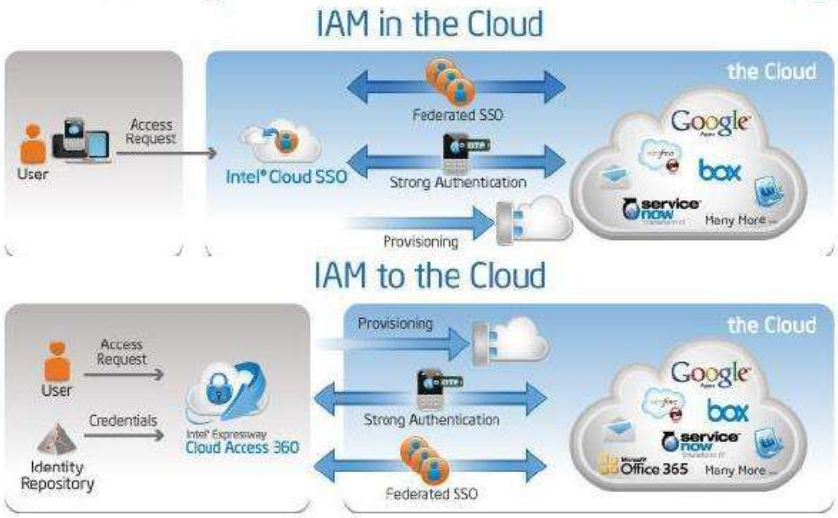


გადაწყვეტილებები

1) Websense (http://www.websense.com/content/cloud-solutions.aspx)
2) Spamina (http://www.spamina.com/eng/products.php?pob=CloudEmailEncryption)
3) BEW GLOBAL (http://www.bewglobal.com/news/clouddlp)

ნახ 7.1.2

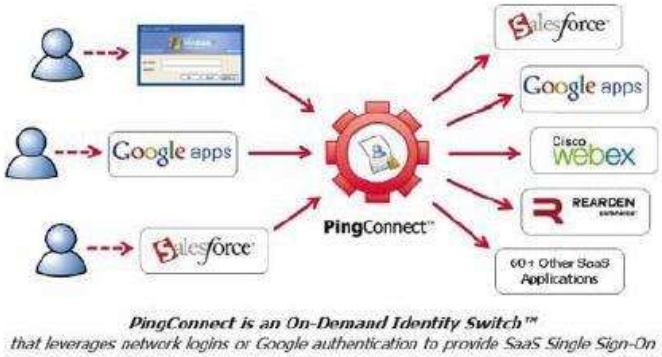
შეღწევის კონტროლი



გადაწყვეტილებები

1) Intel XML Gateway, Application Security & Cloud Identity (http://software.intel.com/en-us/articles/XML-Gateway-Application-Security-Cloud-Identity/)
2) SecureAuth (http://www.gosecureauth.com/)

მიმართვის კონტროლი



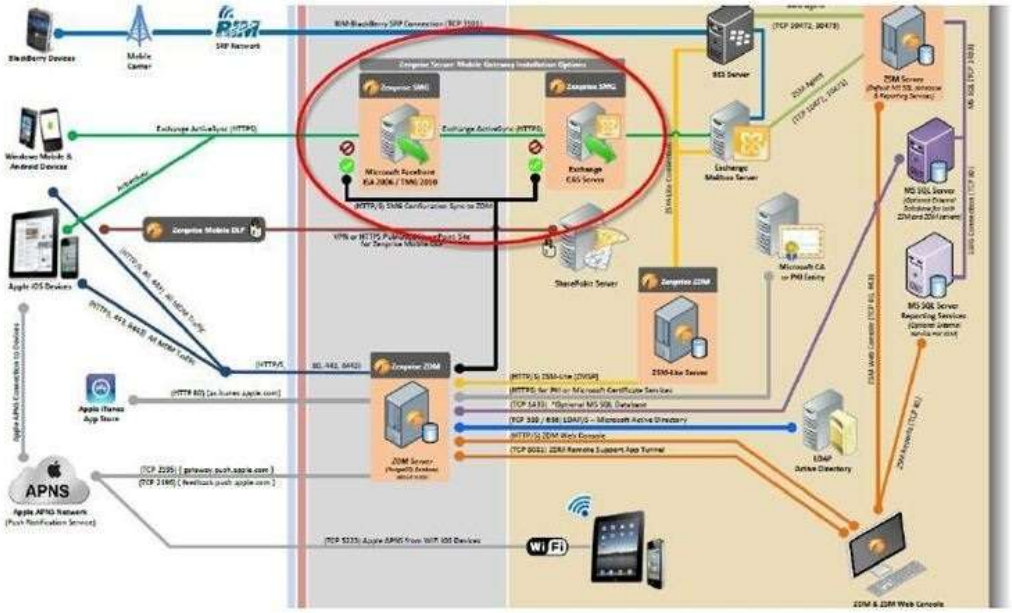
გადაწყვეტილებები

1) Ping Identity (https://www.pingidentity.com/)
2) Simple Cloud Identity Management (http://www.simplecloud.info/)
3) Symplified (http://www.symplified.com/)

ნახ.7.1.3

RISSPA
be professional

მობილური მოწყობილობების მართვა



RISSPA
be professional

მონიტორინგი

გადაწყვეტილებები

1) Splunk>Storm (https://www.splunkstorm.com)
2) Absolute Performance Monitoring (http://www.absolute-performance.com/solutions/cloud-monitoring)
3) Nimsoft Monitor (http://www.nimsoft.com/solutions/nimsoft-monitor/cloud.html)

ნახ7.1.4

ამ არქიტექტურაში უმნიშვნელოვანი ადგილი უჭირავს ე.წ. სერვერული ვირტუალიზაციის ფენას, რომელიც განთავსებული პროგრამული მომსახურებისა და აპარატურული უზრუნველყოფის ფენებს შორის. მისი დადებითი მხარეებია: სისწრაფე ახალი სერვერების განთავსებაში, სარეზერვო კოპიების სწრაფი შექმნა, განახლებების ტესტირება, დატვირთვების დაბალანსება და სხვა. აქ ხშირად არაა მისაზანშეწონილი ან შეუძლებელია ტრადიციული მეთოდებით კონტროლის ჩატარება (მაგ. ტრადიციული ანტივირუსული ტესტების გაშვება). შესაბამისად „ღრუბლოვანი გამოთვლებში“ სერვერული ვირტუალიზაციის გავლენის შესწავლა ინფორმაციულ უსაფრთხოებაზე მნიშვნელოვანი საკითხია. აქ გამოყოფენ რამდენიმე საკითხს (VMware vSphere მაგ-ზე):

ა) **ჰიპერვიზორი**. არის პროგრამა ან აპარატურული გადაწყვეტა, რომელიც უზრუნველყოფს სხვა და სხვა ოპერაციული სისტემის პარალელურ, ერთდროულ შესრულებას ერთ ჰოსტ კომპიუტერზე. იგი განაცალკევებს სხვა და სხვა ოპერაციულ სისტემებს, დაცვას და უსაფრთხოებას, რესურსების განცალკევებას გაშვებულ ოპერაციულ სისტემებს შორის და რესურსების მართვას. ჰიპერვიზორის კომპრომეტაცია იწვევს ყველა ვირტუალური მანქანების კომპრომეტაციას. კონკრეტულ ფიზიკურ მოწყობლობასთან მომუშავე ჰიპერვიზორის კოდით ან ერთად შეიძლება გაშვებული იქნას სხვის მიერ შექმნილი კოდიც, რომლის კონტროლიც ხშირად ვერ ხერხდება. ძირითადი საშიშროებია:

– ბუფერის გადავსება და ჰორიზონტალური კოდის გამოძახება. ცნობილი ასეთი დაუცველობის სახეებია CVE-2012...1516…1517, CVE-2048…2460 (შესასრულებელი მცდარი ბრძანების გამო, რის გამოც გადაივსება მახსოვრობა სხვა კოდის შესრულებით). CVE-2013-3657 (დაშორებული მომხმარებალი აგზავნის სპეციალურ პაკეტს, რომელიც იწვევს გადავსებას და სხვა კოდის შესრულებას ან მომსახურების შეწყვეტას.

- ვირტუალურ მანქანაში მომხმარებლის უფლებების გაზრდა. ძირითადად იგი ხორციელდება ჰიპერვიზორის დაუცველობებთან ერთად წყვილში. ცნობილია შემდეგი მაგალითები:

* CVE-2012-1666 .დაუცველობა განპირობებულია VMware Tools-ითი, ავირუსებს ფაილს tpfc.dll., რის შედეგად ამაღლებს შეღწევის დონეს „სტუმარ“ ოპერაციული სისტემის მომხმარებლისა;

* CVE-2012-1518. იგივე, როცა VMware Tools-ის ბუფერი გადაივსება და როცა უფლებები არასწორად არის გაწერილი;

- მომსახურებაზე უარის თქმა. შედარებით ნაკლებ მნიშვნელოვანია, თუმცა მოქმედებს სისტემაში შეღწევის მახასიათებელზე, ნეგატიურად მოქმედებს „ღრუბლოვანის“ მომსახურე პროვაიდერზე. მაგალითად:

* CVE-2013-5970 .სერვისი hostd-vmdb მწყობრიდან გამოყავთ გაგზავნილი სპეციალურად შექმნილი პაკეტით;

* CVE-2012-5703. გარე სამსახურებთან (vSphere API) მომუშავე API, რომლის დროსაც ხდება მტყუნება API შეკვეთებზე;

ბ) კონსოლი / მართვის სერვერი. „ღრუბლოვან გამოთვლებში“ მონაწილეობს ძალიან დიდი რაოდენობა ვირტუალური მანქანები. ეს კი თხოულობს სპეციალურ მართვის სისტემას, რომელიც თვალყურს ადევნებს ახალი ვირტუალური მანქანების შექმნას, გადატანას და მათი გაუქმების პროცესს. ამ მართვის სისტემაში არასანქცირებულად ჩართვამ შეიძლება გამოიწვიოს „უხილავი“ ვირტუალური მანქანანები, რომლებიც ბლოკირებას უკეთებს ზოგიერთს, ხოლო ზოგიერთს ცვლის. ამის თავიდან ასაცილებლად საჭიროა:

- კონფიგურაციების ცვლილებების კონტროლი;
- ქსელში შეღწევის შეზღუდვა (შეღწევა ქსელის მხოლოდ გარკვეული სეგმენტებიდან ან მართვის სერვერის განთავსება ცალკე სეგმენტში);
- უზრუნველვყოთ რეგულარული განახლებები;
- ხიფათების (სისუსტეების) სკანირება;
- ჟურნალის და მონიტორინგის წარმოება.

გ) ვირტუალური მანქანა და დანართები . აქ სტანდარტული მეთოდები და საშუალებები ხშირად გამოუყენებადია. მაგრამ ვირტუალიზაციის პლატფორმების განვითარება ამ ნაკლს ხანდახან დადებით მხარედ აქცევს. მაგალითად, პლატფორმა VMware შეიცავს ინტერფეისების ნაკრებს Vmsafe, რომელსაც შეუძლია ურთიერთობა სხვების მიერ დამუშავებულ დაცვის სისტემებთან. ამ მიმართულებით დაცვის უზრუნველსაყოფად საჭიროა:

- ანტივირუსული დაცვა მკაცრად განსაზღვრული გაშვებების გრაფიკით;
- ვირტუალური მანქანების დაყოფა ნდობის ზონებად;
- პროგრამული უზრუნველყოფის დროული განახლებები, პერიოდული სკანირება ხიფათების (საშიშროებების) და ინფორმაციული უსაფრთხოების მონიტორინგი;
- უზრუნველვყოთ დაცვის საშუალებების განახლება.

დ) ქსელური ურთიერთ მოქმედებები. დავუშვათ, რომ ბოროტგამზრახველთა ჯგუფმა მოახერხა ვირტუალურ სერვერზე შესვლა „გვირაბოვანი“ ტექნოლოგიის გამოყენებით ან მიიღო ლეგალური გზით ამის განხორციელების შესაძლებლობა ვირტუალურ სერვერებზე, რომლებიც ხშირად შეუმჩნეველია. ამ დროს დაცვა ხორციელდება შემდეგნაირად:

- ძალიან ეფექტურია ვირტუალური შეჭრის საწინააღმდეგო სისტემის მოდულის (VirtualAppliance) გამოყენება და ქსელთაშორისი ეკრანის გამოყენება. ასეთი მოდულები შეიძლება ჰქონდეს ვირტუალიზაციის პლატფორმას ან გამოყენებული იქნეს ცალკეული ფირმების პროდუქცია (Juniper, Check Point);
- ნდობის სხვადასხვა ზონაში განთავსებული ვირტუალური მანქანების იზოლაცია;
- ვირტუალიზაციის პლატფორმის პერიმეტრის ქსელური დაცვა.

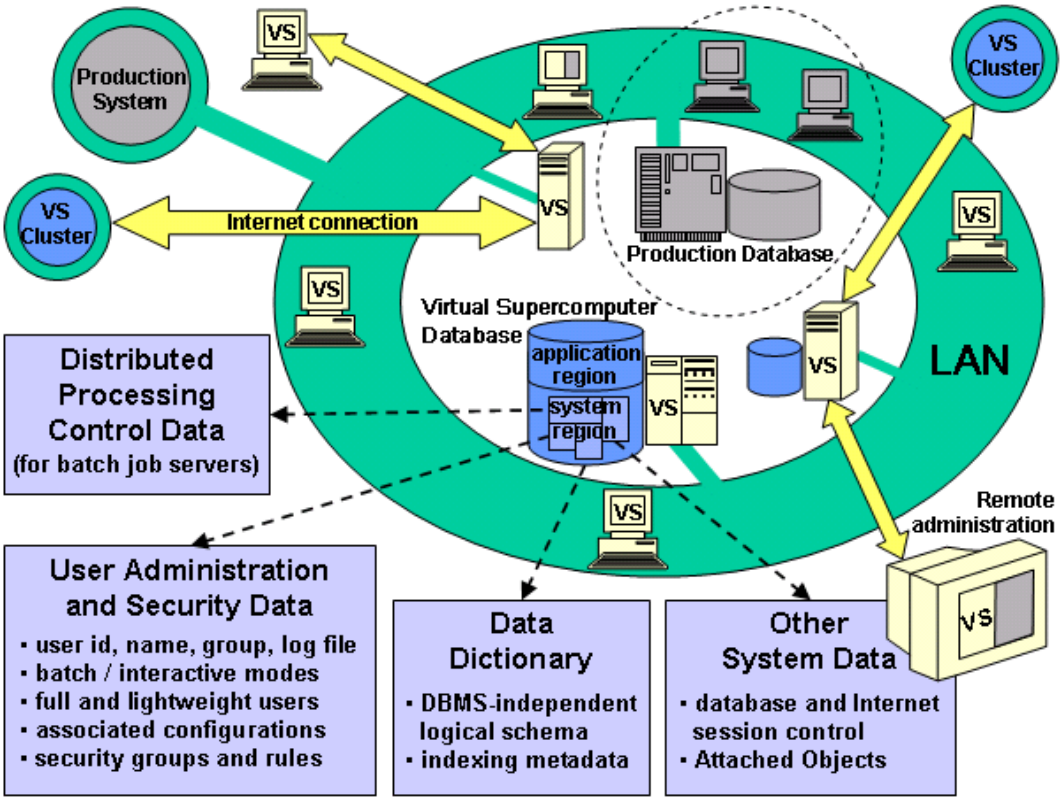
ე) ადმინისტრაციული პრივილეგიები. საწყის ეტაპზე, როცა ხდება არქიტექტურის დადგენა–შექმნა, ხშირად მასში არ მონაწილეობენ ინფორმაციული უსაფრთხოების სპეციალისტები, რაც ხდის შემდგომში უფლებამოსილებების გამიჯვნის ორგანიზაციას შეუძლებელს. ეს ვლინდება ისეთი მანიპულაციების ჩატარებისას, როგორებიცაა კლონირება, კოპირება. ე.ი. აქ საჭიროა უსაფრთხოების სამსახურის ჩართვა თავიდანვე და ადმინისტრირებაში უფლებამოსილებების მკაცრი გამიჯვნა.

ვ) **აუდიტი და ანგარიშგება.** აქაც ამოცანაა, მოხდეს გამართვათა აუდიტი და ვირტუალიზაციის პლატფორმის ლოგფაილების ანალიზი. საწინააღმდეგო შემთხვევაში ბოროტგამზრახველს ხელ-ფეხი გახსნილი აქვს.

§ 7.2 უსაფრთხოების უზრუნველყოფა Grid computing არქიტექტურაში

Grid computing - Grid გამოთვლები (ინგლისურად იგი ნიშნავს ცხაურს, ქსელს). ეს არის ფორმა განაწილებული გამოთვლებისა, რომელშიც „ვირტუალური სუპერკომპიუტერი“ წარმოდგენილია ქსელით შეერთებული კლასტერების სახით შემდგარი ერთგვაროვანი, ერთმანეთთან სუსტ კავშირში მყოფი კომპიუტერებისაგან, რომლებიც ერთად ამუშავებენ უდიდეს ამოცანებს.

Grid computing არქიტექტურა ეს არის გეოგრაფიულად განაწილებული ინფრასტრუქტურა, რომელიც აერთიანებს უამრავ სხვადასხვა სახის რესურს (პროცესორები, ოპერატიული და მუდმივი მახსოვრებები, მონაცემთა ბაზები, ქსელები, მონაცემთა დამგროვებლები) [51]. პრაქტიკული გამოყენების არეალი ძალიან ფართოა: კოსმოსი, არქიტექტურა, სამეცნიერო კვლევები, ელექტრონიკა (პროექტირება), ენერგეტიკა, ფინანსები, ფარმაცევტიკა და სხვა. იხ. ნახ. 7.2.1.



ნახ. 7.2.1

Grid computing არქიტექტურის უარყოფითი მხარეებია:

- იგი არ არის იდეალური, რეალურ დროში, დიალოგიურ რეჟიმში შეკითხვების დასამუშავებლად ;
- პროგრამული საშუალებების დამუშავება საჭიროებს დახვეწას.

დანიშნულების მიხედვით იგი იყოფა :

- გამოთვლითი;
- საინფორმაციო;
- კოლაბორაციული (ურთიერთდამოკიდებული დიდი რაოდენობის, რთული მომხმარებლებისთვის, რომლებიც ახორციელებენ მოდელირებისა და პროექტირების ამოცანებს) ;
- სამთავრობო;
- გამოყენებითი.

Grid computing არქიტექტურის შექმნის ძირითადი საფუძვლებია:

- სტანდარტი „WS-Provisioning";
- რესურსების მართვა (ვებ-სერვისები) WS-RF;
- მოთხოვნები უსაფრთხოების უზრუნველყოფაზე WS-Security, WS-SecureConversation, WS-Trust, WS-Federation, Kerberos;
- მონაცემების დამუშავება WSDL, UDDI, WS-Policy;
- ინტეგრირების უზრუნველყოფა OGSA;
- ბიზნეს პროცესების მართვის ზედა პროგრამული დონე BPEL4WS;
- სტანდარტები WS-Notification (მხარდაჭერილია IBM/HP მიერ) და WS-Eventing (მხარდაჭერილია Microsoft მიერ);
- საკომუნიკაციო პროგრამა SOAP, WSDL, და UDDI;
- მონაცემთა ერთდროულად გამოყენების ენა XML;
- შეტეყობინება საიმედო გადაცემებზე WS-Reliable Messaging;
- ფუნდამენტური განმსაზღვრელი სტანდარტია OGSA (Security Architecture for Open Grid Services).

უსაფრთხოების მხრივ Grid computing არქიტექტურას მოეთხოვება შემდეგი [52]:

<u>პერიმეტრის უსაფრთხოების სისტემის არსებობა</u>. ბევრი ამოცანა ითხოვს, რომ მრავალი გამოყენებითი პროგრამები სრულდებოდეს არა მარტო საკუთარი firewall არეში. ეს გულისხმობს იმას, რომ ხდება სხვადასხვა ორგანიზაციის firewall ზონების ურთიერთ გადაკვეთა. ამ შემთხვევაში, დაცული უნდა იქნეს მოქმედი სტანდარტი OGSA.

<u>იდენტიფიკაცია, ავტორიზაცია;</u>

<u>შიფრაცია;</u>

<u>უსაფრთხოების მრავალ ინფრასტრუქტურას</u>. გამოყენებითი და ქსელური დონეების Firewall-ბი;

<u>სერტიფიკაცია</u>. აქ ძირითადი მოთხოვნები გათვალისწინებულია სტანდარტით X.509

<u>აუტენთიკაცია;</u>

<u>უფლებების გადაცემა;</u>

ერთჯერადი შესვლა. აქ ნაგულისხმევია, რომ სუბიექტმა რომელმაც გაიარა აუტენფიკაციის პროცესი თავისუფლდება ამ პროცესიგან შემდგომი მცდელობების შემთხვევაში გარკვეული პერიოდის განმავლობაში;

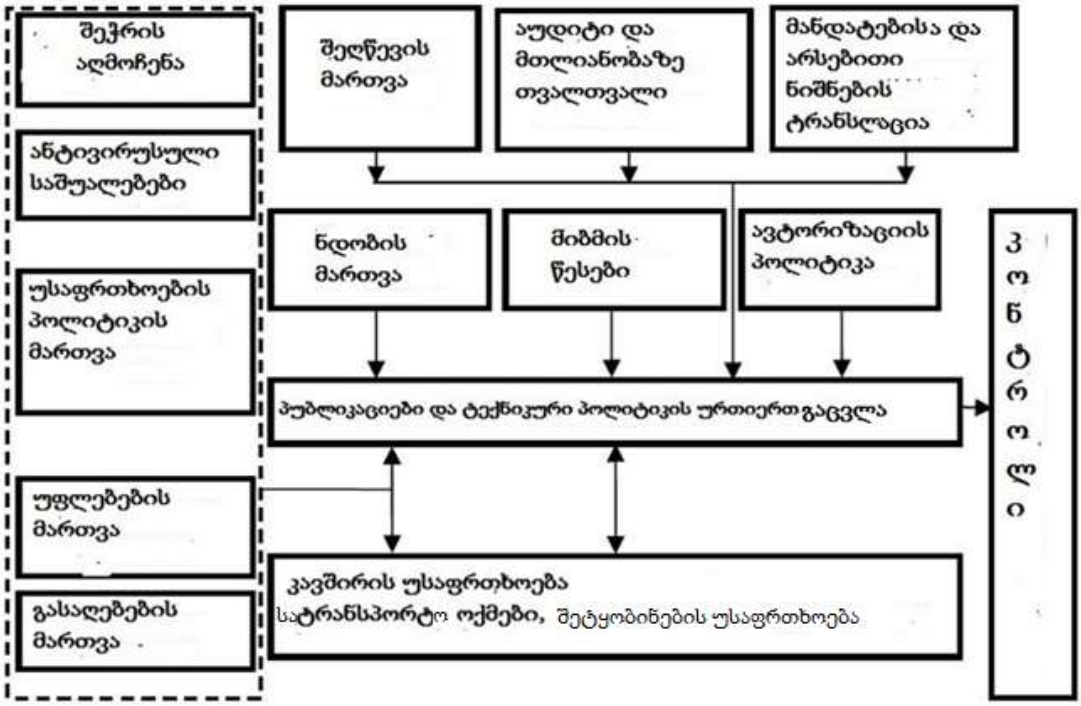
მანდატების მოქმედების ციკლი და მათი განახლება;

კონფიდენციალობა;

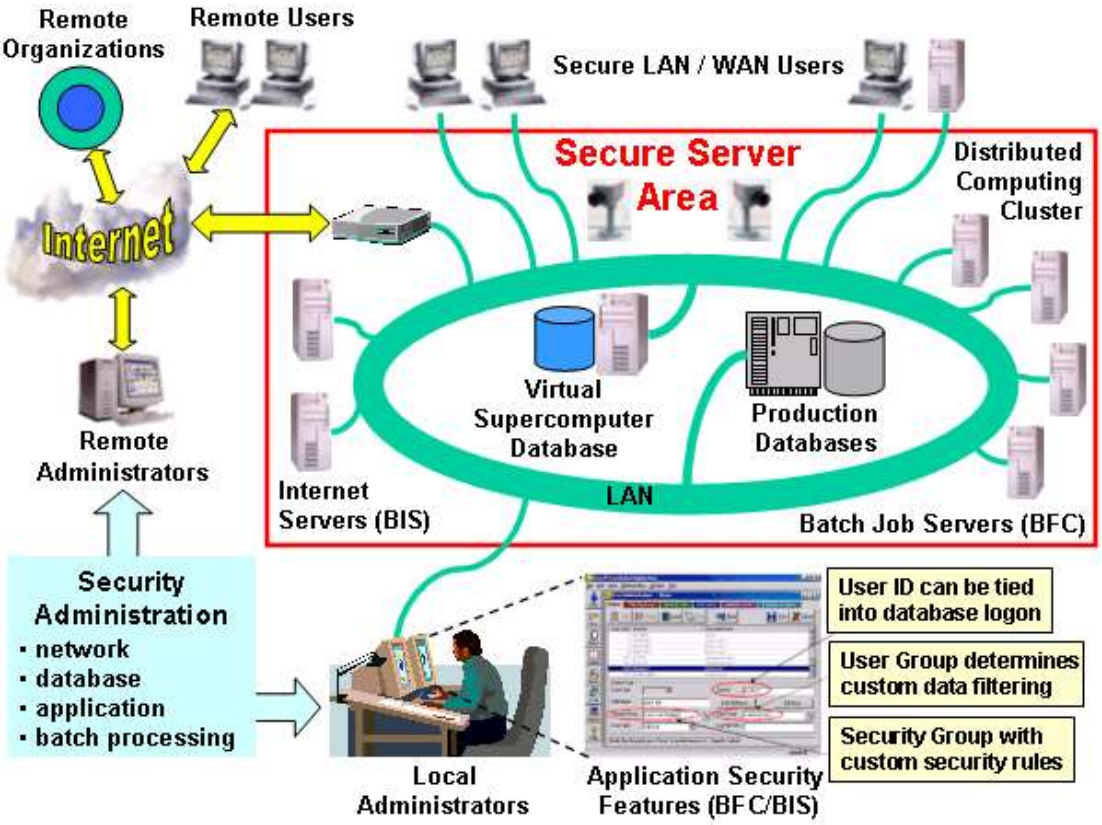
მონაცემთა მთლიანობის უზრუნველყოფა;

ქსელური ეკრანების გამჭვირვალება.

ზოგადად მიღებულია, რომ Grid computing არქიტექტურის უსაფრთხოების სისტემა შედგება ნახ. 7.2.2. და 7.2.3. მოცემული ქვესისტემებისაგან.



ნახ.7.2.2.



ნახ .7.2.3.

Grid computing არქიტექტურის ეფექტური უსაფრთხოების უზრუნველსაყოფად ზოგიერთების მიერ შემოთავაზებულია ე.წ. კვანძებს შორის ნდობისა და ინფორმაციის რანჟირების მეთოდი. მისი არსი შემდეგშია. მიღებულია, რომ *Grid computing* არქიტექტურა შედგება უამრავი კვანძისაგან და ქსელისაგან. ყველა ახალი კვანძი ან ქსელი, რომელიც ერთვება *Grid computing* არქიტექტურაში საწყის ეტაპზე აქვს ნულოვანი ნდობა. ნდობის საკითხი ორმხრივია:

- ძველი, არსებული კვანძების ნდობა ახლის მიმართ;
- ახალი კვანძის პოზიციიდან.

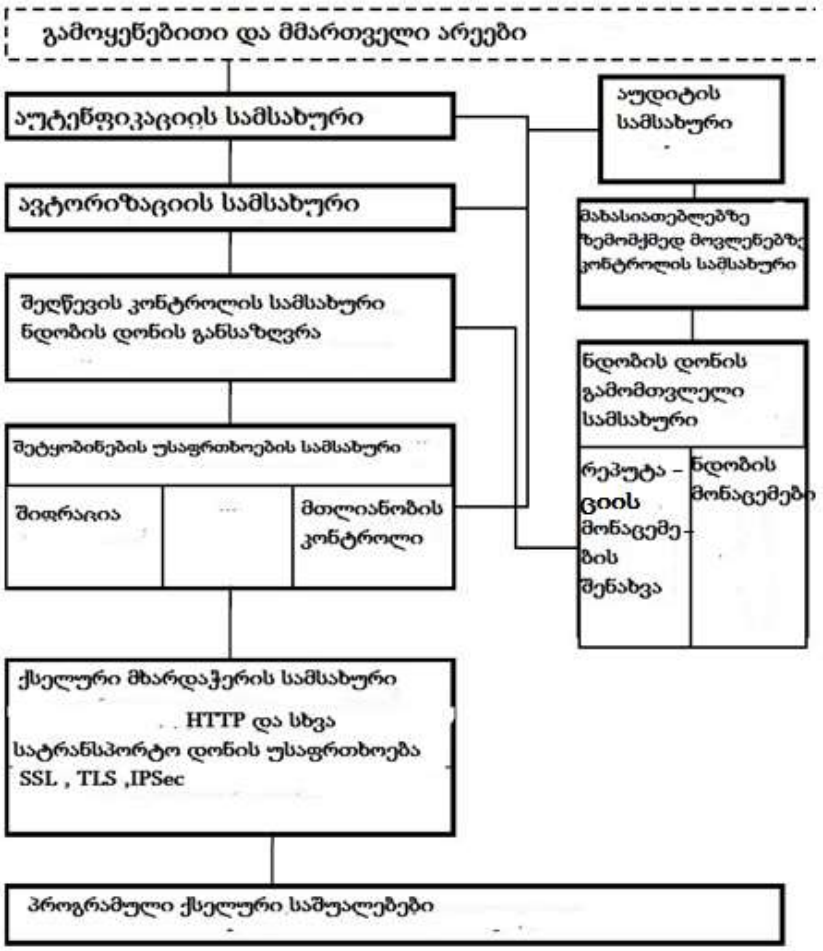
საწყის ეტაპზე, რა თქმა უნდა, არის გარკვეული სიფრთხილე ახალი კვანძისთვის საიდუმლო, კონფიდენციალური, ინფორმაციის გაცემაში. აქ ხდება ე.წ. დეზინფორმაციის მეთოდი, რომლის გადაცემით ახალ კვანძში, ანალიზი უკეთდებაა მის რეაქციას. ნდობის მახასიათებელი მერყეობს 0 და 1-ს შორის და გამოითვლება ფორმულით:

$$T_j = \frac{\sum_{j=1}^{n} \left(1 - W_i\right)}{n_j},$$

სადაც j კვანძის მიერ n_j ტრანზაქციების რაოდენობაა, ხოლო W_i კვანძის რეპუტაციის ვექტორის i-ური მნიშვნელობაა.

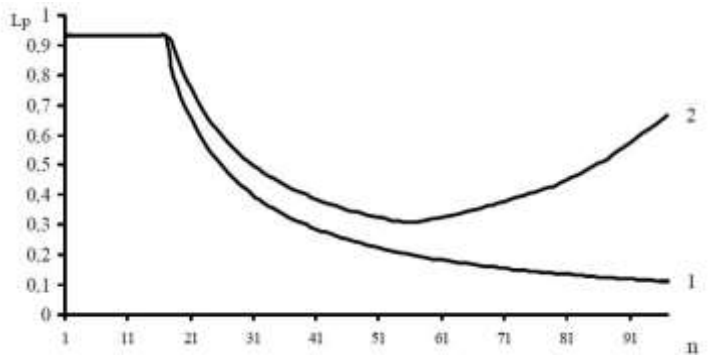
რანჟირების მეთოდის გამოყენებისას ხდება ინფორმაციის რანჟირება მისი ფასეულობის მიხედვით. კვანძში ინფორმაციის გადაცემამდე ხდება ინფორმაციის რანგის და ამ კვანძის ნდობის დონეს შორის შედარება. თუ ეს მონაცემები თანაბარია, ხდება ტრანზაქცია. ყოველი ტრანზაქციის შემდეგ ხდება W_i კვანძის რეპუტაციის ვექტორის მოდიფიკაცია.

*Grid computing*არქიტექტურის დაცვის საშუალებების სტრუქტურა, რომელიც დაფუძნებულია რეპუტაციურ მეთოდზე დამყარებულ ნდობაზე შემდეგია. იხ. ნახ. 6.2.4.



ნახ.7.2.4

ზოგიერთი ავტორის მიერ მოყვანილია გრაფიკული ინფორმაცია, იმის დასამტკიცებლად თუ როგორ იზრდება *Grid computing* არქიტექტურის დაცვის დონე L_p შემოთავაზებული მეთოდიკის გამოყენებით (მრუდი 2) კვანძთა რაოდენობის ზრდასთან ერთად. იხ. ნახ.7.2.5.



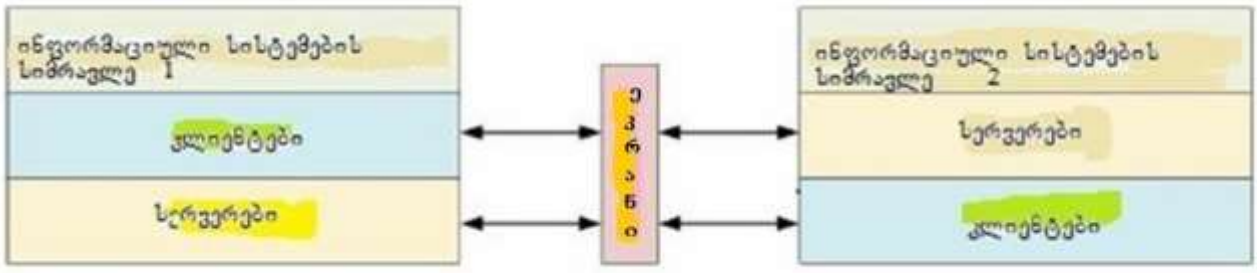
ნახ.7.2.5.

§7.3 ქსელთა შორისი ეკრანები

§7.3.1 ქსელაშორისი ეკრანირების კონცეფცია

ვთქვათ, გვაქვს ორი ინფორმაციული სისტემების (ის) სიმრავლე. ქსელური ეკრანი (ქე)–ეს არის ერთი ქსელის მომხმარებელთა გამიჯვნის საშუალება, არ მიმართონ (ისარგებლონ) მეორე ქსელის სერვერს. ქე თავისი ფუნქციური დანიშნულებით აკონტროლებს ყველა ინფორმაციული ნაკადების ურთიერთგაცვლას ქსელებს შორის [44]. .იხ. ნახ.7.3.1.

აქ **ქე** ძირითადი ფუნქციაა ე.წ. ფილტრაცია. იგი ხორციელდება ფილტრაციის წესების თანამიმდევრული ანალიზის საფუძველზე. აქ ნაგულისხმევია შემდეგი მოქმედებები: საჭიროებისამებრ უკუაგდოს მონაცემთა ნაწილი (პაკეტი), გადასცეს იგი შემდეგ ფილტრს შემდგომი ანალიზისათვის, თუ დაამუშავოს გამგზავნის სახელით იგი და შედეგი დაუბრუნოს გამომგზავნს. იხ. ნახ. 7.3.2.



ნახ. 7.3.1

ნახ.7.3.2.

აქ მითითებული თითოეული წესი კრძალავს ან ნებას რთავს სუბიექტსა და ობიექტს შორის გარკვეული ინფორმაციული ნაკადების ურთიერთცვლას. ქე ყოფს **ის** ცალკეულ სეგმენტებად და ახორციელებს ინფორმაციული ნაკადების ფილტრაციას ადმინისტრატორის მიერ დადგენილი წესებით. იგი აგრეთვე ქმნის შესაბამის ოქმებს.

ქე-ს ფუნქციონირებაში გადამწყვეტი მნიშვნელობა აქვს ე.წ. ფუნქციონირების ფარულ რეჟიმს. ამ დროს მფილტრავ ინტერფეისებს არა აქვთ ქსელური IP მისამართები და ფიზიკური MAC მისამართი. ეს რეჟიმი მიიღწევა:

- ფიზიკურ ინტერფეისებზე ფიზიკური და ლოგიკური მისამართების არქონით;
- ქსელის დაცული სეგმენტებზე მიერთებული ინტერფეისები მუშაობენ ამ სეგმენტებში გადასაცემ **ის** მთლიან ტრაფიკში მიღებისა და გადამუშავების რეჟიმში;
- პაკეტში, რომელიც დამუშავდა და გადაეცემა, არ იცვლება ოქმების თავსართები და გამოყენებადი მონაცემები.

ასეთი მიდგომა არ ცვლის მარშრუტიზაციის ამოცანას ქსელში, თუმცა ქსელში დაცილებული მართვის შემთხვევაში, მმართველ ინტერფეის აქვს ქსელური მისამართები, რაც შესაბამისად ამცირებს დაცულობის ხარისხს. ამ ნაკლოვანების აღმოსაფხვრელად უმჯობესია „უხილავი“ ქე იმართებოდეს ლოკალურად ან გამოყოფილი იქნეს ცალკე კონტროლირებადი არხი, კონტროლირებად ზონაში ჩართვისათვის. ამ დროს გადასაცემი ინფორმაცია კრიპტოგრაფიულად უნდა იქნეს დაცული. ასეთი მიდგომა სხვა და სხვა მიზეზებით ხშირად არაპრაქტიკულია (სიძვირე, ტერიტორია და ა.შ.).

მიღებულია ქე-ს შემდეგი კლასიფიკაცია :

პროგრამული;

პროგრამულ-აპარატურული,

ობიექტის დაცულობის მიხედვით :

სეგმენტური ქე, რომელიც დგება ორი ან მეტი ქსელის საზღვარზე;

ჩაშენებადი ქე, რომლებიც ფუნქციონირებს ერთ პლატფორმაზე დაცულ სერვერებთან ერთად.

მიღებულია ქე კლასიფიკაცია იმის მიხედვით, თუ ქსელების ISO/OSI კლასიფიკაციის რომელ დონეზე ფუნქციონირებს ქე:

- მართვადი კომუტატორები;

- პაკეტების ფილტრები;
- მდგომარეობების ინსპექტორები (StateFull-Inspection FW) სეანსურ დონეზე;
- გამოყენებადი დონის ქე;
- ექსპერტული დონის ქე,

მართვადი კომუტატორები ფუნქციონირებს ISO/OSI მოდელის არხის დონეზე.

პაკეტური ფილტრები აკონტროლებენ ქსელურ და სატრანსპრტო დონეებზე პაკეტების თავსართებს. ეს ხორციელდება თითოეულისათვის წინა კონტროლის შედეგების გაუთვალისწინებლად.

მდგომარეობების ინსპექტორები (სეანსური დონის ქე) ახორციელებს პაკეტების ფილტრაციებს მიმდინარე ვირტუალური შეერთებზე ინფორმაციის არსებობის შემთხვევისათვის (TCP შეერთებები, UDP ხელოვნური ან ICMP შეერთებები). აქ თითოეული პაკეტის ფილტრაციისას მხედველობაში მიიღება წინა პაკეტების შემოწმების შედეგები.

გამოყენებადი დონის ქე ფილტრაციას უკეთებენ თავსართებს ან (და) მონაცემებს გამოყენებით დონეზე.

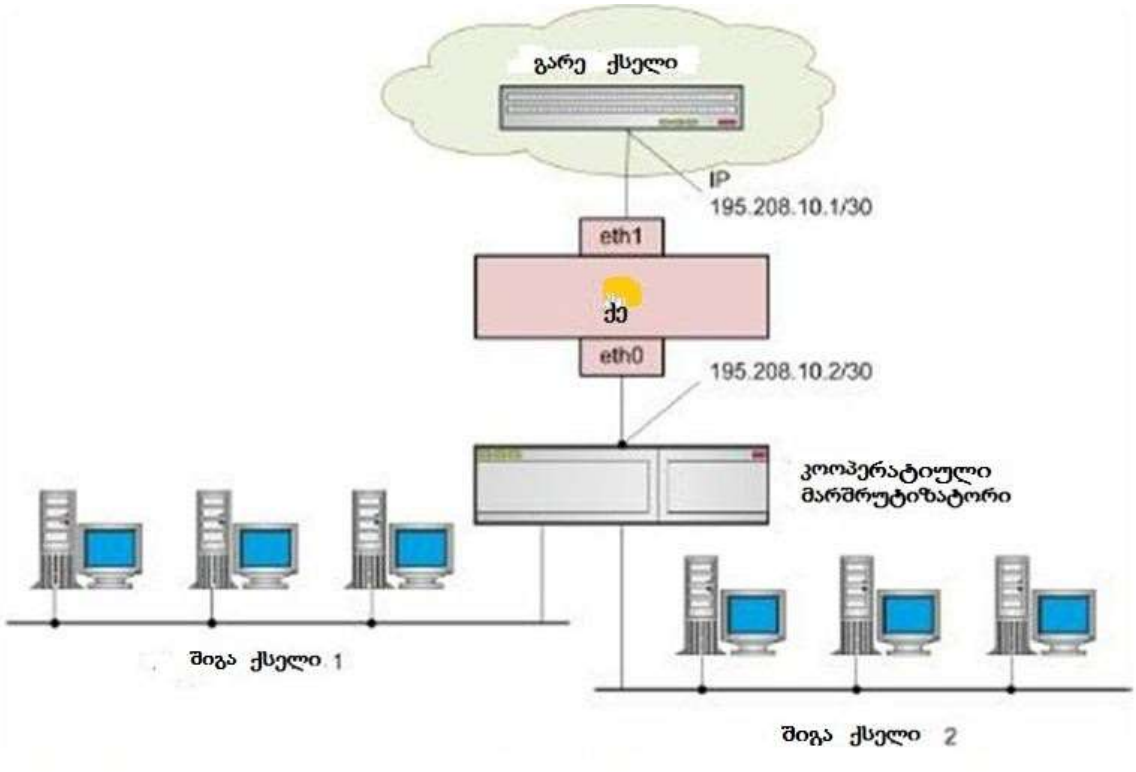
საექსპერტო დონეზე ქე ახორციელებს თითქმის ყველა ჩამოთვლილ დაცვის ტენოლოგიებს. დამატებით აქვთ ჩაშენებული მექანიზმები შეღწევის აღმოსაჩენად, VPN რეჟიმის მხარდასაჭერი მექანიზმები, აუტენფიკაციის სრულყოფილი რთული ალგორითმები და სხვ.

§7.3.2 ქე-ს ძირითადი ფუნქციები და ჩართვის ტიპური სქემები

შეიძლება განისაზღვროს პროგრამულ-აპარატურული ქე ძირითადი ფუნქციები:

- ფარული ფილტრაციის განსახორციელებელი „უხილავი“ Stealth რეჟიმი;
- პაკეტური ფილტრაცია;
- ოქმების ფილტრაცია ვირტუალური შეერთებების გათვალისწინებით;
- NAT მისამართების ტრანსლაცია;
- VLAN მხარდაჭერის უზრუნველყოფა;
- VPN მხარდაჭერა;
- სარკისებრივი ტრაფიკებისა და რეგისტრაციის შესაძლებლობები;
- ტრაფიკის მონიტორინგი და მოვლენების რეგისტრაციის ფაილების ანალიზი;
- „ცხელი“ რეზერვირების უზრუნველყოფა.

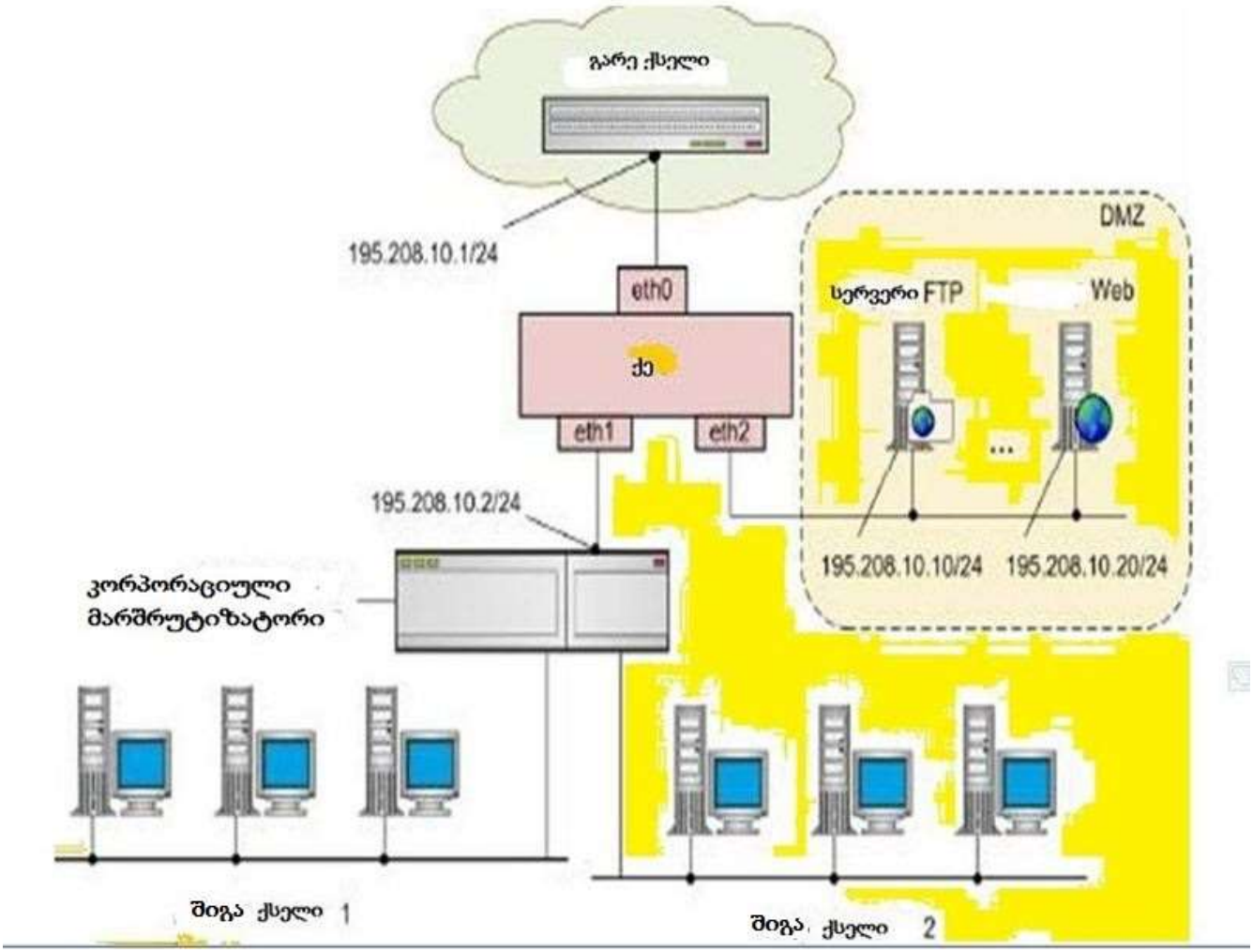
ცნობილია ქე-ს ჩართვის რამდენიმე ძირითადი სქემა. განვიხილოთ ზოგიერთი მათგანი. უმარტივესი სქემა წარმოდგენილია ნახ. 7.3.3.



ნახ. 7.3.3.

ამ სქემაში კორპორაციულ მარშრუტიზატორსა და გარე ქსელს შორის ჩართულია ქე. ჩართვისას უზრუნველყოფილია ქსელური მისამართების უცვლელობა. ტექნიკურ-პროგრამული საშუალებებით ქე უზრუნველყოფს შიგა ქსელს და მის მარშრუტიზატორს გარე ქსელიდან შეტევებისაგან. ამავე დროს ის ახორციელებს შიგა ქსელის მომხმარებლების წვდომას გარე რესურსებთან.

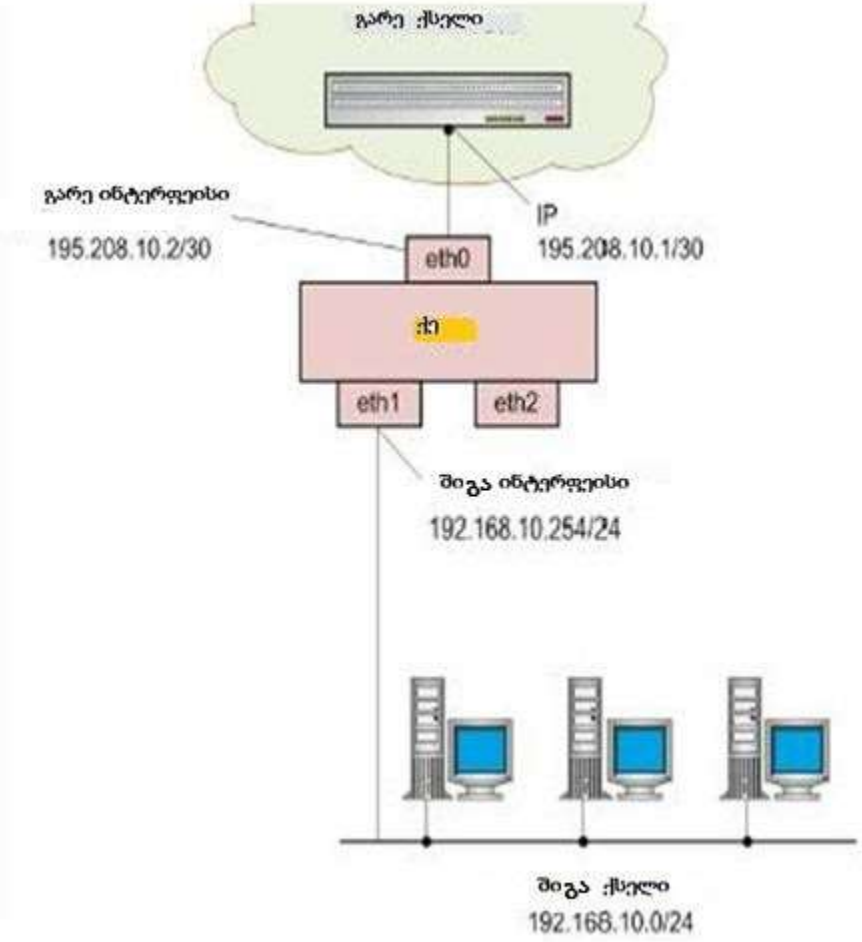
ჩართვის შემდეგი სქემაა ე.წ. დემილიტარიზებული ზონის (დმზ) გამოყენება ნახ.7.3.4. მისი აზრი მდგომარეობს შემდეგში. იქმნება ცალკე ე.წ. დმზ, სადაც განთავსებულია გარკვეული სისტემები და რესურსები, რომელთანაც წვდომის უფლება აქვთ როგორც გარე ისე შიგა მომხმარებლებს ცალმხრივი ან ორმხრივი მიმართულებებით. არსი მდგომარეობს იმაში, რომ თუ მოხდა დმზ სისტემებისა და რესურსების დაცვის გარღვევა, მაშინ შიგა ქსელის მომხმარებლების რესურსებთან და სისტემებთან ბოროტგამზრახველს ავტომატური წვდომა აღარ ექნება.



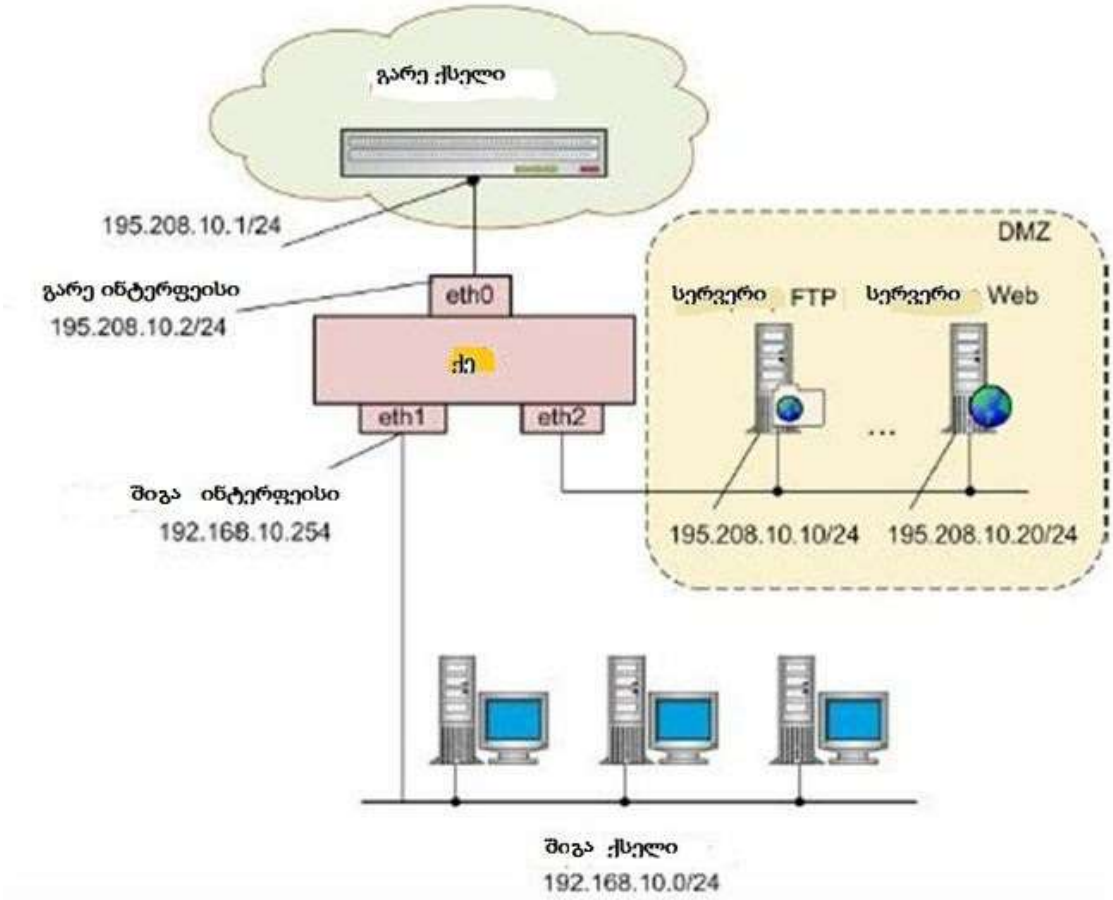
ნახ 7.3.4

ამ სქემაში ქე-ს 3 ან მეტი ინტერფეისი აქვს. ის იცავს შიგა ქსელის მარშრუტიზატორს (შესაბამისად შიგა ქსელის მომხმარებლებს) და დმზ რესურსებს და მომხმარებლებს. იგი უზრუნველყოფს შიგა ქსელის მომხმარებლების წვდომას, როგორც გარე ქსელის ისე დმზ რესურსებთან.

შემდეგი სქემები ანალოგიურია 1 და 2 სქემებისა (იხ ნახ.7.3.1. და ნახ.7.3.2), იმ დაზუსტებით, რომ ამ დროს ქე ახორციელებს ე.წ. NAT ქსელური მისამართების ტრანსლირებას. ამ ვარიანტებისთვის ქე წარმოადგენს მარშრუტიზატორს.


გარე ქსელი
გარე ინტერფეისი
195.208.10.2/30
IP
195.208.10.1/30
eth0
eth1
eth2
შიგა ინტერფეისი
192.168.10.254/24
შიგა ქსელი
192.168.10.0/24

ნახ .7.3.5.

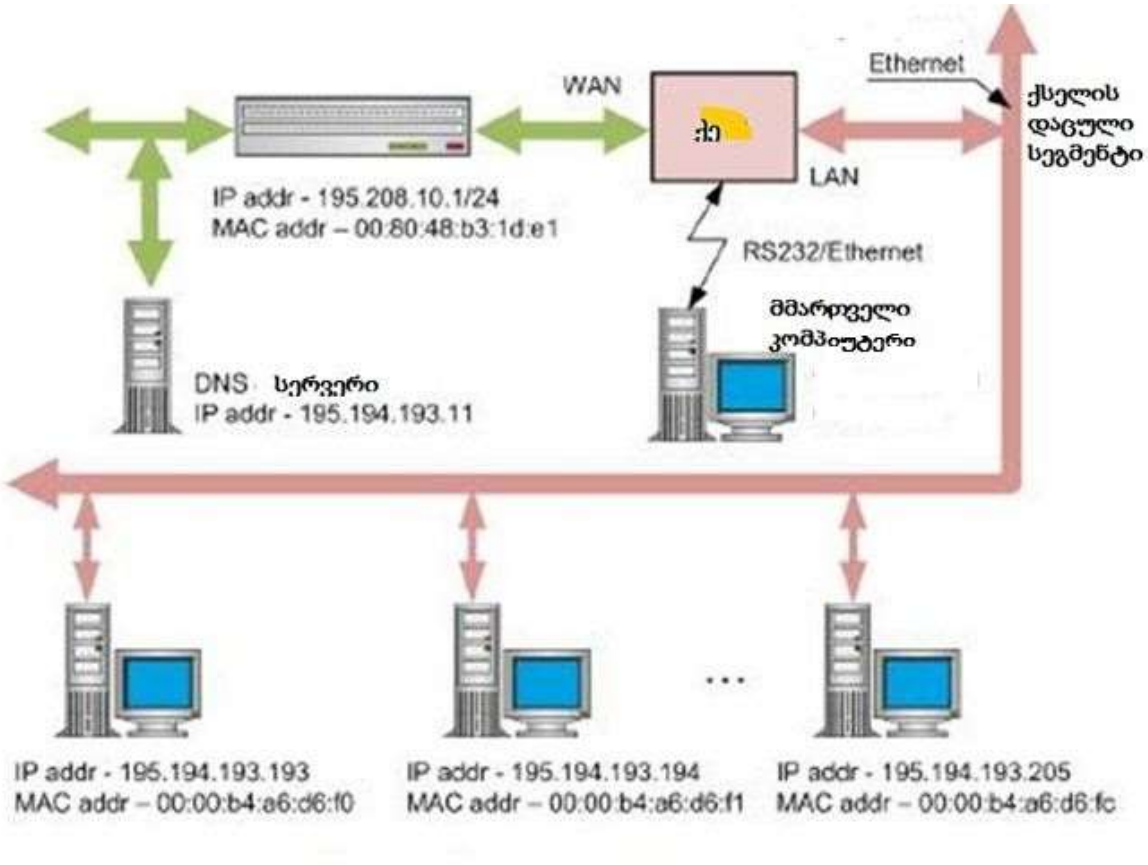


ნახ. 7.3.6

კონკრეტული მაგალითისათვის განვიხილოთ ქე ჩართვა და მისი კონფიგურაცია. ნახ.7.3.7. ამ მაგალითში ერთ IP ლოკალურ ქსელში ჩართულია 13 პერსონალური კომპიუტერი და რაბი.

IP ლოკალურ ქსელში მისამართები შემდეგნაირად არის გაწერილი:

- 195.194.193.192 - ქსელური მისამართებია;
- 255.255.255.240 – ქვექსელის „მასკაა“;
- 195.194.193.193 -195.194.193.204 -პერსონალური კომპუტერების IP მისამართებია;
- 195.194.193.206 - რაბის მისამართია;
- 195.194.193.207 - ღია მისამართებია.



ნახ .7.3.7.

გლობალურ ქსელში შეღწევა ხდება რაბით, რომლის IP მისამართია 195.194.193.206, ხოლო ქსელური პლატაა 00:80:48:b3:1d:e1 MAC მისამართით. სხვა პერსონალურ კომპიუტერებსაც იმავე მწარმოებლის მიერ გამოშვებული ქსელური პლატებია აქვთ MAC მისამართებით 00:80:48:b3:1d:f0- 00:80:48:b3:1d: fc .

DNS სერვერი განთავსებულია ლოკალური გამოთვლითი ქსელის (ლგქ) გარეთ და მისი მასამართია 195.194.193.11. ლგქ დასაცავად ქე იყენებს ორ მფილტრავ ინტერფეისს LAN და WAN . LAN ინტერფეისი მიერთებულია ლგქ-თან, ხოლო WAN რაბთან. მოცემული მაგალითისთვის, განვიხილოთ ლგქ-დან ინტერნეტში კონტროლირებული (დაცული) შეღწევის ორგანიზება. ამოცანა შემდეგია:

- ლგქ ყველა მომხმარებლებს აქვთ შეღწევა FTP სერვერთან, რომლის IP მისამართია 129.12.12.12 ;
- ლგქ ინტერნეტიდან თავისუფალი წვდომა WEB სერვერთან IP მისამართით 185.184.193.199.
- ყველა მომხმარებელს, გარდა ერთისა, მისამართით 195.194.193.193, აქვთ უფლება შეღწევისა ინტერნეტში.

<u>მნიშვნელოვანი დაშვებაა - ის რაც არ არის ნებადართული - აკრძალულია.</u>

ყველა ჯგუფის მოქმედებებში მიღებულია გლობალური წესი-წაშლა. ამ ზოგადი მოთხოვნებებიდან დგება IP წესების ცხრილი 5.3.1.1, რომელსაც იყენებს მართვისას WEB ინტერფეისი.

ცხრ 5.3.1.1.

წესები	მოქმე-დება	ინტერ-ფეისი შემ / გამ	ოქმი	გადამცემი IP მისამართი/მასკა პორტი	მიმღები IP მისამართი/მასკა პორტი	კომენ-ტარი
10 v	გატარება	LANWAN	UDP	195.194.193.192/28 1024-65535	195.194.193.11 53	PC-DNS
11 v	გატარება	WAN LAN	UDP	196.194.193.11 53	195.194.193.192/28 1024-65535	DNS
30 v	გატარება	LAN WAN	TCP	195.194.193.192/28 1024-65535	129.12.12.1220-21	PC-FTP
40 v	გატარება	WAN LAN	TCP	129.12.12.1220-21	195.194.193.192/28 1024-65535	FTP-PC
50 v	გატარება	WAN LAN	TCP	ნებისმიერი 1024-65535	196.194.193.199 80	WAN-Webserver
60 v	გატარება	LAN WAN	TCP	196.194.193.199 80	ნებისმიერი 1024-65535	Webserver WAN
70 v	წაშლა	LAN WAN	TCP	195.194.193.193 1024-65535	ნებისმიერი 80	http-ბლოკირება
80 v	გატარება	LAN WAN	TCP	195.194.193.192/28 1024-65535	ნებისმიერი 80	PC-WAN
90 v	გატარება	WAN LAN	TCP	ნებისმიერი 80	195.194.193.192/28 1024-65535	WAN-PC

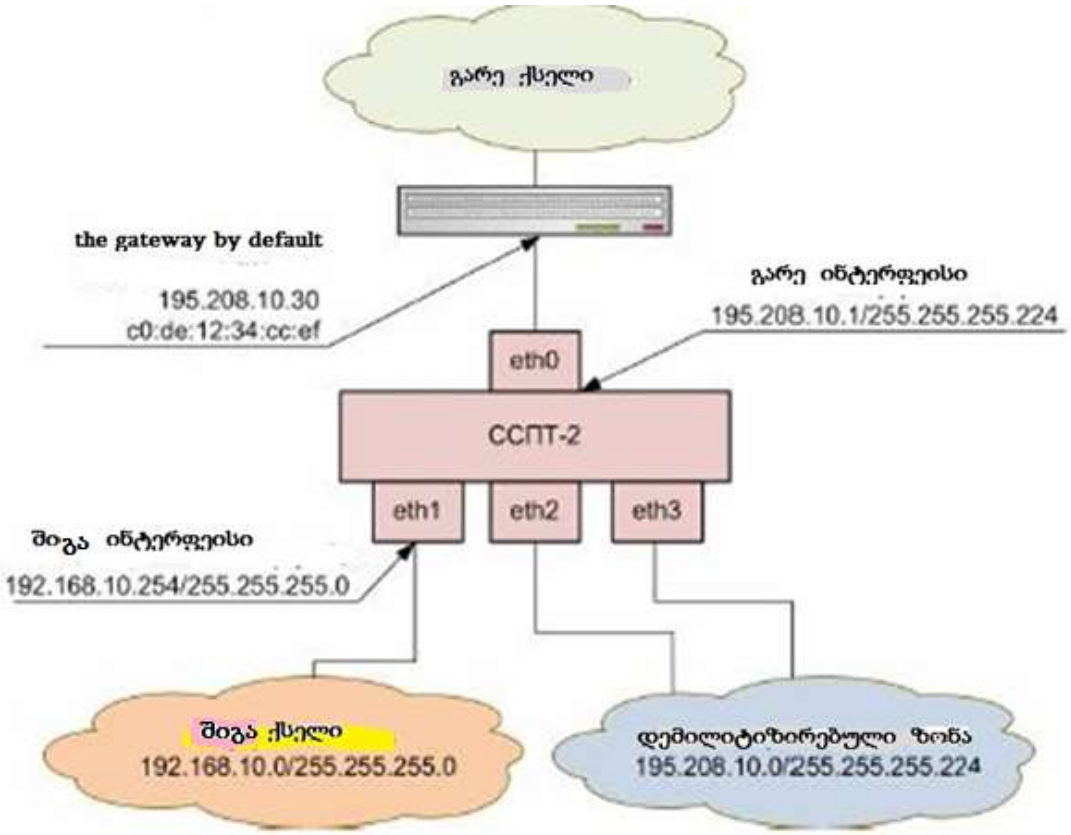
ამ მაგალითში მე-10 წესი განსაზღვრავს პერსონალური კომპუტერიდან (პკ) DNS სერვერამდე მოთხოვნის გავლას, ხოლო მე-11 წესი კი პირიქით. 30-ე და მე-40 წესები განსაზღვრავენ გარე FTP სერვერთან TCP სეგმენტების გავლას (აქტიური რეჟიმი). 50-ე და მე-60 წესებით განსაზღრულია TCP სეგმენტების გატარების უფლება გარე ქსელიდან შიგა WEB სერვერთან. 70-ე წესი ნებას რთავს 195.194.193.193 კვანძს შეაღწიოს გარე ქსელში HTTP ოქმით. მე-80 და 90-ე წესებით (განიხილება ყოველთვის 70-ე წესის შემდეგ) დაკავშირების უფლება ეძლევა ***ლგქ*** ყველა პკ გარეთა ქსელის ნებისმიერი კვანძის 80 პორტთან დაკაცშირებისა.

§ 7.3.3 დაცვის საკითხები NAT ტექნოლოგიის გამოყენებისას.

ქე აქვს NAT ქსელური მისამართების ტრანსლაციის რეჟიმი. ამ რეჟიმში მფილტრავი ინტერფეისები თავისი დანიშნულებით ქმნის:

- გარე ინტერფეისი 0 (მიღებულია eth0). ამ ინტერფეისთან ერთვება გარე ქსელი;
- შიგა ინტერფეისი 1 (მიღებულია eth1). ამ ინტერფეისთან ერთვება შიგა დაცული ქსელი.

დანარჩენი ინტერფეისები მათი არსებობის შენთხვევაში - დემილიტიზული ზონაა DMZ ამ ინტერფეითან ერთვება ხოსტები, როგორც შიგა ისე გარე ქსელები. NAT რეჟიმის რეალიზაციისას გარე და შიგა ინტერფეისებს ენიჭება ვირტუალური მისამართები IP და MAC ნახ.7.3.8.



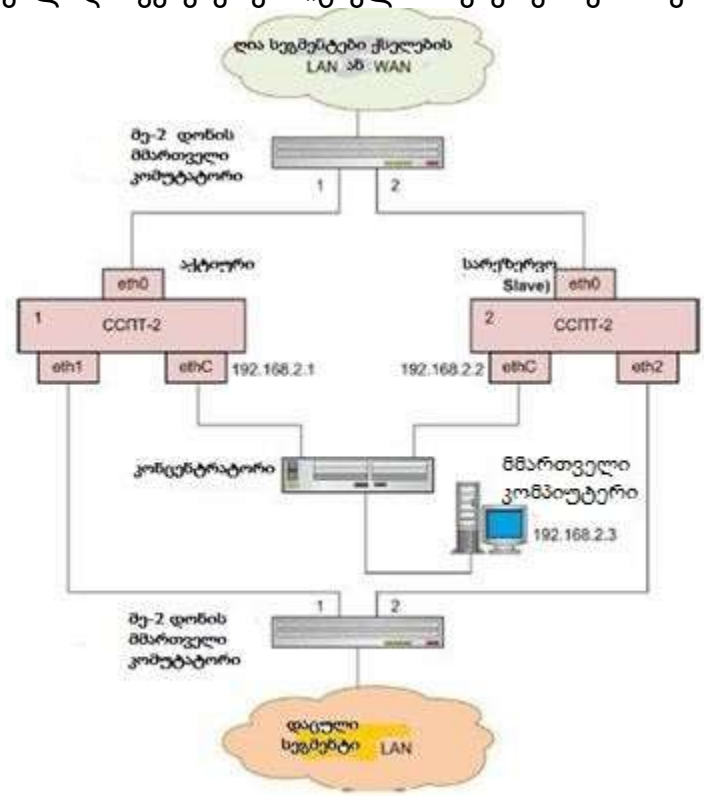
ნახ. 7.3.8.

ამ რეჟიმის დროს ხდება გადამცემის მისამართების IP და MAC შეცვლა ვირტუალურ მისამართებზე პაკეტის ერთი მიმართულებით გადაცემისას და უკუმიმართულებისას

ვირტუალურის IP და MAC მისამართებზე. აქვე შესაძლებელია წყაროს პორტების ნომრების შეცვლა. ქე ამ რეჟიმებს უზრუნველყოფს მხოლოდ შემდეგი ოქმებისთვის TCP, UDP, ICMP (ექო-შეტყობინება და ექო-პასუხი).

§7.3.4 ქე სამედოობის გაზრდა „ცხელი“ რეზერვირების მეთოდის გამოყენებით.

ფილტრაციის მეთოდის გამოყენება თავისთავად ართულებს ქსელში ინფორმაციული ნაკადების ურთიერთგაცვლას, რადგან მას შემოაქვს (იყენებს) დამატებით აპარატურულ-პროგრამული საშუალებები. ეს იწვევს მთლიანად ქსელის საიმედოობის შემცირებას. ამ ეფექტის გასანეიტრალებლად იყენებენ ე.წ. „ცხელი“ რეზერვირების მეთოდს, იხ. ნახ. 7.3.9.



ნახ. 7.3.9.

წარმოდგენილი გადაწყვეტილების სისტემა ფუნქციონირებს სამ რეჟიმში :

- <u>„აქტიური რეზერვირების“ რეჟიმი;</u>

ამ რეჟიმში 2 ქე, ქსელის კომუტატორების გამოყენებით, უერთდება ლგქ სეგმენტებს პარარელურად და მთლიანობაში მუშაობენ, როგორც ერთიანი ლოგიკური სისტემა. ამ ორი *ქე*-დან ერთი (master) მუშაობს აქტიურ რეჟიმში, ხოლო მეორე (slave) „ცხელი რეზერვის“ რეჟიმში. ამ ორი ქე პარალელური მუშაობის სინქრონიზაცია სრულდება მმართველი Ethernet ინტერფეისის (EthC) მოწყობილობით. სისტემა ისეა აგებული, რომ master ქე ნებისმიერი მტყუნება არ იწვევს მთლიანობაში სისტემის მტყუნებას.

- <u>ბალანსირების რეჟიმი;</u>

ამ მეთოდის გამოყენებისას კომუტატორებს შორის ორი ფიზიკური არხი ერთიანდება ე.წ „ტრანკ“ ერთ ლოგიკურ არხში (Link Aggregation - სტანდარტი IEEE 802.3ad). ეს 2 ქე ირთვება კომუტატორებს შორის ფიზიკურ არხებში და მუშაობენ პარარელურად, აქტიურ რეჟიმში, დატვირთვის გადანაწილებას და სინქრონიზაციის ამოცანებს წყვეტს EthC.

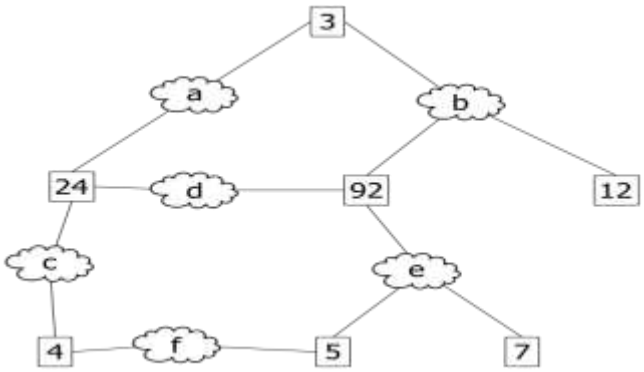
- Spanning Tree Protocol STP (ხის მაგვარი კავშირების ოქმი) რეჟიმი:

  იგი არხის დონის ოქმია. მისი ძირითადი ამოცანაა არ დაუშვას ქსელ Ethernet ტოპოლოგიაში მარყუჟების არსებობა ერთი ან რამდენიმე ქსელური ხიდების არსებობისას. იგი ავტომატურად ბლოკავს ასეთ „ჭარბ, ზედმეტ“ კავშირებს. რეალურად თუ არსებობს ქსელში ასეთი კავშირები, ეს იწვევს მონაცემთა გადაცემის უსასრულო ციკლებს კომუტატორებში ერთი და იმავე კადრის გადაცემისას. ეს კი საბოლოოდ მთლიანობაში ძალიან ამცირებს ქსელის გამტარუნარიანობას. მცირდება ქსელის წარმადობა ძალიან დაბალ დონემდე. STP ოქმი მიეკუთვნება OSI მოდელის მეორე დონეს და აღწერილია სტანდარტში IEEE 802.1d. იგი შექმნილია ***რადია პერლმანის*** მიერ (Radia Perlman).

  მისი ფუნქციონირების ალგორითმია:

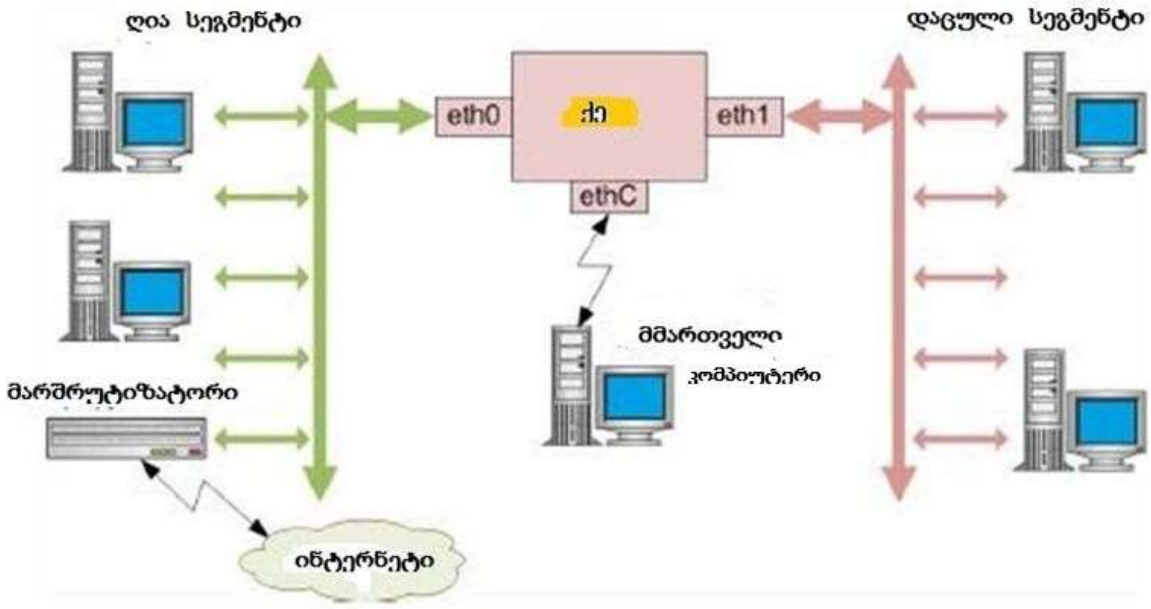
  - ამოირჩევა ერთი ფუძისეული ხიდი (Root Bridge);
  - დგინდება ყველა დანარჩენი ხიდის უმოკლესი მარშრუტი ფუძისეულთან. შესაბამის პორტს ეწოდება ფუძისეული პორტი (Root Port), რომელიც მხოლოდ ერთადერთია;
  - ქსელის ყველა სეგმენტისათვის, რომლებთანაც კავშირი აქვთ ერთ ან მეტ ხიდს - დგინდება უმოკლესი მარშრუტი ფუძისეულ ხიდთან. ხიდი, რომელზეც გადის ეს მარშრუტი ეწოდება „დანიშნული” (Designated Bridge), ხოლო შესაბამის პორტს „დანიშნული“ (Designated port).
  - ყველა სეგმენტში, რომლებთანაც შეერთებულია ხიდის ერთზე მეტი პორტი, ყველა ხიდი ბლოკავს ყველა პორტს, რომლებიც არ არიან ფუძისეულნი ან დანიშნული. შედეგად ვიღებთ ხის მაგვარ სტრუქტურას, რომლის კენწეროა ფუძისეული კომუტატორი. იხ. ნახ. 7.3.10.

აქ კვადრატებით აღნიშნულია კომუტატორები, კვადრატებში მოცემულია ხიდების იდენტიფიკატორები Bridge ID → Bridge ptiority + MAC (აქ Bridge ptiority-ს ნიშნავს ქსელის ადმინისტრატორს), ხოლო ლათინური ასოებით კი ქსელის სეგმენტები. აღნიშნულმა ოქმა განიცადა გაუმჯობესება Rapid Spanning Tree Protocol (RSTP), Per-VLAN Spanning Tree Protocol (PVSTP), Multiple Spanning Tree Protocol (MSTP) ოქმების სახით.



ნახ.7.3.10.

ნახ.7.3.11-ზე მოცემულია ჟ ჩართვის ტიპური სქემა.



ნახ. 7.3.11.

თავი8 უსაფრთხოების უზრუნველყოფა OSI და TCP/IP მოდელებში

OSI (Open Systems Interconnection) **მოდელი** - სატელეკომუნიკაციო და კომპიუტერული ქსელური ოქმების ურთიერთდამოკიდებულებისა და აგების დონეებად დაყოფის აბსტრაქტული სქემაა. მას ასევე უწოდებენ **OSI-ს შვიდდონიან მოდელს.**

OSI მოდელი ჰყოფს ოქმების ფუნქციებს დონეებად. მისი ყოველი დონე შეიცავს მხოლოდ მისთვის საჭირო ფუნქციებს და ემსახურება მხოლოდ საკუთარი პროცესების ურთიერთქმედებას. ჩვეულებრივ ქვედა დონეებს ემსახურება აპარატურული ნაწილი, ხოლო ზედა დონეების დამუშავება ხდება პროგრამული მეთოდით.

OSI მოდელს ხშირად იყენებენ კომპიუტერული ქსელების აგებისას. მისი მთავარი თვისებაა სხვა და სხვა დონის ერთმანეთთან დაკავშირება, რაც ასევე უზრუნველყოფს ერთ დონეზე მომუშავე მწარმოებლის მიერ შემუშავებული აპარატურის სხვა დონეზე მომუშავე აპარატურასთან მუშაობას თუ ამ აპარტურის ყოველი ოქმი დოკუმენტირებულია და მისი აღწერილობა არსებობს. ეს აღწერილობაა TCP/IP-ზე მომუშავე საზოგადოებისათვის ჩვეულებრივ ცნობილია როგორც RFC-ს დოკუმენტაცია (Request for Comments). OSI-ს მოდელი წარმოადგენს შვიდი დონის იერარქიულ სტრუქტურას, რომლითაც განისაზღვრება ორ კომპიუტერს შორის კავშირი. მოდელი განსაზღვრულია სტანდარტების საერთაშორისო ორგანიზაციების მიერ (ISO) 7498-1 სტანდარტში. 1980 წლების ბოლოს, სტანდარტების საერთაშორისო ორგანიზაციამ რეკომენადცია გაუწია მოდელის ქსელურ სტანდარტად დანერგვას. იმ დროისთვის TCP/IP უკვე გამოყენებაში იყო წლების განმავლობაში. ინტერნეტში გამოჩნდა TCP/IP-ის ფუნდამენტური ARPANET და სხვა ქსელები. (TCP/IP-სა და ARPANET-შორის სხვაობის დასადგენად, იხილეთ დოკუმენტი RFC 871.). TCP/IP ოქმი ოთხ დონიანია და შედგება: გამოყენებითი (4), სატრანსპორტო (3), ქსელთაშორისო (2) და ქსელში შეღწევის დონეებისაგან. მიუხედავად იმისა, რომ დღესდღეობით OSI მოდელის

მცირე ნაწილს ექცევა ყურადღება და მისი სპეციფიკაცია ძალიან ჩახლართულია, ადმინისტრატორების დიდი ნაწილი მაინც OSI მოდელს მიჰყვება.

OSI დონეების აღწერა იხ. ნახ. 8-1.

OSI მოდელი			
	მონაცემების ერთეული	**დონე**	**ფუნქცია**
ჰოსტის დონეები	მონაცემები	პროგრამული	ქსელის მიწოდება, პროგრამისათვის
		პრეზენტაციის	მონაცემების შიფრაცია და წარდგენა
		სესიის	კვანძთაშორისი კავშირი
	სეგმენტები	სატრანსპორტო	კავშირი ორ უკიდურეს წერტილს შორის და საიმედოობა
მატარებელი დონეები	პაკეტები	ქსელური	გეზის განსაზღვრა და ლოგიკური მისამართები (IP)
	კადრები	მონაცემთა არხი	ფიზიკური მისამართები (MACდა LLC)
	ბიტები	ფიზიკური	გამტარი ხაზი, სიგნალი და ორობითი გადაცემა

ნახ. 8-1.

განვიხილოთ დონეები და მათში გამოყენებული ოქმები ნახ. 8.2.

დონე		**მაგა-ლითები**	**TCP/IP კრებული**	**SS7**	**AppleTalk კრებული**	**OSI კრე-ბული**	**IPX კრებული**	**SNA**	**UMTS**
#	**სახელი**								

7	პროგრამული	HL7, Modbus, SIP, SSI	HTTP, SMTP,SMPP, SNMP, FTP, Telnet, NFS, NTP, RTP	ISUP, INAP, MAP, TUP, TCAP	AFP	FTAM, X.400, X.500, DAP		APPC	
6	პრეზენტაციის	TDI, ASCII, EBCDIC, MIDI, MPEG	XDR, SSL, TLS		AFP	ISO 8823, X.226			
5	სესიის	Named Pipes, NetBIOS, SAP, SDP	სესიის შესრულება TCP ოქმისთვის		ASP, ADSP, ZIP, PAP	ISO 8327, X.225	NWLink	DLC?	
4	სატრანსპორ-ტო	NetBEUI, nanoTCP, nanoUDP	TCP, UDP, SCTP		ATP, NBP, AEP, RTMP	TP0, TP1, TP2, TP3, TP4, OSPF	SPX, RIP		
3	ქსელური	NetBEUI, Q.931	IP, ICMP, IPsec, ARP, RIP, BGP	MTP-3, SCCP	DDP	X.25 (PLP), CLNP	IPX		RRC
2	მონაცემთაგა დაცემისარხის დონე	Ethernet, 802.11 (WiFi), Token		MTP-2	LocalTalk, TokenTalk, EtherTalk, AppleTalk	X.25 (LAPB), Token Bus	IEEE 802.3 კადრირება, Ethernet II კადრირება	SDLC	MAC (Media Access Control

		Ring, FDDI, PPP, HDLC, Q.921, Frame Relay, ATM, Fibre Channel			Remote Access, PPP				)
1	ფიზიკური	RS-232, V.35, V.34, I.430, I.431, T1, E1, 10BASE-T, 100BASE-TX, POTS, SONET, DSL, 802.11b, 802.11g		MTP-1	RS-232, RS-422, STP, PhoneNet	X.25 (X.21bis , EIA/TIA-232, EIA/TIA-449, EIA-530, G.703)		Twinax	PHY (ფიზიკური დონე)

ნახ. 8-2.

განვიხილოთ დონეების ფუნქციები და საშიშროებები [35,41,42,43,48].

დონე 7: პროგრამულიდონე. პროგრამული დონის გავლით მომხმარებელს შეუძლია ქსელში მოთავსებულ ინფორმაციამდე მიღწევა პროგრამის საშუალებით. პროგრამული დონის ოქმების მაგალითებია: Telnet, File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) და Hypertext Transfer Protocol (HTTP) ოქმები.

დონე 6: პრეზენტაციის დონე. პრეზენტაციის დონე გარდაქმნის მონაცემებს პროგრამული დონის სტანდარტული ინტერფეისისათვის გასაგებ ენაზე. პრეზენტაციის დონეზე წარმოებს MIME კოდირება, მონაცემების შეკუმშვა, მონაცემების კოდირება და მსგავსი მოქმედებები ზემდგომი დონის მოთხოვნის ფარგლებში წარმოდგენისათვის. მაგალითად: EBCDIC-ით კოდირებული ტექსტური ფაილის ASCII- კოდირებულ ფაილად გარდაქმნა, ობიექტების და სხვა მონაცემთა სტრუქტურის XML-ში გარდაქმნა და ა.შ.

დონე 5: სესიის დონე. სესიის დონე აკონტროლებს დიალოგს (სესიებს) კომპიუტერებს შორის. ის იწყებს, მართავს და წყვეტს კავშირებს ადგილობრივ და შორეულ პროგრამებში. ის იძლევა დუპლექსური ან ნახევრადდუპლექსური კავშირის დამყარების საშუალებას და ახდენს საბოლოო კავშირის შესრულების შემოწმებას, რეგულირებას, შეწყვეტას და განახლებას.

ამ დონეზე ძირითადი შეტევის სახეა „უარი მომსახურებაზე“, რომელიც შეიძლება გამოწვეული იყოს კავშირის დამყარების პროცედურების თავისებურებებით TCP ოქმში. შეტევის დასახელებაა SYN-Flood. ამ დროს იგზავნება მხოლოდ ბიტი SYN თვით ინფორმაციის გარეშე. სერვერი ვალდებულია დაამუშაოს იგი (ორგანიზაცია გაუკეთოს მიღების სიგნალის გაგზავნას ინფორმაციის წყაროსათვის და უზრუნველყოს დიალოგის წარმართვა).

დონე 4: ტრანსპორტული დონე. ტრანსპორტული დონე უზუნველყოფს მომხმარებლებს შორის მონაცემების გამჭვირვალე, ეფექტურ გადაცემასა და ამ დავალებისაგან ზედა დონეების განთავისუფლებას. ტრანსპორტული დონე ამოწმებს საიმედოობას ნაკადების მართვით, სეგმენტირებით / დესეგმენტირებით და შეცდომების შემოწმებით. მეოთხე დონის ზოგიერთი ოქმი მოითხოვს ორმაგი კავშირის დამყარებას. ეს ნიშნავს, რომ ტრანსპორტულ დონეს შეუძლია პაკეტების დროებით შენახვა და დაკარგვის შემთხვევაში მათი თავიდან გაგზავნა, მსგავსი ოქმია Transmission Control Protocol (TCP). ეს არის დონე, რომელიც გარდაქმნის შეტყობინებებს TCP, User Datagram Protocol (UDP), Stream Control Transmission Protocol (SCTP) და სხვას პაკეტებში.

ამ დონეზე ძირითადი შეტევის სახეა პაკეტების არასწორი შექმნა (თანამიმდევრობის დარღვევა).

დონე 3: ქსელური დონე. ქსელური დონე უზრუნველყოფს ერთი ან რამდენიმე ქსელის გავლით მონაცემების მიმდევრობების გადაცემას წყაროდან დანიშნულების ადგილამდე ტრანსპორტული დონის მიერ მოთხოვნილი მომსახურების ხარისხის (QoS) დაცვით. ქსელური დონე ახდენს ქსელური მარშრუტიზაციის ფუნქციებს, და ასევე შეუძლია სეგმენტირება/დესეგმენტირება და შეცდომების შეტყობინება. მარშრუტიზატორები მუშაობენ სწორედ ამ დონეზე და აგზავნიან მონაცემებს ერთი ქსელიდან მეორეში, რაც საბოლოოდ შეიძლება ქსელის მომხმარებლის ინტერნეტამდე წვდომას უზრუნველყოფდეს. ასევე არსებობს მესამე დონის კომუტატორები (ხშირად მათ IP-კომუტატორებს უწოდებენ). ეს არის მისამართების ლოგიკური სქემა – მნიშვნელობები შეირჩევა ქსელური ინჟინერის მიერ, მისამართების სქემა იერარქიულია. მესამე დონის ოქმის საუკეთესო მაგალითია ინტერნეტ ოქმი (IP) .

ამ დონეზე ძირითადი შეტევის სახეა პაკეტების მარშრუტიზაციის აღრევა (არასწორი მარშრუტიზაცია).

დონე 2: მონაცემთა გადაცემის არხის დონე. მონაცემთა გადაცემის არხის დონე უზრუნევლყოფს ქსელურ ობიექტებს შორის მონაცემების გადაცემას და ფიზიკურ დონეზე მომხდარი შეცდომების აღმოჩენას და შესაძლო აღმოფხვრას. მისამართების სქემა ფიზიკურია (MAC მისამართები) რაც ნიშნავს, რომ ისინი აპარატურულ ნაწილში ფიქსირდება

წარმოების დროს, სქემა წრფივია. მეორე დონის ოქმის მაგალითიებია: Ethernet, HDLC, ADCCP. შენიშვნა: IEEE 802 სტანდარტის ლოკალურ ქსელებში და ზოგიერთ არა-IEEE 802ქსელებში, მაგალითად, FDDI-ში, ეს დონე იყოფა ორად: MAC დონედ და IEEE 802.2 LLC დონედ, ამ დონეზე მუშაობენ ქსელური ხიდები და კომუტატორები. არსებობს არგუმენტი, რის მიხედვითაც ამ დონეს უწოდებენ „2.5 დონეს", რადგან თვისობრივად ის მეორე დონეს მკაცრად არ უტოლდება.

ამ დონეზე ძირითადი შეტევის სახეა „უარი მომსახურებაზე". ამ დროს ხდება სინქრონიზაციის არევა ან თვით შეტყობინებების დროებითი მტყუნებები.

დონე 1: ფიზიკური დონე. ფიზიკური დონე განსაზღვრავს მოწყობილობების ყველა ფიზიკურ და ელექტრულ თვისებას. ის მოიცავს მავთულების (გამტარების) განლაგებას, მიწოდებულ ძაბვებს და კაბელის პარამეტრებს, ტალღის სიხშირეს და ა.შ. კონცენტრატორები ფიზიკური დონის მოწყობილობებია. ფიზიკური დონის ძირითადი ფუნქცია და დანიშნულებაა:

- ელექტრული კავშირის დამყარება და გაწყვეტა მატარებელთან;
- მრავალ მომხმარებელს შორის საკომუნიკაციო რესურსების ეფექტურად განაწილება. მაგალითად, კავშირის მოთხოვნა და ნაკადების მართვა;
- მოდულაცია, ან ციფრული მონაცემების გადამცემ არხებში გასატარებლად გარდაქმნა. მაგალითად, ეს არის სიგნალები ფიზიკურ კაბელში (როგორც მავთულით, ისე ოპტიკურ-ბოჭკოვანი) და ეთერში.

პარალელური SCSI სალტეები მუშაობენ ასევე ამ დონეზე. ამავე დონეზეა სხვა და სხვა ფიზიკური დონის Ethernet სტანდარტი. Ethernet შეიცავს ფიზიკურ და მონაცემთა გადაცემის არხის დონესაც. ასევე ხდება სხვა ლოკალურ ქსელებში - Token ring, FDDI და IEEE 802.11 (უმავთულო Ethernet კავშირი).

ამ დონეზე ძირითადი შეტევის სახეა „უარი მომსახურებაზე". აგრეთვე „ხმაურის" ორგანიზება არხში, რაც იწვევს კავშირის შეწყვეტას.

აღსანიშნავია, რომ ზოგიერთ გამოყენებად პროგრამას აქვს ინფორმაციის დაცვის საკუთარი მექანიზმები. მაგალითად, ოპ.ს Windows აქვს ფაილების დაშიფრის საშუალება *NTFS* ფაილურ სისტემაში გასაღებით, რომელიც იქმნება მომხმარებლის პაროლით. საოფისე პროგრამა MS Office შეიცავს დოკუმენტების, ელექტრონული ცხრილების დაშიფვრის საკუთარ მექანიზმებს. ასევე მონაცემთა ბაზების მართვის სისტემები მომხმარებელს აძლევს საშუალებას არა მარტო მთლიანად დაშიფვროს მონაცემთა ბაზა, არამედ ცალკეული ცხრილები, ფორმები, მოთხოვნები, პროცედურები და ა.შ.

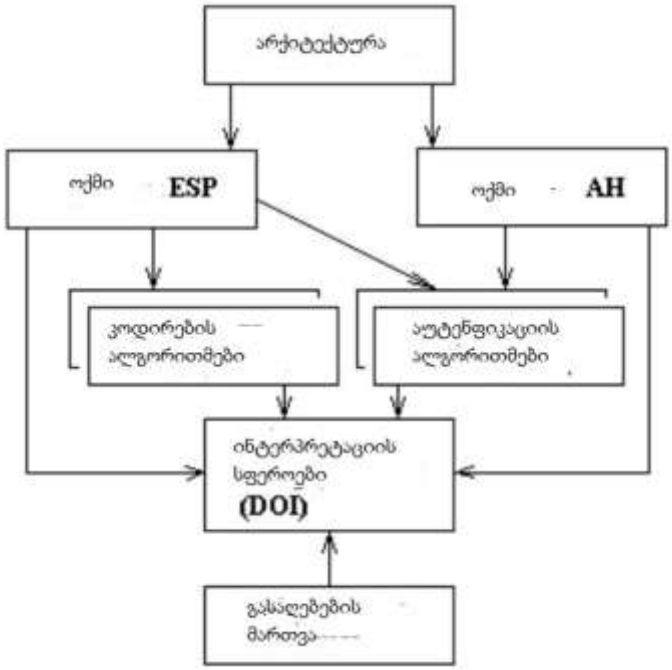
ამ მიდგომას ახასიათებს მნიშვნელოვანი ნაკლიც, რომელიც მდგომარეობს მის არაუნივერსალობაში, მის მოუხერხებელ საექსპლოტაციო თვისებებში. თითოეული ასეთი დანართისათვის საჭიროა ინდივიდუალური დაცვის მეთოდებისა და საშუალებების შექმნა და მათი ინდივიდუალური გაწყობა ქსელში სამუშაოდ.

როგორც ცნობილია, ინფორმაციის დაცვის ერთ-ერთი ხერხია მისი დაშიფვრა. იგი შეიძლება განხორციელებული იქნეს, როგოც დანართების, ისე TCP/IP ოქმების მოქმედების დონეზე. პირველი მიდგომა გულისხმობს პაროლების და სერტიფიკაციების გამოყენებას,

ავტორიზაციასა და აუტენფიკაციას, დაშიფვრა-გაშიფვრის პროცედურების ჩატარებას. ბევრ შემთხვევისათვის ინფორმაციის დაცვა უფრო ეფექტურია, თუ მას განვახორციელებთ არა დანართებში, არამედ ქსელის იმ დონეებში სადაც მოქმედებს TCP/IP ოქმები. ეს შესაძლებელია იმ მომენტისათვის, როცა სხვადასხვა სახის მონაცემი გარდაიქმნება IP პაკეტებში. პაკეტები წარმოადგენს ბიტების ერთობლიობას, რომლებიც შეიცავს IP თავსართებს, გამგზავნის და მიმღების მისამართებს. ამ დროს მიიღწევა უნივერსალობა, რაც საშუალებას გვაძლევს, ერთნაირად დავშიფროთ ყველა მონაცემი მიუხედავად ქსელში მიერთებული კომპიუტერის ტიპისა, გამოყენებული პროგრამული უზრუნველყოფისა და ოპერაციული სისტემისა, რომლის არეშიც ფუნქციონირებს კომპიუტერი.

§8.1 უსაფრთხოების უზრუნველყოფა ოქმების დონეზე, IPsec ოქმი

ერთ-ერთი ყველაზე გავრცელებული ოქმია IPsec, რომლის არქიტექტურა წარმოდგენილია [31] ნახ. 8.1.1.



ნახ. 8.1.1.

IPsec შედგება შემდეგი მოდულებისგან:

- IKE (Internet Key Exhange protocol) ;
- ISAKMP (Internet Security Association and Key Management Protocol);
- AH (Authentication Header Protocol);
- ESP (Encapsulating Security Payload Protocol);
- STS (Station-to-Station Protocol);
- HMAC (Hash Message Authentication Code);
- MD5 (Message Digest 5);
- SHA-1 (Security Hash Algorithm);
- 3DES (Triple Data Encryption Standard);
- XAUTH (Extended Authentication);
- AES (Advanced Encryption Standard),

ჩამოთვლილი ოქმების პროგრამული უზრუნველყოფის ინსტალაცია ხდება სერვერებზე, მიმღების კომპიუტერზე, მარშრუტიზატორებზე ან ბრანდმაუერზე, რომელსაც ხშირად IPsec არქიტექტურაში უწოდებენ უსაფრთხოების რაბებს (security gateway)-ს. შესაბამისად IPsec არქიტექტურაში განარჩევენ ფუნქციონირების ორ რეჟიმს : სატრანსპორტოს და „გვირაბოვანს“.

სატრანსპორტო რეჟიმში 4 და უფრო მაღალ ქსელურ დონეების დაშიფრული პაკეტების ტრანსპორტირება ხდება უშუალოდ ჰოსტ კომპიუტერებს შორის.

„გვირაბოვანი“ რეჟიმი შედარებით რთულია. აქ მიღებულია, რომ ძირითადად კლიენტების მხარეზე (ან სერვერები) არაა უზრუნველყოფილი IPsec ოქმის მოთხოვნები. აქ ფორმირებული პაკეტები ჩვეულებრივი სახისაა. სანამ ასეთი პაკეტები მოხვდება გლობალურ ქსელში, ისინი ხვდება რაბებში. აქ ისინი გარდაიქმნება IPsec ოქმების ფორმატში, იშიფრება პაკეტის ინფორმაცია და IP თავსართი. მიღებული ახალი პაკეტის მარშრუტიზაცისათვის, რაბი აძლევს მას ახალ IP თავსართს და მხოლოდ ამის შემდეგ ეს პაკეტი იგზავნება გლობალურ ქსელში. მიმღებ მხარეზე მიღებული პაკეტი გარდაიქმნება საწყის ფორმაში და მიეწოდება მიმღებს. აღწერილ ტექნოლოგიას ეწოდება „გვირაბიზაცია“. ამ რეჟიმში დაცულია (უხილავია) ჭეშმარიტი IP მისამართები, რაც უდავოდ ამცირებს შეტევების რისკს. Ipsec ოქმის ძირითადი დანიშნულებაა უზრუნველყოს TCP/IP ქსელებში დაცული კავშირი ე.წ. „წერტილი-წერტილთან“. სტანდარტის შემქნელია Internet Engineering Task For (IETF). განვიხილოთ ზემოთ აღნიშნული Ipsec შემავალი ოქმები:

- AH ოქმი არ შიფრავს მონაცემებს, ის უზრუნველყოფს პაკეტების „ელექტრონულ ხელმოწერას“. ამ ოქმის თავსართის ფორმაა იხ. ნახ.8.1.2.

შემდეგი თავსართი	პაკეტის სიდიდე Payload Len	რეზერვისათვის
SPI უსაფრთხოების პარამეტრების ინდექსი		
Sequence Number Field შემდგომი პაკეტის ნომერი		
აუტენფიკაციის მონაცემები		

ნახ. 8.1.2.

ნახ.8.1.3 ბ) ნაჩვენებია როგორ იცვლება სატრანსპორტო რეჟიმში ჩვეულებრივი IP პაკეტი ნახ. 8.1.3 ა) AH ოქმის თავსართით.

IPv4

თავსართის ორიგინალი IP	AH	TCP	მონაცემები

ნახ. 8.1.3 ა)

IPv6

თავსართის ორიგინალი IP	ტრანზიტი, მისამართები, მარშრუტიზაცია, ფრანგმენტაცია	AH	მისამართი	TCP	მონაცემები

ნახ. 8.1.3 ბ)

ნახ. 8.1.4 ა) ნაჩვენებია როგორ იცვლება „გვირაბოვან“ რეჟიმში ჩვეულებრივი IP პაკეტი AH ოქმის თავსართით ნახ. 8.1.4 ბ)

IPv4

ახალი თავსართი IP	AH	თავსართის ორიგინალი IP	TCP	მონაცემები

ნახ. 8.1.4. ა)

IPv6

ახალი თავსართი IP	გაფართოებული თავსართი	AH	თავსართის ორიგინალი IP	TCP	მონაცემები

ნახ. 8.1.4 ბ)

- **ESP** შიფრავს მთლიან მონაცემებს და არა ცალკეულ პაკეტს. ამ ოქმის თავსართის ფორმატი მოცემულია ნახ. 8.1.5.

SPI უსაფრთხოების პარამეტრების ინდექსი
Sequence Number Field შემდგომი პაკეტის ნომერი
ESP Authentication Data დატვირთვის მახასიათებლები
დამატება (0-255 ბაიტი)
აუტენფიკაციის მონაცემები

ნახ. 8.1.5.

აქ გამოიყენება შიფრაციის შემდეგი ალგორითმები 3DES, RC5, IDEA, CAST, Blowfish. ნახ.8.1.6 და ნახ.8.1,7 ნაჩვენებია ESP ოქმით შექმნილი ახალი პაკეტები სატრანსპორტო და „გვირაბოვან“ რეჟიმისათვის.

თავსართის ორიგინალი	ტრანზიტი, მისამართები, მარშრუტიზაცია, დრანგმენტაცია	**ESP** თავსართი	მისამართები	TCP	მონაცემები	**ESP** დაბოლოება	**ESP** აუტენფიკატორი

ნახ. 8.1.6.

თავსართის ახალი მნიშვნელობა	გაფართოებული თავსართი	**ESP** თავსართი	თავსართის ორიგინალი	TCP	მონაცემები	**ESP** დაბოლოება	**ESP** აუტენფიკატორი

ნახ. 8.1.7.

- **IKE** ოქმი უზრუნველყოფს გასაღებების დაცულ ურთიერთგაცვლას, ათანხმებს აუტენფიკაციისა და შიფრაციის გამოყენებულ ალგორითმებს, ადგენს გასაღებების გამოყენების ვადებს.
- **AES, 3DES**, შირაციის სტანდარტებია:
- **SHA-1, MD5, HMAC** ჰეშ ფუნქციების სტანდარტებია:
- **XAUTH** გაფართოებადი აუტენფიკაციის ოქმია. იგი დამატებითი დაცვაა RADIUS ოქმის მიხედვით და ერთჯერადი პაროლების გენერაციით.
- **STS** კრიპტოგრაფიული ოქმია, რომელიც საშუალებას აძლევს ორ მხარეს მიიღოს საერთო საიდუმლო გასაღები ღია არხების საშუალებით. გამოყენებულია დიფი-ჰელმანის ალგორითმი და სიმეტრიული შიფრაციის ალგორითმი. იგი მიეკუთვნება კლასს AKC (*authenticated key agreement with key confirmation).*
- **ISAKMP** მას არა აქვს დამთავრებული ოქმის სახე. იგი წააგავს იმ სამშენებლო მასალების ნაკრებს, რომლებიდანაც შეიქმნება სისტემა გასაღებების ურთიერთგაცვლისა და Security Association (SA) კავშირებისათვის.

შეიძლება მივიღოთ, რომ IPsec ოქმის ფუნქციონირება ორ ფაზიანია. პირველ ფაზაში IKE ოქმით ხდება მხარეებს შორის შეთანხმება გამოსაყენებელ შიფრაციის ალგორითმზე, მთლიანობის შემმოწმებელ ალგორითმზე და თუ როგორ განახორციელებენ მხარეები ერთმანეთის აუტენფიკაციას. აქვე ხდება სპეციალური კავშირების პარამეტრების შეთანხმება. შემდგომში ხდება გასაღებების გაცვლა დიფი-ხელმანის ალგორითმით. ამჟამად ფუნქციონირებს ორი ოქმი IKEv1 и IKEv2. მიღებულია, რომ პირველი ფაზის განხორციელებით ფორმირდება უსაფრთხოების პოლიტიკა IPsec SA (Security Association) და დამხმარე „გვირაბი“.

მეორე ფაზაში, უკვე ერთმანეთის მიმართ ნდობის მქონე მხარეები, თანხვდება ერთმანეთთან მონაცემთა გადასაცემ „გვირაბზე“. ისინი სთავაზობენ ერთმანეთს ვარიანტებს transform-set პარამეტრის საშუალებით და თუ თანხვდებიან - იქმნება ძირითადი „გვირაბი“. დაცულობის ასამაღლებლად ცნობილია ე.წ. 1.5. ფაზა პირველ და მეორე ფაზებს შორის. მისი ორგანიზება ხდება XAUTH ოქმის და ე.წ. მოდეკონფიგურაციის (MODECFG) დახმარებით.

§ 8.2 WEB უსაფრთხოების უზრუნველყოფა

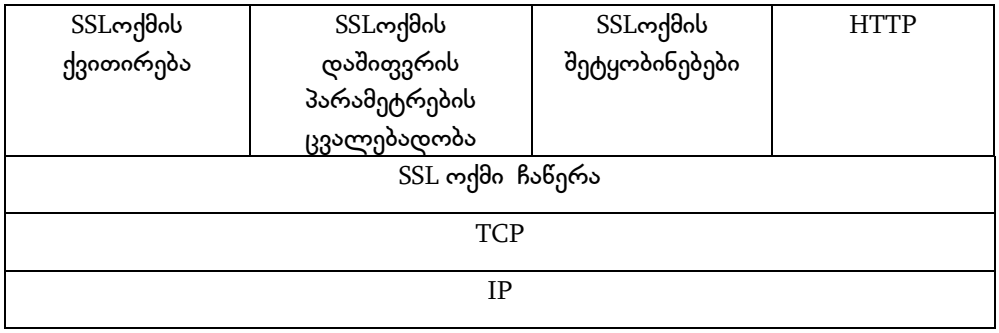
დღეს, ისე როგორც არასდროს, მწვავედ დაგას WEB-ის დაცვის საკითხი [36]. დაცვის ერთ-ერთი საშუალებაა ზემოთ განხილული Ipsec ოქმის გამოყენება, რომელიც გამოირჩევა თავისი უნივერსალობით და საიმედოობით TCP ოქმების გამოყენებისას. თუმცა არსებობს მეორე მიმართულება, რომელიც უზრუნველყოფს დაცვის საშუალებების განთავსებას „ზემოდან“ TCP ოქმებზე. ასეთებია SSL (Secure Socket Layer), TLS (Transport Layer Security) და SET (Secure Electronic Transactions). ცნობილია ამ ოქმების გამოყენების ორი გზა. პირველი, როცა SSL (TLS) ოქმების საშუალებები ჩაშენებულია უშუალოდ სატრანსპორტო ოქმში და მეორე, როცა იგი ჩაშენებულია უშუალოდ გამოყენებად პროგრამებში, მაგ. SET.

- WEB უსაფრთხოება უნდა უზრუნველყოფდეს;
- ინფორმაციის მიღებას მიმდინარე მომხმარებელზე :
- მიმდინარე მომხმარებელზე სახელის მიღებას:
- მიმდინარე მომხმარებლის იდენტიფიკატორის მნიშვნელობის არსებობაზე და სხვ.

გამოყენებული მეთოდებია: მითითებული მომხმარებლისთვის პაროლის შეცვლა, მოთხოვნილი ინფორმაციის არსებობის და მასთან შეღწევის დადასტურება, ადგენს ახალ აღრიცხვის ჩანაწერეს მითითებული მომხმარებლის და მის ახალ პაროლს, ადგენს ახალ პროფილს მითითებული მომხმარებლისათვის, ადგენს ახალ დროსა და თარიღის მნიშვნელობას, ადგენს არასწორად პაროლის შეყვანის მცდელობების რაოდენობას და კიდევ მრავალი.

§8.2.1 SSL ოქმის არქიტექტურა

SSL ოქმი დამუშავდა 1995 წ. Netscape ფირმის მიერ, როგორც გამოყენებითი დონის სერვისული ოქმებს (HTTP, SNTP, FTP და ა.შ.) და სატრანსპორტო ოქმს (TCP/IP) შორის ინფორმაციის დასაცავად. იგი შედგება შემდეგი დონეებისაგან იხ. ნახ. 8.2.1.



ნახ 8.2.1

მისი ფუნქციონირების ალგორითმი შემდეგია იხ. ნახ. 8.2.2:

მიღებული შეტყობონება იშლება ფრაგმენტებად, შესაბამისი ზომის ბლოკებად არა უმეტეს 2^{14} ბაიტი;

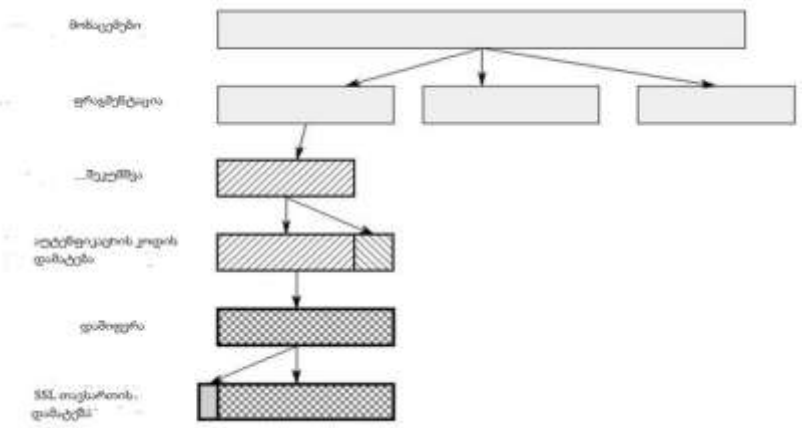
საჭიროების შემთხვევაში ხდება მათი შეკუმშვა. ოქმებში SSLv3 და TLS-სავალდებულოა;

გამოითვლება აუტენთიკაციის კოდი MAC. იგი გამოითვლება საიდუმლო გასაღებით Kc შემდეგი ფორმულით $MAC = h(Kc \| pad_2 \| h(Kc \| pad_1 \| seq\_num \| SSL.message))$.აქ pad_1 ბაიტია (0011 0110), pad_2 ბაიტია (0101 1100), seq_num მოცემული ფრანგმენტის რიგითი ნომერია, SSL.message სამი ველისაგან შემდგარი შეტყობინებაა, **h** ჰეშირების ფუნქციაა (MD5 და SHA1);

მონაცემები იშიფრება სიმეტრიული შიფრაციის მეთოდით (IDEA, RC2-40, DES-40, DES, 3DES, Forteza). Forteza გამოიყენება smart კარტებში;

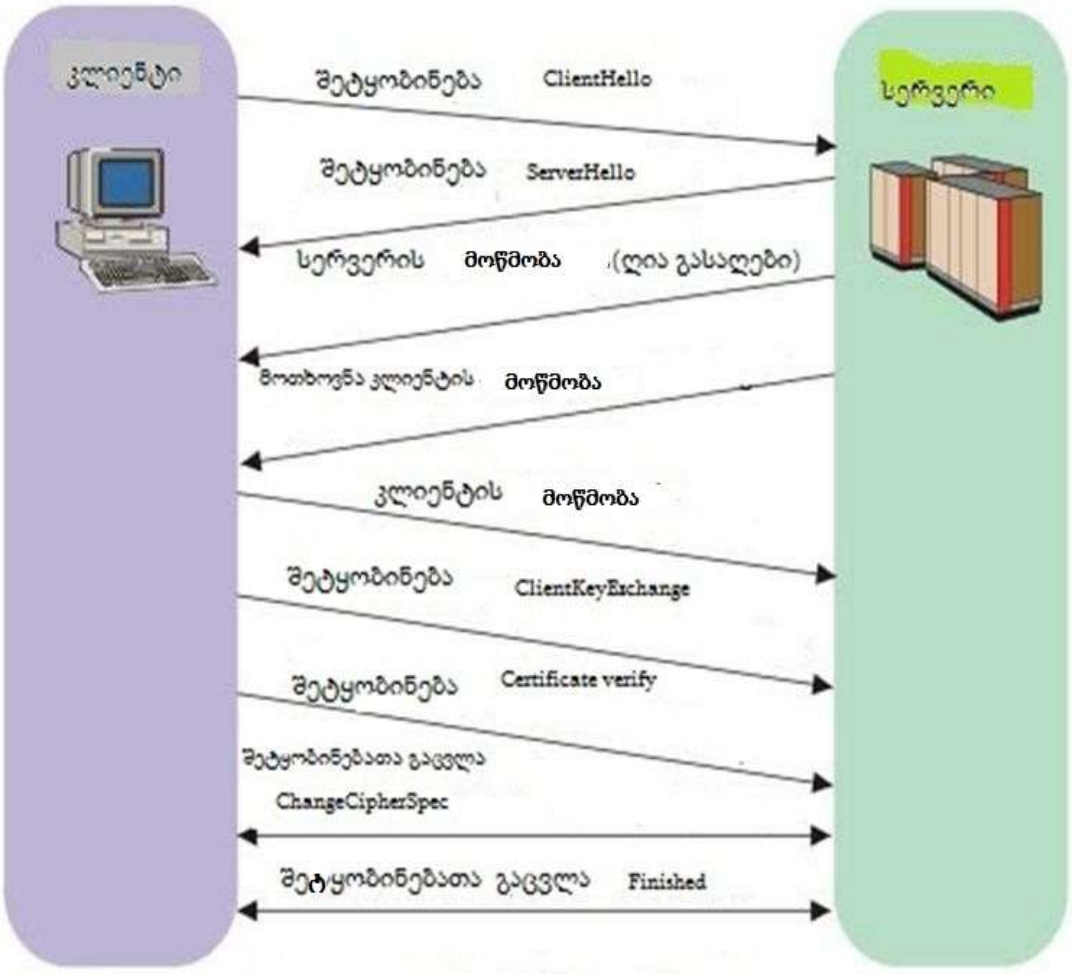
იქმნება ახალი თავსართი;

შექმნილი პაკეტები გადაეცემა TCP სეგმენტს;



ნახ. 8.2.2.

SSL ოქმით სერვერსა და კლიენტს შორის უღთიერთობა გადმოცემულია ნახ. 8.2.3.



ნახ. 8.2.3.

დასკვნის სახით შეიძლება ითქვას, რომ IPsec საყოველთაოდ გავრცელებული ოქმია. მისი პროგრამული რეალიზაცია, პირველად, განხორციელებულია კომპანია Microsoft მიერ ოპერაციულ სისტემაში Windows2000. ხოლო აპარატურულ მხარდაჭერას უზრუნველყოფენ კომპანიები Cisko და Nokia. შედარება Ipsec და SSL ოქმებს შორის მოცემულია ცხრ. 8.2.1.

ცხრ.8.2.1.

მახასიათებლები	Ipsec	SSL
აპარატურული დამოუკიდებლობა	+	+
კოდი	არ ითხოვს ცვლილებებს დანართებისათვის. შეიძლება მოითხოვოს TCP/IP სტეკის საწყის კოდთან შეღწევა	ითხოვს დანართებში ცვლილებებს. შეიძლება მოითხოვოს ახალი DLL ან დანართების საწყის კოდთან შეღწევა
დაცვა	მთლიანად იცავს IP პაკეტს. იცავს უმაღლესი დონის ოქმებს	მხოლოდ დანართების დონის
პაკეტების ფილტრაცია	დაფუძნებულია თავსართების, მისამართების და სხვათა უტენფიკაციაზე, ადვილია, გამოდგება როუტერებისათვის	დაფუძნებულია მაღალი დონეების შინაარსობრივ სახეზე და სემანტიკაზე. რთულია და უფრო ინტელექტუალური.
წარმადობა	გამოირჩევა მონაცემთა გადაადგილების მცირე რაოდენოით	გამოირჩევა მონაცემთა გადაადგილების დიდი რაოდენოით, მონაცემთა დიდი ზომის ბლოკებით, რომლითაც

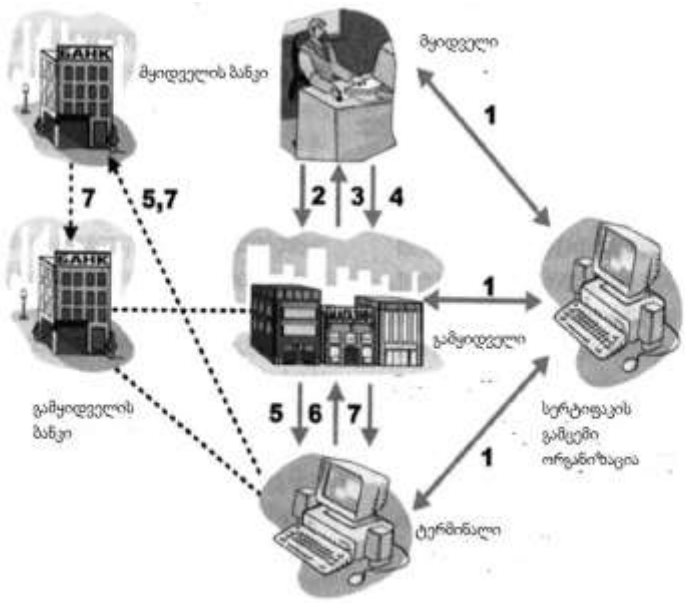
		ჩქარდება კრიპტოგრაფიული ოპერაციები და უმჯობესდება შეკუმშვა.
პლატფორმა	ნებისმიერი სისტემები როუტერების ჩათვლით	ძირითადად კლიენტ/სერვერები და firewalls
Firewalls/VPN	ტრაფიკი მთლიანად დაცულია	დაცულია მხოლოდ გამოყენებითი დონის ტრაფიკი. ICMP, RSVR, QoS და სხვა. შეიძლება დაცული არ იყოს
„გამჭვირვალეობა“	მომხმარებლებისათვის და დანართებისათვის	მხოლოდ მომხმარებლებისათვის
მიმდინარე სტატუსი	სტანდარტი	გამოყენებადია

§8.2.2 SET ოქმი

SET (Security Electronics Transaction) ქსელში ელექტრონული ტრანზაქციების დაცვის უზრუნველყოფის ოქმია, იგი დამუშავებული იყო კომპანიების MasterCard და VISA მიერ. ძირითადი დანიშნულებაა უსაფრთხო ანგარიშსწორების განხორციელება ქსელში (Internet). იგი აგრეთვე უზრუნველყოფს პროგრამულ თავსებადობის შესაძლებლობას ამ ოპერაციებში მონაწილე მომხმარებლების გამოყენებით პროგრამებს შორის. ე.ი. საქმე გვაქვს ე.წ. „ღია“ სპციფიკაციებთან. ***SET*** ოქმი უზრუნველყოფს:

- შეკვეთის ინფორმაციის კონფიდენციალობას და გადახდის მონაცემების საიდუმლოებას;
- გადახდის მონაცემთა მთლიანობას ელექტრონული (ციფრული) ხელმოწერის საშუალებით;
- აუტენფიკაციას ასინქრონული კრიპტოგრაფიის ღია გასაღების საშუალებით;
- მყიდველისა და გამყიდველის აუტენფიკაციას;
- გამყიდველისა და მყიდველის ბანკების აუტენფიკაციას;
- აუტენფიკაციის შედეგის მიხედვით განხორციელდეს ტრანზაქცია;
- მონაცემთა გადაცემისას კრიპტოგრაფიული დაცვის მექანიზმების გამოყენება.

ამ ოპერაციებში მონაწილეთა ურთიერთმოქმედებები გადმოცემულია ნახ. 8.2.3.



ნახ. 8.2.3.

ნახ. 8.2.3. მოცემულია შემდეგი მოქმედებები:

- მონაწილენი უკვეთენ და იღებენ სერტიფიკატებს სერტიფიკატების გამცემი ორგანიზაციებიდან;
- პლასტიკური ბარათის მფლობელი ნახულობს ელექტრონულ კატალოგს და გამყიდველს საქონელს უკვეთს ქსელით;
- გამყიდველი წარუდგენს თანხმობის ნიშნად თავის სერტიფიკატს ბარათის მფლობელს;
- მყიდველი წარუდგენს თავის სერტიფიკატს გამყიდველს;
- გამყიდველი თხოვს ტერმინალს („გადახდის რაბი“) განახორციელოს კონტროლი წარდგენილ ინფორმაციაზე;
- კონტროლის შედეგები ეტყობინება გამყიდველს;
- გამყიდველი ითხოვს ტერმინალისაგან განახორციელოს შესაბამისი გადარიცხვა.

ამ ტექნოლოგიაში გამოყენებულია სპეციფიკაცია Chip Electronic Commerce. სმარტ ბარათების სტანდარტის EMV შემქნელებია Europay, MasterCard და VISA.

თავი9. წამყვან ქვეყნებსა და საქართველოში ინფორმაციის დაცვის მიმართულებით მოქმედი სტანდარტები და კანონები

§9.1. აშშ-ის თავდაცვის სამინისტროს კომპიუტერული სისტემების თავდაცვის კრიტერიუმები („ნარინჯისფერი წიგნი“)

მოთხოვნებში ჩამოყალიბებულია 3 ძირითადი კრიტერიუმი:

- უსაფრთხოების პოლიტიკა;
- აუდიტი;
- კორექტურობა.

ამ კრიტერიუმების საფუძველზე გარკვეულია ექვსი უსაფრთხოების ბაზური მოთხოვნა:

- უსაფრთხოების პოლიტიკა;
- ნიშნულები;
- იდენტიფიკაცია და აუტენფიკაცია;
- რეგისტრაცია და აღრიცხვა;
- დაცვის საშუალებების კორექტური ფუნქციონირების კონტროლი:
- დაცვის უწყვეტობა.

ამ სტანდარტით მიღებულია კომპიუტერული სისტემების უსაფრთხოების 4 კლასი (ჯგუფი) A, B, C, D ნახ. 8.1. უმდაბლესი კლასია D, უმაღლესი A.

უმდაბლესი კლასი D. ზოგადად ეს არის სისტემები, რომლებიც ვერ აკმაყოფილებს დანარჩენი კლასის მოთხოვნებს.

კლასი C. ძირითადი თვისებაა ობიექტების მოქმედებებისა და შეღწევის მართვის შეუზღუდავობა. კლასი იყოფა :

კლასი C1. დისკრეციული დაცვა. სისტემა მრავალმომხმარებლებიანია, რომლებიც სარგებლობენ საიდუმლო ინფორმაციის დაცვის ერთნაირი დონით. მათ აქვთ შესაძლებლობა შეზღუდოს სხვა მომხმარებელი, რათა დაიცვას საკუთარი პრივატული ინფორმაცია.

კლასი C2. შეღწევის მართვა. წარმოებს გამოყოფილი რესურსების და მომხდარი მოვლენების გათვალისწინებით, მომხმარებლების მოქმედებების ინდივიდუალური კონტროლის საშუალებებით, მათი შეღწევის უფრო არჩევითი მართვა, რეგისტრაცია .

კლასი B. მანდატური დაცვა. ძირითადი დანიშნულებაა, უზრუნველოს შეღწევის ნორმატიული მართვა უსაფრთხოების ნიშნულების, მოდელების მხარდაჭერით, უსფრთხოების პოლიტიკით და უსაფრთხოების ბირთვზე სპეციფიკაციების არსებობით. ამ კლასის სისტემებში გათვალისწინებულია მონიტორინგის ჯგუფების არსებობა.

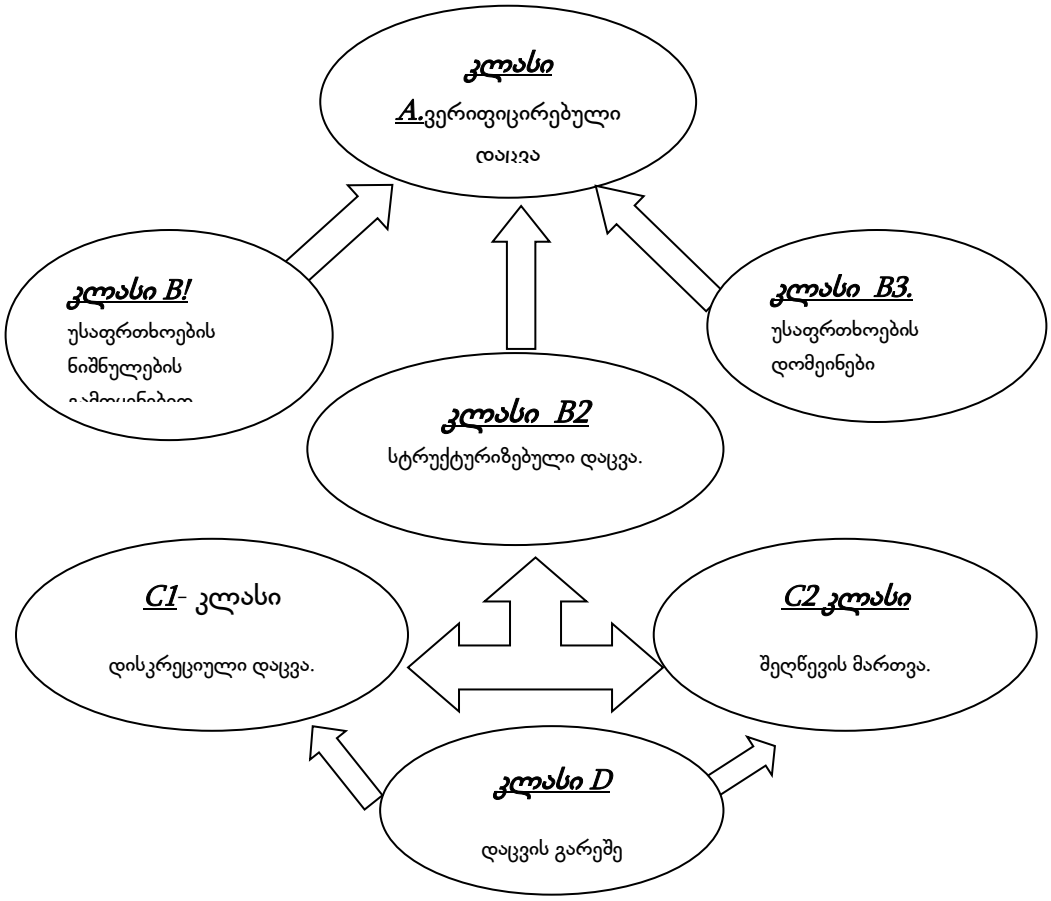
კლასი B1. უსაფრთხოების ნიშნულების გამოყენებით. აქ უზრუნველყოფილია *C2* კლასის ყველა მოთხოვნა, უზრუნველყოფილია უსაფრთხოების მოდელით გათვალისწინებულია მონაცემთა მარკირება და შეღწევის ნორმატიული მართვა. ინფორმაციის ექსპორტის შემთხვევაში ხდება მათი მარკირება, ხოლო ტესტირების დროს აღმოჩენილი ცდომილებები აუცილებლად სწორდება.

კლასი B2. უზრუნველყოფს ყველა მომხმარებლისათვის, უსაფრთხოების მოდელის მიხედვით, ნებისმიერი და დოკუმენტირებული შეღწევის მხარდაჭერას. დამატებით უნდა იქნეს განხილული კონტროლირებადი ინფორმაციის შესაძლო გაჟონვის ფარული არხები. ამ კლასის უსაფრთხოების ბირთვში უნდა იქნენ განსაზღრული უსაფრხოების მიმართ კრიტიკული ელემენტები. მისი ინტერფეისი საშუალებას უნდა იძლეოდეს სატესტო გამოცდების ჩატარებას. უსაფრთხოების მართვა ხორციელდება ადმინისტრატორის მიერ, რომელსაც შეუძლია სისტემის კონფიგურაციის მართვა. ***B1*** კლასისაგან დამატებით, აქ გაძლიერებულია აუტენფიკაციის საშუალებებიც.

კლასი B3. უსაფრთხოების დომეინები. ამ კლასის უსაფრთხოების ბირთვი ახორციელებს მუდმივ მონიტორინგს ყველა სახის ობიექტების შეღწევას ობიექტებთან, რომელთა გვერდის ავლაც შეუძლებელია. იგი ისეთი უნდა იყოს, რომ სისტემიდან გამორიცხოს ისეთი ქვესისტემები, რომლებსაც არ ევალებათ დაცვის ამოცანები. იგი უნდა შეიცავდეს სიგნალიზაციის საშუალებებს ადმინისტრატორისათვის და სამუშაო მდგომარეობის აღმდგენ საშუალებებს.

კლასი A. ვერიფიცირებული დაცვა. ნებისმიერი და ნორმატიული შეღწევის მართვის მექანიზმების, კორექტული მუშაობის ვერიფიკაციის ფორმალური მეთოდების გამოყენება. საჭიროებს ამ ფუნქციის განსახორციელებლად დამატებით დოკუმენტაციას.

კლასი A1 ფორმალური ვერიფიკაცია. მას არ ახასიათებს ***კლასი B3*** ფუნქციურ მოთხოვნებზე უფრო მეტი. განსხვავება მდგომარეობს იმაში, რომ სისტემის დამუშავების პროცესში უნდა იქნეს გამოყენებული ვერიფიკაციის ფორმალური მეთოდები, რათა შედეგად მიღებული იქნეს დაცვის ფუნქციის კორექტული მნიშვნელობები. ვერიფიკაციის მეთოდების უზრუნველსაყოფად იგი უნდა შეიცავდეს კონფიგურაციის მართვის უფრო მძლავრ საშუალებებს და დისტრიბუციის დაცულ პროცედურებს.



ნახ .8.1

§9.2 საინფორმაციო ტექნოლოგიების უსაფრთხოების ევროპული კრიტერიუმები

აქ განიხილება სამი მიმართულება:

- უსაფრთხოების უზრუნველსაყოფი მიზნები (ობიექტები);
- დაცვის ფუნქციის სპეციფიკაციები;
- დაცვის ფუნქციის სპეციფიკაციების უზრუნველსაყოფის მექანიზმები;
- დაცვის ფუნქციის სპეციფიკაციები უნდა შეიცავდეს იდენტიფიკაციას და აუტენფიკაციას, შეღწევის მართვას, ანგარიშგებას, აუდიტს, ინფორმაციის მთლიანობას, მომსახურების საიმედოობას, მონაცემთა გაცვლის უსაფრთხოებას, უტყუარობას.

აქ, აშშ სტანდარტის „ნარინჯისფერი წიგნი“-ან განსხვავებით, განიხილება 10 უსაფრთხოების კლასი, რომელთაგან, 5 ემთხვევა ამერიკულ სტანდარტს (F-C!, F-C2, F-B!, F-B2, F-B3). დამატებითია შემდეგი კლასები:

- ***კლასი F-IN*** . აქ ძირითადად უმაღლესი მოთხოვნები აქვს წაყენებული მთლიანობას (მაგალითად მონაცემთა მართვის სისტემები). მომხმარებლებს აქვს წაკითხვის, ჩაწერის, დამატების, წაშლის, შექმნისა და სახელის შეცვლის უფლებები.
- ***კლასი F-AV.*** აქ გაზრდილია მოთხოვნები სისტემის მუშაობის უნარიანობაზე. მან უნდა უზრუველყოს კრიტიკულ მომენტებშიც სასიცოცხლოდ მნიშვნელოვან მონაცემებთან შეღწევა და სისტემის კომპონენტების სარეზერვოთი ცვლა. დატვირთვის მიუხედავად სისტემას უნდა ჰქონდეს გარე ზემოქმედებებზე რეაქციის გარანტირებული დრო.
- ***კლასი F-D1.*** ინფორმაციის განაწილებულ დამუშავების სისტემებია. ინფორმაციის მიღებისა და გაცვლის დაწყებამდე უნდა წარმოებდეს ურთიერთმხარეების იდენტიფიკაცია და ინფორმაციის უტყუარობა (ჭეშმარიტება). უნდა შეიცავდეს მტყუნებების აღმოჩენისა და მათი გასწორების (აღდგენის) საშუალებებს. შეუძლებელი უნდა იქნეს მონაცემთა მოდიფიკაცია და მათი ხელმეორედ, უკვე გადაცემულის, გადაცემა.
- ***კლასი F-DC.*** აქ ძირითადი მოთხოვნაა კონფიდენციალობის დაცვა. შესაბამისად არხებში ინფორმაცია გადაიცემა დაშიფრული სახით.
- ***კლასი F-DX.*** დამატებითი მოთხოვნაა ინფორმაციის მთლიანობაზე და ტრაფიკის ანალიზის დაცვაზე. თითქოსდა აერთიანებს F-D1 და F-DC კლასებს. კრიპტოანალიზის შესაფერხებლად შეზღუდულია ადრე გადაცემული ინფორმაციასთან წვდომა.

განარჩევენ ადეკვატურობის 3 დონეს:

- მინიმალური ადეკვატურობა. ანალიზი უკეთდება სისტემას დაპროექტებიდან ექსპლუატაციამდე და თანხლებამდე;
- ანალიზი უკეთდება სისტემის მხოლოდ ზოგად არქიტექტურას, ხოლო დაცვის სისტემის ადეკვატურობა მოწმდება ფუნქციური ტესტირებით;
- ანალიზი უკეთდება პროგრამის საწყის ტექსტებს და აპარატურული უზრუნველყოფის სქემებს.

§9.3 საინფორმაციო ტექნოლოგიების უსაფრთხოების დაცვის რუსეთის სახელმწიფო კომისიის მოთხოვნები

იდეოლოგიურად უსაფრთხოების მიმართ მიდგომა კლასიფიცირებულია 2 ჯგუფში:

- დაცვის მახასიათებლები არასანქცირებული შეღწევებისგან თავის დასაცავად;

- მანაცემთა დამუშავების ავტომატიზებული სისტემების დაცვის კრიტერიუმები.

ზოგადად მიღებულია დაცვის 6 კლასი, რომლებზეც მოთხოვნები გადმოცემულია ცხრ.8.3. V გამოსახულებით აღინიშნება არასავალდებულო მოთხოვნა.

ცხრ. 8.3.

+

მახასიათებლების დასახელება	უსაფრთხოების კლასები					
	1	2	3	4	5	6
დისკრეციული პრინციპის შეღწევის კონტროლი	+	+	+	V	+	V
მანდატური პრინციპის შეღწევის კონტროლი	-	-	+	V	V	V
მახსოვრობის გაწმენდა	-	+	+	V	V	V
მოდულების იზოლირება	-	-	+	V	+	V
დოკუმენტების მარკირება	-	-	+	V	V	V
ინფორმაციის განმარტოებული მატარებლის დაცვა ინფორმაციის შეტან-გამოტანაზე	-	-	+	V	V	V
მომხმარებლის და მოწყობილობის შედარება (შეთავსება)	-	-	+	V	V	V
იდენტიფიკაცია და აუტენფიკაცია	+	V	+	V	V	V
პროექტირების გარანტიები	-	+	+	+	+	+
რეგისტრაცია	-	+	+	+	V	V
მომხმარებლი ურთიერთქმედება დაცვის საშუალებების კომპლექსთან	-	-	-	+	V	V
საიმედო აღდგენა	-	-	-	+	V	V
დაცვის კომპლექსის მთლიანობა	-	+	-	+	V	V
მოდიფიკაციის კონტროლი	-	-	-	-	+	V
დისტრიბუციის კონტროლი	-	-	-	-	+	V
გარანტია არქიტექტურაზე	-	-	-	-	-	V
ტესტირება	+	+	+	+	+	V
მომხმარებლის ინსტრუქცია	+	V	V	V	V	V
ინსტრუქციები დაცვის საშუალებების კომპლექსზე	+	+	V	+	+	V

სატესტო დოკუმენტაცია	+	+	+	+	+	V
საკონსტრუქტორო დოკუმენტაცია	+	+	+	+	+	+

§ 9.4 საინფორმაციო ტექნოლოგიების უსაფრთხოების დაცვის მიმართულებით საქართველოში მოქმედი კანონები.

საქართველოს კანონი
ელექტრონული ხელმოწერისა და ელექტრონული დოკუმენტის შესახებ.

ამ კანონის მიზნები და მოქმედების სფეროა: განისაზღვროს ელექტრონული დოკუმენტბრუნვის სისტემის და მასში ელექტრონული ხელმოწერის გამოყენების სამართლებრივი საფუძვლები, უზრუნველყოს ელექტრონული ხელმოწერის უსაფრთხოების პოლიტიკის განხორციელება ამ კანონის რეგულირების სფეროში. იგი არ ვრცელდება ინფორმაციის სახეობებზე, რომლებიც საქართველოს კანონმდებლობით აღიარებულია სახელმწიფო საიდუმლოებად და ექვემდებარება სახელმწიფო დაცვას.

კანონით აღიარებულია პირადი ხელმოწერისა და ელექტრონული („ციფრული ხელმოწერის") თანაბარი იურიდიული ძალა, ჩათვლილია ხელმოწერილი ელექტრონული დოკუმენტის ყველა ეგზემპლარი ორიგინალად, განსაზღვრულია ციფრული ხელმოწერის სერტიფიკატის (მოწმობა) ცნება და მისი მიღების და გაცემის ტექნოლოგია, სერტიფიკატის მფლობელის უფლებები და მოვალეობები, საზღვარგარეთ გაცემული ციფრული ხელმოწერის სერტიფიკატის აღიარება.

კანონი ძალაშია შესული 2008 წლის 14 მარტიდან N 5927 – Iს.

საქართველოსკანონი
ინფორმაციულიუსაფრთხოების შესახებ

ამკანონის მიზანია, ხელი შეუწყოს ინფორმაციული უსაფრთხოების დაცვის ქმედით და ეფექტიან განხორციელებას, დააწესოს საჯარო და კერძო სექტორების უფლება-მოვალეობები ინფორმაციული უსაფრთხოების დაცვის სფეროში, აგრეთვე განსაზღვროს ინფორმაციული უსაფრთხოების პოლიტიკის განხორციელების სახელმწიფო კონტროლის მექანიზმები. განსაზღვრულია, რომ კანონის მოქმედება ვრცელდება ყველა იურიდიულ პირსა და სახელმწიფო ორგანოზე, რომლებიც კრიტიკული ინფორმაციული სისტემის სუბიექტები არიან და მათთან დაკავშირებულ ყველა ორგანიზაციაზე და უწყებაზე, რომლებიც ექვემდებარება ან დაკავშირებულია სუბიექტთან, ამავე კანონით სუბიექტი ვალდებულია მიიღოს ინფორმაციული უსაფრთხოების შინასამსახურებრივი გამოყენების წესები. კანონით განსაზღვრულია ინფორმაციული უსაფრთხოების პოლიტიკის პასუხისმგებელი საქართველოს იუსტიციის სამინისტროს მმართველობის სფეროში მოქმედი საჯარო სამართლის იურიდიული პირი *მონაცემთა გაცვლის სააგენტო* (შემდგომ–მონაცემთა გაცვლის სააგენტო). მასვე ევალება სუბიექტის პასუხისმგებელ პირთან ერთად ინფორმაციული უსაფრთხოების შიგასამსახურებრივი გამოყენების წესების (ინფორმაციული უსაფრთხოების პოლიტიკის) მონაცემთა გაცვლის სააგენტოს მიერ დადგენილ უსაფრთხოების მინიმალურ სტანდარტებთან თავსებადობის შეფასების (ინფორმაციული უსაფრთხოების აუდიტი) ჩატარება.

პირველად განსაზღვრულია ე.წ. ინფორმაციული უსაფრთხოების მენეჯერის ცნება და დაგენილია მისი მოვალეობები. აქვე განსაზღვრულია კომპიუტერული უსაფრთხოების სპეციალისტის ცნება, მისი მოვალეობები და უფლებები.

კანონი ძალაშია შესული 2012 წლის 5 ივნისიდან №6391-Iს

გარდა ზემოთაღნიშნული ორი კანონისა, არსებობს „საიდუმლო“ გრიფიანი კანონი საიდუმლო საქმიწარმოებასთან დაკაშირებით და სტანდარტი „ინფორმაციის დაცვა გამოთვლითი ტექნიკით აღჭურვილ ობიექტებზე“.

ლიტერატურა

1 -2016 Trends in Cybersecurity
https://info.microsoft.com/rs/157-GQE-382/images/EN-MSFT-SCRTY-CNTNT-eBook-cybersecurity.pdf
2 -А.Н. Андрончик, В.В. Богданов, Н.А. Домуховский, А.С. Коллеров, Н.И. Синадский, Д.А. Хорьков, М.Ю. Щербаков .
Защита информации в компьютерных сетях Екатеринбург УГТУ–УПИ 2008
3 -Стратегии защиты данных.4.1. Комплексный подход - необходимое условие надежной защиты корпоративной сети
WWW CITForum.ru
4 -Приложение А. Виды компьютерных угроз
http://download.geo.drweb.com/pub/drweb/mac/doc/wks/ru/dw_mac_threat_types.htm
5 -Защита информации в локальных и глобальных компьютерных сетях
http://www.automationlab.ru/index.php/2014-08-25-13-20-03/467-41
6 -Основы сетевой безопасности. Сеть как объект защиты
http://www.4stud.info/networking/network-security.html
7 -Сетевая безопасностьСеменов Ю.А.
http://book.itep.ru/1/intro1.htm
8 -Борьба с мошенничеством в электронном бизнесе с помощью Oracle Adaptive Access Manager
http://www.oracle.com/technetwork/ru/middleware/id-management/oracle-adaptive-access-manager-427547-ru.html
9 -Л. С. Таганов, В. Г. ЛевинИНФОРМАТИКАУчебное пособиеКЕМЕРОВО 2006
10 -Удаленный доступ к информационным ресурсам_ Аутентификация Константин Кузовкин*журнал"Директор информационной службы" №9, 2003.*
11 - **Международные нормативно-правовые акты по безопасности .** Основные нормативные документы в области информационной безопасности
https://sites.google.com/site/infsecuritycoursesummary/home/gos-ekzamen/osnovnye-normativnye-dokumenty-v-oblasti-informacionnoj-bezopasnosti
12 -Access ControlRoger Needham Rick Maybury
https://www.cl.cam.ac.uk/~rja14/Papers/SE-04.pdf
13 -А.Л.Додохов - руководитель проекта, к.т.н. А.Г.Сабанов - коммерческий директор . **Обеспечение защиты персональных данных в СУБД Oracle** .GlobalTrustHQ
14 - **Oracle Identity Management**

http://www.oracle.com/us/products/middleware/identity-management/access-management/overview/index.html
15 -Oracle Access Manager
http://www.oracle.com/technetwork/ru/middleware/id-management/oam-1539410-ru.html
16 -Александр ПоляковНекоторые вопросы безопасности в Oracle
https://dsec.ru/ipm-research-center/article/nekotorye_voprosy_bezopasnosti_v_oracle_/
17 -Arto Salomaa Public-Key ryptography Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo Hong Kong Barcelona
18 -Таненбаум Э., Уэзеролл Д.Компьютерные сети,5-е изд. —СПб
Питер, 2012. — 960 с.: ил.ISBN 978-5-459-00342-0
19 -СовременныеАлгоритмыШифрованияСергей Панасенко
20 - The Miller-Rabin Randomized Primality TestBobby KleinbergCornell University, 5 May 2010
21 -sistema bastioni (System Bastion) saqpatenti N 4904 2011б.
22 -ASurveyofPublic-KeyCryptosystems.Neal Koblitz. Neal Koblitz
Dept. of Mathematics, Box 354350 Univ. of Washington, Seattle, WA 98195 U.S.A.
koblitz@math.washington.edu. Alfred J. Menezes
Dept. of Combinatorics & Optimization Univ. of Waterloo, Waterloo, Ontario N2L 3G1 Canada
ajmeneze@uwaterloo.ca
August 7, 2004
23 -Квантовая Криптография Д А Кронберг и др
http://sqi.cs.msu.su/store/storage/ss8dw5n_quantum_cryptography.pdf
24 - КВАНТОВАЯ КРИПТОГРАФИЯ: ПРИНЦИПЫ, ПРОТОКОЛЫ, СИСТЕМЫ Д.М. **Голубчиков, К.Е.** Румянцев
http://www.ict.edu.ru/ft/005712/68358e2-st14.pdf
25 -Компьютерная стеганография – защита информации или инструмент преступления?.Владимир Голубев.
http://www.crime-research.ru
26 -ОСНОВНЫЕ ПОЛОЖЕНИЯ СТЕГАНОГРАФИИ. *О. В. Генне, ООО "Конфидент"*
Опубликовано: журнал "Защита информации. Конфидент", №3, 2000
www.confident.ru/magazine
27 -Сунчугашев Иван 4 апреля 2008г *Стеганография.* Московский физико-технический институт (ГУ МФТИ),
http://www.re.mipt.ru/infsec
28 -Descriptions of SHA-256, SHA-384, and SHA-512
http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf
29 - SHA-384, SHA-512 Hashing, Fast (Helion)
http://www.xilinx.com/publications/3rd_party/products/Helion_Fast_SHA384_512_AllianceCORE_data_sheet.pdf

30 -Алексеев Е.Г., Богатырев С.Д. Информатика. Мультимедийный электронный учебник Саранск Морд гос ун-т 2009 http://inf.e-alekseev.ru/
31 -IPSec — протокол защиты сетевого трафика на IP-уровне 14 мая 2001г Станислав Коротыгин, home.al.ru/
32 -*Frequently A s ked Questions about To d ay 's Cry p t o g ra p h yv e r s i on 4.1 - May 2000*1992-2000 RSA Security Inc.
33 -MSRT_Microsoft Malware Protection Center
https://blogs.technet.microsoft.com/mmpc/page/2/
34 -Вредоносная программа
http://www.kaspersky.ru/internet-security-center/threats/malware-classifications
http://www.securitylab.ru/news/tags/%E2%F0%E5%E4%EE%ED%EE%F1%ED%E0%FF+%EF%F0%EE%E3%F0%E0%EC%EC%E0/
35 -Основы сетевой безопасности. Сеть как объект защиты Анатольев А.Г
http://www.4stud.info/networking/network-security.html
36 -*Защищенная платформа для Web-приложений.* Константин Кузовкин
http://old.i-teco.ru/article4.html
37 -Информационная безопасность баз данных Платформа безопасности Oracle. Сергей Базылько
http://www.oracle.com/technetwork/ru/indexes/oracle-pt-joint-1561104-ru.pdf
38 -Мещеряков Р.В. Информационная безопасность и защита информации в сетях ЭВМ.2008 г.ISBN 5-98298-198-2
39 -НФОИРМАЦИОННАЯ БЕЗОПАСНОСТЬ. С. И. МАКАРЕНКО.
Ставрополь СФ МГГУ им. М. А. Шолохова 2009
http://sccs.intelgr.com/editors/Makarenko/Makarenko-ib.pdf
УДК 681.322 ББК 32.973
40 -Yang Xiao,Frank H Li,Hui Chen. HANDBOOK OF SECURITY AND NETWORKS
https://books.google.ru/books?id=yKj8rUxL9JUC&pg=PP1&lpg=PP1&dq=Yang+Xiao,Frank+H+Li,Hui+Chen.+HANDBOOK+OF+SECURITY+AND+NETWORKS&source=bl&ots=Lp58bXxfZA&sig=J-Ym7cgsQXtpxS9XeHoAboScvA&hl=ru&sa=X&ved=0ahUKEwjsn_qcuObPAhVEDJoKHScWBHkQ6AEIMDAC#v=onepage&q=Yang%20Xiao%2CFrank%20H%20Li%2CHui%20Chen.%20HANDBOOK%20OF%20SECURITY%20AND%20NETWORKS&f=false
41 -Защита сетей штатными средствами
http://daybook.org.ua/seti/zashhita-setej-shtatnymi-sredstvami.html
42 -Сетевые ресурсы и их уязвимости
http://ivmai.chat.ru/student/netrvuln/netrvuln.htm
43 -Гладких А.А "Базовые принципы информационной безопасности вычислительных сетей"Ульяновск 2009. УДК 002:34+004.056.5
ББК 67.401+32.973.2-018.2 Г15

44 -ЗАЩИТА ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ С ИСПОЛЬЗОВАНИЕМ МЕЖСЕТЕВЫХ
ЭКРАНОВ ССПТ. САНКТ-ПЕТЕРБУРГ 2010
http://www.npo-rtc.ru/papers/books/nov_2010.pdf
45 -Лекция №18 Защита информации в сетях. Шифрация данных. Защита соединений. Классификация атак
https://www.youtube.com/watch?v=m5PqgJN7A3o
46 -RISSPAЕвгений Климов,
Существующие подходы к защите облачных сервисов
http://www.icl.ru/files/docs/events/itsf2012/Sushhestvujushhie%20podhody%20k%20zashhite%20oblachnyh%20servisov.pdf
47 -ОСНОВНЫЕ УГРОЗЫ БЕЗОПАСНОСТИИНФОРМАЦИИ В ВИРТУАЛЬНЫХ СРЕДАХИ ОБЛАЧНЫХ ПЛАТФОРМАХ
Зубарев Игорь Витальевич, Радин Павел Константинович
elibrary.ru/item.asp?id=21604798
48 -- Сети GRID. Семенов Ю.А. (ИТЭФ-МФТИ)
http://book.itep.ru/1/intro1.htm
49 -Средства защиты GRID-систем на основедифференцирования уровня доверия к узлам Системы.В.Е. Мухин.УДК 004.04
50 - Утечкиконфиденциальнойинформациив России и в мире Итоги 2016 года
www.zecurion.ru | analytics@zecurion.com | +7 495 221-21-60