KAUNO TECHNOLOGIJOS UNIVERSITETAS

Eligijus Sakalauskas,
Narimantas Listopadskis,
Gediminas Simonas Dosinas

KRIPTOGRAFIJOS TEORIJA

Mokomoji knyga

Leidyklos logo

UDK 003.2(075.8)
Sa84

Recenzavo : prof. dr. J.Rimas ir
dr. P.Nefas

Kalbos redaktorius: R.Kupčinskienė

Parengta įgyvendinant projektą „Informacijos saugos studijų modulių rengimas ir dėstytojų kvalifikacijos tobulinimas“
BPD2004-ESF-2.5.0-03-05/0082

Išleista Europos Sąjungos struktūrinių fondų ir Lietuvos Respublikos lėšomis, 2008

ISBN 978-9955-686-75-0

TURINYS

ĮVADAS

Doktorantūros studijų modulis „Kriptografijos teorija“ skirtas suteikti žinių ir įgūdžių, leidžiančių kurti kriptografines sistemas ir atlikti preliminarią šių sistemų saugumo analizę. Prieš studijuojant šį modulį, rekomenduojama išklausyti „Kriptografinių sistemų“ modulį arba susipažinti su kriptografijos pagrindais iš tam skirtų vadovėlių. Kriptografinių sistemų kūrimo įgūdžių įgijęs doktorantas žino pagrindinius kriptografinius primityvus ir remdamasis jais gali sukurti savą arba pagerinti esamą kriptosistemą. Be to, doktorantas gali atlikti pirminį esamų kriptosistemų auditą ir nustatyti bent gana dideles jų kriptografinio saugumo spragas. Gilesnis kriptografinio saugumo įvertinimas reikalauja nepalyginti aukštesnės kvalifikacijos, nei galima įgyti per ketverius doktorantūros metus.

Įsisavinus šį studijų modulį, įgyjama pakankamai aukšta kriptografo kvalifikacija, kuri artimiausiu metu bus gana paklausi naujausių informacinių technologijų rinkoje. Ši rinka neišvengiamai augs, nes tai lemia globaliniai politiniai procesai, kurių metu dabartinė visuomenė virsta informacine.

2005 m. Europos Komisija (EK; *European Commission*) paskelbė komunikatą Tarybai, Europos Parlamentui, Europos Ekonomikos ir socialinių reikalų komitetui ir Regionų komitetui KOM (2005) 229 „i2010 – Europos informacinė visuomenė augimui ir užimtumui skatinti“. Šiuo komunikatu EK pateikė rekomendacijas iki 2010 metų visoms Europos Sąjungos narėms perkelti didžiąją dalį valdžios paslaugų į internetinę terpę. Tokiu būdu Europa įsitraukė į globalinį mums įprastos valdžios transformacijos procesą – elektroninės valdžios, arba e. valdžios. Jeigu iki 2010 metų bus įgyvendintos komunikate numatytos rekomendacijos, naujausios informacinės ir drauge kriptografinės technologijos bus įgyvendintos daugumoje e. valdžios struktūrų.

Paradoksalu, tačiau panašių veiksmų Lietuvos Vyriausybė ėmėsi net kiek anksčiau. 2002 m. gruodžio 31 d. buvo priimtas Lietuvos Respublikos Vyriausybės nutarimas Nr. 2115 „Elektroninės valdžios koncepcija“ (Žin., 2003, Nr. 2 – 54). Pažymėtina, kad e. valdžia nuo to laiko tapo prioritetine Lietuvos valstybės politikos kryptimi.

Lietuva šiuo klausimu nelabai kuo skiriasi nuo kitų valstybių – greičiau ji stengiasi neatsilikti nuo besivystančio pasaulio.

„Kriptografijos teorijos“ doktorantūros studijų modulį rengė KTU Taikomosios matematikos katedros docentų grupė aktyviai padedant dviem doktorantams ir kitiems specialistams bei techniniams darbuotojams, vadovaujant doc. Eligijui Sakalauskui.

Grupės vadovo ir jos narių indėlis į paruoštą „Kriptografijos teorijos“ doktorantūros studijų modulį yra toks.

Doc. Eligijus Sakalauskas kartu su kolegomis dalyvavo ruošiant visas paskaitų medžiagas, joms skirtus savikontrolės, testų ir referatų užduotis.

Doc. Narimantas Listopadskis dalyvavo ruošiant 3, 4, 8, 9 paskaitas, joms skirtus savikontrolės, testų ir referatų užduotis.

Doc. Gediminas Dosinas dalyvavo ruošiant 2 paskaitą, jai skirtus savikontrolės, testų ir referatų užduotis.

Doc. Kastytis Ratkevičius dalyvavo ruošiant 12, 14 paskaitas, joms skirtus savikontrolės, testų ir referatų užduotis.

Doktorantai Kęstutis Lukšys ir Artūras Katvickis dalyvavo ruošiant referatų užduotis, savikontrolės ir testų klausimus.

Inž. Tomas Burba, metodistė Birutė Narijauskaitė, lab. Paulius Vitkus ir bak. Aistė Sakalauskaitė atliko medžiagos pateikimo metodikos ir dizaino sukūrimo darbus.

1. KRIPTOGRAFINIŲ SISTEMŲ SAUGOS ANALIZĖ

Šiuolaikinė kriptografija – tai mokslo šaka, sprendžianti elektroninės informacijos saugos problemas. Kadangi kasmet vis daugiau informacijos siunčiama elektroninėmis ryšio priemonėmis, labai svarbu užtikrinti jos saugumą, mat elektroninė informacija (toliau – informacija) dažniausiai perduodama nesaugiais kanalais, pavyzdžiui, interneto ryšiu, ir gali būti pasiekiama beveik visiems.

Nuo šiol darysime prielaidą, kad informaciją subjektas *A* perduoda subjektui *B*, kuriuos vaizdumo dėlei pavadinsime *Aldona* ir *Broniumi*. Tarkime, *Aldona* nori perduoti *Broniui* informaciją elektroninėmis priemonėmis. Čia kyla keli svarbūs uždaviniai:

1. **Informacijos konfidencialumas (angl. *data confidentiality*).** Reikia užtikrinti, kad *Aldonai* perduodant slaptą informaciją *Broniui*, jos negalėtų perskaityti kitas subjektas, kuriam prieinamas jų naudojamas ryšio kanalas. Tai pasiekiama informaciją šifruojant. *Aldona*, naudodamasi kokiu nors šifravimo algoritmu, užšifruoja informaciją taip, kad tik *Bronius* galėtų ją perskaityti, o kitiems subjektams ji būtų tik beprasmis simbolių rinkinys.
2. **Informacijos vientisumas (angl. *data integrity*).** Reikia užtikrinti, kad duomenų, kuriuos *Aldona* siunčia *Broniui*, būtų neįmanoma suklastoti ar pakeisti. Dažniausiai tam pasitelkiamos specialios santraukos funkcijos (angl. *hash function*), kurios iš turimų duomenų sugeneruoja duomenis atitinkančią santraukos reikšmę (angl. *hash code*). *Aldona*, siųsdama pranešimą *Broniui*, gali prie jo prisegti santraukos reikšmę. Jei *Bronius*, gavęs pranešimą ir jam pritaikęs tą pačią santraukos funkciją, gauna skirtingą reikšmę, galima užtikrintai tvirtinti, kad duomenys pakeliui buvo pakeisti.
3. **Subjekto autentifikavimas (angl. *subject authentification*).** *Broniui,* gavusiam informaciją, turi būti sudaryta galimybė įsitikinti, kad duomenų siuntėjas tikrai yra *Aldona*, o ne koks nors apsimetėlis. Vienas autentifikavimo metodų – elektroninis parašas.

Galima pastebėti, kad šiuolaikinės kriptografijos uždaviniai iš esmės nelabai skiriasi nuo analogiškų, sprendžiamų taikant rašytinius ar spausdintinius informacijos perdavimo būdus.

Reikia pabrėžti, kad kriptografija nesprendžia problemų, susijusių su informacijos perdavimo patikimumu. Siunčiamos informacijos apsaugojimo nuo iškraipymų, perduodant ją triukšmo veikiamomis ryšio linijomis, uždavinius nagrinėja kodavimo teorija. Kriptografijoje veikiančios jėgos kur kas subtilesnės, žmogiškosios prigimties. Tai – įvairūs įsibrovėliai, dėl tam tikrų priežasčių siekiantys sutrukdyti įgyvendinti tris mūsų užsibrėžtus tikslus. Sutarkime, kad įsibrovėliams toliau atstovaus subjektas *Z*, kurį vadinsime *Zose*.

Įsibrovėliai gali būti dviejų tipų:

1. **Pasyvieji.** Tinkle tokie įsibrovėliai nėra aktyvūs – jie tik stebi informacijos mainus tarp *Aldonos* ir *Broniaus*. Svarbiausias jų tikslas – pažeisti informacijos konfidencialumą.
2. **Aktyvieji.** Tokie įsibrovėliai ne tik stebi informacijos srautą, bet ir patys gali įsiterpti į jį, suklastoti, sugadinti ar perimti duomenų paketus, apsimesti legaliais kriptografinės sistemos vartotojais. Bendrąja prasme tokie įsibrovėlių veiksmai vadinami *apsimetimo ataka* (angl. *impersonation attack*), kuri gali būti labai pavojinga kai kurioms kriptosistemoms.

Grįžkime prie mūsų aptartų trijų pagrindinių kriptografijos uždavinių. Sujungę kelis jų į vieną, gauname dar keletą ne mažiau svarbių:

1. **Subjekto anonimiškumas (angl. *subject anonymity*).** Sujungus informacijos konfidencialumo ir subjekto autentifikavimo uždavinius, kyla subjekto anonimiškumo problema, kuri yra itin svarbi kalbant apie elektroninių pinigų ir elektroninio balsavimo sistemas.
2. **Informacijos autentiškumas (angl. *data authenticity*).** Sujungę informacijos vientisumo ir subjekto autentifikavimo uždavinius, galime ne tik užtikrintai pasakyti, kad duomenys yra teisingi, bet ir tai, kad jų autorius yra siuntėjas. Taigi *Aldona*, nusiuntusi duomenis *Broniui*,

vėliau negalės paneigti tai dariusi. Ši savybė kitaip dar vadinama neišsiginamumu (angl. *non-repudiation*).

Visi iki šiol minėti kriptografijos uždaviniai kompiuterių tinkluose (vidiniuose (angl. *intranet*) ir išoriniuose (angl. *internet*)) sprendžiami pasitelkus tam tikrus protokolus. Pavyzdžiui, atliekant banko operacijas internetu, naudojamas HTTPS protokolas, užtikrinantis duomenų konfidencialumą, vientisumą ir autentiškumą. Tiesa, kol kas (2007 metų duomenimis) informacijos ir vartotojo autentifikavimas vyksta gana archajišku būdu – naudojantis bendru bankui ir vartotojui žinomu slaptažodžiu. Tikimasi, kad ilgainiui šį metodą pakeis kur kas pranašesnis elektroninis parašas.

Trumpai apžvelgsime svarbiausius principus ir metodus, kuriais remiantis sprendžiami aukščiau suformuluoti kriptografijos uždaviniai. Pradėsime nuo informacijos konfidencialumo, pasiekiamo šifruojant duomenis.

Tarkime, kad *Aldona* nori perduoti tam tikrus duomenis *Broniui*. Šiuos duomenis nuo šiol vadinsime tekstograma (angl. *plaintext*) ir žymėsime *t*. Naudodamasi tam tikru užšifravimo algoritmu (angl. *encryption*), *Aldona* iš tekstogramos gaus šifrogramą (angl. *cyphertext*), kurią pažymėsime raide *c*. Šifrograma siunčiama *Broniui*, o jis, naudodamas atitinkamą iššifravimo algoritmą (angl. *decryption*), vėl gaus pradinę tekstogramą *t*.

Paprastumo dėlei tarkime, kad *Aldona* ir *Bronius* taiko simetrinį šifravimo būdą: vienas bendras slaptasis raktas (angl. *key*) *k* yra žinomas jiems abiem. Užšifravimo funkciją apibrėšime taip:

$$E_k(t) = c. \tag{1.1}$$

Analogiškai iššifravimo funkcija bus tokia:

$$D_k(c) = t. \tag{1.2}$$

Čia abi funkcijos yra priklausomos nuo to paties parametro *k*.

Nesunku pastebėti, kad

$$D_k(c) = t \Rightarrow D_k(E_k(t)) = t \Rightarrow D_k \equiv E_k^{-1}. \tag{1.3}$$

Atsiranda labai svarbus ryšys tarp užšifravimo ir iššifravimo funkcijų.

1.1. Savybė. Iššifravimo funkcija privalo būti atvirkštinė užšifravimo funkcijai.

Deja, ši savybė nėra pakankama, kad minėtos funkcijos tiktų šifravimui. Jeigu funkcija D_k yra nevienareikšmė, t. y. tą pačią reikšmę *c* atitinka didelė aibė reikšmių $\{t_1, t_2, t_3, ..., t_n\}$, kyla klausimas: kuri jų yra tikroji *t* reikšmė? Analogiški samprotavimai tinka ir užšifravimo funkcijai. Vadinasi, reikia suformuluoti dar vieną savybę:

1.2. Savybė. Funkcija E_k (tuo pačiu ir D_k) turi būti abipusiškai vienareikšmė (bijekcinė).

Jei abi šios sąlygos tenkinamos, vieną *t* visada atitinka viena *c*, ir atvirkščiai, o tai leidžia sėkmingai šifruoti. Tačiau realaus pasaulio sąlygomis ir šių dviejų savybių nepakanka duomenų konfidencialumui užtikrinti.

Dar 1883 metais Auguste'as Kerckhoffsas suformulavo šiais laikais visuotinai kriptografijoje priimtą principą: kriptosistemos saugumas privalo būti pagrįstas jos parametrų (šifravimo raktų) slaptumu, tačiau turi būti nepriklausomas nuo jos algoritmų slaptumo. T. y. kriptosistema privalo užtikrinti duomenų konfidencialumą net ir tada, kai jos algoritmas viešai žinomas. Taigi mūsų atveju tai reiškia, kad funkcijų *E* ir *D* išraiškos turi būti viešai žinomos.

Matome, kad bet kuriam kenkėjui prieinami šie duomenys: šifrograma *c*, užšifravimo funkcija *E* ir iššifravimo funkcija *D*. Akivaizdu, jog kriptosistemą turėtume laikyti nesaugia, jei šių duomenų visiškai pakaktų iššifruoti *c* arba – dar blogiau – apskaičiuoti slaptąjį raktą *k*. Pastaroji situacija kitaip dar vadinama kriptosistemos kompromitacija.

Šifrograma ir šifravimo funkcijos nėra vieninteliai duomenys, kuriuos gali turėti kenkėjas, todėl kriptosistemos turi būti kuriamos taip, kad būtų saugios net į piktavalių rankas nutekėjus daugiau informacijos. Pagal tai, kokią informaciją turi kenkėjas, kriptosistemų atakos (kriptoatakos) skirstomos į kelias grupes:

1. **Šifrogramos ataka.** Tai tokia ataka, kurios metu kenkėjas mėgina įsibrauti į kriptosistemą turėdamas tik mūsų aptartus duomenis. Jei kriptosistema pažeidžiama, ji vadinama absoliučiai nesaugia. Remiantis 1.1 formule galima teigti, kad c vienareikšmiškai atitinka tam tikrą t ir (arba) k, kuriuos kenkėjas gali rasti.
2. **Žinomos tekstogramos ataka.** Tai tokia ataka, kai užpuolikas žino vieną ar daugiau tekstogramos-šifrogramos porų.
3. **Parenkamos tekstogramos ataka.** Tai žinomos tekstogramos atakos porūšis, kai užpuolikas geba parinkti tam tikrą tekstogramą ir gauti jos kriptogramą.
4. **Adaptyviai parenkamos tekstogramos ataka.** Tai pati sudėtingiausia ataka, mat kenkėjui reikia ne vieno, o keleto mėginimų užšifruoti žinomas tekstogramas ir gauti jų šifrogramas, keičiant tekstogramų turinį pagal gautus rezultatus.

Bendruoju atveju paskutines tris atakas galima apibendrinti tokiu uždaviniu: duota E, D, t_1, c_1, t_2, c_2, ..., t_n, c_n, reikia rasti k. Mūsų, kaip kriptosistemų kūrėjų, tikslas – pasiekti, kad šis uždavinys būtų kuo sunkiau išsprendžiamas. Patogumo dėlei perrašykime E funkciją kaip dviejų argumentų funkciją:

$$E(k, t) = c. \tag{1.4}$$

1.3. Apibrėžimas. Funkcija $E(k, t) = c$ yra vadinama vienkrypte VKF (angl. *one-way function*) tada, kai turint k ir t lengva apskaičiuoti c, tačiau turint c labai sunku apskaičiuoti k ir t.

1.4. Apibrėžimas. Funkcija $E(k, t) = c$ yra vadinama stipria vienkrypte (angl. *strong one-way function*) tada, kai turint k ir t lengva apskaičiuoti c, tačiau turint t ir c labai sunku apskaičiuoti k, o turint c ir k labai sunku apskaičiuoti t.

1.5. Pastaba. Pateikti apibrėžimai nėra formalūs. Griežtas VKF apibrėžimo formalizavimas yra susijęs su fundamentaliomis matematikos pagrindų problemomis.

1.6. Pastaba. Daugelyje literatūros šaltinių stipri VKF apibrėžiama kitaip, bet mūsų naudojamas apibrėžimas leis lengviau suvokti dėstomą medžiagą

Dabar jau galime suformuluoti trečiąją savybę, kurią turi tenkinti šifravimo funkcijos, siekiant užtikrinti kriptosistemos saugumą.

1.7. Savybė. Tam, kad šifrosistemos būtų atsparios 2–4 atakoms, užšifravimo funkcija E privalo būti stipri VKF, t. y. turėdamas t ir c, užpuolikas negali lengvai rasti k.

Ši savybė užtikrina kriptografiškai saugų duomenų užšifravimą. Tačiau jei $E_k(t)$ yra stipri VKF, ką daryti, kad duomenis būtų įmanoma iššifruoti? Juk iššifravimui reikia apskaičiuoti funkciją $D_k(t)$, kuri yra atvirkštinė $E_k(t)$. Kadangi pastaroji funkcija – stipri VKF, tai yra labai sudėtingas uždavinys. Tam, kad išvengti šio prieštaravimo, įvedama landinės VKF sąvoka.

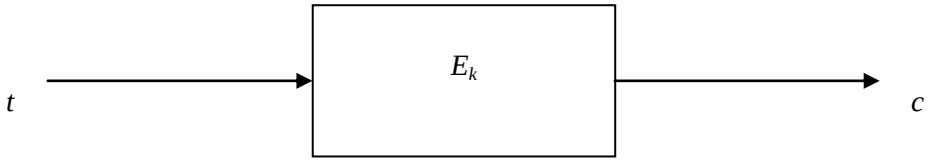
1.8. Apibrėžimas. Landine vienkrypte funkcija (angl. *trap-door one-way function*) vadinama tokia VKF, kuriai duotas tam tikras landinis parametras k, leidžiantis nesunkiai apskaičiuoti atvirkštinę funkciją.

Turėdami šį apibrėžimą galvoje, galime performuluoti svarbiausią užšifravimo funkcijos savybę.

1.9. Savybė. Šifravimo funkcija E_k privalo būti stipri landinė VKF, kurios landinis parametras k leidžia lengvai apskaičiuoti t, kai duota c ir k.

Iki šiol daug kartų vartojome sąvokas „sunku apskaičiuoti“ ir „lengva apskaičiuoti“, tačiau jų neapibrėžėme. Į klausimą, kokios problemos lengvai ar sunkiai sprendžiamos pasitelkus algoritmus, atsako matematikos šaka, vadinama sudėtingumo teorija (angl. *complexity theory*).

Kaip žinome, matematiniai skaičiavimai atliekami naudojant kokį nors algoritmą. Kiekvienas algoritmas turi išeities duomenis, iš kurių po tam tikrų veiksmų gauna rezultatą. Šifravimo funkcija E taip pat realizuojama tam tikru algoritmu, kuris, remdamasis formulių ir veiksmų rinkiniu, iš tekstogramos t gauna šifrogramą c. Simboliškai tai galima pavaizduoti 1.1 pav.



1.1 pav. Šifravimo algoritmo struktūrinė schema

Natūralu tikėtis, kad kuo didesnė tekstograma t, tuo daugiau skaičiavimo veiksmų atliks algoritmas, skaičiuodamas šifrogramą c. Kuo daugiau skaičiavimų bus atliekama, tuo daugiau laiko bus vykdomas algoritmas. Vadinasi, galime įvertinti algoritmo skaičiavimo laiką kokia nors funkcija, priklausančia nuo tekstogramos ilgio $|t|$. Kadangi kompiuteriuose informacija koduojama bitais, sutarsime tekstogramos ilgį $|t|$ taip pat matuoti bitais.

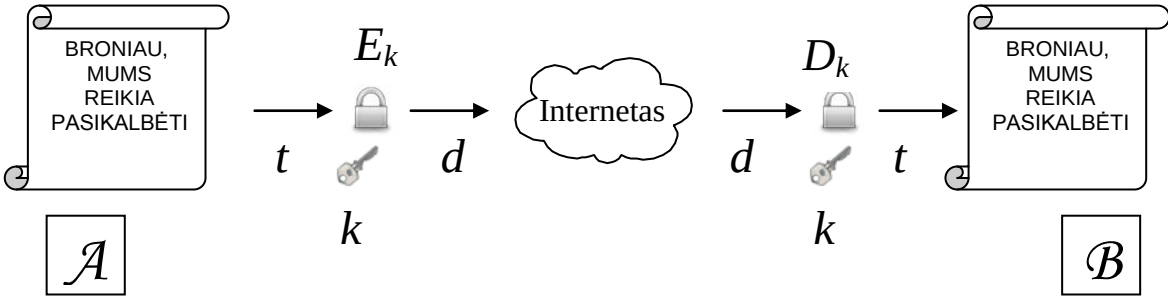
Mums rūpi klausimas: kokia sparta didėja skaičiavimo laikas ilgėjant tekstogramai? Kokia funkcija sieja tekstogramos ilgį su skaičiavimo laiku?

> ***1.10. Apibrėžimas.*** Uždavinys vadinamas lengvai išsprendžiamu, jei galima rasti tokį daugianarį $p(|t|)$, kurio visų $|t|$ ilgio reikšmių skaičiavimo laikas $\tau \le p(|t|)$. Jei tokio daugianario rasti neįmanoma, uždavinys vadinamas sunkiai išsprendžiamu.

Uždaviniai, priklausantys šiai klasei, sprendžiami polinominio laiko algoritmu, todėl ši klasė vadinama P klase.

Uždavinių sudėtingumą išsamiau aptarsime trečiajame skyriuje.

Iki šiol nagrinėjome pavienius duomenų šifravimo algoritmus, tačiau nereikėtų pamiršti, jog viskas vyksta kompiuterių tinkluose, todėl reikia nagrinėti ne atskirus algoritmus, o protokolus. Paprasčiausią kriptografinį protokolą informacijai užšifruoti ir iššifruoti matome 1.2 paveiksle.



1.2 pav. Paprasčiausio šifravimo protokolo pavyzdys

Atsižvelgus į kriptoprotokolų darbo atvirame tinkle (internete) specifiką, atsiranda galimybė surengti prieš juos kriptografines atakas. Paminėsime dvi svarbiausias:

- **Apsimetimo ataka (angl. *impersonation attack*).** Šios atakos metu kenkėja *Zosė* apsimeta teisėta protokolo vartotoja *Aldona* ir atlieka veiksmus jos vardu. Kriptoprotokolas privalo užtikrinti galimybę *Broniui* įsitikinti, kad jis tikrai bendrauja su tikrąja *Aldona*, o ne su apsimetėle.
- **Žodyno ataka (angl. *dictionary attack*).** Ši ataka nukreipta prieš kriptoprotokolus, pagrįstus slaptažodžiais. Jeigu slaptažodis gana paprastas, perrinkus tam tikrus derinius jį galima atspėti ir įsilaužti į kriptosistemą.

Naudojantis bendruoju slaptuoju raktu k, galima ne tik šifruoti duomenis, bet ir atlikti kitas kriptografines funkcijas, pvz., skaičiuoti santraukos funkcijas, kurti elektroninio parašo schemas, generuoti pseudoatsitiktinius skaičius ir t. t. Tuo remdamiesi galime apibrėžti simetrinę kriptosistemą.

> ***1.11. Apibrėžimas.*** Simetrine kriptosistema vadinama tokia sistema, kuri visoms kriptografinėms funkcijoms atlikti naudoja vieną slaptąjį raktą k, žinomą abiem (arba keliems) protokolo dalyviams.

Nors simetrinės kriptosistemos turi nemažai privalumų, tačiau turi ir vieną esminį trūkumą. Tarkime, kad kriptosistemoje yra N kriptosubjektų. Norint išsaugoti kriptosubjektų tarpusavio konfidencialumą, jiems reikia paskirstyti $C^2_N = n(n-1)/2$ raktų porų. Kiekvienas jų turi akylai saugoti N-1 raktą ryšiams su kitais kriptosubjektais. Nesunku įsitikinti, kad, didėjant kriptosubjektų skaičiui, reikalingų raktų porų skaičius didėja kvadratine priklausomybe (pavaizduota 1.1 lentelėje).

1.1 lentelė

Raktų porų skaičiaus didėjimas augant kriptosubjektų skaičiui simetrinėje kriptosistemoje

N	Raktų skaičius
10	45
100	4 950
1 000	499 500
10 000	49 995 000
100 000	4 999 950 000

Matome, kad net ir esant nedidelėms N reikšmėms, generuoti, paskirstyti ir administruoti raktus yra labai sudėtinga organizaciniu atžvilgiu.

1976 metais Whitfieldas Diffie'is ir Martinas Hellmanas išspausdino straipsnį „Naujos kriptografijos kryptys“ (angl. *New directions in cryptography*), sukėlusį lūžį kriptografijos teorijos ir jos taikymo srityje. Šiame straipsnyje pateiktos naujos idėjos leidžia mums pradėti realizuoti elektroninio verslo ir elektroninės valdžios sistemas, nes iš jų išsirutuliojo nauja kriptografijos atmaina – asimetrinė kriptografija.

> ***1.12. Apibrėžimas.*** Asimetrine kriptosistema vadinama tokia sistema, kurioje kriptografinės funkcijos realizuojamos naudojant matematiškai susijusių raktų porą: privatųjį raktą – *PR* (angl. *private key*), žinomą tik jo savininkui, ir su juo matematiškai susijusį viešąjį raktą – *VR* (angl. *public key*), žinomą visiems vartotojams tinkle. Ši raktų pora vadinama asimetriniais raktais.

Asimetrinės kriptosistemos neretai dar vadinamos viešojo rakto kriptosistemomis – VRK (angl. *public key cryptosystem*), o sujungtos su visomis techninėmis ir organizacinėmis priemonėmis joms įgyvendinti – viešojo rakto infrastruktūromis – VRI (angl. *public key infrastructure*).

VRK leidžia spręsti asimetrinio šifravimo, elektroninio parašo, raktų keitimosi ir kitus labai svarbius šiuolaikinei informacinei visuomenei uždavinius. Jau šiuo metu VRI pagrindu kai kuriose valstybėse (tarp jų ir Estijoje) naudojamos ir veikia elektroninio balsavimo sistemos. 2007 metais *EuroCrypt* konferencijoje buvo paskelbtas mokslinis straipsnis, kuriame siūloma praktiškai įgyvendinama ir pakankamai patogi naudoti elektroninių pinigų (e. pinigų) sistema. Tikėtina, kad netrukus būsime liudytojai įspūdingų informacinės visuomenės laimėjimų, iš esmės pakeisiančių mūsų įprastą buitį.

Kol kas apžvelkime VRK taikymo būdus duomenims šifruoti ir elektroniniam parašui realizuoti. Tarkime, kad $\mathcal{A}$*ldona* turi asimetrinių raktų porą VR_A, PR_A, o $\mathcal{B}$*ronius* – atitinkamai VR_B ir PR_B. Iš apibrėžimo seka, kad PR_A ir PR_B yra privatieji raktai, žinomi tik jų savininkams, tuo tarpu VR_A ir VR_B yra viešieji raktai, kuriuos žino visi dalyviai.

Sakykime, kad $\mathcal{B}$, gavęs žinutę iš $\mathcal{A}$, nori jai išsiųsti atsakymą, pasitelkęs asimetrinio šifravimo algoritmą. Jis turi užšifruoti savo siunčiamą informaciją taip, kad ji būtų pasiekiama tik $\mathcal{A}$. Kadangi $\mathcal{B}$ žino jos viešąjį raktą, jis turi galimybę užšifruoti žinutę taip:

$$E(VR_A, t) = E(VR_A) \circ t = c. \qquad (1.5)$$

Čia operacija $\circ$ simboliškai pažymėjome funkcijos poveikį argumentui, nes tai yra labiau simetrinis žymėjimas, kurio mums dar prireiks.

Kadangi iššifruoti atsakymą galės tik $\mathcal{A}$, iššifravimo funkcija D turi priklausyti tik nuo $\mathcal{A}$ žinomo PR_A:

$$D(PR_A, c) = D(PR_A) \circ c = t. \qquad (1.6)$$

Tam, kad užšifravimas ir iššifravimas būtų sėkmingi, turi būti tenkinama ši sąlyga:

$$D(PR_A) \circ c = D(PR_A) \circ E(VR_A) \circ t = t. \quad (1.7)^*$$

Tada iš karto kyla klausimas: kokios sąlygos turi būti tenkinamos, kad 1.7 lygybė galiotų?

1.13. Savybė. *PR* ir *VR* turi būti tarpusavyje susieti bijekcine funkcine priklausomybe:

$$VR_A = \varphi(PR_A). \quad (1.8)$$

Ši sąlyga yra būtina, tačiau toli gražu nepakankama, nes tam tikrus reikalavimus turi tenkinti ir funkcijos *D* bei *E*.

1.14. Savybė. Užšifravimo ir iššifravimo funkcijos turi būti viena kitai atvirkštinės, t. y.

$$E(VR_A) \circ D(PR_A) = I; \quad (1.9)$$

čia *I* – tapačioji funkcija.

Ši, kaip ir kitos šifravimo funkcijų savybės, yra analogiška jau aptartoms simetrinio šifravimo funkcijų savybėms, todėl jos nekartosime.

> ***1.15. Klausimas.*** Kaip apsaugoti tekstogramos konfidencialumą, jei iš 1.8 formulės išvedama formulė, pagal kurią galima apskaičiuoti PR_A?
>
> $$PR_A = \varphi^{-1}(VR_A) \quad (1.10)$$

Akivaizdu, jog nustatyti privatųjį raktą iš viešojo turi būti kuo sudėtingiau. Čia į pagalbą ateina paskaitos pradžioje jau aptartos VKF.

1.16. Savybė. Funkcija φ, siejanti VR_A su PR_A, privalo būti vienkryptė.

Jei ši savybė tenkinama, apskaičiuoti PR_A iš VR_A neįmanoma, todėl užtikrinamas tekstogramos konfidencialumas.

> ***1.17. Klausimas.*** Kokią prasmę gali turėti kitokio tipo užšifravimai ir iššifravimai, naudojant skirtingas PR_A, VR_A, PR_B ir VR_B aibės raktų poras? Kiek tokių derinių gali būti? Kurie jų beprasmiai, šifravimą atliekant *Broniui*?

Panagrinėkime vieną iš kitokių šifravimo galimybių. Tarkime, kad *Bronius*, užuot pasinaudojęs viešuoju *Aldonos* raktu, šifruoja kokią nors tekstogramą *t* taip:

$$E(PR_B) \circ t = s. \quad (1.11)$$

Kadangi viešąjį *Broniaus* raktą žino visi vartotojai, iššifruoti duomenis *s* (mes jų dabar nevadiname šifrograma) gali visi, pasinaudoję tapatybe:

$$D(VR_B) \circ s = D(VR_B) \circ E(PR_B) \circ t = t. \quad (1.12)$$

> ***1.18. Klausimas.*** Kur galima pritaikyti tokį šifravimo būdą?

Kadangi *Bronius* yra vienintelis vartotojas, žinantis PR_B, akivaizdu, jog jis yra vienintelis, galintis sukurti tokius duomenis *s*, kad juos būtų galima iššifruoti pasitelkus VR_B. Be to, kadangi VR_B yra pasiekiamas kiekvienam, bet kas gali iššifruoti tuos duomenis ir įsitikinti, kad juos užšifravo *Bronius*. Vadinasi, jis gali laiško *Aldonai* gale prikabinti pranešimą „Šį laišką tikrai parašiau aš, *Bronius*“ ir jį užšifruoti privačiuoju raktu. *Aldona*, gavusi laišką, iššifruoja šį „pranešimą“ *Broniaus* viešuoju raktu ir įsitikina, kad laiško autorius tikrai yra jis. Pastebėkime, kad 1.11 formulėje pateiktas šifravimas turi įprastam parašui būdingų savybių.

> ***1.19. Apibrėžimas.*** Elektroniniu parašu (e. parašu) vadinamas asimetrinis šifravimas, kai duomenys užšifruojami privačiuoju subjekto raktu, o iššifruojami – viešuoju.

Abstrakčiai panagrinėkime, kaip naudojantis VKF galima sudaryti *VR* ir *PR* raktų porą bei suformuoti elektroninį parašą.

---

*Atkreipkite dėmesį į tai, kad šioje formulėje dirbtinai praplėtėme ◦ operaciją, pritaikę ją ir funkcijų kompozicijai. Nors dėl vietos stokos nepateikiame formalaus tokios jos interpretacijos apibrėžimo, tačiau ji leis greičiau ir geriau suprasti tolesnę medžiagą.

Nagrinėkime stiprią VKF $y = f(x)$, arba ją užrašytą mums patogesnę forma $y = f \circ x$. Prisiminkime, kad stipri VKF yra tokia VKF, kai žinant x ir y algoritmiškai sunku apskaičiuoti f, o žinant y ir f algoritmiškai sunku rasti x. Tokiu atveju asimetrinių raktų porą galime apibūdinti taip:

$$PR = (f),\ VR = (x, y). \tag{1.13}$$

Kadangi funkcija f yra VKF, žinant porą (x, y) neįmanoma (angl. *infeasible*) rasti funkcijos f, nes tai – algoritmiškai sudėtingas uždavinys.

Įveskime dar vieną funkciją h ir pareikalaukime, kad galėtume apskaičiuoti bet kurių funkcijų h ir f iš tam tikros funkcijų klasės kompoziciją $g = h \circ f$. Taip pat pareikalaukime, kad dviejų funkcijų kompoziciją būtų galima pritaikyti ir argumentui x:

$$(h \circ f)(x) = (h \circ f) \circ x = g \circ x. \tag{1.14}$$

1.20. Apibrėžimas. Aibėje $\{h, f, x, \ldots\}$ apibrėžta kompozicijos operacija $\circ$ vadinama asociatyvia, jeigu bet kuriems jos elementams galioja sąryšis:

$$(h \circ f) \circ x = h \circ (f \circ x). \tag{1.15}$$

Tarkime, kad mūsų nagrinėjamoje funkcijų aibėje kompozicijos operacija iš tikrųjų yra asociatyvi. Tuo remdamiesi suformuluokime e. parašo algoritmą, pagrįstą asimetriniu šifravimu subjekto privačiuoju raktu.

Patogumo dėlei formuojamas ne visos tekstogramos, o tam tikros fiksuoto ilgio jos santraukos e. parašas. Sakykime, kad santraukos funkcija yra H (ji dažniausiai ir vadinama H funkcija (angl. *hash function*)). Iš tekstogramos t ji apskaičiuoja santraukos reikšmę h:

$$H(t) = h. \tag{1.16}$$

Tarkime, kad ši santrauka h priklauso 1.20 apibrėžime minimai funkcijų aibei.

Įsivaizduokime, kad $\mathcal{A}$ turi viešąjį raktą $VR_A = (x, y)$ ir privatųjį $PR = (f)$. Ji pasiruošė tekstogramą t, nori ją pasirašyti ir išsiųsti $\mathcal{B}$. Tam $\mathcal{A}$ atlieka šiuos veiksmus:

1. Apskaičiuoja $h = H(t)$.
2. Suformuoja e. parašą

$$s = h \circ f. \tag{1.17}$$

3. Išsiunčia *Broniui* pranešimą (t, s).

Laiškai internete keliauja nesaugiais kanalais, todėl, tarkime, kad $\mathcal{B}$ gavo nebūtinai tą patį pranešimą, kurį pažymėsime (t_B, s_B). Jis perskaito t_B ir nori įsitikinti, kad jo autorius tikrai buvo $\mathcal{A}$. Tam $\mathcal{B}$ turi atlikti šiuos veiksmus:

1. Apskaičiuoti $h_B = H(t_B)$.
2. Patikrinti tapatybę

$$s_B \circ x = h_B \circ y. \tag{1.18}$$

Iš tiesų, jei parašas nesuklastotas ir tekstograma nepakeista, tai $h_B = h$ ir $s_B = s$. Tada

$$s_B \circ x = h_B \circ y \equiv s \circ x = h \circ y \equiv (h \circ f) \circ x = h \circ (f \circ x). \tag{1.19}$$

Kadangi sutarėme, kad mūsų nagrinėjamoje funkcijų aibėje kompozicijos operacija yra asociatyvi, paskutinė tapatybė yra teisinga, o parašas – autentiškas. Lygiai taip pat nesunku įsitikinti, kad jei tekstograma pakeista ($h_B \neq h$) ar parašas suklastotas ($s_B \neq s$), 1.19 tapatybė nebegalioja ir *Bronius* įtaria, kad gautas laiškas yra kokios nors *Zosės* kūrinys.

Mūsų pasiūlytame modelyje *Aldona* iš tiesų parašui naudoja tik savo privatųjį raktą (f), o *Bronius* tikrindamas – *Aldonos* viešąjį raktą (x, y), t. y. 1.17 formulė yra 1.11 atitikmuo, o 1.18 formulė – 1.12. Tai reiškia, kad pasirašymo algoritmas yra asimetrinis šifravimas minėtais raktais.

2. ARITMETIKA IR ALGEBRA

Norint geriau suprasti kriptografiją, reikia susipažinti su daugeliu mokslo sričių ir jų laimėjimais, ypač matematikos, fizikos, ryšio teorijos ir informatikos.

Šiam tikslui būtini matematikos pagrindų elementai: skaičių, grupių, žiedų, laukų teorijos medžiaga pateikiama antrajame skyriuje „Aritmetika ir algebra“.

Suvokus išdėstytus pradmenis ir pasitelkus dideles kompiuterinio skaičiavimo galimybes, galima mėginti kurti ir analizuoti vis plačiau diegiamas informacinės saugos sistemas.

2.1. Teorema. (Dalyba su liekana). Tarkime, kad $a \in \mathbf{Z}$, $b \in \mathbf{N}$. Tokiu atveju egzistuoja tokie vienareikšmiškai apibrėžti skaičiai q, $r \in \mathbf{Z}$, $0 \le r < b$, kad $a = bq + r$.

▲ Pasirinkime didžiausią sandaugą bq, neviršijančią a. Tegu $r = a - bq$. Iš sąlygos $bq \le a < b(q + 1)$ gauname $r < b$. Patikrinsime, ar šie skaičiai vienareikšmiškai apibrėžti. Tarkime, kad $a = bq_1 + r_1$, $0 \le r_1 < b$; tuomet gauname $0 = b(q - q_1) + (r - r_1)$, t. y. $r - r_1$ yra b kartotinis. Kadangi $|r - r_1| < b$, tai turi būti tik tuomet, kai $r - r_1 = 0$. Kitaip tariant, iš $r = r_1$ sąlygos gauname, kad ir $q = q_1$.▼

Teorema teisinga ir bet kuriam sveikajam skaičiui $b \neq 0$, jei aprėžtumas $r < b$ keičiamas į $r < |b|$.

2.2. Apibrėžimas. Bet kuris sveikasis skaičius, dalijantis a ir b, vadinamas jų bendruoju dalikliu. Didžiausias iš bendrųjų a ir b skaičių daliklių vadinamas skaičių a ir b didžiausiu bendruoju dalikliu; žymima DBD(a,b) (angl. *greatest common divider*) arba (a,b).

2.3. Apibrėžimas. Jeigu DBD(a,b) = 1, tai skaičiai vadinami reliatyviai pirminiais.

Faktą, kad skaičius b dalija skaičių a, žymėsime $b|a$. Tada DBD galima užrašyti ir taip:

$$\mathrm{DBD}(a,b) = \max\{k;\ k|a \text{ ir } k|b\}.$$

Reikalaudami, kad DBD būtų teigiamas, turime DBD(a,b) = DBD(a,–b) = DBD(–a,b) = DBD(–a,–b), arba, paprasčiau tariant, DBD(a,b) = DBD(|a|,|b|).

2.4. Teiginys. Sveikieji skaičiai a ir b yra reliatyviai pirminiai tada ir tik tada, kai $au + bv = 1$.

▲ Būtinumas išplaukia remiantis Euklido algoritmo, t. y. jei DBD(a, b) = d, tai $d = au + bv$.

Pakankamumas gaunamas iš dalumo savybės:

jeigu $(a, b) \mid a$ ir $(a, b) \mid b \Rightarrow (a, b) \mid au + bv = 1 \Rightarrow (a, b) = 1$. ▼

Natūralieji skaičiai $p > 1$, turintys tik trivialius daliklius 1 ir p, vadinami pirminiais. Kitu atveju skaičiai vadinami sudėtiniais.

2.5. Teorema. (Euklido) Egzistuoja be galo daug pirminių skaičių.

▲ Tarkime priešingai ir pažymėkime $p_1, p_2, \ldots, p_n$ visus skirtingus pirminius skaičius. Pirminio skaičiaus $p_1p_2p_3 \ldots p_n + 1$ daliklis negali sutapti nė su vienu iš skaičių $p_1, p_2, \ldots, p_n$. Vadinasi, jis dalijasi tik iš 1 ir iš savęs, t. y. jis yra pirminis. Gavome prieštaravimą. ▼

Paminėsime dar keletą pirminio skaičiaus p savybių.

2.6. Savybė. $(p, a) \neq 1, \Rightarrow p \mid a$.

▲ Kadangi DBD (p, a) dalija p, jis lygus arba 1, arba p. Taigi gauname $(p, a) = p \Rightarrow p \mid a$.▼

2.7. Savybė. $p \mid ab \Rightarrow p \mid a$ arba $p \mid b$.

▲ Jeigu turėtume $(p, a) = (p, b) = 1$, tuomet gautume, kad ir $(p, ab) = 1$. Todėl bent vienas daliklis dalijasi iš p. ▼

Savybė lengvai išplečiama kelių skaičių $a, b, c,\ldots$ atveju.

2.8. Teorema. Bet kuris sveikasis skaičius a, didesnis už 1, užrašomas pirminių skaičių, įskaitant jų laipsnį, sandauga: $a = p_1^{k_1} p_2^{k_2} \dots p_s^{k_s}$; čia p_i — pirminiai skaičiai, $k_i \in \mathbf{N}$. Šis užrašas yra vienintelis, jeigu nekreiptume dėmesio į daugiklių išdėstymo tvarką.

▲ Jeigu skaičius $a > 1$ nėra pirminis, tai $a = p_1 a_1$; čia p_1 — mažiausias pirminis daliklis. Jeigu a_1 nėra pirminis, tuomet $a_1 = p_2 a_2$.

Tęsdami samprotavimą, gausime, kad $a_n = 1$. Taigi, skaičių a galima išskleisti pirminių skaičių sandauga

$$a = p_1 p_2 \dots p_n.$$

Sakykime, kad galima ir kitokia sandauga, pvz., $a = q_1 q_2 q_3 \dots q_n$. Tuomet $p_1 p_2 \dots p_n = q_1 q_2 \dots q_n$. Tačiau pagal 2.7 savybę p_1 dalija bent vieną skaičių dešinėje pusėje.

Tarkime $p_1 \mid q_1$, tai reiškia, kad $p_1 = q_1$. Suprastinę abi puses iš p_1, gauname $p_2 p_3 \dots p_n = q_2 q_3 \dots q_n$.

Analogiškai tęsdami samprotavimus, gausime dviejų užrašų tapatumą.▼

2.9. Lyginiai. Tarkime, kad a ir b yra sveikieji skaičiai, m — natūralusis skaičius.

Jeigu sveikųjų skaičių a ir b skirtumas $a - b$ dalijasi iš m, sakome, kad a lygsta b moduliu m, ir rašome

$$a \equiv b (\mathrm{mod}\ m) \Leftrightarrow m \mid a - b.$$

Šį reiškinį vadinsime lyginiu (kongruencija), pvz.,

$$30 \equiv 4 (\mathrm{mod}\ 13);$$
$$15 \equiv -43 (\mathrm{mod}\ 29).$$

Atvejis $m = 1$ – trivialus, todėl toliau laikysime, kad $m > 1$.

Lyginį $a \equiv b (\mathrm{mod}\ m)$ galime keisti ekvivalenčia lygybe $a = b + tm$, $t \in \mathbf{Z}$. Iš reiškinio $b \pmod m = c$ gauname, kad c yra liekana dalijant b iš m.

2.10. Teorema. Du sveikieji skaičiai lygsta vienas kitam moduliu m tada ir tik tada, kai dalijant juos iš m gaunama ta pati liekana.

▲ Remdamiesi dalyba su liekana, rašome lygybes:

$$a = mq + r,\ 0 \le r < m,$$
$$b = mq' + r',\ 0 \le r' < m.$$

Pakankamumas akivaizdus: jeigu $r = r'$, tai

$$a - b = m(q - q'),$$

todėl $a \equiv b (\mathrm{mod}\ m)$.

Būtinumas. Tarkim, $r > r'$, tada

$$a - b = m(q - q') + (r - r').$$

Jeigu $a \equiv b (\mathrm{mod}\ m)$, tai skirtumas $a - b$ dalijasi iš m. Tuomet skirtumas $r - r'$ taip pat turi dalytis iš m, t. y. $r - r' = ms$. Tačiau liekanos r ir r' yra mažesnės už m, o $r - r' \ge 0$. Todėl lygybė $r - r' = ms$ galima tik tada, kai $s = 0$, kitaip tariant, $r = r'$.▼

2.11. Liekanų klasės. Sąryšis „lygsta moduliu m“ $a \equiv b (\mathrm{mod}\ m)$ yra ekvivalentumo sąryšis sveikųjų skaičių aibėje $\mathbf{Z}$, todėl suskaido ją į nesikertančias ekvivalentumo klases $\bar{a}$; $a \equiv b (\mathrm{mod}\ m) \Leftrightarrow \bar{a} = \bar{b}$.

2.12. Apibrėžimas. Visų elementų x, ekvivalenčių duotajam x, poaibis $\bar{x} = \{x' \in X \mid x' \sim x\} \subset X$ vadinamas ***ekvivalentumo*** klase, kuriai atstovauja elementas x. Čia $\sim$ yra binarinis ekvivalentumo sąryšis aibėje X.

2.13. Teorema. Ekvivalentumo klasės arba sutampa arba nesikerta, t.y. aibė $X = \bigcup_{x \in X} \bar{x}$ yra nesikertančių klasių junginys.

▲ Kadangi $x \in \bar{x}$, tai $X = \bigcup_{x \in X} \bar{x}$. Kiekviena klasė $\bar{x}$ bet kuriuo savo atstovu x nusakoma vienareikšmiškai, t. y. $\bar{x} = \bar{x}' \Leftrightarrow x \sim x'$.

Jeigu $x \sim x'$ ir $x'' \in \bar{x} \Rightarrow x'' \sim x' \Rightarrow x'' \in \bar{x}' \Rightarrow \bar{x} \subset \bar{x}'$. Bet $x \sim x' \Rightarrow x' \sim x$. Todėl teisingas ir atvirkščias teiginys, t. y. $\bar{x}' \subset \bar{x}$. Reiškia $\bar{x}' = \bar{x}$. Antra vertus, jeigu $x \in \bar{x}$, tai $\bar{x}' = \bar{x} \Rightarrow x \in \bar{x}' \Rightarrow x \sim x'$.

Jeigu $\bar{x}' \cap \bar{x} \neq \emptyset$ ir $x \in \bar{x}' \cap \bar{x}''$, tai $x \sim x'$ ir $x \sim x''$, bet tuomet $x' \sim x''$ ir $\bar{x}' = \bar{x}''$. Reiškia skirtingos klasės nesikerta. ▼

Ekvivalentumo klasei moduliu m aibėje **Z** priklauso visi sveikieji skaičiai, kurie lygsta vienas kitam moduliu m, t. y. dalijant juos iš m, gaunama ta pati liekana. Iš viso gauname m liekanų klasių moduliu m: $\bar{0}, \bar{1}, \bar{2}, \ldots, \overline{m-1}$. Kiekvieną klasę reprezentuoja bet kuris jos atstovas.

Liekanų klasę $\bar{a}$ moduliu m sudaro sveikieji skaičiai: $\bar{a} = \{a + mt \mid t \in \mathbf{Z}\}$. Ši liekanų klasių aibė dažnai žymima $\mathbf{Z}_m$. Kitaip tariant, visų sveikųjų skaičių aibė **Z** atvaizduojama į baigtinę aibę $\{\bar{0}, \bar{1}, \bar{2}, \ldots, \overline{m-1}\}$, t. y. siurjekcija $\mathbf{Z} \to \mathbf{Z}_m$ (Кострикин, 1977).

Liekanų aibių klasėje galima atlikti šiuos aritmetikos veiksmus:

1^0. $(a(\text{mod } m) + b(\text{mod } m))\text{mod } m = (a + b)\text{mod } m$.

2^0. $(a(\text{mod } m) - b(\text{mod } m))\text{mod } m = (a - b)\text{mod } m$.

3^0. $(a(\text{mod } m) \cdot b(\text{mod } m))\text{mod } m = (a \cdot b)\text{mod } m$.

2.14. Klausimas. Pabandykite šias savybes įrodyti savarankiškai.

2.15. Apibrėžimas. Rinkinys skaičių, paimtų po vieną iš kiekvienos liekanų klasės moduliu m, vadinamas pilnąja liekanų sistema tuo moduliu; žymima p. l. s.

Pilnoji liekanų sistema moduliu 5. Klases sudaro aibės:

$$\bar{0} = \{0 + 5k \mid k \in Z\};$$
$$\bar{1} = \{1 + 5k \mid k \in Z\};$$
$$\bar{2} = \{2 + 5k \mid k \in Z\};$$
$$\bar{3} = \{3 + 5k \mid k \in Z\};$$
$$\bar{4} = \{4 + 5k \mid k \in Z\}.$$

Turime 0, –4, –8, 3, 9; čia $0 \in \bar{0}, -4 \in \bar{1}, -8 \in \bar{2}, 3 \in \bar{3}, 9 \in \bar{4}$. Panašiai gauname rinkinius 0, 1, 2, 3, 4 ir 0, 1, 2, –2, –1, t. y. p. l. s. moduliu 5.

2.16. Teiginys. (p. l. s. sudarymo kriterijus). Kad sveikieji skaičiai sudarytų pilnąją liekanų sistemą moduliu m, būtina ir pakankama, kad jų būtų m ir bet kurie du jų nelygtų vienas kitam moduliu m.

Toliau aptarsime pirmosios eilės lyginių sprendimą.

Tarkime, turime lyginį $ax \equiv b(\text{mod } m)$ ir sąlygą $(a, m) = 1$.

2.17. Apibrėžimas. Lyginio sprendiniu vadiname liekanų klasę moduliu m, kurios vienas elementas (vadinasi, ir visi) tenkina pateiktą lyginį. Kadangi $(a, m) = 1$, rasime tokius sveikuosius skaičius u ir v, kai $1 = au + mv$, t. y. $au \equiv 1(\text{mod } m)$. Iš čia lengvai randame

$$x \equiv bu(\text{mod } m),$$

t. y. turime vieną lyginio sprendinį moduliu m.

Tarkime, kad $(a, m) = d > 1$. Tuomet sąlyga $d \mid b$ yra būtina sąlyga norint išspręsti lyginį $ax \equiv b(\text{mod } m)$. Laikykime, kad ši sąlyga yra išpildyta. Tada $a = a_1 d$, $b = b_1 d$, $m = m_1 d$. Gauname ekvivalentų lyginį $a_1 x \equiv b_1(\text{mod } m_1)$, kuris turi vieną sprendinį $x \equiv x_1 \ (\text{mod } m_1)$. Moduliu m gauname d sprendinių:

$$x_1, x_2 + m_1, x_1 + 2m_1, ..., x_1 + (d-1)m_1.$$

2.18. Teorema. Tarkime, $(a, m) = d$. Tuomet lyginys $ax \equiv b(\text{mod } m)$ negalimas, kai b nesidalija iš d. Jeigu b yra d kartotinis, tuomet lyginys $ax \equiv b(\text{mod } m)$ turi d sprendinių.

▲Kitais žodžiais teisinga tokia teorema: tegul $(a, m) = d$. Lyginys $ax \equiv b(\text{mod } m)$ išsprendžiamas tada ir tik tada, kai $a \mid b$. Tuo atveju turime d sprendinių.

Imkime lyginių sistemą:

$$\begin{aligned} &a_1 x \equiv b_1(\text{mod } m_1); \\ &a_2 x \equiv b_2(\text{mod } m_{21}); \\ &\ldots\ldots\ldots\ldots \\ &a_n x \equiv b_n(\text{mod } m_n). \end{aligned} \qquad (2.1)$$

Galima įrodyti, kad ši sistema suvedama į tokio pavidalo sistemą:

$$\begin{aligned} &x \equiv b_1(\text{mod } m_1); \\ &x \equiv b_2(\text{mod } m_2); \\ &\ldots\ldots\ldots\ldots \\ &x \equiv b_n(\text{mod } m_n). \end{aligned}$$

Šiai sistemai išspręsti reikia mokėti spręsti sistemą

$$\begin{aligned} &x \equiv b_1(\text{mod } m_1), \\ &x \equiv b_2(\text{mod } m_2). \end{aligned} \qquad (2.2)$$

Duotą sistemą spręsime keitimo būdu.

Iš pirmojo lyginio randame $x = b_1 + m_1 t$. Gautą rezultatą įrašome į antrąjį ir gauname $m_1 t = b_2 - b_1(\text{mod } m_2)$. Išsprendžiamumo kriterijus (2.2) sistemai yra $(m_1, m_2) \mid b_2 - b_1$. Tuomet turime vieną sprendinį moduliui $\frac{m_2}{(m_1, m_2)}$, t. y. $t \equiv t_0 \left(\text{mod } \frac{m_2}{(m_1, m_2)} \right)$. Todėl

$$x = b_1 + m_1 \left(t_0 + \frac{m_2}{(m_1, m_2)} s \right) = b_0 + \frac{m_1 m_2}{(m_1, m_2)} t = b_0 + [m_1, m_2] t ,$$

ir (2.2) sistema turi vienintelį sprendinį, jeigu ji išsprendžiama moduliu $[m_1, m_2] = \frac{m_1 m_2}{(m_1, m_2)}$. (2.1) sistema, jeigu ji išsprendžiama, turi vienintelį sprendinį moduliu $[m_1, m_2, ..., m_n]$ (Bulota, et al., 1990).

Tuo atveju, kai visi moduliai $m_1, m_2, ..., m_n$ po du yra reliatyviai pirminiai, (2.1) sistemai taikomas taip vadinamas kinų metodas[1]. Apibrėžkime skaičius $x_1, x_2, ..., x_n$ sąlyga $m_2 \cdot m_3 \ldots m_n x_1 \equiv 1(\text{mod } m_1)$, $m_1 m_3 \ldots m_n x_2 \equiv 1(\text{mod } m_2)$, ..., $m_1 m_2 \ldots m_{n-1} x_n \equiv 1(\text{mod } m_n)$. Tuomet (2.1) sistemos sprendinys yra skaičius

$$x = m_1 m_3 \ldots m_n x_1 b_1 + m_1 m_3 ..., m_n x_2 b_2 + \ldots + m_1 m_2 \ldots m_{n-1} x_n b_n \text{ mod } (m_1 m_2 \ldots m_n).$$

Kitaip tariant, teisinga tokia kinų teorema (Goldwasser, et al., 2001):

Lyginių (2.1) sistema, kai moduliai po du yra reliatyviai pirminiai, turi vienintelį sprendinį moduliu $m = m_1 m_2 \ldots m_n$.▼

2.19. Pavyzdys. Išspręsime sistemą
$x \equiv 2(\text{mod } 3)$,

[1] Manoma, kad šią teoremą įrodė kinų matematikas Sun-Cze apie 100 m. p. m. e..

$x \equiv 3 \pmod 5$,
$x \equiv 2 \pmod 7$.

▲Remdamiesi kinų teorema, turime, kad sistemos sprendinys yra

$x = m_2m_3x_1b_1 + m_1m_3x_2b_2 + m_1m_2x_3b_3$

moduliu $m = m_1m_2 \dots m_n$.

Šiuo konkrečiu atveju imame $m_1 = 3$, $m_2 = 5$, $m_3 = 7$ ir $b_1 = 2$, $b_2 = 3$, $b_3 = 2$. Randame skaičius x_1, x_2, x_3 iš lyginių :

$m_2m_3x_1 \equiv 1 \pmod 3$,
$m_1m_3x_2 \equiv 1 \pmod 5$,
$m_1m_2x_3 \equiv 1 \pmod 7$.

Turime, kad $m_2m_3x_1 \equiv 1 \pmod 3$; $35x_1 \equiv 1 \pmod 3$. Todėl, remdamiesi lyginių savybėmis, gauname, kad $35x_1 \equiv (1 + 3 \cdot 23) \pmod 3$, $35x_2 \equiv 70 \pmod 3$ ir $x_1 \equiv 2 \pmod 3$,

$m_1m_3x_2 \equiv 1 \pmod 5$, $21x_2 \equiv 1 \pmod 5 \equiv (1 + 4 \cdot 5) \bmod 5$ ir $x_2 \equiv 1 \pmod 5$, nes $(21, m_2) = 1$.

$m_1m_2x_3 \equiv 1 \pmod 7$, $25x_3 \equiv 1 \pmod 7 \equiv (1 + 2 \cdot 7) \bmod 7$ ir $x_3 \equiv 1 \pmod 7$, nes $(m_3, 15) = 1$. ▼

2.20. Apibrėžimas. Oilerio funkcija $\varphi(m)$ yra natūraliųjų skaičių, mažesnių už m ir tarpusavyje pirminių su m, skaičius, t. y.

$$\varphi(m) = |\{i \mid 1 \le i < m, (i, m) = 1\}|.$$

Pagal susitarimą: $\varphi(1) = 1$. Tuomet gauname $\varphi(4) = 2$, $\varphi(5) = 4$, $\varphi(6) = 2$.

2.21. Teorema. Jeigu $m = p^\alpha$ yra pirminio skaičiaus p natūralusis laipsnis α, tuomet

$$\varphi(p^\alpha) = p^\alpha\left(1 - \frac{1}{p}\right).$$

Kai $\alpha = 1$, tai

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

▲Tarkime, kad $\alpha \ge 1$. Sekoje $i = 1, 2, 3, \dots, p^\alpha$ sąlyga $(i, p^\alpha) = 1$ pažeidžiama tik kiekvienam p-tajam eilės numeriui. Tokių narių yra $\frac{p^\alpha}{p} = p^{\alpha-1}$, todėl gauname

$$\varphi(p^\alpha) = p^\alpha - p^{\alpha-1} = p^\alpha\left(1 - \frac{1}{p}\right). \text{ ▼}$$

Jeigu modulis m nėra pirminio skaičiaus p natūralusis laipsnis, Oilerio funkcijos $\varphi(m)$ reikšmes apskaičiuoti sudėtingiau. Tuomet praverčia tokia teorema.

2.22. Teorema. Oilerio funkcija yra multiplikacinė, t. y. jeigu a ir b yra reliatyviai pirminiai skaičiai, tai

$$\varphi(ab) = \varphi(a) \cdot \varphi(b).$$

▲ Tarkime, kad $(a, b) = 1$. Skaičiuojame $\varphi(ab)$, sudarę skaičių seką

$$1, 2, \dots, ab - 1, ab.$$

Randame, kiek reliatyviai pirminių skaičių su ab yra šioje sekoje. Surašykime sekos skaičius į lentelę, turinčią a eilučių ir b stulpelių:

$$\begin{array}{ccccc} 1, & 2, & 3, & \dots & b, \\ b+1, & b+2, & b+3, & \dots & 2b, \\ 2b+1, & 2b+2, & 2b+3, & \dots & 3b, \\ \dots & \dots & \dots & \dots & \dots \\ (a-1)b+1, & (a-1)b+2, & (a-1)b+3 & \dots & ab. \end{array}$$

Visi k-tojo stulpelio skaičiai yra šitokie:

$$cb + k,\ 0 \le c \le a - 1,$$

t. y. visi priklauso tai pačiai liekanų klasei moduliu *b*. Kiekvienas stulpelis priklauso vis kitai liekanų klasei moduliu *b*, nes *k* kinta nuo 1 iki *b*.

Skaičiai reliatyviai pirminiai su *b*, gali būti tiktai stulpeliuose, kurių numeriai reliatyviai pirminiai su *b*. Visi tokio stulpelio skaičiai reliatyviai pirminiai su *b*. Tokių stulpelių iš viso $\varphi(b)$. Kiekvienas stulpelis sudaro p. l. s. moduliu *a*. Todėl jame yra $\varphi(a)$ skaičių, reliatyviai pirminių su *a*. Pasinaudoję sąlyga, kad $(i, ab) = 1 \Leftrightarrow (i, a) = (i, b)$ gauname $\varphi(ab) = \varphi(a) \cdot \varphi(b)$. (Bulota, et al., 1990) ▼

Pasinaudoję kanoniniu skaičiaus skaidiniu pirminiais daugikliais $a = p_1^{k_1} \cdot p_2^{k_2} \dots p_s^{k_s}$, gauname, kad

$$\varphi(a) = \left(p_1^{k_1} - p_1^{k_1-1}\right)\left(p_2^{k_2} - p_2^{k_2-1}\right)\dots\left(p_s^{k_s} - p_s^{k_s-1}\right)$$

arba

$$\varphi(a) = p_1^{k_1} p_2^{k_2} \dots p_s^{k_s}\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\dots\left(1-\frac{1}{p_s}\right),$$

$$\varphi(a) = a\prod_{p|a}\left(1-\frac{1}{p}\right).$$

2.23. Klausimas. Kodėl nesutampa skaičiavimai, kai $\varphi(4) = \varphi(2) \cdot \varphi(2) = 1$, bet $\varphi(2^2) = 2^2\left(1-\frac{1}{2}\right) = 2$?

2.24. Teorema (Oilerio). Jeigu $(a, m) = 1$, tai $a^{\varphi(m)} \equiv 1 (\text{mod } m)$.

▲ Tegul $a_1, a_2, \dots, a_k$ – skaičiai, paimti po vieną iš primityviųjų klasių, tuomet $k = \varphi(m)$. Tarkime, kad $(a, m) = 1$. Tuomet $aa_1, aa_2, \dots, aa_k$ taip pat reliatyviai pirminiai *m* atžvilgiu, t. y. priklauso primityvioms klasėms, tuo pačiu tarpusavyje pirminiai, nes iš $aa_i \equiv aa_j (\text{mod } m)$ gauname $a_i \equiv a_j (\text{mod } m)$, tačiau tada turi būti $a_i = a_j$. Taigi, skaičiai $aa_1, aa_2, \dots, aa_k$ yra skirtingose klasėse ir lygsta su skaičiais $a_1, a_2, \dots, a_k$ mod *m*, tiktai, galbūt, užrašytais kita tvarka. Todėl

$$aa_1 \equiv a_{i_1} \ (\text{mod } m),$$
$$aa_2 \equiv a_{i_2} \ (\text{mod } m),$$
$$\dots\dots\dots \ (\text{mod } m),$$
$$aa_k \equiv a_{i_k} \ (\text{mod } m),$$

$i_s \in \{1, 2, 3, \dots, k\}$, $s = \overline{1,k}$.

Sudauginę lyginius, gauname

$$a^k a_1 a_2 \dots a_k \equiv a_1 a_2 \dots a_k \ (\text{mod } m).$$

Skaičiai $a_1 a_2 \dots a_k$ ir *m* reliatyviai pirminiai, todėl lyginį suprastinę, turime

$$a^k \equiv 1 (\text{mod } m);$$

įrašę $k = \varphi(m)$, galutinai gauname $a^{\varphi(m)} \equiv 1 (\text{mod } m)$. Klasių kalba Oilerio teorema užrašoma: jeigu $\overline{a}$ – primityvioji klasė moduliu *m*, tuomet $\overline{a}^{\varphi(m)} \equiv \overline{1}$. (Логачев, et al., 2004) ▼

Oilerio teorema leidžia rasti duotosios primityvios klasės atvirkštinę klasę, būtent $\overline{a}^{-1} = \overline{a}^{\varphi(m)-1}$.

Oilerio teoremą galima taikyti ir lyginiams su kintamaisiais, ieškant lygties $ax \equiv 1 (\text{mod } m)$ sprendinių (Matuliauskas, 1985).

2.25. Pavyzdys. $16x \equiv 1 (\text{mod } 7)$. Tuomet $\varphi(7) = 6$ ir $x \equiv 16^{6-1} \cdot 1 (\text{mod } 7) = 4 (\text{mod } 7)$. Čia $(a, m) = 1$, t. y. $(16, 7) = 1$, todėl sprendinys – vienintelis ir $x \equiv 4 (\text{mod } 7)$. Įsitikinkite.

2.26. Teorema (Ferma). Jeigu p – pirminis skaičius, o a nesidalija iš p, tuomet $a^{p-1} \equiv 1 (\mathrm{mod}\ p)$.

▲ Tai įrodome prisiminę, kad $\varphi(p) = p - 1$, ir pasinaudoję Oilerio teorema. Beje, Ferma teoremą galima užrašyti ir taip: $a^p \equiv a (\mathrm{mod}\ p)$, $(a, p) = 1$. ▼

2.27. Apibrėžimas. ***Grupe*** vadinama netuščia elementų aibė G su apibrėžta dvinare operacija $*$: $G \times G \to G$, t. y. bet kuriai elementų $a, b \in G$ porai (a, b) priskiriamas tos pačios aibės elementas $a * b = c \in G$ ir tenkinamos šios aksiomos:

a) operacija $*$ yra asociatyvioji, t. y. su visais $a, b, c \in G$

$$(a * b) * c = a * (b * c);$$

b) egzistuoja toks vienetinis neutralusis operacijos * elementas e, kad su visais $a \in G$

$$a * e = e * a = a;$$

c) kiekvienam $a \in G$ egzistuoja toks atvirkštinis (simetriškasis) elementas $a^{-1} \in G$, kad

$$a * a^{-1} = a^{-1} * a = e.$$

Grupę žymėsime simboliu $<G; *>$, o grupės $<G; *>$ elementų skaičių, kuris gali būti baigtinis arba begalinis, – $|G|$.

Jeigu grupėje $<G; *>$ tenkinama komutatyvumo aksioma $b * a = a * b$, tuomet sakoma, kad grupė yra ***komutatyvioji*** (***Abelio***).

Dažnai operacijai $*$ grupėse žymėti vartosime daugybos ženklą, t. y. rašysime $a \cdot b$ vietoj $a * b$, ir vadinsime grupę ***multiplikacine***, t. y. $<G; \cdot>$. Jei vartosime sudėties ženklą $<G; +>$, tuomet grupę vadinsime ***adicine*** ir žymėsime $a^{-1} = -a$ bei $e = 0$. Tokie žymėjimai dažnai pasitaiko Abelio grupėse (van der Varden, 1976).

2.28. Pavyzdys. Keitinių grupė $<S_n; \circ>$ – baigtinės aibės bijektyvaus vaizdavimo į save grupė. Keitiniai užrašomi, pvz.,

$$\tau = \begin{pmatrix} i_1 & i_2 & \dots & i_n \\ j_1 & j_2 & \dots & j_n \end{pmatrix}, \quad \begin{matrix} i_s \in \{1,2,\dots,n\}, \\ j_k \in \{1,2,\dots,n\}. \end{matrix}$$

t. y. $j_k = \tau(i_k)$. Operacija $\circ$ tarp keitinių – sąsūka (arba nuoseklus keitinių atlikimas eilės tvarka), pvz.,

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}.$$

Kitaip tariant, $\tau(1) = 3$, $\tau(3) = 2$, t. y. $\tau(\tau(1)) = 1$. Aišku, kad $|S_n| = n!$ (Dosinas, et al., 2006).

2.29. Apibrėžimas. Tarkime, $<G; * >$ yra grupė ir $A \subset G$. Jeigu su bet kuriais $a, b \in A$, gaunama $a * b \in A$, o iš sąlygos $a \in A$ gauname, kad ir $a^{-1} \in A$, tuomet struktūra $<A; *>$ vadinama grupės $<G; *>$ pogrupiu ir žymima $<A; *> \subset <G; *>$. Kitaip tariant, ***pogrupis*** yra grupė, apibrėžta elementų poaibyje $A \subset G$.

2.30. Pavyzdys. n-tojo laipsnio vieneto šaknys $\sqrt[n]{1}$.

Imkime vieneto šaknų aibę $\sqrt[n]{1} = V_n = \left\{ \cos\frac{2k\pi}{n} + i\sin\frac{2k\pi}{n} \,|\, k = \overline{0, n-1} \right\}$, kurią galima gauti, įbrėžus į apskritimą taisyklingą n-kampį, kurio viena viršūnė yra taške (1, 0). Kaip matome, ši grupė $< V_n; \cdot >$ yra baigtinė, t. y. turi baigtinį elementų skaičių, $|V_n| = n$.

Pažymėję $e_k = \cos\frac{2\pi k}{n} + i\sin\frac{2\pi k}{n}$, $k = \overline{0, n-1}$, gauname, kad

$$e_k \cdot e_t = \cos\frac{2(k+t)\pi}{n} + i\sin\frac{2(k+t)\pi}{n} = e_{k+t(\mathrm{mod}\ n)} \in V_n.$$

Be to, $e_k^{-1} = \cos\frac{2(n-k)\pi}{n} + i\sin\frac{2(n-k)\pi}{n} = e_{n-k} \in V_n$; $e_0 = 1 \in V_n$. Vadinasi, struktūra $< V_n; \cdot >$ yra multiplikacinė Abelio grupė.

Kai $n = 6$, turėsime šeštojo laipsnio vieneto šaknų grupę $< V_6; \cdot >$. Elementų daugybos operacijos rezultatas pavaizduotas Keilio lentele:

·	e_0	e_1	e_2	e_3	e_4	e_5
e_0	e_0	e_1	e_2	e_3	e_4	e_5
e_1	e_1	e_2	e_3	e_4	e_5	e_0
e_2	e_2	e_3	e_4	e_5	e_0	e_1
e_3	e_3	e_4	e_5	e_0	e_1	e_2
e_4	e_4	e_5	e_0	e_1	e_2	e_3
e_5	e_5	e_0	e_1	e_2	e_3	e_4

2.31. Pavyzdys. Liekanų klasės $\mathbf{Z}_p^* = \mathbf{Z}_p \setminus \{\bar{0}\}$, čia p – pirminis skaičius, taip pat sudaro grupę daugybos atžvilgiu, atliekant operacijas moduliu p.

Toliau liekanų klases žymėsime be brūkšnelio, kai bus aišku, apie kokią grupę kalbama.

Grupės $<\mathbf{Z}_5^*; \cdot>$ daugybos Keilio lentelė:

·	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

.

Prisiminę Oilerio funkcijos $\varphi(p)$ apibrėžimą, gauname $|\mathbf{Z}_p^*| = \varphi(p) = p - 1$, o $|\mathbf{Z}_p| = p$.

Pastebėsime, kad multiplikacinė grupė $\mathbf{Z}_p^*$ yra Abelio grupė ir $|\mathbf{Z}_p^*| = p - 1$. Kiekvienam grupės $\mathbf{Z}_p^*$ elementui a gauname

$$a^{p-1} \equiv 1 (\mathrm{mod}\ p),$$

kai $(a, p) = 1$. Taigi turime mažosios Ferma teoremos teiginį.

Šis rezultatas yra bendresnės Oilerio teoremos išvada, t. y.

$$a^{\varphi(p)} \equiv 1 (\mathrm{mod}\ p),\ (a, p) = 1.$$

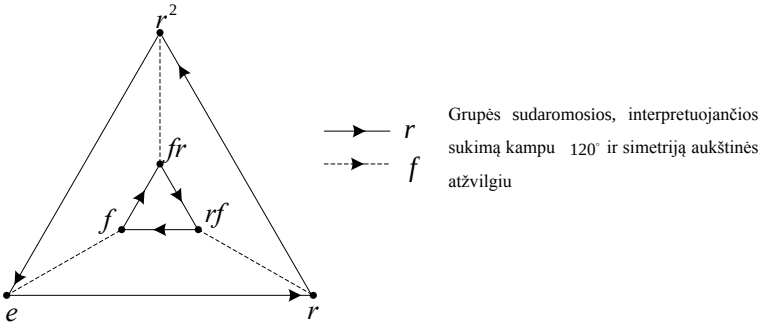
Toliau paminėsime kai kurias svarbias kriptografijoje grupes.

2.32 Pavyzdys. Laisvosios grupės (Магнус, et al., 1974).
Imkime aibę simbolių $s_1, s_2, ..., s_n$ (paprastumo vardan laikykime ją baigtine, nors samprotavimai nuo to nepriklauso). Kiekvienam simboliui s_i priskirkime dar vieną simbolį s_i^{-1}. **Žodžiu** vadinsime vienas greta kito parašytų simbolių s_i ir s_j^{-1} seką. Prijunkime neutralųjį (tuščią) simbolį θ, t. y. tokį, kad visiems s_i turime $s_i\theta = s_i = \theta s_i$. Atvirkštiniu duotam žodžiui vadinsime tokį, kurio visi simboliai parašyti atvirkštine tvarka, keičiant s_i į s_i^{-1} ir s_i^{-1} į s_i. Žodžių A ir B sandauga vadinsime žodį, kuris sudarytas parašius B po A ir išbraukus poras s_i ir s_i^{-1}, kol gaunamas redukuotas žodis, t. y. toks, kuriame nėra greta parašytų simbolių s_i ir s_i^{-1}. Redukuotas žodis gali būti ir θ. Redukuotų žodžių aibė su daugybos operacija sudaro grupę. Ji vadinama laisvąja grupe su n sudaromųjų ir žymima $\mathcal{J}_n$. Akivaizdu, kad žodžiai $s_1, s_2, ..., s_n$, sudaryti iš vieno simbolio, yra grupės sudaromieji simboliai; s_i^{-1} – atvirkštiniai s_i, ir bet kurį žodį galime laikyti skirtingų žodžių s_i ir s_j^{-1} sandauga.
Paprasčiausias laisvosios grupės pavyzdys yra $<\mathbf{Z}; +>$ – sveikųjų skaičių aibė su sudėtimi.

2.33. Apibrėžimas. Grupė G yra vadinama ***cikline,*** jeigu egzistuoja toks elementas $a \in G$, kad kiekvienas grupės elementas x gali būti užrašytas a^n, $n \in \mathbf{Z}$, forma. Jeigu atvaizdis f: $\mathbf{Z} \to G$, $f(n) = a^n$ yra siurjekcija, elementas a vadinamas grupės generatoriumi.

Sakome, kad $m > 0$ yra grupės G eilė, jeigu $x^m = e$, visiems $x \in G$.

2.34. Pavyzdys. Lygiakraščio trikampio simetrijos grupė D_3 (Гроссман, et al., 1971).
Grupę galima pateikti, žinant jos sudaromąsias ir ryšius tarp jų. Pvz., lygiakraščio trikampio simetrijų grupė $D_3 = \langle r, f \mid f^2 = e, r^3 = e, frf = f^2 \rangle$. Nesunku patikrinti, kad simboliai $f^k r^k$, $k = 0, 1$; $m = 0, 1, 2$ sudaro multiplikacinę grupę.
Grupę D_3 sudaro lygiakraščio trikampio sukimas apie centrą kas 120° ir simetrija aukštinių atžvilgiu. Grupę lengva interpretuoti tokiu grafu



Grupės D_3 sudaromoji r sudaro ciklinę grupę C_3.
Čia iškyla rimta problema, kaip žinoti, kad du elementai, užrašyti sudaromųjų raiška, yra lygūs. Ši problema grupių teorijoje vadinama **tapatumo** problema. Bendru atveju jos sprendimo nėra. Tai pagrindė P. S. Novikovas 1952 m.
2.35. Pavyzdys. Tiesioginė grupių sandauga (Matuliauskas, 1985).
Tiesioginė grupių G ir H sandauga $G \times H$, taipogi, yra grupė. Tiesioginė sandauga apibrėžiama elementų poroms: $(a_1, b_1) \cdot (a_2, b_2) = (a_1a_2, b_1b_2)$, $a_1, a_2 \in G$, $b_1, b_2 \in H$. Tiesioginė Abelio grupių sandauga yra Abelio grupė.
Pabrėšime, kad operacijos grupėse G ir H gali būti skirtingos. Čia panaudotas multiplikacinis užrašas grupėms G, H ir $G \times H$.
Griežtai tariant, reikėtų rašyti $(a_1, b_1) * (a_2, b_2) = (a_1 \circ a_1, b_1 \square b_2)$; čia $\circ$, $\square$, $*$ – binarės operacijos aibėse G, H ir $G \times H$, atitinkamai. Aditiniame grupės užraše, pvz. Abelio grupėse, natūralu kalbėti apie tiesioginę sumą $G \oplus H$.
(e_G, e_H) yra neutralusis $G \times H$ elementas. Atvirkštinis elementui $(g, h) \in G \times H$ yra elementas (g^{-1}, h^{-1}).

Jeigu $gg' = g'g$ visiems $g, g' \in G$ ir $hh' = h'h$ visiems $h, h' \in H$, tuomet $(g, h)(g', h') = (gg', hh') = (g'g, h'h) = (g', h')(g, h)$ visiems (g, h) ir (g', h') sandaugoje $G \times H$.
Pavyzdžiui, **b**et kuriam pirminiam p tiesioginė ciklinių grupių $\langle \mathbf{Z}_p; + \rangle$ n sandauga $\mathbf{Z}_p \times \mathbf{Z}_p \times \ldots \times \mathbf{Z}_p$ egzempliorių vadinama elementariąja Abelio grupe, kurios eilė p^n.

2.36. Apibrėžimas. Grupės $\langle G; \cdot \rangle$ pogrupiu $\langle H; \cdot \rangle$ vadiname bet kurį netuščią poaibį $H \subset G$, kuris tenkina šias sąlygas:

a) jeigu $a, b \in H$, tai $ab \in H$;

b) jeigu $a \in H$, tai $a^{-1} \in H$;

c) $e \in H$.

Tarkime G – grupė, o H – jos pogrupis. Sakysime, kad $g_1, g_2 \in G$ lygsta moduliu H, ir rašysime

$$g_1 \equiv g_2 \pmod{H},$$

jeigu $g_1^{-1}g_2 \in H$, t. y. $g_2 = g_1h$, $h \in H$.

2.37. Teorema. Lyginys $g_1 \equiv g_2 \pmod{H}$ yra ekvivalentumo sąryšis.

▲ Visų pirma $g \equiv g \pmod H$, nes $g^{-1}g = e \in H$. Antra vertus, jeigu $g_1 \equiv g_2 \pmod H$, t. y. $g_1^{-1}g_2 \in H$, tai ir $g_2 \equiv g_1 \pmod H$, nes

$$g_2^{-1}g_1 \equiv g_1^{-1}g_2 \in H.$$

Jeigu $g_1 \equiv g_2 \pmod H$ ir $g_2 \equiv g_3 \pmod H$, t. y. $g_1^{-1}g_2, g_2^{-1}g_3 \in H$, tai $g_1 \equiv g_3 \pmod H$, kadangi

$g_1^{-1}g_3 = (g_1^{-1}g_2)(g_2^{-1}g_3) \in H$. ▼

Ekvivalentumo klasės vadinamos (kairiosiomis) ***gretutinėmis klasėmis*** grupėje G pogrupio H atžvilgiu. Akivaizdu, kad gretutinė klasė, kurioje yra elementas g yra aibė

$$gH = \{gh \mid h \in H, g \in G\}.$$

Viena iš gretutinių klasių yra pats pogrupis H.

Kadangi daugyba grupėje nebūtinai komutatyvioji, tai gauname ir kitą ekvivalentumo sąryšį, paėmę analogiškai $g_2^{-1}g_1 \in H$, kuris pavadintas dešiniąja gretutine klase grupėje G atžvilgiu H, t. y.

$$Hg = \{hg \mid h \in H, g \in G\}.$$

Pastebėsime, kad inversija $g \to g^{-1}$ nustato bijektyvų vaizdavimą tarp kairinės ir dešininės klasių. Taigi, gauname $(gH)^{-1} = Hg^{-1}$.

Gretutinių klasių aibę grupėje G pogrupio H atžvilgiu (nesvarbu kairinių ar dešininių), jeigu ji baigtinė, pavadinsime pogrupio H indeksu grupėje G ir žymėsime $|G : H|$ (Кострикин, 2001).

2.38. Teorema. (Lagranžo). Jeigu G – baigtinė grupė ir H – bet kuris jos pogrupis, tai $|G| = |G : H| \cdot |H|$, kitaip pasakius pogrupio eilė dalija grupės eilę.

▲ Visos gretutinės klasės gH turi vienodų elementų skaičių, lygų $|H|$. Kadangi gretutinės klasės suskaido grupę G į nesikertančių ekvivalentumo klasių junginį $G = \bigcup gH$, grupės eilė lygi jų skaičiui, daugintam iš $|H|$. ▼

2.39. Apibrėžimas. Grupės G pogrupis H vadinamas normaliuoju, jeigu:

$$gH = Hg, \text{ su visais } g \in G;$$

arba, ekvivalenčiai užrašius:

$$gHg^{-1} = H.$$

Trumpai rašome $H \triangleleft G$ (arba $G \triangleright H$).

Kad pogrupis H būtų normalusis, pakanka (bet nebūtina), jog kiekvienas grupės G elementas komutuotų su kiekvienu elementu iš H. Abelio grupėje kiekvienas pogrupis yra normalusis.

2.40. Teorema. Lyginys moduliu H yra suderintas su daugybos operacija grupėje G tada ir tik tada, kai pogrupis H yra normalusis.

▲ Lyginio moduliu H suderinamumas su daugybos operacija reiškia, kad

$$g_1 \equiv g_1'(\text{mod } H),\ g_2 \equiv g_2'(\text{mod } H) \Rightarrow g_1g_2 \equiv g_1'g_2'(\text{mod } H);$$

arba su visais $g_1, g_2 \in G$ ir $h_1, h_2 \in H$ turima, kad

$$(g_1h_1)(g_2h_2) \equiv g_1g_2(\text{mod } H).$$

Iš čia, remiantis apibrėžimu, taip pat gauname, kad

$$g_2^{-1}hg_2 \in H,$$

kadangi g_2 gali būti bet kuris grupės G elementas, h – bet kuris pogrupio H elementas. ▼

Tokiu būdu, jei $H \triangleleft G$, tai daugybos operacija grupėje G nusako daugybos operaciją aibėje G / H, remiantis taisykle

$$(g_1H)(g_2H) \equiv (g_1g_2)H.$$

Ši operacija išlaiko asociatyvumo reikalavimą grupėje. Vienetinis elementas yra eH. Kiekviena gretutinė klasė gH turi atvirkštinę – $g^{-1}H$. Taigi, gauname grupę $<G / H; \cdot >$, kuri vadinama grupės G ***faktorgrupe*** H atžvilgiu.

Jeigu grupė yra Abelio, tuomet bet kuri jos faktorgrupė yra Abelio.

2.41. Apibrėžimas. Grupių $<G; *>$ ir $<G'; \square>$ atvaizdis f: $<G; *> \to <G'; \square>$ vadinamas ***homomorfizmu***, jeigu $f(a * b) = f(a) \square f(b)$.

2.42. Teorema. Homomorfizmo f: $G \to G'$ atveju neutralusis G elementas atvaizduojamas į neutralųjį G' elementą.

▲ $f(e) = f(e * e) = f(e) \square f(e)$. Padauginę iš kairės gautą lygybę iš $(f(e))^{-1}$, gauname:

$$e' = (f(e))^{-1} \square f(e) = (f(e))^{-1} \square (f(e) \square f(e)) = ((f(e))^{-1} \square f(e)) \square f(e) = e' \square f(e) = f(e).$$ ▼

2.43. Teorema. Bet kuriam grupės G elementui homomorfizmo f atveju teisinga lygybė

$$f(a^{-1}) = (f(a))^{-1}.$$

▲ $e' = f(e) = f(a * a^{-1}) = f(a) \square f(a^{-1})$ ir $e' = f(e) = f(a^{-1} * a) = f(a^{-1}) \square f(a)$.

Vadinasi, $f(a^{-1}) = (f(a))^{-1}$. ▼

2.44. Apibrėžimas. Homomorfizmo f: $G \to G'$ ***branduoliu*** vadinama aibė grupės G elementų, kuriuos homomorfizmas f atvaizduoja į grupės G' vienetinį elementą. Homomorfizmo branduolys žymimas $Ker\ f$.

2.45. Teiginys. Grupės G branduolys $Ker\ f := \{a \in G \mid f(a) = e\}$ yra normalusis pogrupis grupėje G.

▲ $a, b \in Ker\ f \Rightarrow f(ab) = f(a)f(b) = e^2 = e \Rightarrow ab \in Ker\ f$;

$a \in Ker\ f \Rightarrow f(a^{-1}) = (f(a))^{-1} = e^{-1} = e \Rightarrow a^{-1} \in Ker\ f$ ir $e \in Ker\ f$;

$a \in Ker\ f, g \in G \Rightarrow f(gag^{-1}) = f(g)f(a)f(g^{-1}) = f(g)e(f(g))^{-1} = e \Rightarrow gag^{-1} \in Ker\ f$. ▼

2.46. Pavyzdys. Pažymėkime $GL(n, R)$ visų neišsigimusių matricų su realiaisiais elementais grupę matricų daugybos atžvilgiu. Tuomet vaizdavimas $A \to \det A$, $A \in GL(n, R)$ yra siurjektyvusis atvaizdis į multiplikacinę grupę $<R^*; \cdot >$. Homomorfizmo $GL(n, R) \to R^*$ branduolį sudaro matricų, kurių determinantas lygus vienetui, pogrupis.

2.47. Teiginys. Tarkime, kad G – grupė, o $a \in G$. Imkime homomorfizmą f: $\mathbf{Z} \to G$, $f(n) = a^n$, kurio branduolį žymime $Ker\ f = H$. Galimi šie atvejai (Dosinas, et al., 2006):

1. Branduolys yra trivialusis – $Ker\ f = e$. Tuomet f apibrėžia $\mathbf{Z}$ ir ciklinio grupės G pogrupio, kurį generuoja elementas $a \in G$, izomorfizmą. Šis pogrupis yra begalinis. Jeigu a generuoja G, tai G – ciklinė grupė. Tokiu atveju sakome, kad a yra begalinės eilės (begalinio periodo) elementas.
2. Branduolys nėra trivialusis. Tuomet jeigu G yra baigtinė grupė, kurios eilė $m > 1$, tai bet kurio elemento $a \neq e$ periodas, t. y. mažiausias skaičius d, kad $a^d = e$, $d \in \mathbf{N}\backslash\{1\}$, dalija m. Jeigu grupės G eilė yra pirminis skaičius p, tuomet G – ciklinė grupė, ir kiekvienas nelygus e elementas yra grupės G generatorius, pvz., $GF(2^2)$ multiplikacinėje grupėje.

2.48. Teorema. Grupių homomorfizmas f: $G \to G'$, kurio branduolys – trivialusis, yra injekcinis.

▲ Tarkime, kad $Ker\ f = e$ ir $f(x) = f(y)$; $x, y \in G$. Tuomet

$$f(xy^{-1}) = f(x) \cdot f(y^{-1}) = f(y) \cdot (f(y))^{-1} = e'.$$

Todėl $xy^{-1} \in Ker\ f$ ir $xy^{-1} = e$ arba $x = y$. ▼

Grupės homomorfizmas grupėje vadinamas endomorfizmu, o grupės izomorfizmas į grupę – jos automorfizmų. Grupės homomorfizmas į grupę vadinamas epimorfizmu (siurjektyvusis atvaizdis) (Dosinas, et al., 2006).

2.49. Teorema. *Apie homomorfinį vaizdavimą*. Tarkime, kad atvaizdis $f : G \to G_1$ yra epimorfizmas. Tuomet $G_1 \cong G / Ker\ f$. Jeigu H – normalusis grupės G pogrupis, tai $f : G \to G / H$, kai $f(a) = aH$, yra epimorfizmas ir $Ker\ f = H$.

▲ Jeigu $f(g) = g'$, tai

$$f(gh) = f(g)f(h) = f(g) = g', \text{ kai } h \in Ker\ f.$$

Todėl egzistuoja grupės $g / Ker\ f$ siurjekciją $\bar{f}$ į grupę G_1. Įrodysime injektyvumą.

Jeigu $f(g_1) = f(g_2)$, tai

$$f(g_1)f^{-1}(g_2) = f(g_1g_2^{-1}) = e,$$

tuo pačiu gauname, kad $g_1 \in g_2 Ker\ f$. Taigi, nustatome tokią bijekciją:

$$\bar{f}\ (gKer\ f) = f(g).$$

Sąlyga

$$\bar{f}\ (g_1Ker\ f\ g_2Ker\ f) = \bar{f}\ (g_1Ker\ f) \cdot \bar{f}\ (g_2Ker\ f)$$

lengvai gaunama iš sąlygos $f(g_1g_2) = f(g_1)f(g_2)$. Analogiškai tikrinamas paskutinis teoremos teiginys. ▼

Jeigu grupė G baigtinė, tuomet

$$|G| = |\text{Im } f| \cdot |Ker\ f|.$$

2.50. Teorema. Keilio(Dosinas, et al., 2006)**.** Kiekviena baigtinė grupė G yra izomorfinė simetrijos grupės S_n pogrupiui.

▲ Pažymėkime grupės G, kurios $|G| = n$, elementus $e, g_1, ..., g_{n-1}$ ir nagrinėkime atvaizdį, kai $a \in G$ yra fiksuotas elementas, t. y.

$$\tau_a = \begin{pmatrix} e & g_1 & \dots & g_{n-1} \\ ae & ag_1 & \dots & ag_{n-1} \end{pmatrix}.$$

Kitaip tariant $G \to S_n$ ir $a \to \tau_a$. Turime izomorfinį vaizdavimą, nes skirtingus elementus a ir b atitinka skirtingi keitiniai τ_a ir τ_b; pirmu atveju $e \to a$, antru $e \to b$.

Tikriname sąlygą $\tau_{ab} = \tau_a\tau_b$. Bet kuriam grupės G elementui g turime, kad

$$\tau_{ab} = abg \text{ ir } (\tau_a\tau_b) \cdot g = \tau_a(\tau_b(g)) = abg. \text{ ▼}$$

Iš Keilio teoremos gauname, kad izomorfinių grupių galime neskirti vienų nuo kitų.

2.51. Apibrėžimas. Žiedu vadiname dviveiksmę algebrinę struktūrą, nusakytą netuščioje aibėje $\mathcal{R}$, t. y. $<\mathcal{R}; +; \cdot>$, jeigu:

a) struktūra $<\mathcal{R}; +>$ yra adicinė Abelio grupė;

b) $<\mathcal{R}; \cdot>$ – multiplikacinė asociatyvioji struktūra (pusgrupė);

c) su visais $a, b, c \in \mathcal{R}$ tenkinami distributyvumo dėsniai:

$$a(b + c) = ab + bc \text{ ir } (b + c)a = ba + ca.$$

Adicinės grupės neutralųjį elementą žymėsime 0, priešingą elementui a žymėsime $–a$. Tuomet vietoj $a + (–b)$ paprastai rašysime $a – b$. Nesunku patikrinti, kad $a \cdot 0 = 0 \cdot a = 0$ su visais $a \in \mathcal{R}$. Iš čia gauname $(–a)b = a(–b)$ su visais $a, b \in \mathcal{R}$.

Paprasčiausias žiedo pavyzdys – sveikųjų skaičių žiedas $<\mathbf{Z}; +; \cdot>$.

2.52. Apibrėžimas. Žiedas vadinamas komutatyviuoju, jeigu daugyba struktūroje $<\mathcal{R}; \cdot>$ yra komutatyvi.

2.53. Apibrėžimas. Jeigu egzistuoja toks daugybos neutralusis elementas e (dažnai žymimas tiesiog 1), kad $ea = ae = a$ su visais $a \in \mathcal{R}$, tuomet sakome, kad turime žiedą su vienetu.

2.54. Teorema. Visi apgręžiami asociatyvaus žiedo su vienetu elementai ($1 \neq 0$) sudaro grupę $U(\mathcal{R})$ daugybos atžvilgiu (angl. *Group of units*).

▲ Kadangi $U(\mathcal{R})$ aibėje yra vienetas ir daugybos asociatyvumas išpildomas, tuomet pakanka įrodyti tik uždarumą aibėje $U(\mathcal{R})$, t. y. parodyti, kad bet kurių dviejų elementų a ir b iš $U(\mathcal{R})$ sandauga $a \cdot b$ taip pat priklauso $U(\mathcal{R})$.

Tikrai, $(ab)^{-1} = b^{-1}a^{-1}$, nes

$$(ab)(b^{-1}a^{-1}) = a(bb^{-1})a^{-1} = a \cdot e \cdot a^{-1} = aa^{-1} = e.$$

Taigi ab yra apgręžiamas elementas, t. y. egzistuoja jam atvirkštinis. ▼

Žiede $<\mathbf{Z}_m; +, \cdot >$ grupę $U(\mathcal{R})$ sudaro elementai $\{a \mid (a, n) = 1\}$(Grigutis, 1998).

2.55. Apibrėžimas. Žiedą vadiname ***lauku***, jeigu struktūra $<\mathcal{R}; \cdot>$ yra multiplikacinė Abelio grupė. Akivaizdu, kad bet kuris laukas turi ne mažiau kaip du elementus. Pvz., dviejų elementų laukas $<\{0, 1\}; +, \cdot>$:

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

2.56. Klausimas. Lauke abu distributyvumo dėsniai sutampa. Kodėl?

2.57. Apibrėžimas. Žiedo $\mathcal{R}$ poaibis S yra vadinamas ***požiedžiu*** žiede $\mathcal{R}$, jeigu jis yra uždaras žiedo sudėties bei daugybos operacijų atžvilgiu ir sudaro žiedą.

2.58. Apibrėžimas. Požiedis I žiede $\mathcal{R}$ yra vadinamas ***idealu***, jeigu su visais $a \in I$ ir $r \in \mathcal{R}$ gaunama $ar \in I$ ir $ra \in I$.

Sveikųjų skaičių aibė $\mathbf{Z}$ yra požiedis racionaliųjų skaičių lauke $\mathbf{Q}$, bet nėra idealas. Lengva parinkti tokius sveikąjį ir racionalųjį skaičius, kad sandauga nebūtų sveikasis skaičius.

Tarkime, kad $\mathcal{R}$ – komutatyvusis žiedas, $a \in I$. Imkime poaibį $I = \{ar \mid r \in \mathcal{R}\}$. Tuomet I – žiedo $\mathcal{R}$ idealas, nes $ax + ay = a(x + y) \in I$ ir $(ax)y = a(xy) \in I$; $x, y \in \mathcal{R}$. Pavyzdžiui, sveikieji skaičiai, kartotiniai skaičiui m, žiede $\mathbf{Z}$ sudaro idealą $I = m\mathbf{Z}$.

2.59. Apibrėžimas. Komutatyviajame žiede $\mathcal{R}$ su vienetu idealas $I = \{ar \mid r \in \mathcal{R}\}$, kurį generuoja žiedo $\mathcal{R}$ elementas a, yra vadinamas vyriausiuoju idealu, t. y. $I = a\mathcal{R}$; žymime $<a>$.

Idealas $\{\sum_{i=1}^{n} r_i a_i \mid r_i \in \mathcal{R}\}$ yra vadinamas idealu, kurį generuoja elementai $a_1, a_2, ..., a_n$, ir žymimas $<a_1, a_2, ..., a_n>$.

2.60. Teorema. Žiedo $\mathcal{R}$ netuščias poaibis I yra to žiedo idealas tada ir tik tada, kai:

a) bet kurių dviejų poaibio I elementų a ir b skirtumas a – b priklauso tam poaibiui;

b) poaibio I bet kurio elemento a ir žiedo $\mathcal{R}$ bet kurio elemento r sandaugos ar ir ra priklauso tam poaibiui.

2.61. Faktoržiedas pagal idealą. Žiedo pagrindą sudaro adicinė Abelio grupė. Sudarykime faktoržiedą $\mathcal{R}/I$ idealo I atžvilgiu. Imkime žiedo gretutines klases (sluoksnius) $a + I$ (t. y. lyginius moduliu I); sudėtį ir atimtį apibrėžiame taisyklėmis:

$$(a + I) \oplus (b + I) = (a + b) + I,$$
$$-(a + I) = -a + I.$$

Klasių sandaugos operaciją apibrėžiame taisykle:

$$(a + I) \otimes (b + I) = ab + I.$$

Įsitikinkime daugybos operacijos korektiškumu, t. y. kad sandauga nepriklauso nuo klasių atstovų parinkimo.

Tarkime, kad $a' = a + x$, $b' = b + y$, $x, y \in I$. Tuomet $a'b' = ab + ay + xb + xy = ab + z$, čia $z = ay + xb + xy \in I$, nes I – idealas (dvipusis). Todėl $a'b'$ yra toje pat klasėje kaip ir elementas ab. Tai reiškia, kad sandauga apibrėžta korektiškai. Trumpumo dėlei klasę pažymėkime $\bar{a} = a + I$, kad

$$\bar{a} \oplus \bar{b} = \overline{a+b}, \ \bar{a} \otimes \bar{b} = \overline{ab}.$$

Be to, $\bar{0} = I$ ir $\bar{1} = 1 + I$ (jeigu vienetas yra žiede $\mathcal{R}$). Dabar labai nesunku patikrinti, ar aibėje $\bar{R} = \mathcal{R}/\mathbf{Z} = \{ \bar{a} \mid a \in \mathcal{R} \}$, kurioje apibrėžtos operacijos $\oplus$ ir $\otimes$, išpildomi visi žiedo reikalavimai (Кострикин, 1977).

2.62. Pastaba. Paprastumo dėlei, kalbėdami apie veiksmus su sluoksniais faktoržiede, rašysime + ir $\cdot$ vietoj $\oplus$ ir $\otimes$.

2.63. Apibrėžimas. Vieneto dalikliais ε vyriausiųjų idealų komutatyviajame žiede $\mathcal{R}$ su vienetu vadiname tokius žiedo $\mathcal{R}$ elementus, kuriems žiede $\mathcal{R}$ egzistuoja atvirkštiniai elementai ε^{-1} (imant skyrium, 1 – taip pat vieneto daliklis).

Pavyzdžiui, vieninteliai vieneto dalikliai daugianarių žiede $\mathbf{F}[x]$ lauke $\mathbf{F}$ yra nulinės eilės daugianariai.

Pastebėsime, kad bet kuris elementas $a \in \mathcal{R}$ turi dalytis iš ε ir $a\varepsilon$ (ε – bet kuris vieneto daliklis). Šie dalikliai vadinami trivialiaisiais.

2.64. Teorema. Liekanų klasės moduliu m ($\mathbf{N} \ni m \geq 2$) sutampa su faktoržiedo $\mathbf{Z}/<m>$ elementais; $<m> = m\mathbf{Z}$.

▲ Faktoržiedo $\mathbf{Z}/<m>$ elementą $k + <m>$ (čia $k = 0, 1, 2, ..., m - 1$) sudaro visi sveikieji skaičiai $k + mq$, $q \in \mathbf{Z}$. Tokių skaičių dalybos iš skaičiaus m liekana yra k, todėl $\bar{k} = k + <m>$. Paėmę $k = 0, 1, ..., m - 1$, gausime visus faktoržiedo $\mathbf{Z}/<m>$ elementus ir visas liekanų moduliu m klases: $\bar{0}, \bar{1}, \bar{2}, \ldots, \overline{m-1}$. Todėl faktoržiedas $\mathbf{Z}/<m>$ yra izomorfinis žiedui $\mathbf{Z}_m$; žymime $\mathbf{Z}/<m> \cong \mathbf{Z}_m$. ▼

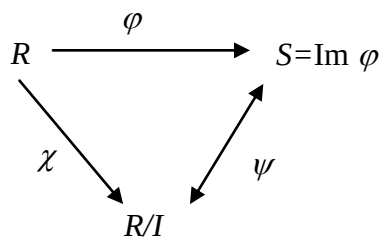
2.65. Apibrėžimas. Tarkime, kad $\mathcal{R}$ ir S yra žiedai su vienetu. Homomorfizmu φ: $\mathcal{R} \to S$ vadiname atvaizdį, kai

$$\varphi(a + b) = \varphi(a) + \varphi(b); \ \varphi(ab) = \varphi(a) + \varphi(b); \ \varphi(e) = e', \text{ su visais } a, b \in \mathcal{R}.$$

Terminai *branduolys, vaizdas, epimorfizmas, monomorfizmas, izomorfizmas, automorfizmas* ir *endomorfizmas* turi tokią pat prasmę kaip ir grupėse.

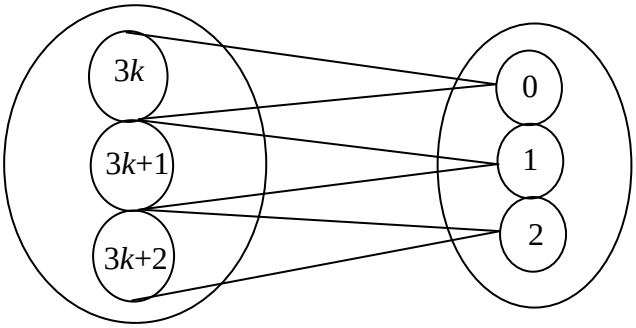
2.66. Teorema. Jeigu φ – homomorfinis žiedo $\mathcal{R}$ atvaizdis į žiedą S (epimorfizmas), t. y. $\varphi(\mathcal{R}) = S$, tuomet $Ker\ \varphi$ – žiedo $\mathcal{R}$ idealas; be to, žiedas S yra izomorfiškas faktoržiedui $\mathcal{R}/Ker\ \varphi$. Ir atvirkščiai, jeigu I – žiedo $\mathcal{R}$ idealas, tai atvaizdis χ: $\mathcal{R} \to \mathcal{R}/I$, nusakytas taisykle $\chi(a) = a + I$, yra $\mathcal{R}$ epimorfizmas į $\mathcal{R}/I$ su branduoliu I (Кострикин, 1977).

Teoremą iliustruoja schema:



$$\psi(\bar{a}) = \varphi(a), \ Ker\ \varphi = \{a \in \mathcal{R} \mid \varphi(a) = 0 \in S\}$$
$$\bar{a} = a + I = \{a + h \mid h \in I\}.$$

2.67. Pavyzdys. Imkime atvaizdį φ: $\mathbf{Z} \to \mathbf{Z}_3$ – siurjekciją, nusakytą taisykle $\varphi(3k) = 0$, $\varphi(3k + 2) = 2$, $\varphi(3k + 1) = 1$. Tuomet gausime homomorfinį žiedų atvaizdį: $< \mathbf{Z}; +, \cdot > \simeq < \mathbf{Z}_3; +, \cdot >$.



$< \mathbf{Z}; +, \cdot >$ $< \mathbf{Z}_3; +, \cdot >$

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

·	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Tikrai, pvz., $\varphi(4 \cdot 7) = \varphi((3 + 1) \cdot (6 + 1)) = \varphi(18 + 6 + 3 + 1)\ \varphi(27 + 1) = 1$.

$\varphi(4 + 7) = \varphi(11) = \varphi(3 \cdot 3 + 2) = 2$.

Kita vertus, $\varphi(4) = 1$, $\varphi(7) = 1$ ir $\varphi(4) \cdot \varphi(7) = 1$, $\varphi(4) + \varphi(7) = 2$.

2.68. Klausimas. Kodėl atvaizdis $\varphi(3k + 1) = 0$, $\varphi(3k) = 2$, $\varphi(3k + 2) = 1$ nėra homomorfinis?

2.69. Apibrėžimas. Žiedo elementai $a \neq 0$, $b \neq 0$ yra vadinami nulio dalikliais, jeigu $a \cdot b = 0$.

2.70. Apibrėžimas. Komutatyvųjį žiedą su vienetu ($1 \neq 0$) ir be nulio daliklių vadiname integralumo sritimi.

Integralumo sritis, kurioje kiekvienas idealas – vyriausiasis, yra vadinama vyriausiųjų idealų žiedu.

2.71. Teorema. Baigtinė integralumo sritis yra laukas.

▲Tarkime, kad $\mathcal{R}$ yra integralumo sritis. Imkime žiedo $\mathcal{R}$ elementus $a_1, a_2, \ldots, a_n$. Paėmę nenulinį elementą $a \in \mathcal{R}$, sudarome elementų seką $aa_1, aa_2, \ldots, aa_n$. Šios sekos elementai yra skirtingi, nes integralumo srityje iš $aa_1 = aa_2$ gautume $a_1 = a_2$, todėl šioje sekoje yra vienetinis elementas $e = a \cdot a_i$. Tuo pačiu įrodyta, kad kiekvienam $a \neq 0$ egzistuoja atvirkštinis, nes $a\,(a_1 - a_2) = 0$, o nulio daliklių nėra.▼

2.72. Teorema. Jeigu m yra pirminis, tai liekanų žiedas $<\mathbf{Z}_m; +; \cdot>$ yra integralumo sritis, taigi – ir laukas.

▲Jeigu m – sudėtinis skaičius, žiede $<\mathbf{Z}_m; +; \cdot>$ yra nulio daliklių, todėl žiedas negali būti laukas. Jeigu $m = p$ – pirminis, tai kiekvienam nenuliniam elementui $a \in \mathbf{Z}_m$ galime rasti atvirkštinį. Tai gauname iš lyginio $ax \equiv 1 (\mathrm{mod}\ p)$ išsprendžiamumo, nes $a \in \{1, 2, ..., p - 1\}$, t. y. $(a, p) = 1$, ir sakome, kad turime faktoržiedą $<\mathbf{Z}_p; +; \cdot>$. ▼

2.73. Apibrėžimas. Euklido žiedu vadiname integralumo sritį $\mathcal{R}$ kartu su norma ν: $\mathcal{R}^* \to \mathbf{Z}_0$ ($\mathbf{Z}_0 = \mathbf{N} \cup \{0\}$), kuri tenkina šias sąlygas:

1) $\nu(a, b) \geq \nu(a)$ su visais $a, b \in \mathcal{R}^*$;
2) visiems $a, b \in \mathcal{R}^*$ egzistuoja tokie elementai q ir r, kad $a = bq + r$, čia arba $r = 0$, arba $\nu(r) < \nu(b)$.

Vienas iš Euklido žiedo pavyzdžių – sveikųjų skaičių žiedas $\mathbf{Z}$, kai norma $\nu(a) = |a|$, $a \in \mathbf{Z}$. Daugianarių žiedą $\mathbf{F}[x]$ bet kuriame lauke $\mathbf{F}$, kai norma yra daugianario eilė, paverčiame Euklido žiedu. Gauso skaičiai $\mathbf{Z}[i]$, t. y. $\mathbf{Z}[i] = \{m + ni \mid m, n \in \mathbf{Z}, i^2 = -1\}$, taip pat sudaro Euklido žiedą, jeigu norma laikysime $\nu(a + bi) = a^2 + b^2$.

Bet kuriame Euklido žiede kiekvienas idealas yra vyriausiasis. Visi idealo elementai yra kartotiniai elementui a, generuojančiam idealą.

Euklido žiede būtinai yra vienetas (Кострикин, 1977).

2.74. Teorema. Jeigu $a \neq 0$ yra skaidomas komutatyviojo žiedo $\mathcal{R}$ elementas, tai sluoksnių žiedas $\mathcal{R}/<a>$ pagal idealą $<a>$ turi nulio daliklius.

▲ Tarkime, kad $a = b \cdot c$, b ir c – netrivialieji a dalikliai. Išnagrinėkime sluoksnius $\bar{b} = b + <a>$; $\bar{c} = c + <a>$. Jie nėra lygūs $\bar{0}$.

Imkime priešingai, pvz., $\bar{b} = \bar{0}$. Tuomet b turi priklausyti idealui $<a>$, o tai reiškia, kad $b = a \cdot d$ yra a kartotinis.

Įrašę šią reikšmę b į lygybę $a = bc$, gauname

$$a = acd$$

arba, suprastinę iš a, $1 = cd$. Vadinasi, c ir d yra vieneto dalikliai, tačiau remiantis prielaida c yra netrivialusis a daliklis. Taigi $\bar{b} \neq \bar{0}$. Panašiai gautume $\bar{c} \neq \bar{0}$.

Sudauginę sluoksnius $\bar{b}$ ir $\bar{c}$ gauname:

$$\bar{b} \cdot \bar{c} = bc + < a >= a + < a >= \bar{0}.$$

Tokiu būdu įrodėme, kad $\bar{b}$ ir $\bar{c}$ yra nulio dalikliai. ▼

2.75. Teorema. Jeigu p – neskaidomas komutatyviojo žiedo $\mathcal{R}$ su vienetu elementas, tai sluoksnių $\mathcal{R}/<p>$ žiedas (faktoržiedas) pagal idealą $<p>$ yra laukas.

▲ Pirmiausia sluoksnių žiede $\mathcal{R}/<p>$ – faktoržiede – egzistuoja bent vienas sluoksnis, kuris nėra $\bar{0}$. Imkime, pvz., $\bar{1} = 1 + <p>$. Jeigu $\bar{1}$ būtų lygus $\bar{0}$, tai 1 priklausytų idealui $<p>$ ir turėtų dalytis iš p. Bet p neturi vieneto daliklių, nes yra neskaidomas, todėl $\bar{1} \neq \bar{0}$.

Kita vertus, įrodysime, kad bet kuriems dviem sluoksniams $\bar{a} = a + <p> \neq \bar{0}$ ir $\bar{b} = b + <p> \neq \bar{0}$ lygtis $\bar{a}\bar{x} = \bar{b}$ išsprendžiama faktoržiede $\mathcal{R}/<p>$.

Imkime idealą $<a, p>$. Kadangi $\mathcal{R}$ yra vyriausiųjų idealų žiedas, tai $<a, p> = <d>$. Iš čia gauname, kad a ir p dalijasi iš d. Kadangi p neskaidomas, todėl galimi tik du atvejai: arba $d = \varepsilon p$, arba $d = \varepsilon$ (ε – vieneto daliklis).

Pirmoji galimybė netinka, nes gauname, kad a dalijasi iš εp:

$d = \varepsilon p \Rightarrow a = dc = \varepsilon pc$, c – žiedo $\mathcal{R}$ elementas. Tuomet turi būti $a = p(\varepsilon c)$, o tai reiškia, kad a dalijasi iš p. Kitaip sakant, a priklauso idealui $<p>$ ir $\bar{a} = a + <p>$ turi būti $\bar{0}$. Gautume $\bar{a} = \bar{0}$, tačiau tai prieštarauja sąlygai $\bar{a} \neq \bar{0}$.

Tuomet lieka, kad $d = \varepsilon$. Gauname $<a, p> = <\varepsilon> = \mathcal{R}$, t. y. jeigu ε – vieneto daliklis, tai vyriausiasis idealas $<\varepsilon>$ yra visas žiedas $\mathcal{R}$, nes kiekvienas žiedo $\mathcal{R}$ elementas yra ε kartotinis. Vadinasi, idealui $<a, p> = <\varepsilon> = \mathcal{R}$ priklauso visi žiedo $\mathcal{R}$ elementai ir tuo pačiu elementas b.

Idealo $<a, p>$ elementus dar galima užrašyti taip: $ar_1 + pr_2$. Tuomet $b = ar_1 + pr_2$. Remdamiesi žiedų homomorfizmu $\mathcal{R} \simeq \mathcal{R}/<p>$ ir paėmę sluoksnius $\bar{a} = a + <p>$; $\bar{b} = b + <p>$, $\bar{r}_1 = r_1 + <p>$ $\bar{r}_2 = r_2 + <p>$ ir $\bar{0}$, užrašome, kad faktoržiede $\mathcal{R}/<p>$ teisinga lygybė $\bar{b} = \bar{a} \cdot \bar{r}_1 + \bar{0} \cdot \bar{r}_2 = \bar{a}\bar{r}_1$.

Iš čia matome, kad lygtis $\bar{a}\bar{x} = \bar{b}$ išsprendžiama, t. y. $\bar{x} = \bar{r}_1$. ▼

Pavyzdžiui, sveikųjų skaičių žiede **Z** neskaidomi skaičiai ±2; ±3; ±5, ... (t. y. pirminiai su ženklu plius arba minus). Daugianarių žiede **F**[x] bus neskaidomi neredukuojami tame lauke **F** daugianariai (Lidl, et al., 1983).

2.76. Baigtiniai laukai. Tarkime, kad $\mathbf{F}_q$ – baigtinis laukas, turintis q elementų. Žinome, kad egzistuoja homomorfizmas $\mathbf{Z} \to \mathbf{F}_q$, kuris $1 \to 1$, o jo branduolys negali būti lygus nuliui. Tai reiškia, kad branduolys yra vyriausiasis idealas, kurį generuoja pirminis skaičius p, nes turime $\mathbf{Z}/p\mathbf{Z}$ atvaizdį lauke $\mathbf{F}_q$, o $\mathbf{F}_q$ neturi nulio daliklių. Tokiu būdu $\mathbf{F}_q$ charakteristika yra p ir egzistuoja laukas, izomorfiškas $\mathbf{Z}/p\mathbf{Z}$, t. y. $\mathbf{Z}/p\mathbf{Z} = GF(p)$.

Pastebėsime, kad laukas $\mathbf{Z}/p\mathbf{Z}$ neturi kitų automorfizmų, išskyrus tapatųjį. Tikrai, kiekvienas automorfizmas turi atvaizduoti 1 į 1 ir kiekvienas elementas faktoraibėje $\mathbf{Z}/p\mathbf{Z}$ pasilieka vietoje, nes elementas 1 adityviai generuoja $\mathbf{Z}/p\mathbf{Z}$. Toliau $\mathbf{Z}/p\mathbf{Z}$ tapatinsime su $GF(p) \subset \mathbf{F}_q$. Tuomet $\mathbf{F}_q$ galima laikyti vektorine erdve virš $GF(p)$. $\mathbf{F}_q$ dimensija yra baigtinė, nes $\mathbf{F}_q$ — baigtinis laukas. Tarkime, kad $dim\ \mathbf{F}_q = n$ ir $w_1, w_2, ..., w_n$ – bazė $\mathbf{F}_q$. Tuomet bet kuris elementas $\mathbf{F}_q$ gali būti užrašomas suma

$$a_1w_1 + a_2w_2 + ... + a_nw_n,$$

čia $a_i \in \mathbf{Z}/p\mathbf{Z}$. Akivaizdu, kad $q = p^n$ yra lauko $\mathbf{F}_q$ elementų skaičius. Pavyzdžiui, $\mathbf{Z}/2\mathbf{Z} = GF(2) \subset \mathbf{F}_4$, $q = 2^2$, $n = 2$. Kitaip sakant, $dim_{GF(2)}\mathbf{F}_4 = 2$ arba $[\mathbf{F}_4: GF(2)] = 2$.

Multiplikacinė grupė $\mathbf{F}^*_q$ lauke $\mathbf{F}_q$ yra q –1 eilės. Kiekvienas elementas $\alpha \in \mathbf{F}^*_q$ tenkina lygtį $x^{q-1} = 1$, todėl kiekvienas $\mathbf{F}_q$ elementas tenkina ir lygtį

$$f(x) = x^q - x = 0.$$

Tai reiškia, kad daugianaris $f(x)$ turi q skirtingų šaknų lauke $\mathbf{F}_q$ – visus $\mathbf{F}_q$ elementus. Iš čia gauname, kad f skaidomas lauke $\mathbf{F}_q$ tiesiniais daugikliais:

$$x^q - x = \prod_{a \in F_q} (x - a)\ .$$

Tuomet sakome, kad $\mathbf{F}_q$ yra polinomo f skaidinio laukas.

2.77. Teiginys. Jeigu pažymėtume $q - 1 = h$, tuomet visi Galua lauko $GF(p^n)$ elementai būtų daugianario $x^h - 1 = 0$ šaknys, t. y. vieneto šaknys. Kadangi h ir p yra tarpusavyje reliatyviai pirminiai, tai visi nenuliniai lauko elementai yra primityviosios vieneto šaknies laipsniai. Kitaip tariant, Galua lauko multiplikacinė grupė yra ciklinė grupė.

Atskirai pastebėsime, kad duotiesiems p ir n visi laukai, turintys p^n elementų, yra izomorfiški.

2.78. Primityvieji (pirminiai) daugianariai. Lygtis $x^{p^n-1} = 1$ turi $\varphi(p^n - 1)$ primityviųjų šaknų lauke $GF(p^n)$. Visos jos yra skirtingos. Kiekviena šaknis generuoja lauką $GF(p^n)$ virš $\mathbf{Z}_p$ ir yra viena iš neredukuojamo tame lauke n-tosios eilės daugianario šaknų.

2.79. Teorema. Lauke $\mathbf{Z}_p$ yra $\varphi(p^n - 1)/n$ daugianarių, kurių eilė n. Vyriausiasis daugianario koeficientas yra vienetas. Šaknų periodas daugianaryje yra $p^n - 1$ (Lidl, et al., 1983).

3. SUDĖTINGUMO TEORIJA

3.1. SPRENDIMO PRIĖMIMO UŽDAVINIAI

Norėdami supaprastinti tolesnį nagrinėjimą, apsiribosime sprendimo priėmimo uždaviniais, t. y. tais, kurių atsakymas yra TAIP arba NE. Kitaip tariant, šių uždavinių rezultatas yra 1 bitas, kurio reikšmė 0 arba 1. Tokių uždavinių pavyzdžiai:

- Ar fragmentas *P* yra simbolių eilutėje *T*?
- Ar dviejų aibių *S* ir *T* sankirta yra tuščia?
- Ar duotame grafe *G* su svoriais egzistuoja kelias iš viršūnės *A* į viršūnę *B*, kurio svoris ne didesnis nei *k*?

Paskutinis pavyzdys rodo, kad dažnai optimizavimo uždavinį galima performuluoti į sprendimo priėmimo: įvedant parametrą *k* ir klausiant, ar optimali reikšmė yra ne didesnė arba ne mažesnė už *k*. Jei parodysime, kad sprendimo priėmimo uždavinys yra „sudėtingas“, tai reikš, kad atitinkamas optimizavimo uždavinys taip pat „sudėtingas“.

3.2. UŽDAVINIAI IR KALBOS

Sakysime, kad algoritmas *A* atpažįsta duomenų eilutę *x*, jeigu jo rezultatas su pradiniais duomenimis *x* yra TAIP. Taigi sprendimo priėmimo (klasifikavimo) uždavinį galime traktuoti kaip visų tokių duomenų eilučių aibę *L*. Iš tikrųjų raide *L* sprendimo priėmimo (klasifikavimo) uždavinį pažymėjome todėl, kad eilučių aibė dažnai yra vadinama kalba. Galime sakyti, kad algoritmas *A* atpažįsta kalbą *L*, jei kiekvienam $x \in L$ algoritmas *A* pateiks atsakymą TAIP, o kitais atvejais – NE. Laikysime, kad su neteisingais pradiniais duomenimis algoritmas *A* pateiks atsakymą NE. Galima numatyti, kad algoritmas *A*, gavęs kai kuriuos pradinius duomenis, „užsiciklins“ ir niekada neduos atsakymo, bet mes apsiribosime algoritmų, baigiančių darbą po baigtinio skaičiaus žingsnių, nagrinėjimu.

> ***3.1. Pavyzdys.*** Panagrinėkime tokį sprendimo priėmimo uždavinį: ar duotas natūralusis skaičius *n* yra pirminis? Abėcėlė Σ, kuria galima užkoduoti natūralųjį skaičių, gali būti {0, 1, 2, ..., 9} arba {0, 1}, t. y. pateikta dešimtaine arba dvejetaine išraiška. Visų skaičių eilutės, sudarytos iš abėcėlės Σ, žymimos Σ^*. Tada visų natūraliųjų skaičių aibė yra Σ^*, visų pirminių skaičių – $L \subset \Sigma^*$. Vadinasi, formaliai nagrinėjamą uždavinį galima apibrėžti taip: duotas $n \in \Sigma^*$. Ar $n \in L$?
> Pavyzdžiui, pasirinkę žodį $x = 31 \in \Sigma^*$, sudarytą iš simbolių $3 \in \Sigma$ ir $1 \in \Sigma$, gausime $x = 31 \in L$. Jei $x_1 = 99 \in \Sigma^*$, akivaizdu, jog $x_1 \notin L$.

Jei nebus nurodyta kitaip, toliau nagrinėsime dvejetainę abėcėlę $\Sigma = \{0, 1\}$.

3.3. SUDĖTINGUMO KLASĖ *P*

Sudėtingumo klasė *P* (polinominė) yra aibė visų sprendimo priėmimo uždavinių ar kalbų *L* atpažinimo uždavinių, kuriems gauti atsakymą TAIP blogiausiu atveju reikia polinominio vykdymo laiko, priklausančio nuo įvesties duomenų ilgio. T. y. egzistuoja toks algoritmas *A*, kad jei $x \in L$, tai *A* pateikia rezultatą TAIP per laiką $p(|x|)$, kur $|x|$ – *x* įvesties duomenų ilgis (pavyzdžiui, išreikštas bitais), o *p* – koks nors polinomas.

Pastebėkime, kad *P* apibrėžime nieko neužsimenama apie vykdymo laiką duomenims, kuriems algoritmas pateikia rezultatą NE. Tokios simbolių eilutės sudaro kalbos *L* papildinį – aibę visų dvejetainių eilučių, nepriklausančių *L*. Turėdami polinominį algoritmą *A*, atpažįstantį kalbą *L*, nesunkiai galime sudaryti

polinominį algoritmą *B*, atpažįstantį *L* papildinį: pradiniams duomenims *x* algoritmas *B* vykdo algoritmą *A* $p(|x|)$ žingsnių. Jei algoritmas *A* pateikia atsakymą TAIP, algoritmo *B* atsakymas – NE, jei algoritmas *A* pateikia atsakymą NE arba nepateikia jokio, tada algoritmo *B* atsakymas yra TAIP. Taigi, jei kalba *L*, atitinkanti kažkokį sprendimo priėmimo uždavinį, priklauso *P*, tai ir *L* papildinys priklauso *P*.

3.4. SUDĖTINGUMO KLASĖ *NP*

Sudėtingumo klasė *NP* (nedeterministiškai polinominė) apima sudėtingumo klasę *P* ir papildomai įtraukia kalbas, galinčias nepriklausyti *P*. *NP* uždavinių algoritmams leidžiama atlikti papildomą pasirinkimo operaciją *pasirinkti*(*b*), kuri nedeterminuotuoju būdu pasirenka reikšmę ir priskiria ją *b*.

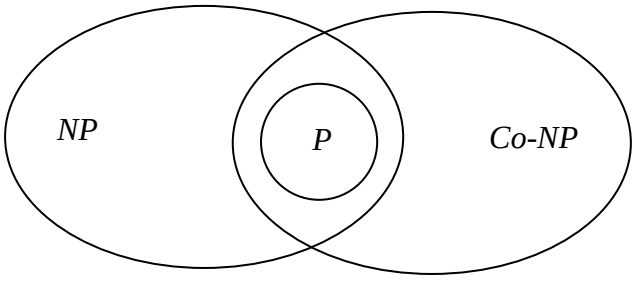
Kai algoritmas *A* naudoja primityvią operaciją *pasirinkti*, *A* vadinamas nedeterministiniu algoritmu. Sakysime, kad algoritmas *A* nedeterministiškai atpažįsta duomenų eilutę *x*, jei egzistuoja operacijos *pasirinkti* kreipinių, kuriuos *A* gali atlikti su pradiniais duomenimis *x*, o rezultatų aibė yra tokia, kad *A* pateiks rezultatą TAIP. Kitais žodžiais, jei nagrinėsime visus galimus operacijos *pasirinkti* kreipinių rezultatus ir pasirinksime tik tuos, pagal kuriuos atpažįstama (jei tokių yra). Pastebėkime, kad tai nėra tas pats, kas atsitiktinis pasirinkimas. Galime įsivaizduoti, kad yra orakulas (išminčius), kuris mums nežinomu (nedeterminuotuoju) būdu atsako į klausimą.

Sudėtingumo klasė *NP* yra aibė sprendimo priėmimo uždavinių arba kalbų *L*, kurios gali būti nedeterministiškai atpažįstamos per polinominį laiką, t. y. egzistuoja toks nedeterministinis algoritmas *A*, kad jei $x \in L$, tai algoritme *A* yra tokia operacijos *pasirinkti* kreipinių rezultatų aibė, kad *A* duoda atsakymą TAIP per laiką $p(|x|)$, kur $|x|$ – *x* ilgis, o *p* – polinomas. Kitais žodžiais, jei polinominį skaičių kartų kreipsimės į orakulą ir gausime tinkamus atsakymus, tai algoritmas *A* pateiks atsakymą TAIP per polinominį laiką.

Pastebėkime, kad *NP* apibrėžime nieko nerašoma apie sprendimo laiką duomenims, kuriems algoritmas pateiks rezultatą NE. Iš tikrųjų, siekiant gauti rezultatą NE, algoritmas *A* gali atlikti gerokai daugiau nei $p(|x|)$ žingsnių. Negana to, kadangi nedeterministinis pripažinimas gali turėti polinominį skaičių operacijos *pasirinkti* kreipinių, tai jei kalba *L* priklauso *NP*, *L* papildinys nebūtinai priklauso *NP*. Iš tikrųjų yra sudėtingumo klasė co-*NP*, sudaryta iš visų kalbų, kurių papildiniai priklauso klasei *NP*. Daugelis mokslininkų tiki, kad co-NP ≠ NP, tačiau formaliai tai neįrodyta.

3.5. *P* = *NP* PROBLEMA

Formaliai neįrodyta, ar *P* = *NP*. Net nežinoma, ar *P* = *NP* ∩ co-*NP*. Manoma, kad *P* skiriasi ir nuo *NP*, ir nuo co-*NP*, ir nuo jų sankirtos (3.1 pav.). Toliau aptarsime *NP* uždavinius, kurie, manoma, nepriklauso *P*, t. y. nežinomas polinominis jų sprendimo algoritmas.



3.1 pav. Sudėtingumo klasių diagrama

Nedeterministinio atpažinimo sąvoka yra gana keista. Kompiuteris negali efektyviai įvykdyti nedeterministinio algoritmo, esant dideliam operacijos *pasirinkti* kreipinių skaičiui. Žinoma, nedeterministinį algoritmą galima sudaryti bandant visus variantus, kuriuos gali pateikti jame esantys *pasirinkti* kreipiniai. Tačiau šis modeliavimas reikalaus eksponentinio laiko bet kuriam nedeterministiniam algoritmui, turinčiam

n^{ε} kreipinių *pasirinkti* bet kokiai konstantai $\varepsilon>0$. Iš tikrųjų yra šimtai *NP* sudėtingumo klasės uždavinių, kuriems polinominis algoritmas yra nežinomas, be to, tikima, kad toks apskritai neegzistuoja.

3.6. POLINOMINIS REDUKAVIMAS

Sakoma, kad kalba *L*, apibrėžianti sprendimo priėmimo uždavinį, yra polinomiškai redukuojama į kalbą *M*, jei egzistuoja polinominio sudėtingumo funkcija *f*, kuri *L* pradinius duomenis *x* transformuoja į *M* pradinius duomenis $f(x)$ taip, kad $x \in L$ tada ir tik tada, kai $f(x) \in M$. Galima žymėti $L \leq_p M$.

M yra *NP* sudėtingas, jei kiekvienam $L \in NP$, $L \leq_p M$. Jei pats $M \in NP$, tai *M* yra *NP* pilnasis.

Jei kas nors įrodytų, kad *NP* pilnajam uždaviniui egzistuoja polinominis algoritmas, tai automatiškai reikštų, kad visa *NP* klasė išsprendžiama per polinominį laiką, t. y. $P = NP$.

Gali pasirodyti, kad *NP* pilnumo apibrėžimas yra pernelyg griežtas, tačiau žemiau pateikta teorema rodo, kad bent vienas *NP* pilnasis uždavinys iš tiesų egzistuoja.

3.7. KUKO TEOREMA

3.2. Teorema (Kuko). Loginės funkcijos įvykdomumo uždavinys yra *NP* pilnasis (angl. *Satisfiability problem* – SAT).

Įrodymo paaiškinimas. Pirmiausia įrodysime, kad SAT priklauso *NP* klasei. Tam reikia pateikti nedeterminuotąją Tiuringo mašiną (NTM), atpažįstančią SAT kalbą. Paprasčiausia NTM turi veikti taip:

a) atspėti visų kintamųjų reikšmes;

b) šias reikšmes įrašyti į formulę;

c) jei formulė įgyja reikšmę 1, pereiti į galutinę būseną.

Matome, kad ši NTM atpažins formulę *f* tada ir tik tada, jei egzistuos kintamųjų reikšmių rinkinys, įvykdantis formulę *f*.

Nagrinėkime bet kurią kalbą $L \in NP$. Reikia įrodyti, kad egzistuoja polinominis redukuojamumas iš *L* į SAT, t. y. $L \leq_p$ SAT. Tai reiškia, kad pagal konkretų kalbos *L* egzempliorių galima sudaryti SAT uždavinio su tuo pačiu atsakymu formulę. Kadangi $L \in NP$, tai egzistuoja NTM *M*, kuri atpažįsta kalbą *L*, atlikdama ne daugiau kaip $p(n)$ perėjimų, kur *p* – tam tikras polinomas.

Ieškomos formulės struktūra:

(*M* pradeda darbą, įvestyje turėdama duotą kalbos egzempliorių) $\wedge$

$\wedge$ (*M* pirmasis žingsnis įvyksta pagal taisykles) $\wedge$

$\wedge$ ... $\wedge$

$\wedge$ (*M* $p(n)$-tasis žingsnis įvyksta pagal taisykles) $\wedge$

$\wedge$ (*M* atėjo į galutinę būseną).

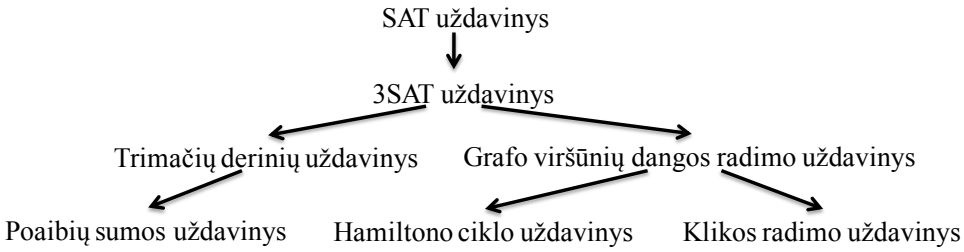
Detalesnis formulės struktūros nagrinėjimas yra gremėzdiškas, jį galima rasti literatūroje (Гэри, et al., 1982). Mes apsiribosime tik pateiktu formulės karkasu.

Tarkime, kad nagrinėjamas kalbos egzempliorius tikrai priklauso *L*. Vadinasi, Tiuringo mašina M, gavusi įvestyje šį egzempliorių, gali apibrėžtam perėjimų pasirinkimui pereiti į galutinę būseną. Tada formulė, aprašanti *M* darbo procesą, turi būti įvykdoma, nes visi teiginiai apie *M* darbą, užkoduoti formulėje, yra teisingi.

Ir atvirkščiai, jei formulė įvykdoma, tai, analizuojant kintamųjų reikšmes kiekvienuose skliaustuose, galima suprasti, koks perėjimas buvo pasirinktas kiekviename žingsnyje, ir įsitikinti, kad Tiuringo mašina atliko tik leistinus perėjimus ir atėjo į galutinę būseną. Vadinasi, įvesties egzempliorius priklauso kalbai *L*.

3.8. KITŲ *NP* PILNŲJŲ UŽDAVINIŲ PAVYZDŽIAI

Skiriami šeši pagrindiniai *NP* pilnieji uždaviniai, kurių redukuojamumo sąsajos (Гэри, et al., 1982) pateiktos 3.2 pav.



3.2 pav. Šešių pagrindinių *NP* pilnųjų uždavinių redukuojamumo diagrama

Loginės formulės įvykdomumo (SAT) uždavinys formuluojamas taip: formulėje yra m disjunktų, kuriuose yra n loginių kintamųjų. Ar egzistuoja toks Būlio kintamųjų reikšmių rinkinys, su kuriuo visi m disjunktai yra teisingi?

3SAT uždavinys – tai toks SAT uždavinys, kai kiekviename disjunkte yra ne daugiau kaip trys kintamieji.

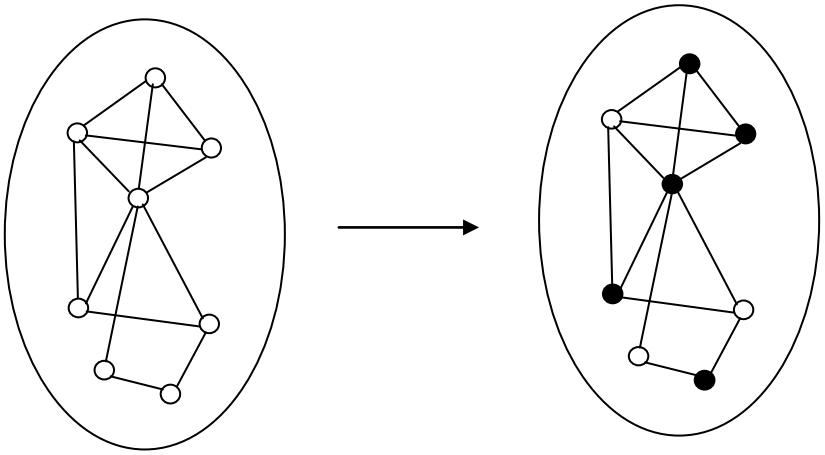
Trimačių derinių uždavinys formuluojamas taip: duota aibė $M = W \times X \times Y$, kur W, X, Y – nesikertančios aibės, turinčios vienodą elementų skaičių q. Ar teisinga, kad M aibėje yra trimačių derinių, t. y. toks poaibis $M' \subseteq M$, kad $|M'| = q$, ir jokie du skirtingi M' elementai neturi nė vienos lygios koordinatės?

> ***3.3. Pavyzdys.*** Pavyzdžiui, duotos aibės $W = \{a, b, c\}$, $X = \{d, e, f\}$, $Y = \{j, h, i\}$ ir $M = \{(a, d, i), (a, e, i), (a, f, j), (b, e, i), (b, f, j), (c, e, h), (c, d, i)\}$. Šis pavyzdys turi sprendinį $M' = \{(a, d, i), (b, f, j), (c, e, h)\} \subseteq M$.

Poaibių sumos uždavinys formuluojamas taip: duota baigtinė aibė A ir svoriai $s(a) \in Z^+$. Ar egzistuoja toks poaibis $A' \subseteq A$, kad

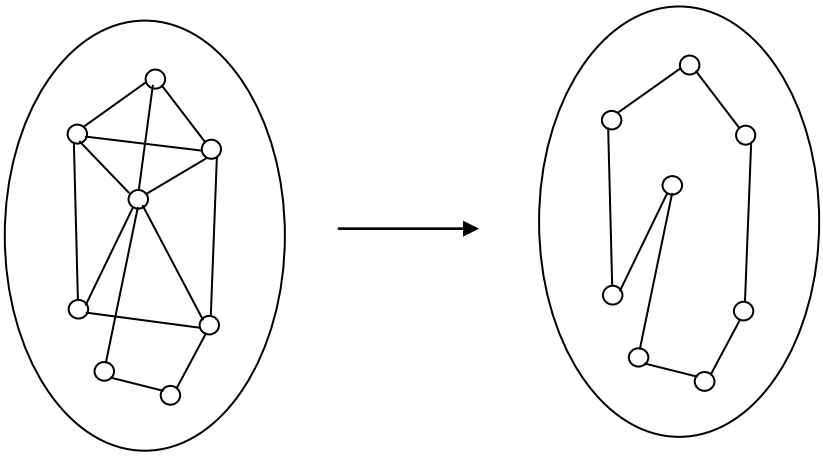
$$\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a) ?$$

Grafo viršūnių dangos uždavinys formuluojamas taip: duotas grafas $G = (V, E)$ ir teigiamas sveikasis skaičius k, $k \le |V|$. Ar egzistuoja grafe G viršūnių danga, kurią sudaro ne daugiau kaip k viršūnių, t. y. toks poaibis $V' \subseteq V$, kad $|V'| \le k$, ir kiekvienos briaunos $\{u, v\} \in E$ bent viena iš viršūnių u arba v priklauso V'? (3.3 pav.)



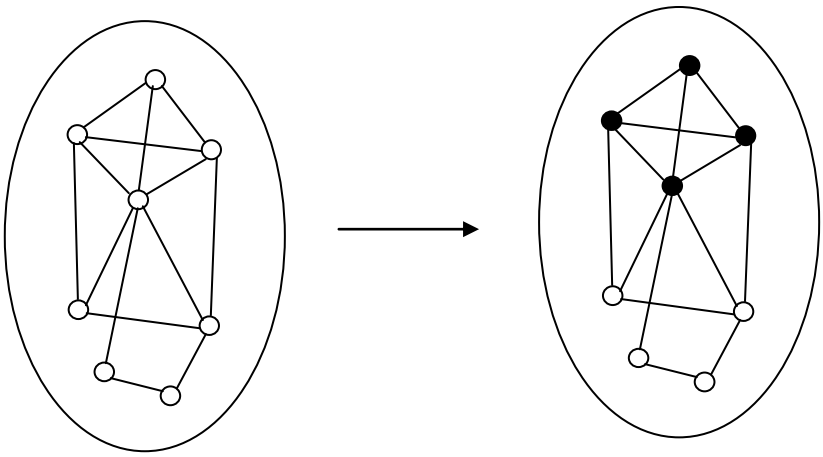
3.3 pav. Grafo viršūnių dangos pavyzdys

Hamiltono ciklo uždavinys: ar neorientuotame grafe egzistuoja ciklas, kurį vykdant kiekviena viršūnė aplankoma lygiai po kartą? (3.4 pav.)



3.4 pav. Hamiltono ciklo pavyzdys

Klikos, arba pilnojo pografio radimo, uždavinys (3.5 pav.): ar egzistuoja grafe pilnasis pografis, turintis ne daugiau kaip k viršūnių?



3.5 pav. Klikos (pilnojo pografio) pavyzdys

3.4. Teorema. 3SAT uždavinys yra *NP* pilnasis.

▲ 3SAT uždavinys priklauso *NP* klasei. Pakanka atspėti loginius kintamuosius, kuriems priskirta reikšmė 1, ir per polinominį laiką patikrinti, ar įvykdoma formulė. Akivaizdu, kad konjunktyvinės normalinės formos reikšmė apskaičiuojama per polinominį laiką.

Parodysime, kaip SAT uždavinys redukuojamas į 3SAT uždavinį.

Kiekvieną k kintamųjų disjunktą galime modeliuoti ne daugiau kaip $3k$ naujų disjunktų, kurių kiekviename yra ne daugiau kaip po tris kintamuosius. Disjunktas

$$\left(p_{i_1} \vee p_{i_2} \vee p_{i_3} \vee \ldots \vee p_{i_k}\right)$$

ekvivalentus disjunktams:

$$\left(q_1 \vee p_{i_3} \vee \ldots \vee p_{i_k}\right)$$

$$q_1 \to (p_{i_1} \vee p_{i_2})$$
$$p_{i_1} \to q_1$$
$$p_{i_2} \to q_1$$

Paskutiniai trys disjunktai nusako ekvivalentumą $q_1 \leftrightarrow (p_{i_1} \vee p_{i_2})$. Pradinis *k* kintamųjų disjunktas ekvivalentus keturiems naujiems disjunktams su *k*-1 kintamuoju. Kintamąjį q_2 apibrėžiame taip: $q_2 \leftrightarrow (q_1 \vee p_{i_3})$. Pradinis disjunktas redukuojamas į $(q_2 \vee p_{i_4} \vee p_{i_5} \vee \ldots \vee p_{i_k})$. Kartodami tuos pačius veiksmus, *j*-tajame etape apibrėžiame q_j: $q_j \leftrightarrow (q_{j-1} \vee p_{i_{j+1}})$. Taigi $(q_j \vee p_{i_{j+2}} \vee \ldots \vee p_{i_k})$ bus redukuotasis disjunktas (j = 3, …, k-3). Paskutinis tokiu būdu gautas disjunktas bus $(q_{k-3} \vee p_{i_{k-1}} \vee p_{i_k})$. Kiekvieno etapo metu gautų disjunktų aibė logiškai ekvivalenti pirmesnio etapo disjunktams. Gavome, kad SAT uždavinys redukuojamas į 3SAT. ▼

3.5. Teorema. Trimačių derinių uždavinys yra *NP* pilnasis.

Teorema įrodoma pasitelkus redukavimą 3SAT $\leq_p$ TRIMAČIAI DERINIAI. Įrodymą galima rasti literatūroje (Гэри, et al., 1982).

3.6. Teorema. Viršūnių dangos uždavinys yra *NP* pilnasis.

Teorema įrodoma pasitelkus redukavimą 3SAT $\leq_p$ VIRŠŪNIŲ DANGA. Įrodymą galima rasti literatūroje (Гэри, et al., 1982).

3.7. Teorema. Hamiltono ciklo uždavinys yra *NP* pilnasis.

Teorema įrodoma pasitelkus redukavimą VIRŠŪNIŲ DANGA $\leq_p$ HAMILTONO CIKLAS. Įrodymą galima rasti literatūroje (Гэри, et al., 1982).

3.8. Teorema. Klikos radimo uždavinys yra *NP* pilnasis.

▲ Klikos radimo uždavinys priklauso *NP* klasei. Pakanka atspėti ne mažesnio kaip *k* rango kliką ir per polinominį laiką patikrinti, ar tas poaibis yra pilnasis.

Įrodysime, kad 3SAT $\leq_p$ *KLIKA*. Tam per polinominį laiką reikia sukurti tokį *G* grafą, kad duoti disjunktai būtų įvykdomi tada ir tik tada, kai *G* grafe yra *k* rango klika. Tarkime, kad yra *n* loginių kintamųjų. Grafo viršūnėms priskirsime etiketes (i_1, i_2, ..., i_n), kur $i_j \in \{0, 1, \#\}$. Etiketę suprasime kaip dalinę disjunktų interpretaciją. Pavyzdžiui, turint (0, 1, #, 1, #), interpretacija bus $x_1 = 0$, $x_2 = 1$, $x_4 = 1$, o x_3 ir x_5 – neapibrėžti. Kiekvienam disjunktui C_i su $l \leq 3$ kintamaisiais sudarome ne mažiau kaip 7 viršūnes $v_{i,j}$. Jos atstovaus visoms dalinėms C_i interpretacijoms. Iš visų galimų 8 interpretacijų (2^3), kai disjunkte yra trys kintamieji, tik su vienintele C_i jis neįvykdomas.

Dabar nagrinėsime simetrinį grafą, kurio sritis

$$D = \{ v_{i,j} \mid i = 1,2, \ldots, m, 1 \leq j \leq 7\}.$$

Jei dvi dalinės interpretacijos $v_{i,j}$, $v_{i',j'}$ suderinamos, t. y. apibrėžtų abiejose interpretacijose loginių kintamųjų reikšmės (0 arba 1) yra tos pačios, tai tas viršūnes sujungiame briauna ($v_{i,j}$, $v_{i',j'}$).

Tarkime, kad *G* – grafas, kuris aprašomas *m* disjunktų, turinčių *n* kintamųjų. Jame yra ne daugiau kaip $O(7m)$ viršūnių. Tarkime, kad $k = m$. Parodysime, kad aibė (G, k) nusako norimą redukciją. Jei disjunktų aibė įvykdoma, tai atsiras interpretacija σ, su kuria įvykdomi visi disjunktai. Kiekvienam $i \leq m$ interpretacija σ įvykdo C_i ir tuo pačiu yra bent vienas modelis $v_{i,j}$. Visos dalinės interpretacijos suderinamos ir sujungtos briaunomis. Taigi egzistuoja *k* rango klika.

Priešingas teiginys. Jei egzistuoja *k* rango klika, tai turime *k* suderinamų interpretacijų. Be to, egzistuoja jas pratęsianti interpretacija *U*, priskirianti reikšmes 0 arba 1 neapibrėžtiems kintamiesiems. Interpretacija *U* tenkina visus disjunktus. Taigi disjunktų aibė yra įvykdoma. ▼

3.9. Teorema. Poaibių sumos uždavinys yra *NP* pilnasis.

▲ Nesunku pastebėti, kad POAIBIŲ SUMA $\in$ *NP*. Nedeterministinis algoritmas turi atspėti tik poaibį $A' \subseteq A$ ir per polinominį laiką patikrinti, ar poaibio A' elementų svorių suma lygi poaibio $A \setminus A'$ elementų svorių sumai.

Įrodysime, kad TRIMAČIAI DERINIAI $\leq_p$ POAIBIŲ SUMA. Tarkime, kad aibės *W*, *X*, *Y*, kai $|W| = |X| = |Y| = q$, ir aibė $M \subseteq W \times X \times Y$ sudaro bet kurį trimačių derinių uždavinio egzempliorių. Šių aibių elementus pažymėsime taip:

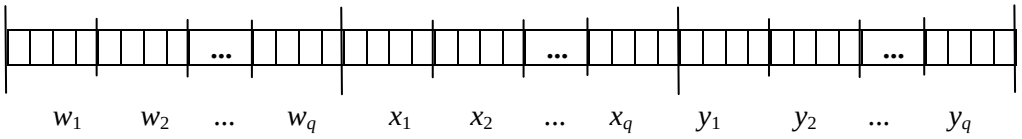
$$W = \{w_1, w_2, \ldots, w_q\},\ X = \{x_1, x_2, \ldots, x_q\},\ Y = \{y_1, y_2, ..., y_q\} \text{ ir } M = \{m_1, m_2, ..., m_k\},$$

kur $k = |M|$. Reikia sudaryti aibę *A* ir visiems $a \in A$ priskirti svorius $s(a) \in Z^+$ taip, kad aibė *A* turėtų poaibį *A'*, kuriam būtų teisinga lygybė

$$\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)$$

tik tuo atveju, kai aibėje *M* bus trimačiai deriniai.

Aibėje *A* bus $k + 2$ elementai ir ji bus sudaryta iš dviejų etapų. Aibės *A* pirmieji *k* elementai bus $\{a_i : 1 \leq i \leq k\}$, kur a_i asocijuojasi su trejetu $m_i \in M$. Elemento a_i svoris $s(a_i)$ apibrėžiamas skaičiaus $s(a_i)$ dvejetainiu pavidalu. Žodis, sudarytas iš 0 ir 1, atitinkantis $s(a_i)$, suskaidomas į $3q$ dalių po $p = \lceil \log_2(k+1) \rceil$ bitų. Kiekviena dalis pažymima tam tikru elementu iš aibės $W \cup X \cup Y$ (3.6 pav.).



3.6 pav. Skaičiaus *s*(*a*) dvejetainis pavidalas

Skaičiaus $s(a_i)$ dvejetainis pavidalas priklauso nuo atitinkamo trejeto $m_i = (w_{f(i)}, x_{g(i)}, y_{h(i)}) \in M$, kur *f*, *g*, *h* – funkcijos, apibrėžiančios trejeto m_i komponenčių indeksus. Skaičiaus $s(a_i)$ dešinieji galai, atitinkantys skaičius $w_{f(i)}$, $x_{g(i)}$, $y_{h(i)}$, pažymėti 1, o visi kiti skaitmenys yra 0. Kitaip tariant,

$$s(a_i) = 2^{p(3q-f(i))} + 2^{p(2q-g(i))} + 2^{p(q-h(i))}.$$

Kadangi kiekvieno skaičiaus $s(a_i)$ dvejetainis pavidalas turi ne daugiau kaip $3pq$ skaitmenų, akivaizdu, kad $s(a_i)$ bus sudaryti per polinominį laiką kiekvienam duotam trimačių derinių egzemplioriui.

Šiame redukavimo etape svarbu pastebėti, kad, sumuojant visus vienos zonos elementus, rezultatas neviršys $2^p - 1$. Vadinasi, sumuojant *s*(*a*) tam tikrame poaibyje $A' \subseteq \{a_i : 1 \leq i \leq k\}$ nebus $\sum_{a \in A'} s(a)$ pernašos į kitą zoną. Iš čia išplaukia, kad jei $B = \sum_{j=0}^{3q-1} 2^{pj}$, tai bet kuriam poaibiui $A' \subseteq \{a_i : 1 \leq i \leq k\}$ lygybė $\sum_{a \in A'} s(a) = B$ galios tada ir tik tada, kai $M' = \{m_i : a_i \in A'\}$ yra trimačiai deriniai aibėje *M*.

Paskutiniame redukavimo etape bus sudaromi likę du aibės *A* elementai. Jie apibrėžiami kaip elementai b_1 ir b_2, kurių svoriai

$$s(b_1) = 2\left(\sum_{i=1}^{k} s(a_i)\right) - B \text{ ir } s(b_1) = \left(\sum_{i=1}^{k} s(a_i)\right) + B.$$

Šių skaičių dvejetainių pavidalų ilgiai neviršija $3pq + 1$ ir gali būti suformuoti per polinominį laiką.

Tarkime, kad egzistuoja toks poaibis $A' \subseteq A$, kad $\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)$. Tada kiekviena iš sumų turi būti lygi $2\left(\sum_{i=1}^{k} s(a_i)\right)$. Be to, vienoje iš aibių *A'* arba $A \setminus A'$ yra elementas b_1 ir nėra elemento b_2. Iš čia išplaukia, kad likę šios aibės elementai priklauso aibei $\{a_i : 1 \leq i \leq k\}$, jų dydžių suma lygi *B* ir jie sudaro poaibį, atitinkantį tam tikrą trimatį derinį $M' \subseteq M$. Jei duotas trimatis derinys $M' \subseteq M$, tai

aibė $\{ b_1\} \cup \{ a_i : m_i \in M'\}$ sudaro ieškomą poaibį A' kiekvienam individualiam trimačių derinių uždavinio egzemplioriui. ▼

Šiame skyrelyje apžvelgėme sudėtingumo teorijoje nagrinėjamas uždavinių klases. Daugiausia dėmesio skyrėme *NP* pilniesiems uždaviniams bei redukuojamumo sąvokai, pasitelkiamai *NP* pilnumui įrodyti. Redukavimo pavyzdžiai pateikti įrodytose teoremose, o ten, kur redukavimo aprašymas yra ilgas ir sudėtingas, pateiktos nuorodos į literatūrą.

4. VIENKRYPČIŲ FUNKCIJŲ APVERTIMU PAGRĮSTA KRIPTOANALIZĖ

Kriptografijoje dažniausiai naudojamos vienkryptės funkcijos (VKF), pagrįstos daugybe algoritmiškai sunkiai sprendžiamų uždavinių. Jau minėjome, kad tokie uždaviniai trumpiau vadinami sudėtingais, nes manoma, kad jie gali priklausyti tam tikroms uždavinių sudėtingumo klasėms.

Dažniausiai kriptografijoje naudojami sudėtingi uždaviniai pateikti 4.1 lentelėje.

4.1 lentelė

Uždavinio pavadinimas	Sąlyginis žymėjimas	Duota	Rasti
Sveikųjų skaičių faktorizacijos uždavinys	*FACTORING*	n	$p_i, e_i : n = p_1^{e_1} \cdot \ldots \cdot p_k^{e_k}$, kur p_i – poromis pirminiai skaičiai, $e_i \geq 1$
RSA uždavinys	*RSA*	$n = pq$, e : DBD(e, (p-1)(q-1))=1, $c \in Z$	$M : m^e \equiv c \bmod m$
Kvadratinių liekanų uždavinys	*QRP*	n – nelyginis sudėtinis sveikasis skaičius, $a \in Z : \left(\frac{a}{n}\right) = 1$, kur $\left(\frac{a}{n}\right)$ – Jakobio simbolis (žr. pastabą)	$a \in QR_n$?
Kvadratinės šaknies traukimo moduliu n uždavinys	*SQROOT*	n – sudėtinis sveikasis skaičius, $a \in QR_n$	$x : x^2 \equiv a \bmod n$
Diskretinio logaritmo uždavinys	*DLP*	p – pirminis skaičius, α – Z_p^* generatorius, $\beta \in Z_p^*$	$x : 0 \leq x \leq p$-2, $\alpha^x \equiv \beta \bmod p$
Apibendrintas diskretinio logaritmo uždavinys	*GDLP*	G – baigtinė ciklinė grupė, $\lvert G \rvert = n$, α – G generatorius, $\beta \in Z$	$x : 0 \leq x \leq n$-1, $\alpha^x \equiv \beta$
Diffie'io ir Hellmano uždavinys	*DHP*	p – pirminis skaičius, α – Z_p^* generatorius, $\alpha^a \bmod p$, $\alpha^b \bmod p$	$\alpha^{ab} \bmod p$
Apibendrintas Diffie'io ir Hellmano uždavinys	*GDHP*	G – baigtinė ciklinė grupė, α – G generatorius, α^a, α^b	α^{ab}
Diffie'io ir Hellmano sprendimo uždavinys	*DDHP*	p – pirminis skaičius, α – Z_p^* generatorius, $\alpha^a \bmod p$, $\alpha^b \bmod p$, $\alpha^c \bmod p$	$\alpha^{ab} \equiv \alpha^c \bmod p$?
Sustiprintas RSA uždavinys	*SRSA*	$n = pq$, $z \in Z_n^*$	$r = r(z) > 1$, r – atsitiktinis skaičius, $y \in Z_n^* : y^r = z$

Kuprinės uždavinys	*SUBSET*	$\{a_1, a_2, ..., a_n\}$ – teigiamų sveikųjų skaičių aibė, s – teigiamas sveikasis skaičius	$\sum_{j\in\{1,\ldots,n\}} a_j = s$?

4.1. Pastaba. Jakobio simbolio reikšmė: jei $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k}$ yra sudėtinis skaičius, tai

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1}\left(\frac{a}{p_2}\right)^{\alpha_2}\ldots\left(\frac{a}{p_k}\right)^{\alpha_k};$$

čia $\left(\frac{a}{p}\right)$ yra Ležandro simbolis:

$$\left(\frac{a}{p}\right) = \begin{cases} 0, \text{ jei } p \text{ yra } a \text{ daliklis,} \\ 1, \text{ jei egzistuoja toks } k, \text{ kad } k^2 \equiv a \bmod p, \\ -1, \text{ jei neegzistuo ja toks } k, \text{ kad } k^2 \equiv a \bmod p. \end{cases}$$

Panagrinėsime keletą 4.1 lentelėje nurodytų uždavinių.

4.1. SVEIKŲJŲ SKAIČIŲ FAKTORIZAVIMO ALGORITMAI

Skaičiams faktorizuoti gali būti naudojami šie algoritmai: bandomosios dalybos, Pollardo *ro*, Pollardo *p*-1, Viljamso *p*+1, Lenstros elipsinių kreivių faktorizacijos, specialaus skaičių lauko rėčio.

Bandomosios dalybos algoritmo idėja yra paprasčiausia: pasitelkus dalybos operaciją, reikia išbandyti visus pirminius skaičius iki $\sqrt{n}$. Skaičiavimo apimčiai sumažinti gali būti tikrinami tik visi nelyginiai skaičiai.

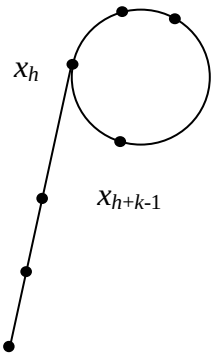
4.1 algoritmas. Bandomosios dalybos algoritmas

(įvestis: sveikasis skaičius n)

$x \leftarrow 3$.
Jei n mod $x = 0$, grąžinti 2.
Kol $x < \sqrt{n}$, vykdyti:
    jei n mod $x = 0$, grąžinti x
    $x \leftarrow x + 2$
Grąžinti „Pirminis skaičius“.

(išvestis: daugiklis x arba skaičius yra pirminis)

Pollardo *ro* algoritmo idėja. Tarkime, duota funkcija $f : Z_n \to Z_n$. Pasirinkime $x_1 \in Z_n$ ir sudarykime seką $\{x_i\}$, kuri apskaičiuojama rekursiniu būdu pagal formulę $x_{i+1} = f(x_i)$. Kadangi Z_n – baigtinė aibė, tai skaičiai x_i negali būti visi skirtingi. Todėl atsiras tokių natūraliųjų skaičių h ir k, kad skaičiai $x_1, \ldots, x_h, x_{h+1}, \ldots, x_{h+k-1}$ bus skirtingi, o $x_{h+k} = x_h$. Toliau naudojant funkciją f, elementai $x_h, x_{h+1}, \ldots, x_{h+k+1}$ kartosis, t. y. seka $\{x_i\}$ yra periodinė, kurios periodas – k, pradedant $i = h$. Skaičius h vadinamas įvesties į periodą indeksu. Jei $h = 1$, seka vadinama periodine, o jei $h > 1$ – beveik periodine. Beveik periodinė seka schemoje vaizduojama taškais, esančiais ant kreivės, primenančios graikišką raidę ρ (4.1 pav.). Iš čia ir kilęs šio metodo pavadinimas.



4.1 pav. Beveik periodinės sekos schema

4.2. Teorema. Atsitiktinai pasirinktiems atvaizdžiams $f : Z_n \to Z_n$ ir pradiniam nariui $x_1 \in Z_n$ numatomas vidutinis iteracijų skaičius iki pirmojo pasikartojimo sekoje $\{x_i\}$, $x_{i+1} = f(x_i)$, įvertinamas kaip $O(\sqrt{n})$.

▲ Teoremoje atsitiktinai pasirenkamos funkcijos $f : Z_n \to Z_n$ ir pradinis narys x_1. Tokiu būdu elementariųjų įvykių aibė Ω – aibė porų (f, x_1). Iš viso turime n^n atvaizdžių $f : Z_n \to Z_n$ ir n elementų, kuriuos galime imti kaip $x_1 \in Z_n$. Vadinasi, aibė Ω yra sudaryta iš n^{n+1} elementų.

Tarkime, $\{x_i\}$ yra seka su pradiniu nariu x_1, generuojama funkcijos f, t. y. $x_{i+1} = f(x_i)$, k – sekos periodas, h – indeksas, $s = h+k-1$ – iteracijų skaičius iki pirmojo pasikartojimo, $1 \le s, h, k \le n$. Erdvėje Ω yra įvykis S. Teoremoje gaunamas šio atsitiktinio įvykio S matematinio vidurkio įvertis.

Rasime tikimybę $p(h, k)$, kad seka $\{x_i\}$, generuojama atsitiktinės poros (f, x_1), turi indeksą h ir periodą k. Tam reikia nustatyti tokių porų skaičių. Jei (f, x_1) yra tokia pora, tai sekos pradinė atkarpa $(x_1, x_2, ..., x_{h+k-1})$ yra sudaryta iš skirtingų aibės Z_n elementų. Parametrai h ir k vienareikšmiškai apibrėžia f poveikį aibėje $(x_1, x_2, ..., x_{h+k-1})$. Papildinyje $Z_n \setminus (x_1, x_2, ..., x_{h+k-1})$ funkcija f gali veikti laisvai. Tokių funkcijų f kiekvienam aibės $\{x_1, x_2, ..., x_{h+k-1}\}$ elementui yra $n^{n-(h+k-1)} = n^{n-s}$. Šią aibę vadinsime atkarpa. Atkarpų $(x_1, x_2, ..., x_{h+k-1})$, sudarytų iš skirtingų elementų, skaičius lygus gretinių iš n po $s = h+k-1$ skaičiui ir yra $n(n-1) ... (n-s+1)$. Vadinasi, mus dominančių elementariųjų įvykių (t. y. porų (f, x_1)) skaičius lygus $n^{n-s}\, n(n-1) ... (n-s+1)$. Iš čia gauname

$$p(h,k) = \frac{n^{n-s+1}(n-1)\ldots(n-s+1)}{n^{n+1}} = \frac{1}{n}\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\ldots\left(1-\frac{s-1}{n}\right).$$

Matome, kad tikimybė $p(h, k)$ priklauso tik nuo $s = h+k-1$. Apskaičiuosime atsitiktinio įvykio S matematinį vidurkį.

$$E(S) = \sum_{s=1}^{n} s\left(\sum_{h+k-1=s} p(h,k)\right) = \sum_{s=1}^{n} s^2 \frac{1}{n}\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\ldots\left(1-\frac{s-1}{n}\right).$$

Pažymėkime $Q(n)$ sumą:

$$Q(n) = \sum_{s=1}^{n}\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\ldots\left(1-\frac{s-1}{n}\right) = 1+\left(1-\frac{1}{n}\right)+\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)+\ldots+\left(1-\frac{1}{n}\right)\ldots\left(1-\frac{n-1}{n}\right).$$

Parodysime, kad $Q(n) = E(S)$.

Akivaizdu, kad

$$s^2 \frac{1}{n}\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\ldots\left(1-\frac{s-1}{n}\right) = s\left(1-\frac{1}{n}\right)\ldots\left(1-\frac{s-1}{n}\right) - s\left(1-\frac{1}{n}\right)\ldots\left(1-\frac{s-1}{n}\right)\left(1-\frac{s}{n}\right).$$

Vadinasi,

$$E(S)=\sum_{s=1}^{n} s^2\,\frac{1}{n}\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\dots\left(1-\frac{s-1}{n}\right)=$$
$$=\sum_{s=1}^{n} s\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\dots\left(1-\frac{s-1}{n}\right)-\sum_{s=1}^{n} s\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\dots\left(1-\frac{s}{n}\right)=$$
$$=1+2\left(1-\frac{1}{n}\right)+3\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)+\dots-\left(1-\frac{1}{n}\right)-2\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)-\dots=$$
$$=1+\left(1-\frac{1}{n}\right)+\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)+\dots\left(1-\frac{1}{n}\right)\dots\left(1-\frac{n-1}{n}\right)=Q(n).$$

Tai, kad teoremos teiginys yra teisingas, rodo Knuto asimptotinis įvertis:

$$Q(n)=\sqrt{\frac{\pi n}{2}}-\frac{1}{3}+\frac{1}{12}\sqrt{\frac{\pi}{2n}}-\frac{4}{135n}+\frac{1}{288}\sqrt{\frac{\pi}{2n^3}}+O\left(n^{-2}\right).\ \blacktriangledown$$

Tarkime, kad m – hipotetinis skaičiaus n daliklis, $1 < m < n$. Iš sekos $\{x_i\}$, $x_i \in Z_n$, $x_{i+1} = f(x_i)$, galima sudaryti seką $\{x'_i\}$, $x'_i \in Z_n$, $x'_i \equiv x_i \bmod m$. Remiantis 4.2 teiginiu, reikės $\sqrt{m}$ žingsnių iki pasikartojimo sekoje $\{x'_i\}$. Tai reiškia, kad yra toks r, kurio eilė $\sqrt{m}$ ir $x'_r = x'_s$, tam tikram $s < r$. Iš x'_i apibrėžimo gauname $m|(x_s - x_r)$, bet m yra n daliklis, vadinasi, m dalija $d = \mathrm{DBD}(|x_s - x_r|, n)$, todėl $d > 1$. Jei $d \neq n$, radome netrivialųjį skaičiaus n daliklį. Sudėtinis skaičius n turi daliklį $1 < m < \sqrt{n}$, todėl netrivialusis daliklis gali būti aptiktas per iteracijų skaičių, kurio eilė $\sqrt{m} \le \sqrt{\sqrt{n}} = n^{\frac{1}{4}}$

Pollardo *ro* algoritmui realizuoti reikia funkcijos f, kurią siūloma pasirinkti kaip daugianarį $f(x) \in Z[x]$. Tačiau reikalingas toks daugianaris, kuris generuotų seką $\{x_i\}$, labai panašią į atsitiktinę funkciją ne tik žiede Z_n, bet ir žiede Z_m, kai $m|n$. Skaičiavimai rodo, kad daugianariai $x^2 + a$ ($a \neq 0, 2$) tenkina šiuos reikalavimus, todėl jie dažniausiai ir naudojami taikant Pollardo *ro* metodą.

4.2 algoritmas. Pollardo *ro* algoritmas

(įvestis: sveikasis skaičius n)

$x \leftarrow 2$, $y \leftarrow 2$; $d \leftarrow 1$
Kol $d = 1$:
    $x \leftarrow f(x)$,
    $y \leftarrow f(f(y))$,
    $d \leftarrow \mathrm{DBD}(|x - y|, n)$.
Jei $d = n$, grąžinti KLAIDA,
Priešingu atveju grąžinti d.

(išvestis: daliklis d arba pranešimas KLAIDA)

4.3. Apibrėžimas. Tarkime, kad B yra teigiamas sveikasis skaičius. Sveikasis skaičius n yra vadinamas B glodžiu arba glodžiu ribos B atžvilgiu, jei jo visi pirminiai daugikliai yra mažesni už B.

Naudojant Pollardo p-1 algoritmą, galima rasti bet kurį skaičiaus n pirminį daugiklį, jei p-1 yra glodus tam tikros ribos B atžvilgiu.

Panagrinėkime Pollardo p-1 algoritmą. Tarkime, kad B yra glodumo riba. Tegu Q yra mažiausias bendrasis kartotinis visų pirminių skaičių, mažesnių už B, laipsnių, kurių reikšmės mažesnės už n. Jei $q^l \le n$, tai $l \ln q \le \ln n$, ir taip pat $l \le \left\lfloor \frac{\ln n}{\ln q} \right\rfloor$. Taigi $Q = \prod_{q \le B} q^{\lfloor \ln n / \ln q \rfloor}$. Jei p yra toks skaičiaus n pirminis daugiklis, kad p-1 yra B glodus, tai p-1$|Q$, todėl su bet kuriuo a tenkinama sąlyga $\mathrm{DBD}(a, p)$=1. Iš Ferma teoremos išplaukia, kad $a^Q \equiv 1 \bmod p$. Taigi, jei d=DBD(a^Q-1, n), tai $p|d$. Gali būti ir $d = n$. Tokiu atveju algoritmas pateikia klaidą, ir tai pasitaiko tada, kai skaičius n turi mažiausiai du didelius skirtingus pirminius daugiklius.

4.3 algoritmas. Pollardo *p*-1 algoritmas

(įvestis: sveikasis skaičius n)

1. Pasirinkti ribą B.
2. Pasirinkti atsitiktinį a, $2 \le a \le n-1$, $d := \text{DBD}(a, n)$. Jei $d \ge 2$, grąžinti d.
3. Kiekvienam pirminiam $q \le B$ vykdyti:
   3.1. $l := \lfloor \ln n / \ln q \rfloor$;
   3.2. $a := a^{ql} \bmod n$.
4. $d = \text{DBD}(a-1; n)$.
5. Jei $d = 1$ arba $d = n$, grąžinti KLAIDA. Priešingu atveju grąžinti d.

(išvestis: daliklis d arba pranešimas KLAIDA)

Šio algoritmo sudėtingumas yra $O(B \cdot \log B \cdot \log^2 n)$.

Viljamso p+1 algoritmas. Šis algoritmas gerai veikia tada, kai skaičius n turi vieną ar daugiau tokių daugiklių p, kad p+1 yra glodus, t. y. p+1 turi tik mažus daugiklius. Algoritmui realizuoti pasirenkame sveikąjį skaičių $A > 2$, kuris apibrėžia seką:

$V_0 = 2$, $V_1 = A$, $V_j = (AV_{j-1} - V_{j-2}) \bmod n$.

Tada bet kuris pirminis p dalija $\text{DBD}(n, V_M\text{-}2)$, kai tik M yra kartotinis p-$\left(\frac{D}{p}\right)$, čia $D = a^2\text{-}4$ ir $\left(\frac{D}{p}\right)$ yra Ležandro simbolis. Reikalaujame, kad $\left(\frac{D}{p}\right) = -1$. Kadangi p iš anksto nežinomas, gali būti reikalinga daugiau nei viena A reikšmė sprendiniui rasti. Jei $\left(\frac{D}{p}\right) = 1$, tai šis algoritmas degeneruoja į lėtą Pollardo p-1 algoritmo versiją. Taigi skirtingoms M reikšmėms skaičiuojame $\text{DBD}(n, V_M\text{-}2)$. Kai rezultatas nelygus 1 arba n, gauname netrivialųjį n daugiklį.

4.4 algoritmas. Viljamso *p*+1 algoritmo fragmentas

(įvestis: sveikasis skaičius n ir M)

$x = B$.
$y = (B\text{^}2\text{-}2) \bmod n$.
Kiekvienam skaičiaus M bitui iš kairės į dešinę vykdyti:
  jei bitas lygus 1,
    $x = (x*y\text{-}B) \bmod n$,
    $Y = (y\text{^}2\text{-}2) \bmod n$;
  priešingu atveju,
    $y = (x*y\text{-}B) \bmod n$,
    $x = (x\text{^}2\text{-}2) \bmod n$.
$V=x$.

(išvestis: V)

Elipsinių kreivių faktorizacijos algoritmas yra Pollardo p-1 algoritmo apibendrinimas, grupę Z_p^* pakeičiant atsitiktine elipsinių kreivių grupe virš Z_p. Jo sudėtingumas $O\left(\exp\left(\left(1+o(1)\right)(\ln p \ln\ln p)^{\frac{1}{2}}\right)\right)$, čia p – mažiausias skaičiaus n daugiklis.

Skaičių lauko rėčio algoritmo nenagrinėsime. Šiuo metu jis yra sparčiausias iš žinomų faktorizacijos algoritmų, o jo sudėtingumas – $O\left(\exp\left((c+o(1))(\log n)^{\frac{1}{3}}(\log\log n)^{\frac{2}{3}}\right)\right)$.

4.2. DISKRETINIO LOGARITMO SKAIČIAVIMO ALGORITMAI

Diskretinis logaritmas gali būti skaičiuojamas išsamios paieškos algoritmu, „Mažo žingsnio – didelio žingsnio“ algoritmu, Pollardo *ro* algoritmu logaritmams, taip pat Pollardo lambda, Pohligo-Hellmano, indeksų skaičiavimo ir skaičių lauko rėčio algoritmais.

Akivaizdžiausias diskretinio logaritmo skaičiavimo algoritmas – išsamios paieškos, reikalaujantis nuosekliai skaičiuoti reikšmes g^0, g^1, g^2, ... mod p, kol bus gauta reikšmė $y = g^x \bmod p$. Šis metodas reikalauja $O(n)$ daugybos veiksmų, kur n yra skaičiaus g eilė, t. y. $g^n = 1 \bmod p$.

Panagrinėkime „Mažo žingsnio – didelio žingsnio“ algoritmą. Tarkime, kad $m = \lfloor \sqrt{n} \rfloor$, čia n yra α eilė. Jei $\beta = \alpha^x$, tai galime rašyti $x = im + j$, kur $0 \le i,j \le m$. Vadinasi, $\alpha^x = \alpha^{im}\alpha^j$, o iš to išplaukia, kad $\beta(\alpha^{-m})^i = \alpha^j$.

4.5. „Mažo žingsnio – didelio žingsnio“ algoritmas

(įvestis: ciklinės grupės G, kurios eilė n, generatorius α ir elementas $\beta \in G$)

1. $m \leftarrow \lceil n^{1/2} \rceil$.
2. Visiems j, kur $0 \le j < m$, vykdyti:
   2.1. Skaičiuoti α^j ir rašyti į lentelę poras (j, α^j).
3. Skaičiuoti α^{-m}.
4. $y \leftarrow \beta$.
5. Visiems i, kur $0 \le i < m$, vykdyti:
   5.1. Tikrinti, ar y yra kaip antroji komponentė suformuotoje lentelėje.
   5.2. Jei taip, grąžinti $im + j$.
   5.3. Jei ne, $y \leftarrow y \cdot \alpha^{-m}$.

(išvestis: $im + j$)

Šio algoritmo sudėtingumas yra $O(n^{1/2})$.

Pollardo *ro* algoritmas yra tikimybinis, o jo sudėtingumas toks pat kaip ir „Mažo žingsnio – didelio žingsnio“, tačiau jis reikalauja nedidelio atmintinės kiekio. Paprastumo dėlei tarkime, kad G yra ciklinė grupė, kurios eilė p – pirminis skaičius.

Yra ciklinė grupė G, kurios generatorius – α. Duota $\beta \in G$. Reikia rasti x iš lygties $\beta = \alpha^x$, t. y. $x = \log_\alpha \beta$. Grupė G padalijama į tris aibes – S_1, S_2 ir S_3, kurių galios apytiksliai lygios ir apibrėžtos tam tikra lengvai patikrinama savybe. Numatyti tam tikri dalijimo apribojimai, pavyzdžiui, $1 \notin S_2$. Apibrėšime grupės elementų seką x_0, x_1, x_2, ..., kur $x_0 = 1$ ir

$$x_{i+1} = f(x_i) = \begin{cases} \beta x_i, & jei\ x_i \in S_1, \\ x_i^2, & jei\ x_i \in S_2, \\ \alpha x_i, & jei\ x_i \in S_3, \end{cases} \quad \text{čia } i \ge 0.$$

Ši grupės elementų seka apibrėžia dvi sveikųjų skaičių sekas a_0, a_1, a_2, ... ir b_0, b_1, b_2, ..., tenkinančias sąlygą $x_i = \alpha^{a_i}\beta^{b_i}$, kur $i \ge 0$. Tada $a_0 = 0$, $b_0 = 0$ ir visiems $i \ge 0$

$$a_{i+1} = \begin{cases} a_i, & jei\ x_i \in S_1, \\ 2a_i \bmod n, & jei\ x_i \in S_2, \\ a_i + 1 \bmod n, & jei\ x_i \in S_3, \end{cases}$$

bei

$$b_{i+1} = \begin{cases} b_i + 1 \bmod n, & jei\ x_i \in S_1, \\ 2b_i \bmod n, & jei\ x_i \in S_2, \\ b_i, & jei\ x_i \in S_3. \end{cases}$$

Floido ciklo radimo algoritmas gali būti pasitelktas ieškant tokių dviejų grupės elementų x_i ir x_{2i}, kad $x_i = x_{2i}$. Vadinasi, $\alpha^{a_i}\beta^{b_i} = \alpha^{a_{2i}}\beta^{b_{2i}}$, todėl $\beta^{b_i - b_{2i}} = \alpha^{a_{2i} - a_i}$. Išlogaritmavę abi lygybės puses pagrindu α, gauname:

$(b_i - b_{2i}) \log_\alpha \beta \equiv (a_{2i} - a_i) \bmod n$.

Esant reikalavimui, kad $b_i \not\equiv b_{2i}$, ši lygtis gali būti efektyviai išspręsta ieškant $\log_\alpha \beta$.

4.6 algoritmas. Pollardo *ro* algoritmas

(įvestis: ciklinės grupės G, kurios eilė – pirminis skaičius n, generatorius α ir elementas $\beta \in G$)

1. $x_0 \leftarrow 1$, $a_0 \leftarrow 0$, $b_0 \leftarrow 0$.
2. Kiekvienam $i = 1, 2, \ldots$ vykdyti:
   2.1. Naudojant anksčiau apskaičiuotas reikšmes x_{i-1}, a_{i-1}, b_{i-1} ir x_{2i-2}, a_{2i-2}, b_{2i-2}, pagal aukščiau pateiktas formules apskaičiuoti x_i, a_i, b_i ir x_{2i}, a_{2i}, b_{2i}.
   2.2. Jei $x_i = x_{2i}$, atlikti šiuos veiksmus
   $r \leftarrow b_i - b_{2i} \bmod n$.
   Jei $r = 0$, grąžinti KLAIDA,
   Priešingu atveju skaičiuoti $x = r^{-1}(a_{2i} - a_i) \bmod n$ ir grąžinti x.

(išvestis: diskretinis logaritmas $x = \log_\alpha \beta$)

Šio algoritmo sudėtingumas yra $O\left(\sqrt{n}\right)$ grupės operacijų.

Pohligo-Hellmano algoritmas skirtas skaičiuoti diskretiniam logaritmui multiplikacinėje grupėje, kurios eilė yra glodus sveikasis skaičius. Šis algoritmas remiasi kinų liekanų teorema, o jo sudėtingumas yra polinominis.

4.7 algoritmas. Pohligo-Hellmano algoritmas

(įvestis: p, g, e)

1. Pasitelkę Oilerio funkciją, randame grupės eilės pirminius daugiklius:
   $\varphi(p) = p_1 \cdot p_2 \cdot \ldots \cdot p_r$.
2. Iš kinų liekanų teoremos žinome, kad $x = a_1 p_1 + b_1$. Naudodami „Mažo žingsnio – didelio žingsnio“ algoritmą, galime rasti reikšmę b_1, kuriai galioja tokia lygtis:
   $e^{\varphi(p)/p_1} \equiv \left(g^x\right)^{\varphi(p)/p_1} \bmod p \equiv \left(g^{\varphi(p)}\right)^{a_1} g^{b_1\varphi(p)/p_1} \bmod p \equiv \left(g^{\varphi(p)/p_1}\right)^{b_1} \bmod p.$
   Jei $g^{\varphi(p)/p_1} \equiv 1 \bmod p$, tai g eilė yra mažesnė už $\varphi(p)$, ir $e^{\varphi(p)/p_1} \bmod p$ turi būti lygus 1, kad egzistuotų šios lygties sprendinys. Šiuo atveju bus daugiau nei vienas sprendinys x, mažesnis už $\varphi(p)$, bet kadangi nepaisome kitų sprendinių, galime reikalauti, kad $b_1 = 0$.
   Tokia pat operacija atliekama ir visiems kitiems p_i.
   Nedidelė modifikacija reikalinga tada, kai pirminis daugiklis yra kartotinis. Tarkime, kad p_j kartojasi $k + 1$ kartą. Tada jau žinome c_j lygtyje $x = a_j p_j^{k+1} + b_j p_j^k + c_j$ ir rasime b_j kaip ir anksčiau.
3. Pasitelkę kinų liekanų teoremą, iš pakankamo skaičiaus gautų lygybių galime rasti x.

(išvestis: toks sveikasis skaičius x, kad $e \equiv g^x \bmod p$)

Šio algoritmo sudėtingumas yra $O\left(\sum_{i=1}^{r} e_i\left(\lg p + \sqrt{p_i}\right)\right)$ grupės daugybos operacijų, kai e_i – daugiklio kartotinumo skaičius. Iš sudėtingumo įverčio formulės matome, kad Pohligo-Hellmano algoritmas efektyvus tik tuo atveju, jei pirminiai daugikliai yra sąlygiškai maži.

Toliau nagrinėsime efektyviausią iš žinomų diskretinio logaritmo skaičiavimo algoritmų –indeksų skaičiavimo. Jis nepritaikomas visoms grupėms, bet jei tik galima tai padaryti, tai jo skaičiavimo laikas yra subeksponentinis.

4.8 algoritmas. Indeksų skaičiavimo algoritmas

(įvestis: ciklinės grupės G, kurios eilė – pirminis skaičius n, generatorius α ir elementas $\beta \in G$)

1. Faktorbazės parinkimas. Parenkame tokį grupės G poaibį $S = \{p_1, p_2, ..., p_t\}$, kad daug G elementų galėtų būti išreikšti S elementų sandaugomis.
2. Surenkame tiesinius sąryšius, siejančius S elementų logaritmus.
   2.1. Parenkame atsitiktinį skaičių k, $0 \le k \le n-1$, ir apskaičiuojame α^k.
   2.2. Bandome α^k išreikšti S elementų sandauga:

$$\alpha^k = \prod_{i=1}^{t} p_i^{c_i}, \quad c_i \ge 0.$$

Jei tai pavyksta, išlogaritmavę abi šios lygybės puses, gauname tiesinį sąryšį:

$$k = \sum_{i=1}^{t} c_i \log_\alpha p_i \mod n.$$

   2.3. Kartojame 2.1 ir 2.2 žingsnius tol, kol gauname $t + c$ tiesinių sąryšių (čia c – mažas sveikasis skaičius, pavyzdžiui, toks c = 10, kad gauta $t + c$ tiesinių sąryšių sistema, esant didelei tikimybei, turėtų vienintelį sprendinį).
3. Randame S elementų logaritmus. Atlikdami skaičiavimus moduliu n, sprendžiame $t + c$ tiesinių lygčių sistemą (su t nežinomųjų) ir randame $\log_\alpha p_i$, $1 \le i \le t$.
4. Skaičiuojame y.
   4.1. Parenkame atsitiktinį skaičių k, $0 \le k \le n-1$, ir apskaičiuojame $\beta\alpha^k$.
   4.2. Bandome $\beta\alpha^k$ išreikšti S elementų sandauga:

$$\beta\alpha^k = \prod_{i=1}^{t} p_i^{d_i}, \quad d_i \ge 0.$$

Jei to padaryti nepavyksta, kartojame 4.1 žingsnį.
Priešingu atveju, išlogaritmavę 4.2 žingsnyje gautą lygybę, gauname

$$\log_\alpha \beta = \sum_{i=1}^{t} d_i \log p_i - k \mod n.$$

(išvestis: diskretinis logaritmas $y = \log_\alpha \beta$)

Lauke Z_p, kur p – pirminis skaičius, faktorbazė S gali būti pirmieji t pirminiai skaičiai. Skaičiuojant α^k ir siekiant nustatyti, ar jis yra S elementų sandauga, pasitelkiamas bandomosios dalybos algoritmas.

4.4. Pavyzdys. Tarkime, kad $p = 229$. Skaičius $\alpha = 6$ yra grupės Z_{229}^*, kurios eilė $n = 228$, generatorius. Raskime $\log_6 13$, naudodami indeksų skaičiavimo algoritmą.

1. Faktorbaze pasirinkime pirmuosius penkis pirminius skaičius: S = {*2*, 3, 5, 7, 11}.
2. Raskime šešis sąryšius, siejančius faktorbazės S elementus (nesėkmingų bandymų nerodome):

$$6^{100} \bmod 229 = 180 = 2^2 \cdot 3^2 \cdot 5,$$
$$6^{18} \bmod 229 = 176 = 2^4 \cdot 11,$$
$$6^{12} \bmod 229 = 165 = 3 \cdot 5 \cdot 11,$$
$$6^{62} \bmod 229 = 154 = 2 \cdot 7 \cdot 11,$$
$$6^{143} \bmod 229 = 198 = 2 \cdot 3^2 \cdot 11,$$
$$6^{206} \bmod 229 = 210 = 2 \cdot 3 \cdot 5 \cdot 7.$$

Iš šių sąryšių gauname šešias lygtis su faktorbazės S elementų logaritmais:

$$100 \equiv 2\log_6 2 + 2\log_6 3 + \log_6 5 \pmod{228},$$
$$18 \equiv 4\log_6 2 + \log_6 11 \pmod{228},$$
$$12 \equiv \log_6 3 + \log_6 5 + \log_6 11 \pmod{228},$$
$$62 \equiv \log_6 2 + \log_6 7 + \log_6 11 \pmod{228},$$
$$143 \equiv \log_6 2 + 2\log_6 3 + \log_6 11 \pmod{228},$$
$$206 \equiv \log_6 2 + \log_6 3 + \log_6 5 + \log_6 11 \pmod{228}.$$

3. Spręsdami šešių lygčių su penkiais nežinomaisiais sistemą, gauname faktorbazės S elementų logaritmus: $\log_6 2 = 21$, $\log_6 3 = 208$, $\log_6 5 = 98$, $\log_6 7 = 107$, $\log_6 11 = 162$.
4. Tarkime, pasirinktas sveikasis skaičius $k = 77$. Kadangi $\beta\alpha^k = 13 \cdot 6^{77} \bmod 229 = 147 = 3 \cdot 7^2$, tai

$$\log_6 13 = (\log_6 3 + 2\log_6 7 - 77) \bmod 228 = 117.$$

Parodysime, kaip veikia indeksų skaičiavimo algoritmas Galua grupėje $GF^*(2^m)$.

4.5. Pavyzdys. Polinomas $f(x) = x^7 + x + 1$ yra pirminis virš lauko Z_2. Atlikę $Z_2[x]$ faktorizaciją pirminiu polinomu $f(x)$, gausime Galua grupę $GF^*(2^7)$. Elementas x yra grupės $GF^*(2^7)$ generatorius. Tarkime, kad $\beta = x^4 + x^3 + x^2 + x + 1$. Rasime $y = \log_x\beta$, naudodami indeksų skaičiavimo algoritmą.

1. Faktorbaze pasirinksime visų neredukuojamų polinomų grupėje $GF^*(2^7)$ aibę, kurios laipsnis ne didesnis kaip 3: $S = \{x, x + 1, x^2 + x + 1, x^3 + x + 1, x^3 + x^2 + 1\}$.
2. Rasime penkis sąryšius, siejančius faktorbazės S elementus:

$$x^{18} \bmod f(x) = x^6 + x^4 = x^4(x + 1)^2,$$
$$x^{105} \bmod f(x) = x^6 + x^5 + x^4 + x = x(x + 1)^2(x^3 + x^2 + 1),$$
$$x^{72} \bmod f(x) = x^6 + x^5 + x^3 + x^2 = x^2(x + 1)^2(x^2 + x + 1),$$
$$x^{45} \bmod f(x) = x^5 + x^2 + x + 1 = (x + 1)^2(x^3 + x + 1),$$
$$x^{121} \bmod f(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 = (x^3 + x + 1)(x^3 + x^2 + 1).$$

Iš šių sąryšių gauname penkių tiesinių lygčių su penkiais nežinomaisiais sistemą faktorbazės S elementų logaritmų atžvilgiu. Pažymime $p_1 = \log_x x$, $p_2 = \log_x(x + 1)$, $p_3 = \log_x(x^2 + x + 1)$, $p_4 = \log_x(x^3 + x + 1)$, $p_5 = \log_x(x^3 + x^2 + 1)$. Gauname tokią sistemą:

$$18 \equiv 4p_1 + 2p_2 \pmod{127},$$
$$105 \equiv p_1 + 2p_2 + p_5 \pmod{127},$$
$$72 \equiv 2p_1 + 2p_2 + p_3 \pmod{127},$$
$$45 \equiv 2p_2 + p_4 \pmod{127},$$
$$121 \equiv p_4 + p_5 \pmod{127}.$$

3. Išsprendę šią sistemą, gauname faktorbazės elementų logaritmų reikšmes: $p_1 = 1$, $p_2 = 7$, $p_3 = 56$, $p_4 = 31$, $p_5 = 90$.
4. Tarkime, pasirenkame $k = 66$. Kadangi

$$\beta\alpha^k = (x^4 + x^3 + x^2 + x + 1)x^{66} \bmod f(x) = x^5 + x^3 + x = x(x^2 + x + 1)^2,$$

gauname

$$\log_x(x^4 + x^3 + x^2 + x + 1) = (p_1 + 2p_3 - 66) \bmod 127 = 47.$$

4.2 lentelėje pateiksime dar kelių diskretinio logaritmo skaičiavimo algoritmų pavadinimus ir jų sudėtingumo įverčius.

4.2 lentelė

Algoritmas	Sudėtingumo įvertis
Skaičių lauko rėtis	$O\left(\exp\left(c\sqrt[3]{\ln p(\ln\ln p)^2}\right)\right)$, čia $c < 2$.
Funkcijų lauko rėtis	$O\left(\exp\left(c\sqrt[3]{\ln q(\ln\ln q)^2}\right)\right)$, čia $c = \sqrt[3]{\frac{32}{9}} + o(1)$.

5. BLOKINIO ŠIFRAVIMO SISTEMŲ KŪRIMAS IR KRIPTOGRAFINĖ ANALIZĖ

Naujas šifras turi būti atsparus kriptoanalizės atakoms. Naujų šifrų struktūra darosi vis sudėtingesnė, todėl vien faktas, kad šifravimo algoritmas yra labai sudėtingas, saugumo neužtikrina. Praeityje jau ne kartą buvo lengvai „nulaužti“ net itin sudėtingi šifrai. Deja, mes negalime būti tikri, kad mūsų pasirinktas šifras yra saugus. Geriausias būdas įvertinti šifravimo algoritmo saugumą – viešai jį pateikti analizuoti plačiai kriptografinei bendruomenei.

Šifro autorius ar būsimas vartotojas turėtų atsižvelgti į įvairius šifro saugumo kriterijus, nes būtent jie atspindi bendras šifro savybes ir jo atsparumą žinomoms atakoms. Pirmiausia šifravimo raktų aibė turi būti pakankamai didelė, kad visiško perrinkimo ataka būtų neįmanoma ar bent jau per brangi realizuoti. Vienas svarbiausių kriterijų: neturėtų būti jokios sąsajos tarp įvesties bitų ar rakto bitų ir išvesties bitų. Kitaip tariant, šifrogramoje neturėtų būti jokių užuominų apie tekstogramą ar raktą. Tai yra itin svarbu, nes jei kas nors pradėtų bandyti skirtingus raktus, negalėtų įvertinti, kaip arti tikrojo rakto esama. Net jei šifravimo raktas skirtųsi vienu bitu, juo šifruotos šifrogramos turėtų būti labai skirtingos.

Šiame skyriuje apžvelgsime dvi pagrindines blokinių šifrų architektūras, galimas ir veiksmingiausias kriptografinių atakų rūšis. Taip pat pateiksime šifrų atsparumo žinomoms kriptoanalizės atakoms kriterijų apžvalgą. Blokinio šifro kriptoanalizę dažniausiai traktuosime kaip šifravimo rakto demaskavimą.

5.1. BLOKINIŲ ŠIFRŲ ARCHITEKTŪROS

Blokiniuose šifruose atskiras šifruojamo teksto elementas vadinamas bloku. Blokinis šifras paprastai apibrėžto ilgio tekstogramos blokus atvaizduoja į tokio pat ilgio šifrogramos blokus, todėl tekstogramų (T) ir šifrogramų (C) aibės sutampa: $T = C = \{0, 1\}^n$, čia n – apibrėžtas bloko ilgis bitais, o aibė $\{0, 1\}^n$ nusako n ilgio bitų eilutę.

> ***5.1. Apibrėžimas.*** Blokinis šifras – tai simetrinis šifras, kuriuo užšifruotas teksto blokas priklauso tik nuo įvedamo teksto bloko ir slaptojo rakto, t. y. šifras neturi vidinės būsenos. Kiekvienas duomenų blokas šifruojamas vienodu algoritmu.

Blokiniais šifrais užšifruojami dideli fiksuoto ilgio duomenų blokai (pvz., 64, 128 ar 512 bitų).

Dauguma šiuolaikinių šifrų kuriami remiantis laiko patikrintomis architektūromis – sukeitimų ir perstatymų tinklais bei Feistelio šifru.

5.1.1. SP TINKLAI

Sukeitimų ir perstatymų tinklai buvo pasiūlyti siekiant išvengti kriptoanalizės, pagrįstos statistine analize. Geras šifras turi visiškai paslėpti tekstogramos statistines savybes. C. Shannonas pasiūlė naudoti sukeitimų ir perstatymų tinklus, kurie užtikrintų šifruojamos informacijos sumaišymą ir paskleidimą. Paskleidimo mechanizmu siekiama, kad vieno tekstogramos bito pokytis turėtų įtakos visiems šifrogramos bitams. Sumaišymu siekiama ryšius tarp šifrogramos statistinės informacijos ir šifravimo rakto padaryti kuo sudėtingesnius.

Sukeitimų ir perstatymų tinklai (SP tinklai) sudaryti iš kelių sukeitimų (S blokų) ir perstatymų blokų (P blokų) lygių, einančių vienas paskui kitą. Nors sukeitimai kriptografiniu atžvilgiu yra gana nesaugūs, tačiau derinant juos su perstatymo blokais galima pasiekti neblogų sumaišymo rezultatų: sukeitimai leidžia duomenis lokaliai sumaišyti, o perstatymai juos sujungia ir paskleidžia. P blokais atliekamos tiesinės ar afiniosios transformacijos, o S blokais – netiesinės, didinančios šifro atsparumą žinomoms atakoms.

Priminsime pagrindinius kriptografinius primityvus, naudojamus SP tinkluose.

5.2. Apibrėžimas. Sukeitimas (angl. *substitution*) – simbolių ar objektų pakeitimas kitais simboliais. Simbolius galima keisti po vieną arba jų grupes.

5.3. Apibrėžimas. S blokas (sukeitimo blokas, angl. *S-box*) – pagrindinis simetrinių šifrų komponentas. S blokai sukeičia simbolius ir užmaskuoja pradinio bei šifruoto tekstų ryšius.

5.4. Apibrėžimas. Perstatymas (angl. *permutation*) – objektų ar simbolių eilės tvarkos perstatymas. Perstatymas – abėcėlės abipusiškai vienareikšmis atvaizdis į save patį, kuris vadinamas P bloku.

Taigi SP tinklus sudaro S blokai ir P blokai. Formaliai SP tinklai nusakyti 5.5 apibrėžimu.

5.5. Apibrėžimas. Sukeitimų ir perstatymų tinklai (SP tinklai) yra blokinių šifrų architektūra, sudaryta iš kelių sukeitimų ir perstatymų blokų lygių, einančių vienas paskui kitą.

Nors savaime sukeitimai kriptografiniu atžvilgiu yra gana nesaugūs, tačiau juos derinant su perstatymo blokais galima gauti neblogus sumaišymo rezultatus: sukeitimai leidžia lokaliai sumaišyti, o perstatymai – juos globaliai paskleisti.

SP tinklus paprastai sudaro kelios panašios nuosekliai vykdomos operacijos (lygiai). Visa vieno lygio operacijų realizacija vadinama raundu (angl. *round*). Pagrindinės raundo sudedamosios dalys – raundo funkcija ir raundo raktas. Šifruojama atliekant iteracijas, t. y. pereinant nuo vieno raundo prie kito, pasitelkus prieš tai gautus duomenis.

5.6. Apibrėžimas. Raundo funkcija – tai blokinio šifro vienos iteracijos šifravimo funkcija.

Raundo funkciją gali sudaryti įvairūs S ir P blokų deriniai. Raundo funkcijos savybės priklauso nuo konkrečios architektūros. Vienais atvejais raundo funkcija turi būti apverčiama, kitais – vienkryptė.

5.7. Apibrėžimas. Raundų raktų sąrašas (angl. *round key schedule*) – tai rinkinys raktų, sugeneruotų raktų sąrašo algoritmu, pasitelkus pradinį vartotojo duotą raktą, ir naudojamų kiekviename raunde pagal šifro algoritmą.

Raundų raktų sąrašas – tai išplėstas iki reikiamo šifruoti fiksuoto ilgio slaptasis raktas. Iš šio sąrašo imamas kiekvieno raundo raktas.

Kiekvieną S bloką galima nagrinėti kaip *n* kintamųjų vektorinę Būlio funkciją *f* su *m* komponenčių: *f*: $F_2^n \rightarrow F_2^m$, kur $F_2 = \{0, 1\}$. Bet kurią Būlio funkciją vienareikšmiškai aprašo jos teisingumo lentelė.

5.8. Teorema. Kiekviena Būlio funkcija *f* gali būti išreiškiama *n* kintamųjų polinomu virš lauko *GF*(2), kurio eilė pagal kiekvieną kintamąjį neviršija 1 (Логачев, et al., 2004).

Tokia Būlio funkcijos išraiška vadinama algebrine normaline forma (ANF). ANF kitaip dar galima vadinti daugianariu polinomu (angl. *multivariate polynomial*). Kiekvienas šio polinomo adicinis narys vadinamas termu. Iš funkcijos *f* teisingumo lentelės galima gauti ANF, naudojant Mėbiuso transformaciją (Pommerening, 2005).

Tarkime, yra du vektoriai $x, y \in F_2^n$. Skaliarinę sandaugą (x, y) apibrėšime taip:

$$(x, y) = x \cdot y = x_1 y_1 \oplus \ldots \oplus x_n y_n.$$

Remdamiesi skaliarine sandauga, galime apibrėžti tiesinę Būlio funkciją:

$$l_a(x) = a \cdot x = a_1 \cdot x_1 \oplus \ldots \oplus a_n \cdot x_n;$$

čia $a \in F_2^n$ yra fiksuotas vektorius. Afinioji funkcija $l_{a,b}$ apibrėžiama taip:

$$l_{a,b}(x) = a \cdot x + b;$$

čia $b \in F_2$.

Tiesinių funkcijų aibę žymėsime $\mathcal{L}_n$, afiniųjų – $\mathcal{A}_n$, visų Būlio funkcijų – $\mathcal{F}_n$.

Nagrinėjama erdvė F_2^n yra vektorinė, todėl galima nagrinėti tam tikras jos metrines savybes.

5.9. Apibrėžimas. Vektoriaus $x \in F_2^n$ Hemingo svoris – nenulinių vektoriaus koordinačių skaičius, kurį žymėsime *wt*(*x*).

5.10. Apibrėžimas. Hemingo atstumu tarp dviejų vektorių $x, y \in F_2^n$ vadinsime dydį $dist(x, y)$, kuriuo nusakomas jų skirtingų koordinačių skaičius:

$$dist(x, y) = wt(x \oplus y).$$

Dydžiai $wt()$ ir $dist()$ tenkina metrinės erdvės normos savybes.

5.11. Apibrėžimas. Atstumas tarp Būlio funkcijų f, g apibrėžiamas lygybe

$$dist(f, g) = \#\{x \in F_2^n \mid f(x) \neq g(x)\};$$

čia # žymi aibės elementų skaičių.

5.12. Apibrėžimas. Būlio funkcijos f atstumas iki erdvės $\mathcal{F}_n$ poerdvio $\mathcal{M} \subset \mathcal{F}_n$ apibrėžiamas taip:

$$dist(f, \mathcal{M}) = \min_{g \in \mathcal{M}} dist(f, g).$$

Saugus S blokas turi atitikti šiuos pagrindinius kriterijus:

- Balansuotumą: šis kriterijus rodo, kad S blokas neperduoda išėjimo signalams statistinės informacijos apie įėjimo signalus. Perdavę į šifravimo bloką visus tekstogramos derinius, jo išvestyje taip pat gausime visus įmanomus šifrogramos derinius.
- Netiesiškumas rodo funkcijos atstumą iki afiniųjų funkcijų poerdvio:

$$N_f = dist(f, \mathcal{A}).$$

- Diferencinis potencialas.

Iš šių kriterijų yra išvedami papildomi kriterijai:

- Koreliacinis imunitetas (angl. *correlation immunity* – CI): šis kriterijus rodo įėjimo ir išėjimo signalų koreliacijos laipsnį.
- Griežtas lavininis kriterijus (angl. *strict avalanche criterion* – SAC): S blokas tenkina SAC, jei tikimybė, kad vieno įėjimo signalo pakeitimas lems bet kurio išėjimo signalo pasikeitimą, yra lygi 0,5.
- Sklidimo kriterijus (angl. *propagation criterion*): tai apibendrintas SAC kriterijus, kuriuo įvertinami Būlio funkcijos reikšmių pokyčiai, atsižvelgiant į dalies argumentų pokytį.

Kriptografiniu atžvilgiu saugus S blokas turi užtikrinti:

- Duomenų sumaišymą.
- Pakitus vienam įvesties bitui, vidutiniškai turi pasikeisti pusė išvesties bitų.
- Kiekvienas išvesties bitas turi priklausyti nuo visų įvesties bitų.

SP tinkluose šifruojamų duomenų blokas padalijamas į mažesnes dalis, ir būtent jos yra S blokų įvestys. Tokiu būdu S blokai užtikrina tik lokalų duomenų sumaišymą, tuo tarpu P blokai SP tinkluose turi garantuoti S blokų sudaryto lokalaus sumaišymo išsklaidymą.

5.1.2. FEISTELIO TIPO ŠIFRAI

H. Feistelis[2] pasiūlė blokinių šifrų architektūrą, taip pat pagrįstą C. Shannono SP tinklų koncepcija. Ši architektūra leidžia lengvai sudaryti blokinį šifrą, pasitelkus įvairius šifravimo etapus ir funkcijas. Remiantis Feistelio šifro architektūra, galima sukurti sudėtingus šifrus su sudėtingomis raundo funkcijomis, o nesudėtingą iššifravimą užtikrina dvi involiucijos.

H.Feistelio šifro architektūra leidžia kurti šifrus su sudėtingomis raundo funkcijomis, o nesudėtingą iššifravimą užtikrina dvi naudojamos involiucijos.

[2] Horst Feistel (1915–1990) – vokiečių kriptografas, dirbęs ir gyvenęs JAV.

5.13. Apibrėžimas. Involiucija – tai pati sau atvirkštinė funkcija, t. y. $G(G(x)) = x$.

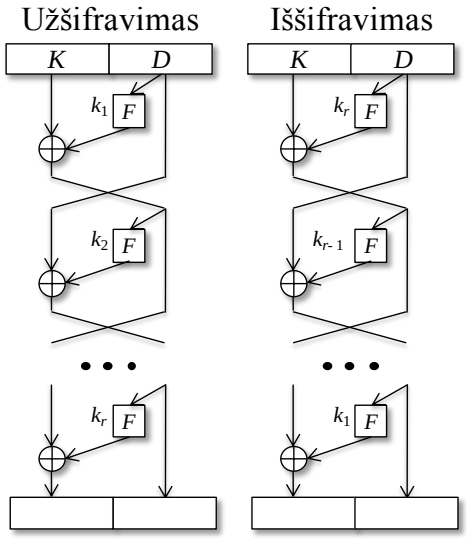
5.14. Uždavinys. Sugalvokite funkcijas, kurios būtų involiucijos.

5.15. Klausimas. Ar galima parinkti tokią vienkryptę funkciją, kuri būtų involiucija?

Feistelio šifras – viena iš iteratyvių blokinių šifrų klasių. Viename Feistelio šifro raunde naudojamos dvi involiucijos. Tai leidžia užšifruojant ir iššifruojant taikyti tokią pat schemą (paprastai pakanka pakeisti raundų raktų eilės tvarką). Duotą $2n$ bitų duomenų bloką Feistelio šifras padalija į dvi lygias dalis – K (kairę) ir D (dešinę). Pasitelkus tam tikrą funkciją $F(D, k)$, perskaičiuojami dešinės pusės bitai, gautas rezultatas sudedamas moduliu 2 su K (pirmoji involiucija) ir atliekamas sukeitimas (antroji involiucija):

$$(K, D) \rightarrow (D, K \oplus F(D, k));$$

čia k – raundo raktas, gautas raundo raktų sąrašo algoritmu.



5.1 pav. Feistelio šifro užšifravimo ir iššifravimo algoritmai

Svarbiausia Feistelio šifrų savybė ta, kad visada gaunamas perstatymas, kad ir kokia būtų funkcija F, t. y. gaunamas abipusiškai vienareikšmis atvaizdis. Raundinė funkcija F turi būti kriptografiškai saugi, o jos apverčiamumas, vienkryptiškumas, vienareikšmiškumas ar kitos savybės nebūtinos. Svarbu tai, kad ji leistų gerai užmaskuoti tekstogramos ir šifravimo rakto statistinius požymius. Ypatinga Feistelio schemos savybė ta, kad F gali būti siurjekcinė. Nepaisant F siurjektiškumo, užtikrinamas vienareikšmis šifrogramos iššifravimas.

Feistelio šifro šifravimo ir iššifravimo algoritmai pateikti 5.1 paveiksle.

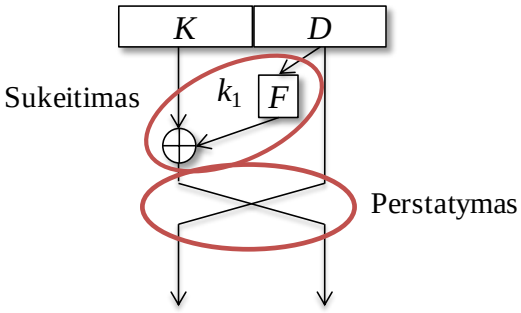
Paskutiniame raunde pusės nekeičiamos, todėl užšifravimo ir iššifravimo algoritmai sutampa. Iššifruojant imamas tas pats šifravimo algoritmas, tik atvirkštine raundų raktų tvarka.

Žinomiausias Feistelio šifras – praeityje labiausiai paplitęs DES, tačiau jis jau nėra saugus. Jau prieš aštuonerius metus DES šifro raktas buvo „nulaužiamas“ per 23 valandas (Oppliger, 2005)[3].

Kiti žinomi Feistelio šifrai arba modifikuoti Feistelio šifrai: *Blowfish*, *Camellia*, CAST-128, *FEAL*, *ICE*, *KASUMI*, LOKI97, *MARS*, *MAGENTA*, MISTY1, RC5, TEA, Triple DES, *Twofish*, XTEA.

Feistelio šifruose sukeitimą (S bloką) sudaro duomenų paveikimas funkcija F ir sudėtis moduliu 2, o perstatymą (P bloką) – gauto S bloko rezultato sukeitimas su dešiniąja duomenų dalimi (5.2 pav.).

[3] http://www.distributed.net/



5.2 pav. Feistelio šifro vienas raundas

Įrodyta, jog pakanka trijų raundų, kad Feistelio šifras su kriptografiškai saugia raundine funkcija *F* taptų pseudoatsitiktiniu keitiniu. Šifruojant keturiais ir daugiau raundų, šifras tampa „stipriu“ pseudoatsitiktiniu keitiniu (Luby, et al., 1988). Vėlesniuose darbuose buvo įrodyta, kad Feistelio šifro saugumui užtikrinti pakanka septynių raundų (Patarin, 2003).

Taip pat galimi apibendrinti Feistelio šifro variantai (Vaundenay, 2006):

- Dalijimą pusiau galima pakeisti dalijimu į keturias, aštuonias ar daugiau dalių.
- Vietoj sudėties moduliu 2 galima naudoti kokią nors kitą involiuciją (operaciją). Vienintelė būtina šios operacijos sąlyga – reguliarumas (t. y. jei $a * x = a * y$, tai $x = y$), o komutatyvumas ir asociatyvumas nebūtini.
- Duomenų bloką galima dalyti į skirtingo dydžio dalis (pvz., kaip BEAR ir LION šifruose).
- Į Feistelio schemą galima įtraukti papildomus apverčiamus S blokus (pvz., kaip BLOWFISH šifre).

Žinomi šie apibendrinti Feistelio šifrai: CAST-256, *MacGuffin*, RC2, RC6, *Skipjack*.

Skirtingoms to paties tekstogramos bloko šifrogramoms gauti blokiniai šifrai gali būti naudojami įvairiais režimais.

Primename penkis pagrindinius blokinių šifrų šifravimo režimus:

- Elektroninės šifrų knygos režimas (angl. *Electronic codebook* – ECB): pranešimo blokai šifruojami nepriklausomai vienas nuo kito.
- Šifro bloko grandininis režimas (angl. *Cipherblock chaining* – CBC): pirmiausia pranešimo bloko bitai sudedami moduliu 2 su ankstesnio etapo šifravimo rezultato bitais, o paskui šifruojami.
- Grįžtamojo ryšio šifro režimas (angl. *Cipher feedback* – CFB): pagal raktą ir jau užšifruotus blokus generuojama pseudoatsitiktinių bitų seka ir sumuojama moduliu 2 su pranešimu.
- Grįžtamojo ryšio išvesties režimas (angl. *Output feedback* – OFB): pagal raktą ir jau sugeneruotus bitų blokus generuojama pseudoatsitiktinių bitų seka ir sudedama moduliu 2 su pranešimu.
- Skaitiklio režimas (angl. *Counter* – CTR): pagal raktą ir skaitiklį (bloko indeksą) generuojama pseudoatsitiktinių bitų seka ir sudedama moduliu 2 su pranešimu.

ECB režimas yra nesaugiausias, mat juo kiekvienas blokas šifruojamas atskirai, tad kenkėjas gali atkurti prasmingą tekstogramą iš atskirų žinomų tekstogramos ir šifrogramos blokų porų. Be to, net užšifruoti atskiri blokai gali perduoti kenkėjui tam tikrą informaciją apie tekstogramą. Pavyzdžiui, jei šifruojant grafinius duomenis atskiras taškas atitiks vieną duomenų bloką, ECB režimu kiekvienas tokios pat spalvos taškas bus atvaizduotas į kitą tokios pat spalvos tašką. Tokiu būdu visi pavaizduoti objektai skirsis tik spalva ir nebus užmaskuoti.

Siekiant to išvengti, buvo pasiūlyti kiti šifravimo režimai. Konkretaus režimo pasirinkimas priklauso nuo pritaikymo srities. Vieni jų veikia sparčiau, kitus galima vykdyti lygiagrečiai (pvz., CTR). Atsižvelgiant į duomenų svarbą, taip pat reikia įvertinti šifravimo klaidos įtaką visai šifrogramai.

5.2. KRIPTOGRAFINĖS ATAKOS

Visas kriptografines atakas kenkėjo atžvilgiu galima skirstyti į du tipus:

- Pagal kenkėjo galimybes, t. y. pagal skaičiavimo pajėgumus ir pagal tai, kokią informaciją kenkėjas turi.
- Pagal kenkėjo tikslus, t. y. pagal tai, ko jis siekia. Paprastai kenkėjo tikslas – paslapties atskleidimas, tačiau kartais tikslas gali būti konkretus, pvz., kenkėjas gali siekti atkurti raktą ar iššifruoti šifrogramą.

Pagal kenkėjo galimybes dažniausiai skiriamos šios atakos:

- Šifrogramos ataka (angl. *ciphertext-only attack*): kenkėjas bando „nulaužti“ kriptosistemą, perimdamas šifruotus pranešimus. Jis nieko nežino nei apie tekstogramas, nei apie raktą.
- Žinomos tekstogramos ataka (angl. *known-plaintext attack*): kenkėjas gauna šifrogramas atitinkančias tekstogramas (su nežinomu raktu) ir kriptosistemą bando „nulaužti“ naudodamasis šia papildoma informacija.
- Pasirinktos tekstogramos ataka (angl. *chosen-plaintext attack*): kenkėjas turi prieigą prie šifravimo sistemos ir gali užšifruoti norimas tekstogramas (su nežinomu raktu) bei gauti atitinkamas šifrogramas. Kriptosistemą mėginama „nulaužti“ bandant įvairius derinius.
- Pasirinktos šifrogramos ataka (angl. *chosen-ciphertext attack*): kenkėjas turi prieigą prie šifravimo sistemos ir gali iššifruoti bet kokias pasirinktas šifrogramas (su nežinomu raktu) bei gauti atitinkamas tekstogramas. Kriptosistemą bandoma „nulaužti“ naudojant įvairias gautas tekstogramų ir šifrogramų poras. Siekiama arba iššifruoti kitas šifrogramas, nesinaudojant šifravimo sistema, arba nustatyti šifravimo raktą.

Paprastai šifravimo sistemose taikomas Kerckhoffso principas: kenkėjas žino šifravimo algoritmą, o sistemos saugumas priklauso tik nuo šifravimo rakto slaptumo.

Jei kenkėjas žino, kokia šifravimo sistema naudojama, jis gali įvykdyti tik šifrogramos ataką, bandydamas atspėti šifravimo raktą. Tokia ataka gali būti vykdoma lygiagrečiai, t. y. kenkėjas gali lygiagrečiai perrinkti galimus variantus. Visų galimų raktų aibę pažymėkime K. Jei kenkėjas turi n skaičiavimo vienetų (procesorių ar kompiuterių) ir vienam rakto variantui patikrinti reikia t laiko, tai viena atakos realizacija vidutiniškai truks

$$\frac{|K|t}{2n}.$$

Ši ataka vadinama visišku raktų perrinkimu (angl. *exhaustive key search*), arba jėgos ataka (angl. *brute-force attack*). Aišku, šiai atakai įvykdyti nepakanka vien perrinkti visus galimus variantus. Norėdamas sėkmingam ją baigti, kenkėjas turi gebėti nustatyti, ar konkretus perrenkamas raktas yra teisingas, t. y. ar gauta tekstograma yra prasmingas simbolių rinkinys.

> ***5.16. Uždavinys.*** Tarkime, raktą $k \in K$ sudaro 128 bitai. Tada $|K| = 2^{128}$. Sakykime, kad vienas raktas perrenkamas ir patikrinamas per 1 μs, pasitelkus vieną skaičiavimo vienetą. Kiek reikės lygiagrečių skaičiavimo vienetų n, siekiant nustatyti šifravimo raktą vidutiniškai per 24 valandas?

Toliau apžvelgsime kriptografines atakas, kurios pastaraisiais metais darė didžiausią įtaką blokinių šifrų plėtros tendencijoms. Panagrinėsime tiesinę, diferencinę ir algebrinę kriptoanalizes.

5.2.1. TIESINĖ KRIPTOANALIZĖ

Tiesinė kriptoanalizė remiasi tiesinėmis Būlio funkcijų išraiškomis, kuriomis aproksimuojami S blokai. Aproksimacijos tikslumas priklauso nuo to, kiek S bloko funkcija yra nutolusi nuo tiesinių funkcijų poerdvio. Šią ataką pasiūlė M. Matsui, kuris "nulaužė" DES naudodamas 50 kompiuterių ir 2^{43} žinomų tekstogramų (Matsui, 1994). Tiesinė kriptoanalizė yra žinomos tekstogramos ataka, t. y. daroma prielaida, jog užpuolikas žino tam tikrą tekstogramų ir jas atitinkančių šifrogramų porų aibę, tačiau neturi priemonių konkrečioms tekstogramoms (ir atitinkamoms šifrogramoms) pasirinkti. Daugeliu atvejų priimtina laikyti, kad užpuolikas žino atsitiktinį tekstogramų ir jas atitinkančių šifrogramų porų rinkinį.

Tiesinė kriptoanalizė taip aproksimuoja netiesinę šifro dalį tiesinėmis lygtimis, kad jas atitinkantis tiesinis šifras daugeliu atvejų duotų tokius pat rezultatus kaip ir netiesinis, tačiau kai kuriais atvejais pateiktų ir klaidingus rezultatus. Kadangi tiesinės aproksimacijos atitikimas realiai funkcijai yra tikimybinis, ši aproksimacija reikalauja labai daug tekstogramų. Šis skaičius dar labiau padidėja aproksimacijos tikimybei esant labai arti 1/2. Šiuo atveju aproksimacija yra tik truputį pranašesnė už rakto bito spėliojimą.

Taikant tiesinę kriptoanalizę blokiniams šiframs, atliekami du žingsniai:

1. Randamos šifrui tinkamos tiesinės aproksimacijos.
2. Pritaikomas žinomos tekstogramos atakos algoritmas.

Pažymėkime *i*-tąjį duomenų bloko A bitą $A[i]$, o lygiškumo bitą (angl. *parity bit*), kuris apskaičiuojamas sudėjus visus bitus $A[i_1] \oplus A[i_2] \oplus ... \oplus A[i_k]$, – $A[i_1, i_2, ..., i_k]$. Paprastoms tiesinėms operacijoms, tokioms kaip sudėtis moduliu 2 su raktu ar bitų perstatymas, galima užrašyti nesudėtingas tiesines išraiškas, kurios galioja esant vienetinei tikimybei. Netiesiniams elementams (pvz., S blokams) bandoma rasti tiesines aproksimacijas, galiojančias esant tikimybei p, kuri maksimizuoja absoliutinį skirtumą $|p - 1/2|$. Paskui atskirų šifravimo operacijų aproksimacijos sujungiamos į šifro vieno raundo aproksimacijas. Atitinkamai sujungus gautas aproksimacijas, kenkėjas galiausiai gauna viso tokio tipo šifro aproksimacijas, susiejančias tekstogramos (T), šifrogramos (C) ir rakto(K) lygiškumo bitus:

$$T[i_1, i_2, ..., i_a] \oplus C[j_1, j_2, ..., j_b] = K[k_1, k_2, ..., k_c], \tag{5.1}$$

čia $i_1, i_2, ..., i_a, j_1, j_2, ..., j_b$ ir $k_1, k_2, ..., k_k$ žymi fiksuotų bitų pozicijas. Šios aproksimacijos užpuolikui naudingos tik tuomet, kai $p \neq 1/2$. Pavyzdžiui, DES šifrui M. Matsui nustatė tokias aproksimacijas esant $1/2 + 2^{-24}$ tikimybei. Naudojant šią aproksimaciją, vienam rakto lygiškumo bitui $K[k_1, k_2, ..., k_c]$ rasti galima pasitelkti nesudėtingą algoritmą, paremtą didžiausio tikėtinumo principu. Tarkime, duota N atsitiktinių tekstogramų aibė ir N_T žymi skaičių tekstogramų, su kuriomis kairioji (5.1) reiškinio pusė lygi 0. Tada:

- jei $(N_T - N / 2) \cdot (p - 1/2) > 0$, tai $K[k_1, k_2, ..., k_c] = 0$;
- priešingu atveju $K[k_1, k_2, ..., k_c] = 1$.

Tam, kad būtų galima teisingai atkurti lygiškumo bitą $K[k_1, k_2, ..., k_c]$ esant priimtinai tikimybei, M. Matsui parodė, kad tekstogramų skaičius N turi būti $|p - 1/2|^{-2}$ eilės. M. Matsui taip pat yra pateikęs ir veiksmingesnių tiesinės kriptoanalizės algoritmų, kuriais randama daugiau raktų bitų (Matsui, 1994).

Pirmuoju tiesinės kriptoanalizės žingsniu paprastų komponentų, tokių kaip S blokai, labiausiai paslinkta aproksimacija (tikimybė p yra labiausiai nutolusi nuo 1/2) nesunkiai gali būti rasta nuodugniu būdu. Tačiau bandant ekstrapoliuoti šį metodą visam šifrui, kyla praktinių problemų. Taikant tiesinę kriptoanalizę, pirmoji problema yra susijusi su tiesinės aproksimacijos tikimybės skaičiavimu. Iš esmės, norint apskaičiuoti tikimybę, reikia, kad užpuolikas (ar kriptoanalitikas) išbandytų visus įmanomus tekstogramų ir šifrogramų variantus. Akivaizdu, jog tai – neįmanoma. Ši problema sprendžiama darant tam tikras prielaidas ir remiantis vadinamąja *sukaupimo lema* (angl. *piling-up lemma*).

5.17. Lema (sukaupimo lema). Tarkime, duota n nepriklausomų atsitiktinių kintamųjų $X_1, X_2, ..., X_n \in \{0, 1\}$. Sumos $X = X_1 \oplus X_2 \oplus ... \oplus X_n$ tikimybės nuokrypis $e = p - 1/2$ apskaičiuojamas taip:

$$e = 2^{n-1} \prod_{j=1}^{n} e_j ; \tag{5.2}$$

čia e_1, e_2, ..., e_n yra atitinkami kintamųjų X_1, X_2, ..., X_n tikimybių nuokrypiai nuo 1/2 (pvz., $e_1 = p_1 - 1/2$, kur p_1 yra tikimybė, kad kintamasis X_1 bus lygus 1) (van Tilborg, 2005).

Siekiant apskaičiuoti tiesinės aproksimacijos tikimybę pasitelkus sukaupimo lemą, bendroji aproksimacija išreiškiama atskirų šifro dalių tiesinių aproksimacijų grandine. Ši grandinė vadinama *tiesine charakteristika*. Darant prielaidą, kad šių dalinių aproksimacijų nuokrypiai yra statistiškai nepriklausomi ir lengvai apskaičiuojami, bendrąjį nuokrypį galima apskaičiuoti pagal (5.2) formulę.

Nors sukaupimo lema daugumoje praktinių atvejų pateikia labai gerus įvertinimus, tačiau būtina pasakyti, kad kai aproksimacijos nėra griežtai nepriklausomos, galima sulaukti netikėtų padarinių. Realus tikimybės nuokrypis gali būti ir gerokai mažesnis, ir daug didesnis už lemos įvertį.

Sėkmingai blokinio šifro tiesinei atakai pakanka, kad būtų vienintelė tiesinė charakteristika, kurios tikimybė p būtų pakankamai nutolusi nuo 1/2. Todėl pagrindinis šifro kūrėjo tikslas – užtikrinti, kad tokia charakteristika neegzistuotų. Tai paprastai daroma pasirenkant didelio netiesiškumo S blokus ir argumentuojant, kad bitų paskirstymas šifre susieja visas charakteristikas su pakankamai dideliu minimaliu S blokų skaičiumi.

Toks metodas suteikia gerų euristinių argumentų, pagrindžiančių šifro atsparumą, tačiau, siekiant visiškai įrodyti šifro atsparumą tiesinei atakai, reikia taip pat įvertinti įvairius fenomenus, pavyzdžiui, tiesinio lukšto efektą (angl. *linear hull effect*) (Nyberg, 1994). Tokią panašių į DES šifrų (Feistelio) saugumo analizę atliko L. Knudsenas ir K. Nybergas (Nyberg, 1994)(Nyberg, et al., 1995). Jų gauti rezultatai leido sukurti daugelio šiuo metu vartojamų šifrų struktūrą: MISTY (KASUMI), *Rijndael/AES*, *Camellia* ir kt. Vėliau buvo paskelbti ir analogiški SP tinklų šifrų rezultatai (Hong, et al., 2000).

Šifro atsparumą tiesinei kriptoanalizei galima įvertinti ir nagrinėjant atskirų jo S blokų savybes. Skiriami šie pagrindiniai S blokų atsparumo tiesinei kriptoanalizei kriterijai:

- Netiesiškumas (N_F).
- Tiesinis potencialas (LP_F) – didžiausia tiesinio profilio, kuriuo įvertinamas tiesinę aproksimaciją tenkinančių porų skaičius, reikšmė (Pommerening, 2005).

Tarp Būlio funkcijos F netiesiškumo N_F ir tiesinio potencialo LP_F yra toks sąryšis:

$$N_F = 2^{n-1}(1 - \sqrt{LP_F}\ ).$$

S bloko (Būlio funkcijos) tiesinis potencialas tenkina nelygybę:

$$\frac{1}{2^n} \le LP_F \le 1.$$

Kuo mažesnis tiesinis potencialas, tuo atsparesnis S blokas.

Skaitytojams, norintiems detaliau susipažinti su tiesine kriptoanalize, rekomenduojame pastudijuoti šią medžiagą (Heys, 2001).

5.2.2. DIFERENCINĖ KRIPTOANALIZĖ

Diferencinė kriptoanalizė – tai simetrinių kriptografinių primityvų analizės metodika, itin gerai pritaikoma blokinių šifrų kriptoanalizei. Pirmą kartą ją viešai paskelbė E. Bihamas ir A. Shamiras 1990 m. apribotų raundų skaičiaus DES šifro atakos atveju (Biham, et al., 1991).

Tarkime, t žymi dvejetainiu kodu užkoduotą tekstogramą, t. y. $t \in \{0, 1\}^n$, o c – ją atitinkančią šifrogramą nežinomu šifravimo raktu k:

$$c = E_k(t).$$

Sakykime, t^* žymi antrąją tekstogramą, o c^* – atitinkamą šifrogramą tuo pačiu raktu k:

$$c^* = E_k(t^*).$$

Tekstogramų skirtumą apibrėžkime taip:

$$\Delta t = t \oplus t^*,$$

o šifrogramų skirtumą – taip:

$$\Delta c = c \oplus c^*;$$

čia $\oplus$ žymi bitų sudėtį moduliu 2 (operacija XOR). Taip pat apibrėžiamas ir bet kurių tarpinių šifravimo duomenų x, naudojamų šifravimo algoritme (pvz., tarp raundų išvesties reikšmių), skirtumas:

$$\Delta x = x \oplus x^*.$$

Diferencinė kriptoanalizė nagrinėja, kaip minėti skirtumai (šiuo atveju – sudėtis moduliu 2) kinta įvairiuose šifro raunduose ir šifravimo operacijose. Tiesinės ir afiniosios operacijos nekeičia skirtumų arba jų keitimus galima nuspėti. Bitų perstatymo operacija, perrikiuojanti x bitus į $P(x)$, taip pat perrikiuoja ir skirtumus, t. y.

$$P(\Delta x) = P(x) \oplus P(x^*).$$

Pasirinkus tam tikrus duomenų bitus, taip pat pasirenkami ir atitinkami skirtumai. Kintamųjų x ir y sudėties moduliu 2 operacija taip pat sudeda moduliu 2 ir jų skirtumus:

$$\Delta x \oplus \Delta y = (x \oplus y) \oplus (x^* \oplus y^*).$$

Nesunku pastebėti, kad į duomenis įterpto tarpinio (raundinio) rakto bitai gali būti atmesti skirtumų atžvilgiu. Pavyzdžiui, jei tarpinis raktas sudedamas su tarpiniu kintamuoju moduliu 2, t. y. $y = x \oplus k$, tai šifruojant antrąją porą (x^*, y^*) gaunama $y^* = x^* \oplus k$. Tokiu atveju raktų įterpimo operacijos skirtumas lygus

$$\Delta y = y \oplus y^* = (x \oplus k) \oplus (x^* \oplus k) = \Delta x,$$

t. y. šis skirtumas nepriklauso nuo tarpinio rakto. Tokiu būdu galima nepaisyti šios raktų įvedimo operacijos, skaičiuojant galimus šifravimo funkcijos skirtumus.

Galima skaičiuoti ir netiesinių šifravimo operacijų, tokių kaip S blokai, skirtumus. Akivaizdu, kad jei S blokų įvesčių skirtumas – 0, tai jie yra lygūs, išvestys taip pat bus lygios, t. y. jų skirtumas bus nulinis. Kai skirtumas nenulinis, negalima tiksliai pasakyti, koks bus išvesties skirtumas, nes jis priklausys nuo skirtingų įvesčių. Tačiau galima nustatyti išvesties skirtumų statistinę informaciją, priklausančią nuo įvesties skirtumų.

Taikant diferencinę kriptoanalizę, sudaroma kiekvieno S bloko (ar kitos netiesinės operacijos) *skirtumų skirstinio lentelė* (angl. *difference distribution table*). Šios lentelės eilutės – visi galimi įvesties skirtumai, stulpeliai – visi galimi išvesties skirtumai, o lentelės įrašai rodo, kiek įvesties ir išvesties porų atitinka konkrečius skirtumus. Pildant skirtumų skirstinio lentelę, perrenkamos visos įmanomos įvesties duomenų poros, turinčios konkretų skirtumą. Pavyzdžiui, jei įvestis yra 4 bitų ilgio, tai iš viso bus $16 = 2^4$ skirtingų tekstogramų. Šiuo atveju bus ir 16 skirtingų skirtumų, todėl skirtumų skirstinio lentelę sudarys 16 eilučių. Stulpelių skaičius priklausys nuo išvesties bitų skaičiaus. Norint visiškai užpildyti lentelę, reikia apskaičiuoti S bloko išvestį su visomis tekstogramomis ir visus galimus įvesties bei išvesties skirtumus. Jei ir išvestis bus 4 bitų, iš viso reikės apskaičiuoti 128 įvesčių ir 128 išvesčių skirtumus, nes kiekvieną skirtumą atitinka $2^{4-1} = 8$ poros.

Diferencinė kriptoanalizė apibrėžia skirtumų galimų kitimų charakteristikas įvairiuose šifro raunduose ar operacijose. Kiekviena charakteristika atitinka tekstogramos skirtumą, kuriam ji prognozuoja skirtumą kituose raunduose. Tekstogramų poros, kurių pradinis skirtumas ir šifravimo funkcijos išvesties skirtumas yra tiksliai toks, kokį rodo charakteristika, yra vadinamos *geromis* (angl. *right pairs*). Visos kitos poros vadinamos *blogomis* (angl. *wrong pairs*). Tikimybė, kad charakteristika teisingai nurodys skirtumą, t. y. kad atsitiktinė pora, turinti reikiamą skirtumą, bus gera, priklauso nuo kiekvieno S bloko (ar operacijos) įvesties ir išvesties skirtumų ryšių. Viso šifro bendra tikimybė, kad su pasirinktu tekstogramos skirtumu bus gautas reikiamas šifrogramos skirtumas, lygi visų raundų tikimybių sandaugai, jei visos tikimybės yra nepriklausomos (dažniausiai taip ir būna). Priešingu atveju gaunama tik tikimybės aproksimacija.

Dydis $P(f(t + \Delta t) = f(t) + \Delta c)$ rodo tikimybę, kad atsitiktinė tekstograma t, turinti fiksuotą skirtumą Δt, bus atvaizduota į šifrogramą, turinčią fiksuotą skirtumą Δc.

Turint tikėtiną priešpaskutinio raundo tarpinių duomenų skirtumą, gali būti įmanoma nustatyti nežinomą raktą, pasitelkus statistinę analizę. Diferencinė kriptoanalizė yra *pasirinktos tekstogramos ataka*, kuri vykdoma dviem etapais:

- Duomenų rinkimo metu užpuolikas reikalauja užšifruoti didelį kiekį tekstogramų porų, kurių skirtumai parenkami pagal nustatytą skirtumų kitimo charakteristiką.
- Duomenų analizės metu užpuolikas išgauna raktą iš gautų šifrogramų.

Tarkime, charakteristikos tikimybė yra p, t. y. laukiama, kad p dalis porų bus geros. Tokiu būdu priešpaskutiniame raunde p dalies porų skirtumai atitiks charakteristiką. Paprasčiausias (neefektyvus) būdas nustatyti paskutinio raundo raktą – išbandyti visus galimus variantus. Kiekvieno galimo rakto tinkamumui patikrinti visoms šifrogramoms atliekamas vienas iššifravimo raundas ir apskaičiuojami kiekvienos poros duomenų skirtumai iki paskutinio raundo įvesties. Tikėtina, kad, esant neteisingiems rakto variantams, charakteristikos nurodytas skirtumas pasitaikys retai, o esant teisingam raktui, šis skirtumas turėtų pasitaikyti p ar didesnėje dalyje porų. Jei tikimybė p nėra per maža, tikėtina, kad būtent esant teisingam rakto variantui reikiamas skirtumas pasitaikys dažniausiai.

Norint diferencinei kriptoanalizei gauti pakankamai daug gerų tekstogramos ir šifrogramos porų, paprastai reikia $k(1/p)$ pasirinktų tekstogramų porų; čia p – atakai naudojamos charakteristikos tikimybė. Šis užšifruotų duomenų kiekis gali būti labai didelis (pavyzdžiui, DES kriptoanalizės atveju reikia 2^{47} pasirinktų tekstogramų blokų). Todėl dažniausiai duomenų surinkimo etapas yra sudėtingesnis už duomenų analizės etapą. Vien dėl itin didelio pasirinktų tekstogramų skaičiaus ataka gali būti nepraktiška, nes didesnė atakos skaičiavimų dalis perduodama užpultajai šaliai, kuri turi užšifruoti daug tekstogramų, kad užpuolikas galėtų realizuoti savo ataką. Dėl to įprasta diferencinės atakos sudėtingumą vertinti reikalingų pasirinktų tekstogramos ir šifrogramos porų skaičiumi.

Geriausias būdas apsisaugoti nuo diferencinės atakos – sumažinti geriausios charakteristikos (ar skirtumo) tikimybę. Šifro kūrėjas, norėdamas įrodyti, kad diferencinė ataka negali būti taikoma, taip apriboja geriausios charakteristikos (ar skirtumo) tikimybę p, kad skaičius $1/p$ būtų didesnis už tam tikrą saugumo slenkstį, net didesnis už tekstogramų aibės galią (tokiu atveju net perrinkus visą tekstogramų aibę neįmanoma įvykdyti atakos). Šios ribos nustatymas leidžia įrodyti šifro saugumą diferencinės kriptoanalizės atžvilgiu. Tada sakoma, kad sukurtas šifras turi įrodomos apsaugos nuo šios atakos savybę.

Šifro atsparumą diferencinei kriptoanalizei galima įvertinti ir nagrinėjant atskirų jo S blokų charakteristikas. Svarbiausias kriterijus, rodantis S bloko atsparumą, yra diferencinis potencialas (DP_F) (Pommerening, 2005). Jį apibūdina didžiausia diferencinio profilio, kuriuo įvertinamas fiksuotus skirtumus tenkinančių porų skaičius, reikšmė. Diferencinis potencialas tenkina šią nelygybę:

$$\frac{1}{2^m} \le DP_F \le 1.$$

Kuo mažesnis diferencinis potencialas, tuo atsparesnis S blokas.

Nors ir atrodo, kad tiesinė bei diferencinė atakos naudoja skirtingas šifro (S blokų) charakteristikas, tačiau iš tikrųjų egzistuoja ryšys tarp šių kriptoanalizių (Chabaud, et al., 1995).

Galimi įvairūs diferencinės atakos patobulinimo būdai, tačiau jų nenagrinėsime. Pateiksime tik nuorodas į atitinkamą literatūrą: kelių skirtingų charakteristikų sujungimas į vieną (Biham, et al., 1991), sutrumpinti skirtumai (Knudsen, 1995), aukštesnių eilių skirtumai (Knudsen, 1995). Taip pat yra tokių atakų, kurios skirtumus pasitelkia kaip sudedamąsias dalis ir juos sujungia įvairiais būdais: bumerango (Wagner, 1999), sustiprinta bumerango (Kelsey, et al., 2000) ir stačiakampio atakos (Biham, et al., 2002).

Skaitytojams, norintiems detaliau susipažinti su diferencine kriptoanalize, rekomenduojame pastudijuoti medžiagą (Heys, 2001).

5.2.3. ALGEBRINĖ KRIPTOANALIZĖ

2002 m. buvo paskelbtas naujas kriptoanalizės metodas, vadinamas algebrine kriptoanalize. Jis pakeitė nusistovėjusį požiūrį į tradicinį kriptografinį atsparumą (Courtois, et al., 2002). Pirmosios algebrinės kriptoanalizės idėjos pasirodė 1983 m. (Schaumüller-Bichl, 1983). Nors kai kurie šifrai pasižymi gerais kriterijais įprastų atakų atžvilgiu, tačiau paaiškėjo, kad juos gali pažeisti algebrinė kriptoanalizė. Šie šifrai realizuojami eksponentinėmis šifravimo funkcijomis, tokiomis kaip *Gold*, *Kasami*, atvirkštine ir kt. (Cheon,

et al., 2004). Šios funkcijos gali būti aprašomos gana paprastomis algebrinėmis lygtimis, taigi jos yra jautrios algebrinei kriptoanalizei.

Taikant algebrinę kriptoanalizę, blokinį šifrą arba atskirus jo S blokus galima aprašyti kaip įvesties ir išvesties sistemą algebrinėmis lygtimis, susiejančiomis sistemos įvesties, išvesties ir raundinio rakto kintamuosius. Gavus šią lygčių sistemą ir turint vieną ar daugiau išvesties bei šifruotų duomenų porų, galima ieškoti gautos lygčių sistemos sprendinio ir nustatyti šifravimo raktą. Kitaip nei tradicinės kriptografinės atakos atveju, algebrinei atakai nereikia daug tekstogramų ir šifrogramų porų. Jai užtenka vienos ar kelių porų, o tai gerokai didina algebrinės kriptoanalizės efektyvumą. Pasak N. Courtoiso, šifrų kūrimas jau niekada nebus toks, koks buvo iki šiol, nes seni blokinių šifrų saugumo postulatai keičiasi (Courtois, et al., 2005).

Pagrindiniai nauji postulatai:

1. Šifro sudėtingumas nedidėja eksponentiškai šifravimo raundų skaičiaus atžvilgiu.
2. Kriptoanalizei gali būti reikalinga nedaug tekstogramų ir šifrogramų porų (pavyzdžiui, 1 ar 2).

Nors AES šifras yra beveik optimalus tiesinės ir diferencinės kriptoanalizės atžvilgiu, tačiau jis aprašomas gana paprastomis algebrinėmis lygtimis. Iki šiol šis šifras nėra sukompromituotas, tačiau NESSIE projekto vadovai perspėja, kad AES šifrui būdinga gana paprasta algebrinė struktūra (Preneel, et al., 2004).

Trumpai panagrinėsime šifrų aprašymo metodiką algebrinėmis lygtimis. Remiantis AES šifro aprašu, supaprastintai AES raundą galima užrašyti tokiomis lygtimis:

$$y = g(x) = Ax + b, \tag{5.3}$$
$$z = f(y) = y^{-1}; \tag{5.4}$$

čia g – tiesinė funkcija vektorinėje erdvėje F_2^8, f – netiesinė funkcija, išreiškiama atvirkštine eksponente Galua lauke $GF(2^8)$.

Iš (5.4) lygties gauname:

$$zy = 1, \tag{5.5}$$

o vietoj y įrašę jo reikšmę iš (5.3) lygties gauname

$$z(Ax + b) = 1. \tag{5.6}$$

Kintamasis z gali būti interpretuojamas kaip S bloko išvestis ir kaip tarpiniai šifruoti duomenys. Kintamasis x sudedamas moduliu 2 su išeities duomenimis ir šifravimo raktu. Kaip matome, (5.6) lygtis yra bitiesinių polinominių lygčių sistema virš lauko $GF(2^8)$, susiejanti įvesties, išvesties kintamuosius ir raundinius raktus.

Tiesiogiai išspręsti (5.6) lygčių sistemos negalima, nes nežinomųjų yra daugiau nei lygčių. Tačiau N. Courtoisas pasiūlė metodą, kaip padidinti lygčių skaičių ir gauti jų daugiau nei nežinomųjų (Courtois, et al., 2002). Gautos lygtys yra priklausomos, tačiau netiesiškai, todėl sprendiniui gauti galima pritaikyti ištiesinimo metodus bei efektyvų Gauso algoritmą. Papildomų lygčių sudarymas pagrįstas vadinamaisiais XL arba patobulintais XSL metodais, kurie priskiriami žinomos tekstogramos tipo atakoms (Courtois, et al., 2002)(Cid, et al., 2005). XSL atakos metu lygtys sistemoje dauginamos iš specialiai parinktų monomų, kurių eilė neviršija tam tikro nustatyto skaičiaus D – 2. Monomai parenkami taip, kad atliekant šią daugybą atsirastų kuo daugiau monomų, jau esančių kitose lygtyse. Tokiu būdu atliekama tam tikra optimizacija, siekiant išvengti daugelio papildomų kintamųjų atsiradimo. Atliekant ištiesinimo procedūrą, gaunama tiesinių lygčių sistema.

Naujų lygčių sudarymą galima pailiustruoti pavyzdžiu. Turėdami (5.3) lygtį, ją padauginame, pavyzdžiui, iš $y, y^3, ..., z, z^3, ...$ (Cheon, et al., 2004). Gauname tokias papildomas lygtis:

$$zy^2 = y,$$
$$zy^3 = y^2, \ldots$$

Siūlomi du AES šifro atakų scenarijai. Mes apžvelgsime antrąjį, kuriame naudojamas raktų grafikas (angl. *key schedule*) ir reikalaujama bent vieno žinomo išeities teksto (Courtois, et al., 2002). Papildomų lygčių sudarymas problemos neišsprendžia, nes jos turi būti tinkamiausios ištiesinti. Tokios lygtys yra daugianarės kvadratinės (angl. *multivariate quadratic* – MQ), o jų sudarymas vadinamas MQ ataka.

Remiantis šia ataka, AES-128 šifrui gauta 8000 kvadratinių lygčių sistema su 1600 kintamųjų. Kaip matome, ši sistema yra perteklingai apibrėžta (angl. *overdefined*) kintamųjų atžvilgiu. Tačiau laikoma, kad MQ lygčių sprendimas priklauso NP pilnųjų uždavinių klasei. Todėl labai stengiamasi rasti sprendimo algoritmus, kurių sudėtingumas būtų bent jau subeksponentinis.

Apžvelgsime AES-128 šifro antrojo tipo XSL ataką (Cid, et al., 2005). Pradinių lygčių skaičius – 1600, kintamųjų – 3200, monomų – 11200. Tačiau kol kas konstatuota, kad XSL algoritmas negali išspręsti gautos lygčių sistemos. Nepaisant to, stengiamasi sudaryti polinominių algebrinių lygčių sistemas ir kitiems žinomiems šiframs, pasižymintiems gerais atsparumo tiesinei ir diferencinei kriptografinėms atakoms kriterijais.

A. Biryukovo straipsnyje pateiktas blokinių šifrų *Khazad, Misty 1, Kasumi, Camellia-128*, AES-128 ir *Serpent-128* palyginimas pagal tai, kiek išeities polinominių lygčių galima jiems sudaryti (Biryukov, et al., 2003). Lygtys buvo sudaromos individualiai kiekvienam šifro komponentui, o paskui sujungtos į vieną sistemą. Gana paprasta AES algebrinė struktūra leidžia šį šifrą aprašyti labai išretinta kvadratinių lygčių sistema virš lauko $GF(2^8)$ (Murphy, et al., 2002). Šifrą galima įterpti į praplėstą šifrą, vadinamą didele šifravimo sistema (angl. *Big encryption system* – BES), pakeičiant kiekvieną baitą jo vektoriniu jungtiniu elementu. Tokiu būdu buvo sudarytos AES ir *Camellia* šifrų lygtys. Remiantis gautais rezultatais, beveik visi tradiciniai šifrai, pagrįsti fiksuoto laipsnio eksponentinėmis funkcijomis, gali būti aprašomi daugianarėmis polinominėmis lygtimis (angl. *multivariate polynomial equations*) o tai yra rimtas pavojus sprendžiant šių šifrų saugumo klausimus.

Tiriant algebrinių atakų veiksnumą pasitelkus XL, XSL atakas ir jų modifikacijas, pastebėta, kad atakos sudėtingumas priklauso ne tik nuo kintamųjų ir monomų skaičiaus, bet ir nuo balanso tarp lygčių skaičiaus ir monomų skaičiaus, kuris yra tose lygtyse. Todėl spręsti lygčių sistemas, kurios yra perteklingai apibrėžtos ir išretintos, gerokai lengviau nei kitas tokių pat matmenų sistemas.

Dėl to buvo suformuoti du algebrinio imuniteto (AI) kriterijai:

$$\Gamma(r, n, t) = (t/n)^{\lceil t/r \rceil},$$
$$\Gamma'(r, n, t) = ((t-r)/n)^{\lceil (t-r)/n \rceil};$$

čia r – lygčių skaičius; n – baigtinio lauko eilė, virš kurio apibrėžta lygčių sistema; t – monomų skaičius lygčių sistemoje.

Šie du kriterijai yra susieti su dviem XSL atakų versijomis (Courtois, et al., 2005). Manoma, kad algebrinei atakai atsparus S blokas turi tenkinti nelygybę $\Gamma > 2^{32}$.

Nenuostabu, kad atsiradus naujai kriptoanalizės atakai atsirado ir naujų kriterijų, nusakančių šifro atsparumą šiai atakai. Tačiau įdomu, kaip šie nauji kriterijai dera su kriterijais, nusakančiais šifro atsparumą tradicinėms atakoms, tokioms kaip tiesinė ir diferencinė kriptoanalizė? Čia iškyla didelis prieštaravimas.

APN funkcijų atsparumas tradicinei kriptoanalizei yra artimas optimaliai reikšmei. Tačiau tai šių APN funkcijų atsparumas algebrinei kriptoanalizei yra labai blogas (Cheon, et al., 2004). Iš to galime daryti išvadą, kad „labai geros“ eksponentinės funkcijos, pasižyminčios labai gerais kriterijais (puikiu netiesiškumu, t. y. priklauso ANP funkcijų klasei) tiesinės ir diferencinės kriptoanalizių atžvilgiu, yra „labai blogos“ algebrinės kriptoanalizės atžvilgiu.

Plačiau apie šifrų aprašymo metodiką algebrinėmis lygtimis galima paskaityti kituose šaltiniuose (Murphy, et al., 2002)(Biryukov, et al., 2003)(Cheon, et al., 2004)(Courtois, et al., 2005).

5.3. BLOKINIŲ ŠIFRŲ STANDARTIZAVIMAS

Šio skyriaus pradžioje jau minėjome, kad geriausias šifravimo algoritmo įvertinimas – jo pateikimas kriptografinės bendruomenės analizei. Įvairios vyriausybinės ir tarptautinės organizacijos pagal šį principą yra rengusios įvairius šifrų standartizavimo projektus. Šių projektų tikslas – pateikti rekomendacijas potencialiems šifrų vartotojams. Taip pat jais siekiama paskatinti naujų šifrų kūrimą ir jau egzistuojančių tobulinimą. Projektų metu pasiūlyti šifrai analizuojami, tikrinamas jų atsparumas žinomoms atakoms,

įvertinamos skaičiavimų sąnaudos. Tuo paprastai užsiima specialiai pasamdyti specialistai, tačiau savo tyrimus gali atlikti visi norintieji. Jau tapo įprasta, kad kiekvieno projekto metu organizuojamos konferencijos, kuriose pristatomi pasiūlyti šifrai, gauti kriptoanalizės rezultatai ir kitų etapų planai.

Garsiausi blokinių šifrų tyrimų, rekomendacijų ir standartizavimo projektai:

- AES – NIST organizuotas atvirasis konkursas (1997–2000 m.), kurio metu buvo pasirinktas *Rijndael* algoritmas (NIST, 2001).
- NESSIE (angl. *New European Schemes for Signatures, Integrity and Encryption*) – Europos Komisijos remtas projektas (2000–2003 m.), skirtas kriptografinių algoritmų analizei ir rekomendacijoms (NESSIE Consortium, 2003).
- ECRYPT STVL – ECRYPT (angl. *European Network of Excellence for Cryptology*) projekto komitetas, sudarytas simetrinių kriptografinių sistemų analizei (ECRYPT, 2005). Tai Europos Komisijos finansuojamas ketverių metų projektas, pradėtas 2004 m.
- CRYPTREC – tai Japonijos vyriausybės remtas projektas, analogiškas NESSIE ir AES. CRYPTREC komitetas pradėjo savo veiklą 2000 m. ir ją tęsia iki šiol (CRYPTREC, 2007).

6. SRAUTINIO ŠIFRAVIMO SISTEMŲ KŪRIMAS IR KRIPTOANALIZĖ

Atsižvelgiant į šiuolaikinės kriptografijos laimėjimus ir srautinius šifrus realizuojančią mikroprocesorių techniką, formalus srautinio šifro apibrėžimas tampa gana dirbtinis. Riba tarp srautinio ir blokinio šifrų tampa labai neraiški. Skaitytojams, norintiems apie tai sužinoti daugiau, siūlome paskaityti A. Shamiro pranešimą, parengtą „AsiaCrypt 2004“ konferencijai.

Vis dėlto norėdami išskirti srautinį šifrą, galime pateikti tokį srautinio šifro apibrėžimą:

6.1. Apibrėžimas. Srautinis šifras yra simetrinis šifras, kuriuo užšifruojamas duomenų srautas.

Šiuo apibrėžimu iš karto atskiriame srautinį šifrą nuo blokinio, skirto duomenų rinkmenoms užšifruoti. Ir blokinis, ir srautinis šifrai priklauso simetrinių šifrų klasei, tačiau jų taikymas skiriasi. Srautiniam šifrui keliami du papildomi reikalavimai, kuriuos galima nusakyti šiomis savybėmis.

6.2. Savybė. Srautinis šifras turi užtikrinti pakankamą užšifravimo spartą.

6.3. Savybė. Srautinis šifras turi užtikrinti darbą prijungties (angl. *on-line*) režimu.

NESSIE projekte nėra pateiktos rekomendacijos, kokį srautinį šifrą galima saugiai naudoti. Kadangi visi pasiūlyti šifrai turėjo rimtų saugumo problemų. Dėl to EUROCRYPT tinklas 2004 m. lapkričio mėn. paskelbė naują projektą eSTREAM, kuriuo siekiama nustatyti tinkamą ir plačiai pritaikomą srautinį šifrą: http://www.ecrypt.en.org/stream/. Projektą numatoma baigti 2008 m. sausio mėn. Laukiami dviejų tipų pasiūlymai šiam konkursui:

1. Didelės spartos srautiniai šifrai, skirti programinei realizacijai.
2. Srautiniai šifrai, skirti aparatinei realizacijai su ribotais skaičiavimo ištekliais, t. y. su ribota atmintine, loginių elementų skaičiumi ir energijos sąnaudomis.

Konkursui galima pateikti abiejų tipų siūlymus. Pirmojo tipo šifro siūlymas priimamas tuo atveju, jei užšifravimo sparta yra didesnė už AES-128 užšifravimo spartą CTR režimu.

Kadangi vienoje paskaitoje neįmanoma išsamiai apžvelgti pagrindinių perspektyvių srautinių šifrų sudarymo metodų, pasistengsime daugiau panagrinėti tuos, kurie pagrįsti blokinių šifrų architektūra. Tai leis geriau susisteminti medžiagą ir įsigilinti į šiuos šifrus.

Iš pradžių panagrinėkime Vernamo šifrą, kuris tam tikromis sąlygomis gali būti absoliučiai saugus.

Tarkime, šifruojama tekstograma yra $t = \{t_i\} \in \{0,1\}^n$ yra n – bitų eilutė. Tada užšifravimo raktas bus užšifravimo gama $\gamma = \{\gamma_i\} \in \{0,1\}^n$, t. y. tokio pat ilgio bitų eilutė. Šifrogramą pažymėsime raide $c = \{c_i\} \in \{0,1\}^n$.

Vernamo šifru kiekvienas šifrogramos bitas c_i apskaičiuojamas taip:

$$c_i = t_i \oplus \gamma_i; \tag{6.1}$$

čia $\oplus$ yra bitų sudėtis moduliu 2, arba XOR operacija.

Šifrogramai dešifruoti pasitelkiama formulė:

$$c_i \oplus \gamma_i = t_i \oplus \gamma_i \oplus \gamma_i = t_i \oplus 0 = t_i. \tag{6.2}$$

Matome, kad užšifruoti tekstogramą ir iššifruoti šifrogramą naudojama gana paprasta ir algoritmiškai efektyviai realizuojama XOR operacija. Ši operacija pasižymi dar dviem labai svarbiomis savybėmis: ji yra subalansuota ir pati sau atvirkštinė. Norėdami tuo įsitikinti, užrašykime XOR operaciją kaip dviejų argumentų funkciją:

$$t_i \oplus \gamma_i = f(t_i, \gamma_i) = f_{\gamma_i}(t_i). \tag{6.3}$$

Funkcijos $f(t_i, \gamma_i)$ teisingumo lentelėje matome, kad dviem (t_i, γ_i) reikšmėms (0, 0) ir (1, 1) funkcijos reikšmė $f(t_i, \gamma_i) = 0$, o likusioms dviem reikšmėms (0, 1) ir (1, 0) – 1. Remiantis apibrėžimu, tai ir yra funkcijos ***subalansuotumo*** savybė, t. y. atsitiktinai pakeitus argumento reikšmes, funkcijos reikšmė gali būti lygi tiek 0, tiek 1. Priminsime, kad subalansuotumo reikalavimas yra vienas svarbiausių sudarant blokinių šifrų S blokų funkcijas.

Jau buvo minėta, kad pati sau atvirkštinė funkcija yra vadinama ***involiucija***. Remiantis apibrėžimu, involiucijai galioja tokia lygybė:

$$f_{\gamma_i}(f_{\gamma_i}(t_i)) = t_i . \tag{6.4}$$

Kadangi $f_{\gamma_i} = f_{\gamma_i}^{-1}$, tai $f_{\gamma_i}(f_{\gamma_i}(t_i)) = f_{\gamma_i}^{-1}(f_{\gamma_i}(t_i)) = (f_{\gamma_i}^{-1} \circ f_{\gamma_i})(t_i) = I(t_i) = t_i$. Čia $\circ$ žymėjome funkcijų kompoziciją, o I – tapačiąją funkciją.

Vernamo šifrui realizuoti galima naudoti ir kitas involiucijas, tačiau tai nebūtina, nes vargu ar galima sudaryti kitą funkciją, kuri būtų efektyviau algoritmiškai realizuojama.

C. Shannonas įrodė, kad Vernamo šifras yra absoliučiai saugus, jei šifravimo gama tenkina šias sąlygas:

1. Šifravimo gamos (rakto) ilgis turi būti lygus tekstogramos ilgiui, t. y. $|\gamma| = |t|$.
2. Šifravimo gama (raktas) susideda iš atsitiktinių tolygiai pasiskirsčiusių ir statistiškai nepriklausomų bitų (skaičių) γ_i.
3. Šifravimo gama (raktas) turi būti naudojama tik kartą.

Jei Shannono teoremos sąlygos tenkinamos, jokia informacija apie tekstogramą *t* neperduodama šifrogramai *c*, t. y. turint tik *c*, nieko negalima pasakyti, koks galėtų būti *t* (pvz., kokį tikimybinį skirstinį galėtų turėti *t*). Dar kartą pakartosime, kad tam didžiausios įtakos turi XOR operacijos subalansuotumas.

Gali susidaryti įspūdis, kad Vernamo šifras yra labai paprastas ir lengvai realizuojamas. Anaiptol, tai nėra taip paprasta, todėl ir šis faktas turėtų sudominti šią temą studijuojančius žmones.

Kadangi šiuolaikinė elektronika leidžia efektyviai atlikti operacijas su bitais ir žodžiais (baitų grupėmis), tai kartu su bitų operacijomis nagrinėsime ir operacijas su skaičiais.

Pasirinkus Vernamo šifrą, sudėtingiausia gauti šifravimo gamą, tenkinančią Shannono teoremos sąlygas. Kaip jau minėjome, nuo šiol sakysime, kad gamą γ ir tekstogramą *t* sudaro arba bitų, arba skaičių sekos.

Atrodytų, atsitiktiniams skaičiams generuoti galima rasti principinį sprendimo būdą, pavyzdžiui, naudojant fizikinį šiluminio triukšmo generatorių, pagrįstą elektriniu triukšmo signalu rezistoriuje, tranzistoriuje ir t. t., arba registruojant reliktinį visatos šiluminį spinduliavimą, atsiradusį nuo Didžiojo sprogimo momento, kai susikūrė mūsų visata.

Tačiau toks sprendimas yra labai neefektyvus ir reikalauja didelių materialinių sąnaudų. Pavyzdžiui, visam DVD diskui užšifruoti reikia sugeneruoti atitinkamą kiekį gamos elementų. Tačiau didžiausia problema ta, kad šią gamą reikia saugiai laikyti ir – svarbiausia – saugiai paskirstyti tarp kriptografinių subjektų, o tai yra labai sudėtinga.

Norint rasti veiksmingesnį sprendimo būdą, pirmiausia būtina suprasti, kas yra „tikra“ atsitiktinių skaičių seka, o paskui mėginti šią seką gauti ne fizikiniu, o algoritminiu būdu. Būtų labai naudinga turėti kokį nors ne itin ilgą slaptąjį skaičių, kaip tam tikrą sėklą (*seed*), ir iš jo gauti kokio norima ilgio atsitiktinių skaičių seką, pasitelkus kokį nors algoritmą. Deja, to padaryti negalima, nes, remiantis Kolmogorovo ir Chinčino kriterijumi, atsitiktinė skaičių seka gali būti apibūdinama ir realizuojama tik ne mažesnio ilgio skaičiumi ir algoritmu nei ji pati yra, o ne trumpu ir patogiu naudoti skaičiumi ir tokiu pat trumpu algoritmu.

Dėl minėtų teorinių problemų negalima kalbėti apie algoritmiškai gaunamas atsitiktinių skaičių sekas, todėl norint išvengti loginių prieštaravimų yra įvedama pseudoatsitiktinių (*pseudorandom*) skaičių sekos (PASS) sąvoką.

6.4. Apibrėžimas. Skaičių seka yra vadinama PASS, jei joks kenkėjas negali jos atskirti (angl. *distinguish*) nuo tikros atsitiktinės skaičių sekos

Atskiriamumas (*distinguishability*) yra fundamentali matematikos ir kriptografijos sąvoka, su kuria toliau teks susidurti. Iš apibrėžimo matome, kad pseudoatsitiktinumas (angl. *pseudorandomness*) yra subjektyvi sąvoka, susijusi su kriptoanalitiko skaičiavimo ištekliais, tačiau tuojau pamatysime, kad tai yra konstruktyvi išeitis iš gana keblios padėties. Nors šiuo atveju nekalbama apie absoliučiai saugaus šifro sukūrimą, tačiau galima nagrinėti šifrą, subjektyviai artimą absoliučiai saugiam ir tinkamą duomenų srautui užšifruoti.

Siūlomas toks sprendimo būdas: kriptografiniai subjektai *Aldona* ($\mathcal{A}$) ir *Bronius* ($\mathcal{B}$) turi simetrinį slaptąjį raktą k, kurį naudoja gana ilgai šifravimo gamai generuoti. Šiam uždaviniui spręsti ir pasitelkiami pseudoatsitiktinių skaičių generatoriai (PASG).

6.5. Apibrėžimas. Pseudoatsitiktinių skaičių generatorius (PASG) yra deterministinis polinominio laiko algoritmas, iš sėklinės eilutės (angl. *seed string*) $k \in \{0,1\}^k$ sugeneruojantis ilgesnę eilutę $\xi(k)$, kurios ilgis išreiškiamas daugianariu $p(k)$ ir $p(k) = |\xi| > k$.

6.6. Klausimas. Kodėl PASG generuotos eilutės ξ ilgis išreiškiamas daugianariu, o ne, pvz., eksponente?

PASG saugumui įvertinti suformuluojamas paskesnio bito kriterijus. Jis gali būti apibendrintas ir paskesnio skaičiaus kriterijumi, tačiau pastaroji formuluotė būtų gerokai sudėtingesnė.

6.7. Apibrėžimas. PASG tenkina paskesnio bito kriterijų, jeigu kenkėjas, žinodamas $n + 1$ ilgio seką $\xi_{i-n}, ..., \xi_{i-1}, \xi_i$, gali prognozuoti (*predict*) paskesnį bitą ξ_{i+1}, esant tikimybei, nykstamai viršijančiai ½, t. y.

$$\text{Prob}[\xi_{i+1} = b] = ½ + \varepsilon; \tag{6.5}$$

čia $b \in \{0,1\}$, ε – nykstamasis dydis, kuris bendruoju atveju yra priklausomas nuo k, t. y. $\varepsilon = \varepsilon(k)$.

Kriterijus, pagal kurį nagrinėjamas paskesnio bito neprognozuojamumas, nėra labai griežtas ir pakankamas, todėl jį reikia papildyti PASS neatskiriamumo nuo „tikros“ atsitiktinės sekos kriterijumi.

Tarkime, kad simetrinis raktas k yra $|k|$ ilgio bitų eilutė. Šis raktas atsitiktinai parenkamas iš aibės $\{0,1\}^{|k|}$, t. y. $k \in_R \{0,1\}^{|k|}$.

6.8. Apibrėžimas. Statistinis testas T yra tikimybinis polinominio laiko algoritmas T, kuris, gavęs įvesties eilutę, pateikia atsakymą TAIP arba NE (arba 0 ir 1).

6.9. Apibrėžimas. PASG, kuris gavęs sėklą k generuoja $p(|k|)$ ilgio bitų seką ξ, tenkina statistinį testą T, jeigu seka ξ ir „tikra“ atsitiktinė seka η, kai $|\eta| = p(|k|)$, tenkina lygybę

$$|\text{Prob}[T(\xi) = 1] - \text{Prob}[T(\eta) = 1]| \le \varepsilon. \tag{6.6}$$

Šis apibrėžimas yra šiek tiek tikslesnis ir labiau pritaikomas praktiškai.

6.10. Apibrėžimas. PASG tenkina neatskiriamumo kriterijų, jeigu naudojant bet kuriuos statistinius testus ji neatskiriama nuo „tikros“ atsitiktinės sekos.

6.11. Teorema. Bitų generatorius yra pseudoatsitiktinis tada ir tik tada, jei jis tenkina visus atsitiktinumo patikros statistinius testus (Yao, 1982), (Goldwasser, et al., 2001).

Paskutinis nepatogumas, kurį reikia įveikti, tai – būtinybė šioje teoremoje žodį visus kaip nors pakeisti į žodį pagrindinius, siekiant gauti kokį nors praktiškai naudingą rezultatą. Nors statistiniai testai ir nėra pakankama sąlyga, kad PASG būtų kriptografiškai saugus, tačiau atitikimas šiems testams yra būtinas. Bet koks statistinis dėsningumas atsitiktinių skaičių sekoje gali būti panaudotas būsimų generatoriaus reikšmių prognozei. Jeigu statistinių testų metu teigiamai įvertinamas PASG tinkamumas, toliau reikia tirti generatoriaus struktūrą. Atliekant šifro kriptoanalizę (turint pakankamai daug tekstogramos ir šifrogramos porų ir esant nesudėtingai PASG struktūrai), generatoriaus reikšmių prognozavimas taip pat gali būti sėkmingai atliktas taikant tam tikrus kriptoanalizės metodus.

Kita PASG savybė nusakoma sugeneruotos sekos entropija, kuri yra atvirkštinis informacijos, esančios toje sekoje, dydis. Akivaizdu, kad informacijos kiekis sekoje turi būti kuo mažesnis, o entropija – kuo

didesnė. Taigi entropija yra sekos išankstinio (*a priori*) neapibrėžtumo matas. Kadangi PASG generuojama seka yra visiškai apibūdinama pradine reikšme, tai tos sekos entropija negali būti didesnė už pradinės reikšmės entropiją, ir ji nepriklauso nuo sugeneruotos sekos ilgio. Tuo PASG skiriasi nuo tikros atsitiktinių skaičių sekos. Tačiau jei pradinė reikšmė yra pakankamai ilga, kad tokios reikšmės paieška perrenkant visas galimas reikšmes būtų neįmanoma atsižvelgiant į šių dienų skaičiavimo pajėgumus, tai minėtas principinis PASG trūkumas gali būti ignoruojamas.

Siūlomi keli PASG testavimo metodai, iš kurių žymiausi šie:

1. Knutho atsitiktinumo testai.
2. FIPS 140–1 testas – JAV nacionalinio standartų ir technologijų instituto (NIST) priimtas serijų testų standartas, sukurtas Knutho testų pagrindu.
3. Maurerio universalusis statistinis testas, pagal kurį galima nustatyti didelę defektų klasę. Papildomai šis testas pateikia atsitiktinių skaičių entropijos įvertį.
4. FIPS 140–2 standartu apibrėžiami išsamesni rinkinių testai, tarp jų ir Maurerio universalusis statistinis.

Laikoma, kad PASG tenkina neatskiriamumo (angl. *undistinguishability*) savybę, jeigu juo sugeneruoti bitai atitinka FIPS-140-2 standartą. KTU Taikomosios matematikos katedra turi programinę įrangą, kuri pagal FIPS-140-2 standartą realizuoja numatytus statistinius testus. Tokiu būdu, sukūrus arba gavus PASG, juos galima testuoti ir nustatyti, ar jie tinkami kriptografiniam taikymui pagal neatskiriamumo kriterijų.

Pagrindinės srautinių šifrų sudedamosios dalys: 1) šifravimo rakto (gamos) generatorius (angl. *key stream generator*); 2) tekstogramos bei gamos kompozicijos blokas, kuris dažniausiai realizuojamas kaip jų sudėtis pabičiui moduliu 2.

PASG galima aprašyti tokia rekursine lygtimi:

$$\xi_i = f(\xi_{i-1}); \tag{6.7}$$

čia f – tam tikra funkcija, nuo kurios pasirinkimo priklauso generuojamų skaičių statistinės savybės ir generatoriaus kriptografinis saugumas. Generavimo operacijai reikia nurodyti pradinę sąlygą ξ_0, kuri mūsų supaprastintu atveju buvo lygi simetriniam raktui k, t. y. $\xi_0 = k$. Generuojami skaičiai ξ_i turi būti baigtinėje aibėje, kurios elementai koduojami baigtinio ilgio bitų seka, pavyzdžiui, $|\xi_i| = |k_i|$ ir $\xi_i \in \{0,1\}^{|k|}$.

Pateiksime keletą PASG algoritmų, kurių pagrindu sudaromi srautiniai šifrai.

1. Tiesiniai kongruenciniai (TK) generatoriai.
2. Tiesinių grįžtamojo ryšio postūmio registrų (TGRP; angl. *linear feed-back shift registers* – LFSR) generatoriai.
3. Netiesinio dinaminio chaoso (NDC) generatoriai.
4. Blokinio šifravimo režimų OFB, CFB ir CTR generatoriai.
5. Vienkryptėmis funkcijomis (VKF) paremti generatoriai.

Algoritmų, paremtų 1 ir 2 punktuose nurodytais principais, nenagrinėsime dėl laiko stokos, atsižvelgdami į tai, kad jie šiuo metu nelaikomi perspektyviais. Besidomintieji gali paskaityti šaltinį (Menezes, et al., 2001).

NDC generatoriai iš pradžių buvo laikomi labai perspektyviais. Jei funkcija f atvaizduoja realųjį skaičių ξ_{i-1} į realųjį skaičių ξ_i, tai, tinkamai parinkus f, toks atvaizdavimas gali būti artimas fizinio chaoso sistemoms, tokioms kaip turbulencija, ir pasižymi pakankamai didele entropija. Deja, šifruojant reikia gauti diskrečiuosius bitus ir baitus, todėl realieji skaičiai turi būti pakeisti diskrečiaisiais. Tokiu atveju diskrečiosios dinaminės sistemos būsenų erdvė tampa baigtinė, todėl diskretizuota dinaminė sistema generuoja ciklines sekas. Ilgą laiką nebuvo gauta jokių teorinių rezultatų, galinčių padėti įvertinti diskretinės

sekos periodą. Yra ir kitų trūkumų, kurių čia neminėsime. Tačiau darbai šioje srityje tęsiami. Projektui eSTREAM pateiktas K. Boesgaardo ir R. Vesteragerio šifras „Rabbit“, paremtas diskretizuota dinaminio chaoso sistema[4]. Diskretizuojant atsitiktiniai skaičiai ξ_{i-1}, ξ_i, ... yra diskretieji, koduojami baigtiniu bitų skaičiumi. Be abejo, ir šio šifro atveju neįmanoma gauti neperiodinės gamos, tačiau autoriai pateikia minimalaus gamos periodo įvertinimą, lygų 2^{64} bitams. Tai reiškia, kad su viena ir ta pačia sėkla ξ_0 galima saugiai šifruoti 2^{64} bitus, o tai kur kas daugiau, nei užšifruoti DVD filmus realiuoju laiko masteliu visą mėnesį.

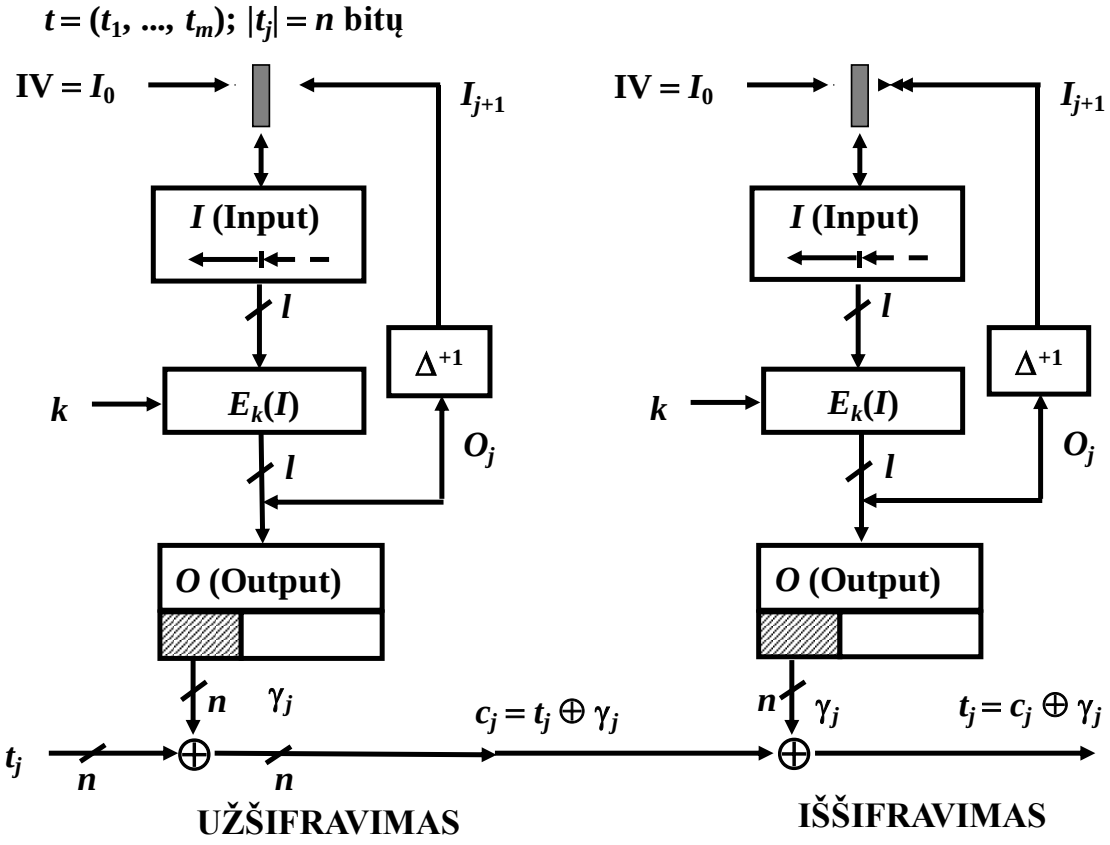
„Rabbit“ šifre naudojami 32 bitų kintamieji, todėl jis yra efektyviai realizuojamas mikroprocesoriuose ir dėl efektyvių šifravimo operacijų pasižymi didele sparta. Todėl galima pasiekti, kad, pvz., DVD filmas būtų užšifruojamas ir iššifruojamas realiuoju laiku, esant labai nedideliam vėlinimui. Be abejo, šiuo atveju nekalbame apie informacijos perdavimo spartą, pavyzdžiui, interneto tinklu.

Manome, kad šiuo metu daugiausia dėmesio galime skirti blokinio šifro OFB, CFB ir CTR režimams, kuriais remiantis galima sudaryti PASG ir tuo pačiu sukurti srautinį šifrą. Toks sprendimo būdas šiuo metu skaitomas gana perspektyviu.

Srautinio šifro schema, pagrįsta blokinio šifro OFB režimu, pateikta 6.1 paveiksle.

Šiuo atveju gamai generuoti pasitelkiamas blokinis šifras, kurio šifravimo funkcija yra $E_k(I)$. Čia k yra simetrinis blokinio šifro raktas, o I – blokinio šifro įvesties duomenys, kurių ilgis bitais yra lygus l. Šifruoti pradedama tada, kai pradiniai įvesties duomenys yra inicijavimo vektorius IV, t. y. $I = I_0 = IV$. Tada išvestyje O gauname tokio pat ilgio l užšifruotus įvesties bitus, kuriais galime šifruoti tekstogramos bitus t_0. Tačiau to nedarome – gautus išvestyje O_0 duomenis siunčiame į įvestį (žymime I_1) ir vėl šifruodami funkcija $E_k(I_1)$ gauname išvesties bitus O_1, kuriuos ir galime panaudoti tekstogramai užšifruoti. Bet kuriuo i-tuoju laiko momentu gauname tokią tapatybę:

$$O_j = E_k(I_j). \qquad (6.8)$$



6.1 pav. OFB režimo srautinis šifras

[4] http://www.exrypt.eu.org/stream/rabbit.html

Nebūtinai visus l išvesties bitus galima naudoti tekstogramai t_j užšifruoti. Jeigu $|t_j| = n < l$, išrenkame n išvesties O_j bitų, kuriuos pavadiname šifravimo gama γ_j, ir pabičiui sumuojame moduliu 2 su atitinkamais t_j bitais. Po šios operacijos gauname šifrogramos elementą c_j, kurį galime išreikšti taip:

$$c_j = t_j \oplus \gamma_j. \tag{6.9}$$

Šifrograma iššifruojama tokia pat tvarka. Turint tą patį *IV* ir *k*, atkuriama ta pati gamos seka $\{\gamma_j\}$, tad panašiai kaip Vernamo šifre:

$$t_j = c_j \oplus \gamma_j. \tag{6.10}$$

Pagrindinės šio srautinio šifro savybės:

1. Tai asinchroninis šifras, todėl klaidos sklaida yra minimali. Jei klaida atsiras viename šifrogramos simbolyje c_j, tai bus klaidingai iššifruotas tik atitinkamas simbolis t_j.
2. Praradus bet kurį šifrogramos simbolį c_j, klaidos ištaisyti neįmanoma likusiuose simboliuose. Tokiu atveju visi iššifruoti simboliai t_{j+1}, t_{j+2}, ... tampa klaidingi.
3. Šifravimo gama yra nepriklausoma nuo tekstogramos t. Tai labai gera savybė, didinanti šifro saugumą.
4. Dėl inicijavimo vektoriaus *IV*, šifruojant vienodas tekstogramas, gaunamos skirtingos šifrogramos, jei joms šifruoti naudojami skirtingi *IV*.

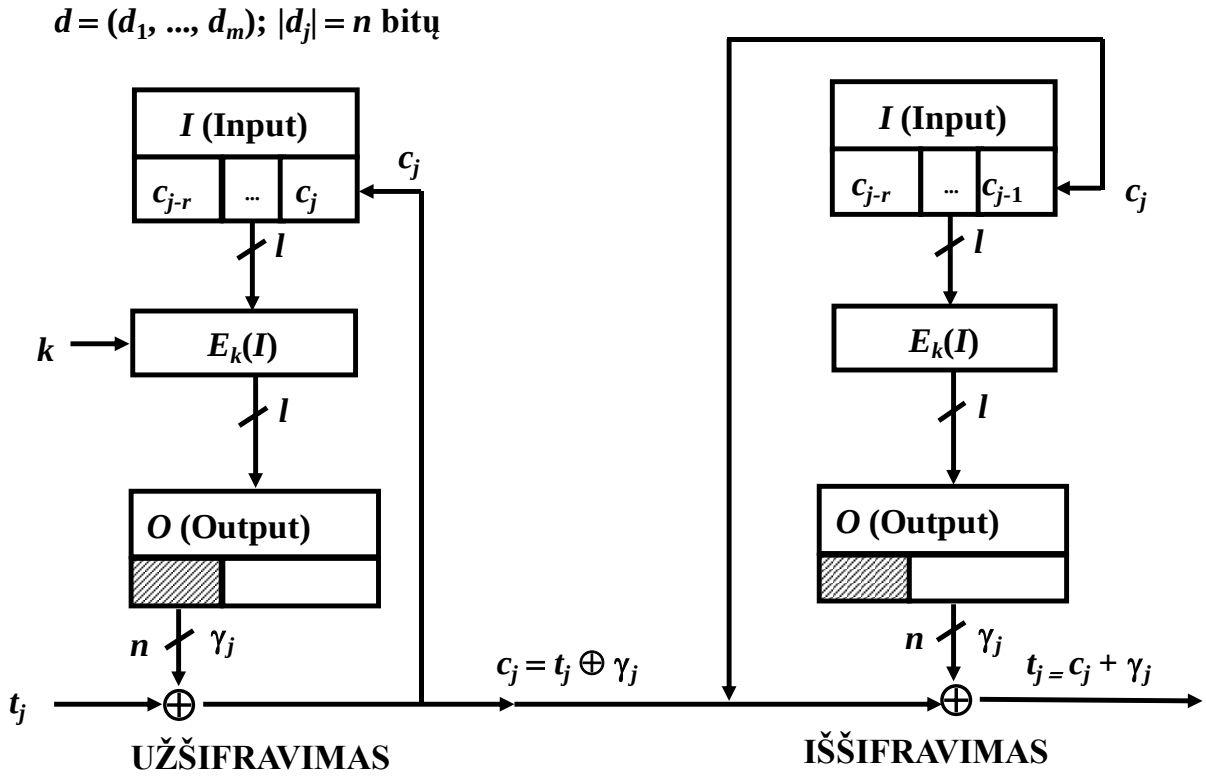
Vienas tokio šifro pavyzdžių su tam tikromis modifikacijomis – A. Biryukovo pasiūlytas srautinis LEX, pagrįstas AES blokinio šifro OFB režimu[5]. Kitas pavyzdys – P. Canniere'o ir B. Preneelo šifras „Trivium", struktūriškai panašus į blokinį šifrą OFB režimu[6]. Šie šifrai jau įveikė eSTREAM projekto pirmąjį etapą ir yra nagrinėjami antrajame etape (*focus phase 2 cipher*).

Srautinio šifro schemą, pagrįstą CFB blokinio šifro režimu, pritaikysime sinchroninio šifro atvejui. Ši schema pateikta 6.2 paveiksle. Inicijavimo vektorius *IV* naudojamas kaip ir OFB atveju, todėl paveikslėlyje jo nenurodėme. Kaip matome, blokinio šifro įvestis I yra postūmio registro reikšmės c_{j-r}, ..., c_j. Kiekvieno takto j metu postūmio registro dešiniajame bloke rašoma nauja c_j reikšmė, o kitos reikšmės pastumiamos per 1 bloką į kairę. Reikšmė c_{j-r-1} dingsta iš registro, o į jos vietą įrašoma c_{j-r} reikšmė. Užšifravimo metu registras saugo $r + 1$ šifravimo reikšmes c_{j-r}, ..., c_{j-1}, c_j.

Tam, kad vyktų užšifravimo procedūra, reikia užpildyti įvesties registrą I. Tam naudojamas inicijavimo vektorius, sudarytas iš $r + 1$ komponenčių, – IV_1, IV_2, ..., IV_{r+1}. Paskui sistemą galima perjungti į šifravimo režimą.

Galima pastebėti, kad šifravimo bloko įvestį bendruoju atveju sudaro $(r + 1)l$ bitų eilutė, o išvestį O – tik l bitų ilgio eilutė. Todėl užšifravimo funkcija E_k apibrėžia atvaizdavimą $E_k : \{0,1\}^{(r+1)l} \rightarrow \{0,1\}^l$. Šiuo atveju galimos įvairios variacijos. Viena jų – į užšifravimo bloką galima įvesti kokį nors vieną šifrogramos simbolį, pvz., c_{j-r+i}, $0 \le i \le r$. Visos kitos operacijos yra panašios kaip ir OFB režimu.

[5] http://www.ecrypt.en.org/sream/lex.html
[6] http://www.ecrypt.en.org/stream/trivium.html



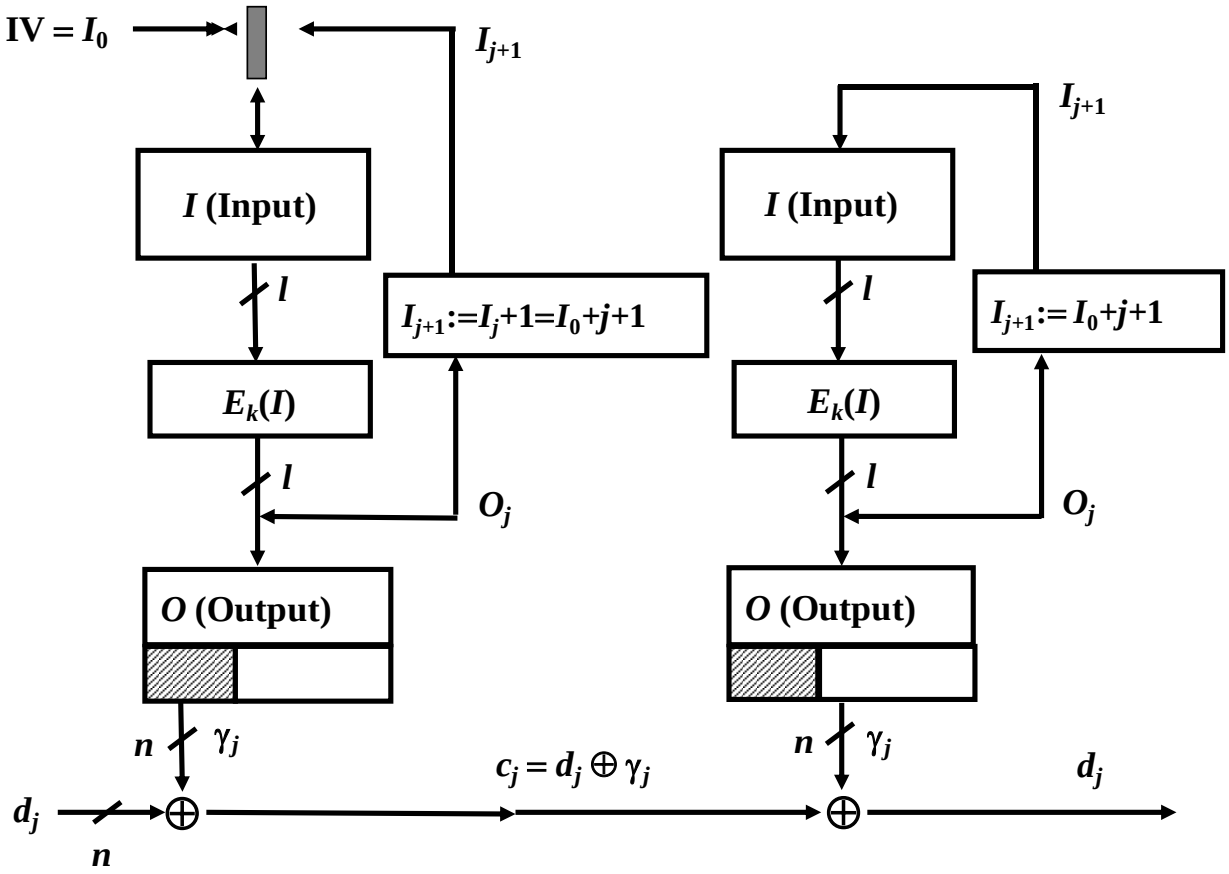
6.2 pav. CFB režimo srautinis šifras

Pagrindinės srautinio šifro savybės:

1. Tai yra sinchroninis srautinis šifras, todėl, esant klaidai šifrogramos simbolyje c_j, visi tekstogramos blokai t_j, ..., t_{j+r} bus dešifruojami klaidingai, o blokai, pradedant t_{j+r+1}, – teisingai. Taip yra todėl, kad dešifruojant tekstogramos blokas t_{j+r+1} jau nepriklauso nuo klaidingo simbolio c_j.
2. Gausime lygiai tokį pat rezultatą kaip ir 1 punkte, jeigu šifrogramos simbolis c_j bus prarastas ir vietoj jo į iššifravimo įrenginį atkeliaus simbolis c_{j+1}. Todėl šis šifras turi klaidų ištaisymo savybę.
3. Šifravimo gama yra priklausoma nuo tekstogramos t, todėl užšifravimo funkcijai E_k turi būti keliami papildomi reikalavimai, kad kiekvieno tekstogramos bloko t_{j-r}, ..., t_j bitas būtų tolygiai išsklaidytas visuose γ_{j-r}, ..., γ_j bituose.
4. Dėl skirtingų inicijavimo vektorių IV, šifruojant vienodas tekstogramas, gaunamos skirtingos šifrogramos.

Panagrinėsime CTR režimu pagrįstą srautinį šifrą, kuris sudaromas supaprastinus OFB režimo šifrą. Šiuo atveju OFB įvestis I tampa skaitiklio (*counter*) išėjimu. Kitaip sakant, išėjimo grįžtamas ryšys dirba skaitiklio režimu.

Taigi CTR režimo srautinis šifras – tai supaprastintas šifras, šifro įvesties blokui veikiant skaitiklio režimu: $I_{j+1} = I_j + 1$.



6.3 pav. CTR režimo srautinis šifras

6.3 pav. matome, kad šifro grįžtamasis ryšys realizuoja funkciją:

$$I_{j+1} := f(I_0, j) = I_0 + j + 1. \tag{6.11}$$

Tokiu būdu kiekvienas gamos simbolis γ_j yra abipusiškai vienareikšmiškai susietas su kiekvienu tekstogramos bloko t_j indeksu j, nes, remiantis apibrėžimu, užšifravimo funkcija E_k turi būti bijekcinė. Tada kiekvieną indeksą j atitiks vienintelis γ_j ir atvirkščiai. Tai labai naudinga savybė, tačiau apie jos pritaikymą pakalbėsime vėliau.

Užšifravimo ir iššifravimo funkcijos realizuojamos taip pat kaip ir OFB režimu.

CTR režimo srautinis šifras yra sinchroninis, o pagrindinės jo savybės tokios pat kaip ir OFB režimo srautinio šifro. Tačiau šis šifras pasižymi labai svarbia papildoma savybe, dėl kurios jis yra pranašesnis už OFB ir CFB režimu šifrų variantus.

OFB ir CFB režimais paremti sinchroniniai ir asinchroniniai šifrai pasižymi tuo, kad užšifravimo ir iššifravimo veiksmai vyksta nuosekliai. Norint užšifruoti ar iššifruoti simbolį t_j ar c_j, reikia šią procedūrą pradėti nuo pradžių, t. y. atitinkamai nuo simbolio d_1 ar c_1. Tai labai nepatogu, pvz., šifruojant simbolį t_j, kai $j = 2^{47} + 1$.

Kitaip nei OFB ir CFB režimai, CTR režimo srautinis šifras leidžia užšifruoti ir iššifruoti t_j ir c_j, kai j yra bet koks, nepradedant šios procedūros nuo pradžių. Tam j tam indeksui priskiriama tikra reikšmė ir suformuojamas inicijavimo vektorius IV_j. Dėl šios savybės CTR režimo srautinis šifras dar vadinamas atsitiktinės prieigos šifru (angl. *random access cipher*). Juo gali būti užšifruojami ir iššifruojami dideli informacijos srautai arba masyvai, siekiant užšifruoti ir iššifruoti atskirus jų fragmentus. Tokiu būdu galima sutaupyti laiko. Pavyzdžiui, prireikus šifruoti tam tikrus diskinio kaupiklio fragmentus, šių fragmentų pradžiai priskiriami atitinkami indeksai j_1, .., j_n, ir jie šifruojami tiesioginiu būdu, o ne nuosekliai pradedant nuo pradžių.

Vienkryptėmis funkcijomis (VKF) (angl. *one-way function* – OWF) paremtus PASG ir atitinkamus srautinius šifrus panagrinėsime tolimesniuose skyriuose.

Pateiksime keletą OFB, CFB ir CTR režimų srautinių šifrų sudarymo rekomendacijų, kurios leidžia kurti šifrus, atsparius trims pagrindinėms kriptoanalizės atakoms: tiesinei, diferencinei ir algebrinei.

Daugelyje mokslinių publikacijų įrodyta, kad blokiniuose šifruose naudojama multiplikacinė inversinė S-blokų funkcija pasižymi beveik optimaliais parametrais, užtikrinančiais kriptografinį atsparumą tiesinei ir diferencinei kriptoanalizėms. Tokia multiplikacinė inversinė funkcija yra realizuota AES šifre. Ji atliekama $GF(2^8)$ lauko elementų, koduojamų 8 bitais, inversija. Ši operacija vykdoma gana greitai, turint visų 256 lauko elementų atvirkštinių reikšmių lentelę. Sudarant srautinį šifrą blokinio šifro pagrindu, rekomenduojama į šifravimo funkciją įtraukti inversinę funkciją. Deja, inversinė funkcija aprašoma labai paprastomis lygtimis, siejančiomis funkcijos argumento (įvesties) reikšmes su jos reikšmių (išvesties) reikšmėmis. Šis sąryšis nusakomas lygčių sistema, sudaryta iš daugelio kintamųjų kvadratinių daugianarių. Tai reiškia, kad tokios pavienės netiesinės funkcijos naudojimas gali susilpninti šifrą algebrinės kriptoanalizės atakos atveju.

Bendras principas kovojant su algebrine kriptoanalize – naudoti kuo sudėtingesnes netiesines funkcijas ir aritmetines operacijas įvairiose algebrinėse struktūrose. Be to, reikia pasitelkti skirtingas raundines funkcijas, t. y. N raundų šifras gali turėti N skirtingų raundų. Saugumas dar labiau padidėtų, jei raundai būtų parenkami naudojant papildomą PASG.

Be įprastų XOR operacijų, S blokuose galima naudoti (jau yra naudojamos) ciklinio postūmio į dešinę (ar į kairę) operacijas, kurios žymimos simboliais >> ir <<. Prie jų paprastai nurodomas pastumiamų bitų ar baitų skaičius n, t. y. $>>_n$ ir $<<_n$.

Kitos naudingos operacijos – skaičių suma ir skirtumas moduliu 2^m, kurios žymimos + ir –. Naudojant 32 bitų mikroprocesorius, parenkamas $m = 32$. Visos išvardytos operacijos tokiuose mikroprocesoriuose vykdomos labai sparčiai.

Taip pat galimos ir lėtesnės operacijos, pavyzdžiui, skaičių daugyba moduliu 2^m arba eksponentinė funkcija lauke $GF(2^m)$.

Atliekant skaičių dalybą moduliu 2^m, apibrėžiamas laukas, kuriame vykdoma ši operacija, ir nustatoma dalybos iš 0 taisyklė (tai daroma ir AES atveju).

Turint arba sudarius blokinį šifrą, atliekama jo saugumo analizė, apskaičiuojant penktajame skyriuje minėtus kriterijus: tiesinį, netiesinį profilius ir N. Courtoiso bei J. Pieprzyko gama kriterijus.

Nagrinėjant S blokus kaip vektorines Būlio funkcijas, dažnai Būlio kintamųjų skaičius yra gana didelis, tad tiesiogiai minėtų kriterijų apskaičiuoti neįmanoma. Tokiu atveju patartina rinktis žaislinį šifrą (angl. *toy cipher*), turintį mažai kintamųjų. Gautos kriterijų reikšmės suteikia tam tikrą objektyvią informaciją apie jį atitinkančio, tikro šifro saugumą.

7. SANTRAUKOS FUNKCIJOS IR DUOMENŲ VIENTISUMAS

Santraukos funkcijos (angl. *hash function*) yra vienos svarbiausių šiuolaikinių kriptografinių primityvų. Santraukos funkcijos argumentas – bet kokio baigtinio ilgio bitų eilutė, o rezultatas – fiksuoto ilgio bitų eilutė. Svarbiausia šių funkcijų paskirtis – užtikrinti duomenų vientisumą (angl. *data integrity*). Be to, santraukos funkcijos gali būti naudojamos kuriant pseudoatsitiktinių skaičių generatorius (PASG), mat jos pasižymi PASG būdingomis svarbiomis savybėmis, t. y. įvesties ir išvesties duomenys yra beveik nekoreliuoti.

7.1. Apibrėžimas. Santraukos funkcija – tai kriptografinė transformacija, iš bet kokio ilgio pranešimo sugeneruojanti fiksuoto ilgio duomenų bloką, kuris yra vadinamas santrauka (angl. *hash value* arba *digest*).

Pagrindinė santraukos funkcijų savybė – spartus (efektyvus) pranešimo santraukos apskaičiavimas ($t \rightarrow H(t)$) ir itin sudėtingas (ar net neįmanomas) pranešimo atkūrimas iš santraukos ($H(t) \rightarrow t$).

Santraukos funkcijos algoritmai veikia taip, kad net dėl menkiausių duomenų pakeitimų gaunama visiškai skirtinga santraukos reikšmė. Tai neleidžia (ar bent padaro gerokai sudėtingiau) kenkėjams rasti kito pranešimo, kurio santrauka būtų tokia pat. Situacija, kai randami du pranešimai, kurių santraukos funkcijos sutampa, vadinama kolizija. Saugios santraukos funkcijos turi būti atsparios kolizijai.

Santraukos funkcijos skirstomos į priklausančias nuo slaptojo parametro arba rakto ir nepriklausančias nuo jo. Nuo rakto priklausančios funkcijos kitaip dar vadinamos pranešimo autentifikavimo kodais (angl. MAC – *Message Authentication Code*). Nuo rakto nepriklausančios santraukos funkcijos plačiai naudojamos e. parašo schemose. Toliau santraukos funkciją, kurios argumentas x, žymėsime $H(x)$.

Tam, kad santraukos funkcijos būtų naudojamos kriptosistemose, turi būti tenkinamos šios savybės:

- Santraukos funkcija yra siurjekcinis (suspaudžiantysis) bet kokio ilgio duomenų bloko atvaizdis į fiksuoto ilgio bitų eilutę.
- Santraukos funkcijos reikšmė turi būti algoritmiškai lengvai apskaičiuojama, neatsižvelgiant į pradinių duomenų kiekį.
- Santraukos funkcija pasižymi pirmavaizdžio atsparumo savybe, t. y. beveik neįmanoma rasti įvesties x, žinant santraukos funkcijos $H(x)$ reikšmę.
- Santraukos funkcija pasižymi antrojo pirmavaizdžio atsparumo savybe, t. y. bet kuriai įvesčiai x beveik neįmanoma rasti kitos tokios įvesties y, kad $H(x) = H(y)$.
- Santraukos funkcija pasižymi atsparumo kolizijoms savybe, t. y. beveik neįmanoma rasti tokių dviejų įvesčių x ir y, kad $H(x) = H(y)$.

Pirmosios dvi savybės atspindi minimalius santraukos funkcijų reikalavimus, kad būtų galima jas taikyti pranešimui autentifikuoti.

Trečioji savybė užtikrina santraukos funkcijų vienkryptiškumą, ketvirtoji reikalinga apsaugoti pranešimą nuo klastojimo, o penktoji nusako santraukos funkcijų atsparumą atakoms, pagrįstoms gimtadienio dienos paradoksu.

Santrauka – tai lyg unikalūs pranešimo pirštų atspaudai, pagal kuriuos identifikuojamas pranešimas. Kiekvienas konkretus pranešimas turi vienintelę jį atitinkančią tam tikros funkcijos santrauką. Paprastai pranešimas iš didesnės aibės atvaizduojamas į mažesnę, todėl konkrečią santrauką atitinka daugybė pranešimų. Jei santraukos funkcija yra kriptografiškai saugi, beveik neįmanoma rasti dviejų pranešimų, turinčių vienodą santrauką.

Santraukos funkcijų pritaikymo schemos:

- Apskaičiuota pranešimo santraukos funkcijos reikšmė užšifruojama kartu su pranešimu. Tokiu būdu santraukos funkcija naudojama ir kaip klaidų aptikimo priemonė.

- Apskaičiuota pranešimo santraukos funkcijos reikšmė užšifruojama asimetrinio šifro privačiuoju raktu PR, neužšifruojant paties pranešimo. Tokiu būdu užtikrinamas duomenų vientisumas ir siuntėjo autentifikavimas. Toks šifravimas atitinka e. parašo schemą.
- Nuo rakto priklausančios santraukos funkcijos reikšmė prijungiama prie pranešimo. Šiuo atveju šifruoti nebūtina, mat raktą žino tik siuntėjas ir gavėjas. Asmuo, nežinantis rakto, negali gauti santraukos funkcijos reikšmės.
- Tam, kad nereikėtų saugoti paties slaptažodžio, kai kuriose sistemose saugoma tik slaptažodžio santrauka. Vartotojui įvedus slaptažodį, apskaičiuojama jo santrauka ir patikrinama, ar ji sutampa su esančia duomenų bazėje. Jei santraukos sutampa, vartotojui suteikiama prieiga prie sistemos, naudojant įvestą slaptažodį. Tokiu principu veikia *Microsoft Windows* operacinių sistemų registracija.

Santraukos funkcijos skirstomos į vienkryptes (angl. OWHF – *One-Way Hash Function*); atsparias kolizijoms (angl. CRHF – *Collision Resistant Hash Function*) ir pranešimo autentifikavimo kodus (MAC).

7.2. Apibrėžimas. Santraukos funkcija yra vadinama vienkrypte (OWHF), jei funkcija yra siurjekcinis atvaizdis, atvaizduojantis bet kokio ilgio duomenų bloką į fiksuoto ilgio bitų eilutę ir pasižymintis pirmavaizdžio bei antrojo pirmavaizdžio savybėmis.

7.3. Apibrėžimas. Santraukos funkcija vadinama atsparia kolizijoms (CRHF), jei funkcija yra siurjekcinis atvaizdis, atvaizduojantis bet kokio ilgio duomenų bloką į fiksuoto ilgio bitų eilutę ir pasižymintis pirmavaizdžio, antrojo pirmavaizdžio bei atsparumo kolizijoms savybėmis.

7.1. SANTRAUKOS FUNKCIJŲ ALGORITMAI

Labiausiai paplitusios santraukos funkcijos – MD5 ir SHA-1. Santraukos funkcijos MD5 rezultatas – 128 bitų eilutė. Dar visai neseniai tai buvo bene labiausiai paplitusi santraukos funkcija, tačiau, remiantis „Eurocrypt 2007“ konferencijos medžiaga, ji yra nesaugi.

SHA-1 algoritmas buvo sukurtas 1995 metais, o visi jo aprašai pateikti FIPS 180-1 standarte. SHA-1 algoritmo santrauka yra 160 bitų, o sutraukiamas pranešimas turi neviršyti 2^{64} bitų. Trumpai apžvelgsime SHA-1 algoritmo struktūrą. Algoritmą sudaro penki etapai, o tarpiniai veiksmai atliekami su 512 bitų ilgio blokais:

- Pirmojo etapo metu pranešimas papildomas iki L ilgio bitų eilutės, kad L (mod 512) = 448. Pirmajam bitui priskiriamas 1, visiems kitiems – 0.
- Antrojo etapo metu pridedamas 64 bitų blokas. Šis blokas interpretuojamas kaip sveikasis teigiamas skaičius, lygus sutraukiamo pranešimo ilgiui.
- Trečiuoju žingsniu inicializuojamas pradinių reikšmių vektorius – penki 32 bitų ilgio blokai A, B, C, D, E.
- Ketvirtasis etapas sudarytas iš 4 raundų po 20 žingsnių. Raundo įvestys – 512 bitų ilgio pranešimo blokas ir 160 bitų buferis, žymimas ABCDE. Kiekvieno raundo išvestis yra naujas buferio ABCDE turinys.
- Paskutinio etapo metu santraukos funkcijos reikšmei priskiriama buferio reikšmė, gauta po ketvirtojo etapo.

SHA-1 funkcija tinka įprastiems dokumentams apsaugoti, tačiau siekiant didesnio saugumo patartina naudoti kitus santraukų funkcijų algoritmus. NESSIE rekomenduoja rinktis šiuos santraukų funkcijų algoritmus (NESSIE Consortium, 2003): SHA-256, SHA-384, SHA-512, *Whirpool*. SHA-256 generuoja 256 bitų ilgio santraukas, SHA-384 – 384 bitų, o SHA-512 ir *Whirpool* – 512 bitų.

7.2. SANTRAUKOS FUNKCIJŲ ATAKOS

Kaip ir bet kuris kitas kriptografinis primityvas, santraukos funkcijos turi būti atsparios įvairioms atakoms. Skiriamos šios pagrindinės santraukos funkcijų atakos: gimtadienio paradoksu paremta ataka (angl. *Birthday Attack*), atsitiktinio pirmavaizdžio ataka (angl. *Random Preimage Attack*), susitikimo viduryje ataka (angl. *Meet-in-the-Middle Attack*), pataisančio bloko ataka (angl. *Correcting-Block Attack*), nejudamojo taško ataka (angl. *Fixed Point Attack*).

Gimtadienio paradoksu pagrįsta ataka gali būti veiksminga tuo atveju, jeigu potencialus kenkėjas turi galimybę modifikuoti pranešimą prieš jį pasirašant siuntėjui. Jei santraukos funkcijos rezultatas yra m bitų eilutė, tai generuojamos dvi aibės, turinčios po $2^{m/2}$ pranešimų. Vienoje aibėje yra pranešimai, pagal prasmę atitinkantys tikrąjį pranešimą, kitoje – suklastoti. Apskaičiuojamos visų pranešimų santraukos funkcijų reikšmės, ir, remiantis gimtadienio paradoksu, kenkėjas, esant 0,5 tikimybei, gauna du skirtingus pranešimus, kurių santraukos funkcijos reikšmė yra vienoda. Pranešimą, pagal prasmę atitinkantį tikrąjį, kenkėjas duoda pasirašyti siuntėjui, o vėliau šį pranešimą pakeičia į suklastotą. Kadangi jų santraukos funkcijų reikšmės sutampa, toks pakeitimas neturės įtakos parašo patikros funkcijos rezultatui. Generuoti skirtingus pranešimus, turinčius vienodą prasmę, nesudėtinga. Kadangi pranešime vieno bito pakeitimas, esant 0,5 tikimybei, keičia bet kurį santraukos funkcijos rezultato bitą, tai užtektų pakeisti keliasdešimt žodžių sinonimais, o vieną žodį antonimu, pvz., vietoj „taip“ parašyti „ne“, t. y. A neskolingas B.

Atsitiktinio pirmavaizdžio ataka atitinka visų galimų variantų visišką perrinkimą. Kenkėjas pasirenka pranešimą ir tikisi, kad jo santraukos funkcijos reikšmė įgis reikiamą reikšmę. Tokios sėkmės tikimybė yra 2^{-m}, kai m – santraukos funkcijos rezultato bitų skaičius. Šią ataką sudėtinga realizuoti, nes jeigu m = 160 bitų, reikėtų sugeneruoti 2^{160} pranešimų, o tai – neįmanoma.

Susitikimo viduryje atakos metu sugeneruojama dalis pranešimo ir bandoma sukurti likusią dalį pagal turimą santraukos reikšmę.

Pataisančio bloko atakos metu sugeneruojami du pranešimai X ir X' ($X \neq X'$) bei ieškomi tokie Y ir Y', kad $H(X \mid X') = H(Y \mid Y')$. Ši ataka dažniausiai taikoma paskutiniajam blokui, bet iš esmės gali pasitaikyti ir kituose.

Nejudamojo taško atakos metu ieškomas duomenų blokas, nekeičiantis santraukos funkcijos apskaičiavimo algoritmo tarpinių rezultatų.

Trys paskutinės išvardytos atakos gali būti taikomos tik santraukos funkcijoms, apskaičiuojamoms naudojant iteracinį algoritmą, t. y. turinčioms aukščiau minėtus tarpinius rezultatus. Iteracinis algoritmas aprašomas šiomis lygtimis:

$$H_0 = IV;\ H_i = f(X_i, H_{i-1}),\ i = 1, 2, ..., t;\ H(X) = g(H_t);$$

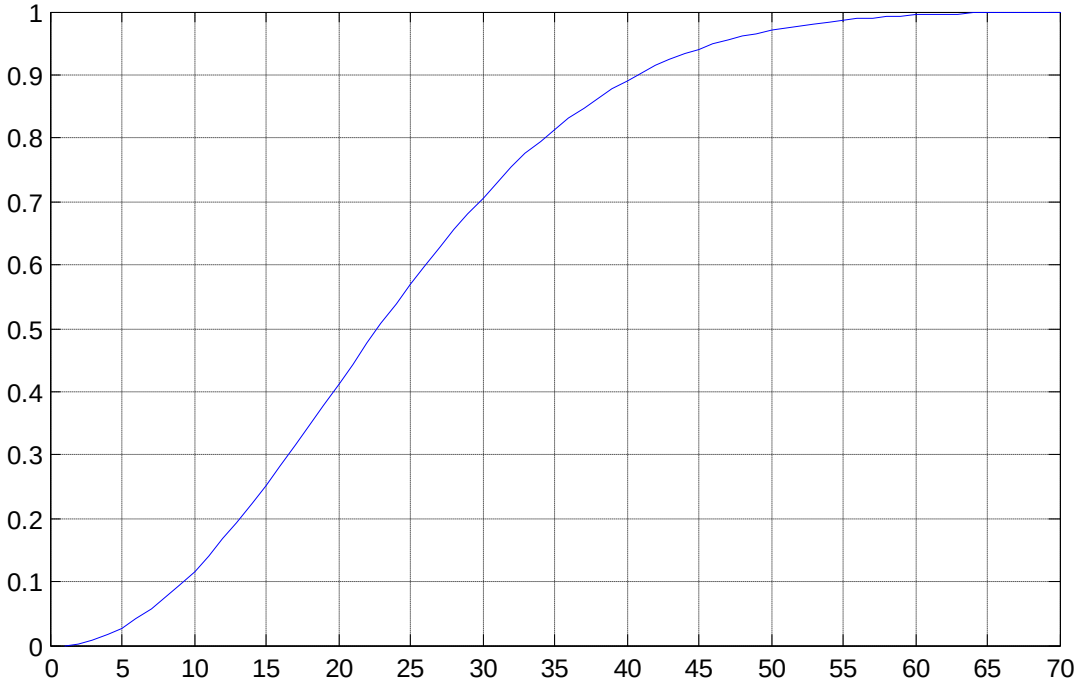
čia IV – inicijavimo vektorius, X – pranešimas, suskaidomas į t blokų H_i, f ir g yra vidinės algoritmo funkcijos.

7.2.1. GIMTADIENIO DIENOS PARADOKSAS

Gimtadienio dienos paradoksas – puikus uždavinys, rodantis, jog žmonių intuityvi ir subjektyvi supančio pasaulio samprata kartais prieštarauja objektyviems tikimybių teorijos rezultatams. Gimtadienio dienos uždavinį galima suformuluoti taip: kiek turi būti mažiausiai atsitiktinai susirinkusių žmonių (skaičius k), kad tikimybė, jog bent du iš šios grupės švęstų savo gimtadienį tą pačią dieną, būtų ne mažesnė nei 0,5?

Pažymėkime nurodytą tikimybę Prob(k) ir ją apskaičiuokime. Tarkime, kad kiekvieni metai turi 365 dienas (t. y. nenagrinėkime vasario 29 dienos). Tuomet asmens gimtadienis, esant vienodai tikimybei, gali būti bet kurią dieną. Apskaičiuokime priešingo įvykio tikimybę: k žmonių grupėje visi yra gimę skirtingomis dienomis. Apskaičiuojame remdamiesi klasikiniu tikimybės apibrėžimu: $1 - \text{Prob}(k) = \frac{n(k)}{N(k)}$, čia $n(k)$ – palankių įvykių skaičius, $N(k)$ – visų galimų įvykių skaičius. Visų galimų įvykių yra $N(k) = 365^k$, nes bet

kuris asmuo gali gimti bet kurią dieną. Palankių įvykių skaičius yra $n(k) = \frac{365!}{(365-k)!}$. Tuomet ieškoma tikimybė lygi Prob(k) = $1 - \frac{365!}{365^k (365-k)!}$. Šios tikimybės grafikas pateiktas 7.1 pav.



7.1 pav. Gimtadienio dienos paradoksas

Iš grafiko aišku, kad surinkus grupę iš 23 žmonių, tikimybė, jog du švęs gimtadienį tą pačią dieną, yra 0,5. Nors iš pirmo žvilgsnio grupėje, tenkinančioje uždavinio sąlygą, turėtų būti apie 100 žmonių. Taip yra todėl, kad konkrečiam asmeniui tikimybė, jog dar kažkas švęs gimtadienį tą pačią dieną, yra gana maža. Tuo tarpu uždavinyje nagrinėjama, ar atsiras kokia nors asmenų, gimusių tą pačią dieną, pora, nefiksuojant nei gimimo dienos, nei konkretaus asmens. 23 žmonių grupėje galima sudaryti 253 skirtingas poras, todėl ir tikimybė yra pakankamai reikšminga.

Šį uždavinį galime apibendrinti taip: turime k atsitiktinių dydžių seką, kurioje kiekvienas atsitiktinis dydis įgyja sveikąsias reikšmes nuo 1 iki n, esant tikimybei $\frac{1}{n}$. Kam lygi tikimybė Prob(n, k), kad sekoje bus bent dvi vienodos reikšmės? Analogiškai samprotaudami gauname Prob(n, k) = $1 - \frac{n!}{n^k (n-k)!}$. Be to, galime įrodyti, kad kai $k \approx \sqrt{n}$, tai Prob(n, k) $\approx 0{,}5$.

7.3. DUOMENŲ VIENTISUMAS

Duomenų vientisumas – tai viena iš kriptografijos krypčių, dar vadinama informacijos autentifikavimu. Ji su tam tikru patikimumu turi garantuoti, kad duota informacija yra autentiška, t.y. kad informacija nebuvo taisyta ar pakeista.

Autentifikavimo sistemos modelis susideda iš siuntėjo, gavėjo ir galimo kenkėjo. Paskutinysis gali stebėti visą informaciją perduodamą tarp siuntėjo ir gavėjo. Laikoma, kad kenkėjas žino viską, netgi pradinį (nešifruotą) pranešimą (tai vadinama autentifikavimu be apsaugos), bet jis nežino koks raktas buvo panaudotas.

Autentifikavimo tikslais naudojamos santraukos funkcijos priklausančios nuo slaptojo rakto *K*, dar vadinamos autentifikavimo kodais (angl. MAC –Message Authentication Code) MAC = $H_K(M)$. Slaptąjį raktą *K* žino tik siuntėjas ir gavėjas.

Siunčiant pranešimą *M*, siuntėjas apskaičiuoja MAC kaip funkcija nuo pranešimo *M* ir slaptojo rakto *K* ir gautą rezultatą prijungia prie paties pranešimo. Gavėjas atlieka analogiškus veiksmus naudojant tą patį slaptąjį raktą ir sulygina apskaičiuotą MAC reikšmę su gautąją. Jei reikšmės sutampa, tai gavėjas yra tikras, kad nebuvo modifikuotas siuntimo metu.

Jei užtikrinamas slaptojo rakto saugumas, tai ši schema pasižymi tokiomis savybėmis:

- Gavėjas yra tikras, kad pranešimas nebuvo modifikuotas. Jei galimas kenkėjas pakeistų pranešimą, bet nepakeistų MAC reikšmės, tai gavėjo apskaičiuota reikšmė nesutaps su esama. O dėl santraukos funkcijų savybių, kenkėjas, nežinant slaptojo rakto, negali tinkamai pakeisti MAC reikšmės;
- Gavėjas yra tikras, kad pranešimas gautas iš nurodyto siuntėjo. Nes slaptąjį raktą žio tik siuntėjas ir gavėjas.

Reikia pastebėti, kad autentifikavimo kodas užtikrina tik duomenų vientisumą. Pats pranešimas nėra šifruojamas ir norint užtikrinti ir duomenų konfidencialumą reikia naudoti šifravimo algoritmus.

8. IDENTIFIKACIJA IR AUTENTIFIKACIJA

Šiame skyriuje supažindinsime su subjekto autentifikacija ir pateiksime autentifikacijos protokolų pavyzdžius. Subjekto autentifikacija skirstoma į tris grupes: slaptažodinę, užklausos ir reakcijos bei nulinio atskleidimo. Panagrinėsime atakas prieš autentifikacijos protokolus ir patarsime, kaip jų išvengti.

> ***8.1. Apibrėžimas.*** Subjekto autentifikacija — tai procesas, kurio metu vienas subjektas (tikrintojas) įsitikina kito subjekto (pareiškėjo) tapatybe, esant tam tikrai garantijai. Šis įsitikinimas užtikrinamas reikalaujant iš pareiškėjo pateikti jo tapatybę patvirtinančius įrodymus. Tapatybė gali būti pateikta tikrintojui kaip kokio nors protokolo dalis arba žinoma iš anksto.

Bendruoju atveju subjektas tvirtina tam tikrą savo tapatybę ir atlieka savo autentifikaciją: vienokiu ar kitokiu būdu įrodo tikrintojui, kad tapatybė tikrai priklauso jam. Procesas, kurio metu pareiškėjas tvirtina tam tikrą tapatybę, vadinamas identifikacija. Autentifikacijos metu pareiškėjas įrodo tikrintojui, kad pateikta tapatybė tikrai priklauso jam. Kai kuriuose literatūros šaltiniuose šie terminai vartojami kaip sinonimai, todėl reikia atidžiai pažiūrėti, kuria prasme jie yra vartojami. Bet kuris asmuo gali identifikuoti save bet kuriuo kitu ir teigti, kad jam priklauso atitinkama tapatybė, tačiau greičiausiai jis negalės autentifikuoti savęs kuo nors kitu, jei autentifikacijos mechanizmas bus pakankamai saugus kriptografiniu atžvilgiu.

Matome, kad identifikacija ir autentifikacija yra labai susiję procesai. Toliau šiame skyriuje nagrinėsime autentifikaciją plačiąja prasme, kartu apimančią ir identifikaciją, mat neįmanoma subjektui autentifikuotis prieš tai savęs neidentifikavus.

Tapatybės autentifikavimas gali būti vienpusis, kai vienas subjektas autentifikuojasi kitam, ir abipusis, kai subjektai autentifikuojasi vienas kitam. Abiem atvejais tikrintojas turi žinoti autentišką identifikacinį parametrą, pvz., bendrą slaptąjį raktą, viešojo rakto sertifikatą ar kokią nors biometrinę informaciją, pagal kurią autentifikuojama.

Įrodymu pagrįsta subjekto autentifikacija remiasi tapatybės atributais arba autentifikacijos kodais. Šie kodai paprastai skaičiuojami remiantis vienu iš šių įvesties tipų:

- *Tai, kas žinoma*. Paprastai tai yra pareiškėjui suteiktas slaptažodis arba asmeninis identifikacinis kodas (PIN). Jis arba tiesiogiai pateikiamas tikrintojui, arba naudojamas tapatybės atributams apskaičiuoti, kurie pateikiami tikrintojui ir paskui patvirtinami.
- *Tai, kas turima*. Šiam įvesties tipui priklauso ir fiziniai įrenginiai, kurie naudojami tapatybės atributams, pateikiamiems tikrintojui, apskaičiuoti, ir programinė įranga, kurią pareiškėjas privalo turėti, kad būtų apskaičiuoti tapatybės atributai. Fiziniai įrenginiai – tai lustinės kortelės, kortelės su magnetinėmis juostelėmis ir vienkartinio naudojimo slaptažodžių generatoriai. Programinės įrangos rinkmenose paprastai saugomi privatieji raktai, kurie naudojami tapatybės atributams apskaičiuoti. Šie raktai gali būti ir fiziniuose įrenginiuose.
- *Tai, kas esama*. Šiai kategorijai priskiriamas įvesties tipas, kuris vadinamas biometrija. Tapatybės atributams apskaičiuoti pasitelkiami žmogaus fiziniai požymiai, pvz., pirštų atspaudai, akies tinklainė ar ranka rašytas tekstas.

Subjektas autentifikuojamas siekiant išvengti nesankcionuotos prieigos prie saugomų informacijos šaltinių. Tai gali būti nuotolinė prieiga prie kompiuterinių sąskaitų, patekimas į interneto svetaines ar prieiga prie banko sąskaitų bankomatuose. Jei subjekto autentifikacija taikoma šiais tikslais, ji turi būti suderinta su prieigos kontrolės metodu, siekiant apriboti prieigą. Pareiškėjas paprastai iš pradžių autentifikuojamas, o paskui prieigos kontrolė ar privilegijuoto valdymo technika nustato leistinos prieigos lygį.

Didžiausias tapatybės autentifikavimo ir pranešimo autentifikavimo, naudojant elektroninį parašą arba pranešimo autentifikavimo kodą, skirtumas tas, kad pranešimo autentifikavimas nepriklauso nuo laiko momento, kada pranešimas buvo sukurtas, tuo tarpu tapatybė autentifikuojama pareiškėjui ir tikrintojui bendraujant realiuoju laiku. Be to, tapatybės autentifikavimas neatskleidžia jokios svarbios informacijos, išskyrus identifikatorių, reikalingą tapatybei nustatyti. Pranešimo autentifikavimas pateikia daug informacijos, pavyzdžiui, patį pranešimą.

Autentifikacija skirstoma į tris grupes: slaptažodinę (arba silpną), užklausos ir reakcijos (arba stiprią) ir nulinio atskleidimo.

8.1. SLAPTAŽODINĖ (SILPNA) AUTENTIFIKACIJA

Standartinė slaptažodinės autentifikacijos sistema remiasi laiko atžvilgiu nekintančiais slaptažodžiais. Svarbiausia šios sistemos idėja – su kiekvienu vartotoju susieti slaptažodį, kurį paprastai sudaro 6 – 10 ar daugiau simbolių ir kurį vartotojas geba įsiminti. Slaptažodis – bendra vartotojo ir sistemos paslaptis, tad tokią autentifikaciją galima priskirti simetrinėms kriptosistemoms. Norėdamas pasiekti sistemą, vartotojas pateikia identifikatorių, nusakantį jo tapatybę, ir slaptažodį, kuris yra tapatybės įrodymas. Sistema patikrina, ar slaptažodis atitinka turimus duomenis apie pareiškėją. Jei viskas sutampa, pareiškėjas sėkmingai autentifikuojamas ir gauna prieigą prie informacijos išteklių.

Pateiksime keletą paprastų slaptažodinių autentifikacijos sistemų.

8.1.1. FIKSUOTŲ SLAPTAŽODŽIŲ SISTEMOS

Paprasčiausia vartotojo slaptažodį laikyti paprasto teksto pavidalu sistemos slaptažodžių rinkmenoje, apsaugotoje nuo skaitymo ir rašymo. Vartotojui įvedus slaptažodį, sistema palygina jį su atitinkamo vartotojo slaptažodžiu, esančiu rinkmenoje. Tačiau tokia sistema neapsaugo nuo privilegijuotų vartotojų, kurie gali atversti slaptažodžių rinkmeną. Kadangi tokioje sistemoje kriptografiniai metodai netaikomi, ji laikoma nekriptografine.

Norint, kad slaptažodžiai nebūtų laikomi kaip paprastas tekstas, galima taikyti santraukos funkcijas kiekvienam slaptažodžiui ir rinkmenoje saugoti tik slaptažodžių santraukas. Norėdama patikrinti vartotojo įvestą slaptažodį, sistema apskaičiuoja slaptažodžio santrauką ir palygina ją su atitinkamo vartotojo santrauka, saugoma rinkmenoje. Taigi pakanka, kad slaptažodžių rinkmena būtų apsaugota tik nuo rašymo. Svarbu pabrėžti, kad taip užšifruotas slaptažodis yra labai silpnai apsaugotas siunčiant jį viešaisiais ryšio kanalais.

Nuspėjami slaptažodžiai neatsparūs žodyno tipo atakoms, kurias aptarsime 8.1.2 skyrelyje. Siekiant to išvengti, kai kuriose sistemose yra numatytos slaptažodžių taisyklės. Dažniausiai jose nurodomas trumpiausias galimas slaptažodžių ilgis (pvz., 8 arba 12 simbolių), taip pat reikalaujama, kad slaptažodžiai būtų sudaryti iš simbolių, priklausančių skirtingoms kategorijoms (pvz., didžiųjų raidžių, skaičių, nestandartinių simbolių), arba būtų tikrinama, kad į slaptažodžio turinį nepatektų duomenų apie vartotojo identifikatorių.

Tačiau žinodamas slaptažodžių sudarymo taisykles kenkėjas gali taip modifikuoti savo žodyno ataką, kad būtų tikrinami slaptažodžiai, bent minimaliai atitinkantys nustatytas taisykles.

Dar vienas būdas pagerinti slaptažodžių saugumą – nustatyti jų galiojimo laiką. Kiekvienam slaptažodžiui priskiriamas naudojimo laiko intervalas (pvz., 30 arba 90 dienų), tad jie periodiškai keičiami.

Siekiant sulėtinti atakas, kurių metu bandoma daug slaptažodžių, galima nustatyti intervalus tarp pakartotinio slaptažodžių pateikimo.

Kitas būdas geriau apsisaugoti nuo žodyno atakos – prie kiekvieno slaptažodžio pridėti t bitų ilgio atsitiktinę eilutę prieš taikant santraukos funkciją. Ir slaptažodžio santrauka, ir atsitiktinė eilutė saugomos slaptažodžių rinkmenoje. Vartotojui pakartotinai įvedus slaptažodį, sistema prie jo prideda atsitiktinę eilutę ir, apskaičiavusi modifikuoto slaptažodžio santrauką, tikrina, ar ši santrauka sutampa su sistemos turima. Tai nė kiek neapsunkina visiško perrinkimo atakos, nes atsitiktinė eilutė saugoma slaptažodžių rinkmenoje paprasto teksto pavidalu. Tačiau tai apsunkina žodyno ataką, nes reikalaujama, kad žodyne būtų 2^t kiekvieno bandomo slaptažodžio variantų skaičius.

Norint sukurti sudėtingesnius slaptažodžius, atsižvelgiant į žmogaus atminties galimybių ribas, slaptažodžius galima praplėsti – sudaryti slaptas frazes. Šiuo atveju vartotojas autentifikuodamasis vietoj trumpo žodžio pateikia frazę arba sakinį. Slaptai frazei pritaikoma santraukos funkcija ir ji panaudojama kaip

ir anksčiau minėti slaptažodžiai. Slaptą frazę žmogus įsimena geriau nei atsitiktinių skaičių seką, tačiau reikia įvesti daugiau simbolių.

8.1.2. ATAKOS PRIEŠ FIKSUOTŲ SLAPTAŽODŽIŲ SISTEMAS

Vienas iš fiksuotų slaptažodžių sistemų trūkumų – pakartotinis slaptažodžių naudojimas. Kenkėjas gali sužinoti jūsų slaptažodį, jums jį neatsargiai įvedus, arba rasti jį kur nors užrašytą. Dar viena saugumo spraga – vartotojo įvestas slaptažodis ir slaptažodžio santrauka gali būti siunčiami sistemai neužšifruoti viešuoju tinklu. Slaptažodis taip pat laikinai egzistuoja sistemoje paprasto teksto pavidalu, kol yra tikrinamas. Kenkėjas gali išsaugoti tokius slaptažodžius ir pakartotinai juos panaudoti.

Dėl minėtų priežasčių fiksuotų slaptažodžių sistemos tinkamos tuo atveju, jei su sistema vartotojas bendrauja saugiu ryšiu, o vengtinos tada, kai slaptažodis siunčiamas viešuoju tinklu. Tarkime, kad vartotojas iš namų jungiasi prie darbo kompiuterio internetu. Slaptažodis siunčiamas viešuoju tinklu atviro teksto pavidalu ir yra matomas kenkėjams.

Kenkėjas gali įvykdyti labai paprastą ataką: atsitiktinai arba naudodamas kokią nors sistemą, jis gali mėginti po vieną visus įmanomus slaptažodžius, taip siekdamas prisijungti prie sistemos. Tokiai visiško perrinkimo atakai (angl. *exhaustive search attack*) galima priešintis parenkant ilgesnius slaptažodžius, ribojant neteisingų bandymų skaičių per tam tikrą laiko tarpą bei lėtinant slaptažodžių tikrinimo procesą. Kita vertus, gali būti atvejis, kai iki tam tikro momento su slaptažodžius tikrinančia sistema nereikia kontaktuoti. Šį atvejį dabar ir aptarsime.

Tarkime, slaptažodžių rinkmenoje saugomos vartotojų slaptažodžių santraukos. Kenkėjas gali įveikti sistemą bandydamas visus galimus slaptažodžius ir lygindamas gautas santraukas su saugomomis slaptažodžių rinkmenoje. Teoriškai tai yra įmanoma. Jei kenkėjas žino slaptažodžių santraukas, jis randa ir slaptažodį. Galima bandyti laikyti slaptažodžių rinkmeną neprieinamą, tačiau tai sudėtinga padaryti, mat negalima tikėtis, kad slaptažodžių rinkmena bus absoliučiai neprieinama. Tokia sistema bus saugi tik tol, kol bus rastas bent vienas slaptažodis.

Visiško perrinkimo ataką galima paspartinti. Užuot tikrinęs visą galimą slaptažodžių aibę, kenkėjas gali tikrinti ją slaptažodžio panaudojimo tikimybės mažėjimo tvarka. Nors idealiai bet kokia *n* ilgio eilutė gali būti naudojama kaip slaptažodis, tačiau dauguma vartotojų pasirenka savo slaptažodžius iš mažo visų galimų slaptažodžių poaibio (pvz., trumpi slaptažodžiai, žodyno žodžiai, pavadinimai, mažųjų raidžių eilutės). Tokius lengvus slaptažodžius galima lengvai atspėti. Atlikus tyrimus paaiškėjo, kad daug vartotojų pasirinktų slaptažodžių galima rasti 150 000 žodžių žodyne, nors šis žodynas sudaro tik mažą dalį visų galimų *n* simbolių slaptažodžių.

Internetiniuose ar kitokiuose žodynuose randamus slaptažodžius gali panaudoti ir kenkėjas, bandydamas perrinkti visus žodžius, esančius žodyne. Tai yra vadinamoji žodyno ataka (angl. *dictionary attack*) prieš slaptažodį. Be tradicinių žodynų, gali būti naudojami ir užsienio kalbų arba specialieji, pvz., muzikos, filmų ar kiti, žodynai. Kenkėjas, turintis daug išteklių, gali parengti savo žodyną, kuriame būtų žodžių ar dažnesnių slaptažodžių santraukų funkcijų reikšmės.

Žodyno atakos bendruoju atveju nėra sėkmingos ieškant vieno konkretaus vartotojo slaptažodžio, tačiau jos yra pavojingos, jei yra nukreiptos prieš daugelį visos sistemos slaptažodžių.

8.1.3. SUDĖTINGESNĖ SLAPTAŽODINĖ AUTENTIFIKACIJA

Sudėtingesnei fiksuotų slaptažodžių sistemai sukurti galima naudoti asmeninį identifikacinį kodą, kuris dažniausiai naudojamas su atributu „*tai, ką turime*“. Tai gali būti kortelė su magnetine juostele arba lustinė kortelė. Autentifikacijai naudojant kortelę, jos savininkas privalo įvesti asmeninį identifikacinį kodą. Tai suteikia antrojo lygmens apsaugą kortelę pametus ar pavogus.

Dėl istorinių priežasčių ir vartotojų patogumui asmeninis identifikacinis kodas dažniausiai yra trumpas ir sudarytas iš skaičių (pvz., nuo 4 iki 8 skaitmenų). Siekiant panaikinti visiško perrinkimo galimybę mažoje galimų kodų aibėje, reikia papildomų apribojimų. Pavyzdžiui, bankomatai konfiskuoja banko kortelę įvedus tris klaidingus asmeninius identifikacinius kodus.

Kitas būdas yra pagrįstas vartotojo slaptažodžio santraukos funkcijos apskaičiavimu. Funkcijos reikšmė paverčiama kriptografiniu šifravimo raktu. Tokie šifravimo raktai naudojami autentifikacijai reikalingiems duomenims šifruoti, kuriuos vartotojas siunčia sistemai. Sistema taip pat turi žinoti slaptažodį, tačiau būtina užtikrinti, kad jis būtų pakankamai sudėtingas, ir visiško perrinkimo ataka, taikoma slaptažodžiui, nebūtų lengvesnė nei visiškas šifravimo raktų perrinkimas.

Norint, kad iš slaptažodžių gaunami kriptografiniai raktai nebūtų nuolatiniai, kol yra nuolatiniai slaptažodžiai, galima naudoti papildomus eilės numerius, kurie ir keičia raktus. Sistema saugo eilės numerį kartu su vartotojo slaptažodžiu. Kiekvieną kartą vartotojui jungiantis prie sistemos, sistema viešai nusiunčia eilės numerį vartotojui. Eilės numeris pridedamas prie slaptažodžio. Nors numeris yra visiems matomas, tačiau kiekvieną kartą gaunamas kitas šifravimo raktas, tad kenkėjui sudėtingiau iššifruoti duomenis.

Vartotojų patogumui slaptažodžiai, iš kurių gaunami kriptografiniai raktai, turi būti ilgalaikiai. Tai pasiekiama apribojant jų naudojimą tik autentifikacijai. Tuo pačiu reikia užtikrinti, kad sistemoje esantys slaptažodžiai būtų gerai apsaugoti.

Jau buvo minėta, kad viena didžiausių fiksuotų slaptažodžių problemų – pakartotinis jų naudojimas. Šią problemą iš dalies galima spręsti pasitelkus vienkartiniais slaptažodžiais paremtas sistemas. Tokios sistemos galėtų būti:

- Bendras vienkartinių slaptažodžių sąrašas. Vartotojas ir sistema turi vienkartinių slaptažodžių sąrašą. Kiekvienas slaptažodis skirtas vienai autentifikacijai ir yra iš anksto paskirstomas vartotojams. Tokios sistemos trūkumas – sąrašo naudojimo būdas. Jei slaptažodžiai iš sąrašo parenkami ne iš eilės, sistema gali tikrinti įvestą slaptažodį tarp visų nepanaudotų. Vienas galimų šios problemos sprendimo būdų – pasitelkti užklausos ir reakcijos lentelę, kurią sudaro slaptažodžių poros. Šios poros skirtos abipusei vartotojo ir sistemos autentifikacijai ir gali būti naudojamos tik kartą.
- Nuolat atnaujinami vienkartiniai slaptažodžiai. Iš pradžių sistema ir vartotojas turi vieną slaptažodį. Kiekvieną kartą autentifikuodamasis vartotojas sukuria naują slaptažodį ir jį, užšifruotą dabartiniu slaptažodžiu, nusiunčia sistemai. Kitą kartą vartotojui jungiantis prie sistemos, bus naudojamas jau naujas slaptažodis. Tačiau šį metodą sudėtinga įgyvendinti nutrūkus ryšiui, mat tada neįmanoma nusiųsti naujo slaptažodžio sistemai.
- Vienkartinių slaptažodžių sekos, pagrįstos santraukos funkcijomis. Iš pradžių sistema ir vartotojas turi vieną slaptažodį w. Slaptažodžių sekai gauti naudojama vienkryptė funkcija H. Galima sudaryti seką: w, $H(w)$, $H(H(w))$, ..., $H^t(w)$. Tada i-tojo autentifikacijos seanso ($1 \le i \le t$) slaptažodis bus $w_i = H^{t-i}(w)$. Šis metodas yra pranašesnis už nuolat atnaujinamus vienkartinius slaptažodžius ir nesukelia nepageidaujamų padarinių nutrūkus ryšiui.

Vienkartinio slaptažodžio sistemos vis dar yra pažeidžiamos. Kenkėjas gali apsimesti sistema ir tokiu būdu sužinoti vartotojo slaptažodį, o vėliau jį panaudoti jungdamasis prie tikrosios sistemos. Todėl laikomasi tokio bendro principo: slaptažodis gali būti atskleistas tik autentiškam subjektui.

8.2. UŽKLAUSOS IR REAKCIJOS (STIPRI) AUTENTIFIKACIJA

Remiantis užklausos ir reakcijos protokolu, pareiškėjas „įrodo“ savo tapatybę tikrintojui, parodydamas, kad pareiškėjas žino su savo tapatybe susijusią paslaptį. Ši paslaptis vykdant protokolą neatskleidžiama. Autentifikacija vyksta pareiškėjui atitinkamai reaguojant į nuo laiko priklausančias tikrintojo užklausas. Reakcija priklauso ir nuo pareiškėjo paslapties, ir nuo užklausos. Užklausa dažniausiai yra skaičius, kurį tikrintojas pasirenka atsitiktinai ir slaptai protokolo vykdymo pradžioje. Kenkėjui stebint ryšio kanalą, pareiškėjo autentifikacijos reakcija į užklausą neturi suteikti jokios naudingos informacijos kitai pareiškėjo autentifikacijai.

8.2.1. NUO LAIKO PRIKLAUSANTYS PARAMETRAI

Siekiant geriau apsisaugoti nuo pakartotinio panaudojimo ir kombinacijų atakų, autentifikacijos protokoluose gali būti naudojami nuo laiko priklausantys parametrai. Šie parametrai taip pat yra naudingi – suteikia autentifikacijos protokolams unikalumo ir galimybę panaudoti realiu laiku apsaugant nuo pasirinkto teksto atakos. Daugiau informacijos apie atakas prieš autentifikacijos protokolus galima rasti 8.4 skyrelyje.

Nuo laiko priklausantys parametrai užklausos ir reakcijos protokoluose dažniausiai apibūdinami kaip „atsitiktiniai“ skaičiai, tačiau jų savybės skiriasi. Dažniausiai naudojami šie nuo laiko priklausantys parametrai: atsitiktiniai skaičiai *r*, serijos numeriai *n*, laikrodžio rodmenys *t*. Šiuos parametrus aptarsime žemiau.

Atsitiktiniai skaičiai, naudojami užklausos ir reakcijos protokoluose, suteikia nenuspėjamumą ir padeda apsisaugoti nuo pasirinkto teksto atakų. Tačiau tam reikia generuoti kriptografiškai saugius atsitiktinius skaičius. Naudojant pseudoatsitiktinių skaičių generatorius, generatoriaus sėkla (angl. *seed*) turi pasižymėti pakankama entropija.

Užklausos ir reakcijos protokoluose serijos numeriai naudojami pranešimų unikalumui identifikuoti, ir dažniausiai jie yra skirti pakartotinio panaudojimo atakoms aptikti. Serijos numeriai yra atskiri kiekvienai subjektų porai. Atskiros serijos taip pat reikalingos siunčiant pranešimus tarp subjektų $\mathcal{A}$ ir $\mathcal{B}$ bei tarp $\mathcal{B}$ ir $\mathcal{A}$.

Abu subjektai, numeruodami pranešimus, turi vadovautis anksčiau nustatytomis taisyklėmis. Pranešimas priimamas tuo atveju, jei jo serijos numeris anksčiau nebuvo naudotas (arba nebuvo naudotas tam tikrą laiko tarpą) ir atitinka nustatytas taisykles. Paprasčiausia taisyklė – pradėti seriją nuo nulio ir su kiekvienu sėkmingu pranešimu didinti serijos numerį vienetu. Mažiau ribojanti taisyklė – monotoniškai didinti serijos numerį. Tokiu būdu galima aptikti pranešimus, dingusius dėl ryšio kaltės, tačiau trukdoma nustatyti tuos, kurie dingo dėl kenkėjo veiksmų.

Vienas iš serijos numerių trūkumų – pareiškėjo būtinybė ilgai prisiminti visų tikrintojų informaciją apie anksčiau naudotus ir dar galimus serijos numerius. Gali prireikti net specialių procedūrų siekiant atkurti dėl ryšio linijos kaltės nutrūkusią seriją. Priverstinio vėlavimo atakos bendruoju atveju taip pat nesusekamos. Dėl minėtų priežasčių serijos numeriai tinkamiausi mažoms, artimoms žmonių grupėms.

Laikrodžio rodmenys naudojami užtikrinti panaudojimą realiu laiku ir unikalumą bei aptikti pranešimų pakartojimus. Jie taip pat gali būti naudojami siekiant užtikrinti laiko apribotą prieigą ir aptikti priverstinio vėlavimo atakas.

Subjektas, sukūręs pranešimą, prie jo kriptografiškai prijungia vietinio laikrodžio rodmenis. Gavęs tokį pranešimą, kitas subjektas nuskaito savo laikrodžio rodmenis ir gauna pranešimo laikrodžio rodmenis. Pranešimas bus priimtinas, jeigu:

1) laikrodžio rodmenų skirtumas bus priimtinas;
2) iš siuntėjo nebuvo gauta pranešimo su tokiu pat laikrodžio rodmeniu.

Kita vertus, būtina užtikrinti, kad laiko rodmenis naudojančių subjektų laikrodžiai būtų sinchronizuoti ir apsaugoti nuo modifikavimo. Taip pat reikia saugoti didelį sąrašą visų siuntėjų laiko rodmenų.

8.2.2. UŽKLAUSOS IR REAKCIJOS AUTENTIFIKACIJA, PAGRĮSTA SIMETRINIO ŠIFRAVIMO METODAIS

Užklausos ir reakcijos sistema, pagrįsta simetriniu šifravimu, reikalauja, kad pareiškėjas ir tikrintojas žinotų tą patį kriptografinį raktą. Uždarose sistemose, kuriose yra mažai vartotojų, kiekvienai subjektų porai tokias raktų poras galima nesunkiai padalyti. Didelėse sistemose dažniausiai veikia patikimas serveris, subjektams padalijantis autentifikacijai reikalingus raktus.

Laikydami, kad subjektai $\mathcal{A}$ ir $\mathcal{B}$ jau turi kriptografinius raktus, pateiksime tris paprastus užklausos ir reakcijos autentifikavimo metodus. Šie metodai detaliai aptarti ISO/IEC 9798-2 standarte.

Pirmiausia pateiksime žymenis. Pareiškėjas yra subjektas $\mathcal{A}$, o tikrintojas – subjektas $\mathcal{B}$. Subjekto atsitiktinį skaičių ir jo laiko rodmenis atitinkamai žymime *r* ir *t*. Šiuose protokoluose laiko rodmenys gali

būti pakeisti sekos numeriu n, taip suteikiant protokolui kitokių savybių. E_K žymi simetrinio šifravimo algoritmą su kriptografiniu raktu K. Darome prielaidą, kad abu subjektai žino vienas kito tapatybes. Kintamieji A ir B atspindi atitinkamų subjektų identifikatorius. Neprivalomi laukai bus žymimi žvaigždute „*“, o simbolis „||“ reikš sujungimo (angl. *concatenation*) operaciją. Rodyklės kryptis rodys, kam siunčiamas pranešimas.

1. Vienpusė autentifikacija, pasitelkus laikrodžio rodmenis:

$$\mathcal{A} \rightarrow \mathcal{B}\text{: } E_K(t_A \,||\, B^*).$$

Subjektas $\mathcal{B}$, gavęs ir iššifravęs pranešimą, patikrina, ar laiko rodmenys yra priimtini, ir pasirinktinai patikrina, ar pranešimas buvo skirtas jam. Kartu užšifruotas subjekto $\mathcal{B}$ identifikatorius neleidžia kenkėjui tos pačios žinutės persiųsti subjektui $\mathcal{A}$, nes jų identifikatoriai nesutampa.

2. Vienpusė autentifikacija, naudojant atsitiktinius skaičius:

$$\mathcal{A} \leftarrow \mathcal{B}\text{: } r_B$$

$$\mathcal{A} \rightarrow \mathcal{B}\text{: } E_K(r_B \,||\, B^*).$$

3. Abipusė autentifikacija, naudojant atsitiktinius skaičius:

$$\mathcal{A} \leftarrow \mathcal{B}\text{: } r_B$$

$$\mathcal{A} \rightarrow \mathcal{B}\text{: } E_K(r_A \,||\, r_B \,||\, B^*)$$

$$\mathcal{A} \leftarrow \mathcal{B}\text{: } E_K(r_B \,||\, r_A).$$

Pastebėsime, kad abipusę autentifikaciją galima atlikti pasitelkus bet kurią vienpusę autentifikaciją du kartus abiem kryptimis. Tačiau taip panaudotos dvi vienpusės autentifikacijos nėra tarpusavyje susijusios ir negali būti laikomos vienalyčiu protokolu.

Aptartuose autentifikacijos algoritmuose šifravimo algoritmas gali būti pakeistas santraukos ar vienkrypte funkcija, pvz., pranešimo autentifikavimo kodu. Tokie pakeitimai naudingi tuomet, kai šifravimo algoritmai dėl didelės skaičiavimų apimties negalimi ar nepriimtini. Vienkryptėmis funkcijomis paremti autentifikacijos protokolai detaliai aptarti ISO/IEC 9798-4 standarte.

Kaip pavyzdį pateiksime SKID3 protokolą, naudojantį pranešimo autentifikavimo kodą h_K. Šiuo atveju abipusės autentifikacijos protokolas sudaromas iš trijų siunčiamų pranešimų:

$$\mathcal{A} \leftarrow \mathcal{B}\text{: } r_B$$

$$\mathcal{A} \rightarrow \mathcal{B}\text{: } r_A, h_K(r_A, r_B, B)$$

$$\mathcal{A} \leftarrow \mathcal{B}\text{: } h_K(r_B, r_A, A).$$

Praleidę trečiąjį pranešimą, gausime SKID2 protokolą, atliekantį vienkryptę autentifikaciją.

8.2.3. UŽKLAUSOS IR REAKCIJOS AUTENTIFIKACIJA, PAREMTA ASIMETRINIO ŠIFRAVIMO METODAIS

Asimetrinio šifravimo metodus galima taikyti užklausos ir reakcijos autentifikacijos protokole, kai pareiškėjas įrodo žinąs privatųjį raktą (PR) vienu iš dviejų būdų:

- Pareiškėjas iššifruoja tikrintojo atsiųstą užklausą, užšifruotą jo viešuoju raktu (VR).
- Pareiškėjas elektroniniu būdu pasirašo tikrintojui siunčiamą užklausą.

Rekomenduojama, kad VR ir PR pora, naudojama tokiose sistemose, nebūtų taikoma kitais tikslais.

Pateiksime keletą autentifikacijos protokolo pavyzdžių, kai pareiškėjas yra subjektas $\mathcal{A}$, užklausa užšifruota pareiškėjo viešuoju raktu, o tikrintojas yra subjektas $\mathcal{B}$. Kintamieji A ir B žymi atitinkamų subjektų identifikatorius, E – asimetrinį šifravimą viešuoju raktu, r – pasirinktą atsitiktinį skaičių, o H – santraukos funkciją.

- Vienpusė autentifikacija:

$$\mathcal{A}\leftarrow\mathcal{B}\text{: } H(r_B),\ B,\ E(VR_A,\ r_B \,||\, B)$$

$$\mathcal{A}\rightarrow\mathcal{B}\text{: } r_B.$$

Subjektas $\mathcal{A}$, gavęs pranešimą, iššifruoja atsitiktinį dydį r_B savo privačiuoju raktu PR_A ir apskaičiavęs santraukos funkciją palygina ją su gautu dydžiu $H(r_B)$. Jei lygybė galioja, subjektas $\mathcal{A}$ autentifikuojasi, atsakydamas į užklausą ir nusiųsdamas tikrintojui r_B.

- Abipusė autentifikacija:

$$\mathcal{A}\rightarrow\mathcal{B}\text{: } E(VR_B,\ r_A \,||\, A)$$

$$\mathcal{A}\leftarrow\mathcal{B}\text{: } E(VR_A,\ r_A \,||\, r_B)$$

$$\mathcal{A}\rightarrow\mathcal{B}\text{: } r_B.$$

Taip pat pateiksime tris autentifikacijos protokolus, pagrįstus elektroniniu parašu. Šie protokolai plačiau aptarti ISO/IEC 9798-3 standarte. *S* žymi elektroninio parašo funkciją, o *C* – elektroninio parašo sertifikatą su viešuoju raktu.

1. Vienpusė autentifikacija, pasitelkus laikrodžio rodmenis:

$$\mathcal{A}\rightarrow\mathcal{B}\text{: } C_A,\ t_A,\ B,\ S(PR_A,\ t_A \,||\, B).$$

2. Vienpusė autentifikacija, naudojant atsitiktinius skaičius:

$$\mathcal{A}\leftarrow\mathcal{B}\text{: } r_B$$

$$\mathcal{A}\rightarrow\mathcal{B}\text{: } C_A,\ r_A,\ B,\ S(PR_A,\ r_A \,||\, r_B \,||\, B).$$

3. Abipusė autentifikacija, naudojant atsitiktinius skaičius:

$$\mathcal{A}\leftarrow\mathcal{B}\text{: } r_B$$

$$\mathcal{A}\rightarrow\mathcal{B}\text{: } C_A,\ r_A,\ B,\ S(PR_A,\ r_A \,||\, r_B \,||\, B)$$

$$\mathcal{A}\leftarrow\mathcal{B}\text{: } C_B,\ A,\ S(PR_B,\ r_B \,||\, r_A \,||\, A).$$

8.3. NULINIO ATSKLEIDIMO AUTENTIFIKACIJA

Jau minėjome, kad slaptažodžių autentifikacijos sistemose pareiškėjas pateikia savo slaptažodį tikrintojui. Jį žinodamas, tikrintojas gali bandyti apsimesti pareiškėju, o tai dažniausiai nepageidautina. Šią saugumo spragą bando įveikti užklausos ir reakcijos autentifikavimo sistemos. Pareiškėjas realiuoju laiku atsakinėja į tikrintojo užklausas, nesuteikdamas papildomos informacijos, kurią būtų galima pakartotinai ir tiesiogiai panaudoti. Deja, dalis informacijos apie pareiškėjo paslaptį vis tiek gali būti atskleista.

Siekiant išspręsti šią problemą, kuriami nulinio atskleidimo protokolai, leidžiantys pareiškėjui įrodyti, jog jis žinąs paslaptį, ir tuo pačiu neatskleisti tikrintojui jokios informacijos, kurią jis galėtų panaudoti kenkėjiškais tikslais. Šią idėją gali realizuoti tik vienas siunčiamas bitas, rodantis, kad pareiškėjas žino savo paslaptį.

Kadangi autentifikuodamasis pareiškėjas atsako į tikrintojo užklausą, sudarytą iš vieno atsitiktinai parinkto bito, kenkėjas gali taip pat bandyti atsakinėti į užklausas atsitiktiniais bitais. Siekiant užtikrinti protokolo patikimumą, apsikeičiama keletu pranešimų. Vadinasi, naudodami nulinio atskleidimo protokolus, mes negalime būti visiškai tikri, kad pareiškėjas nėra kenkėjas, tačiau galime užtikrinti, kad tokia tikimybė būtų kiek norima artima 1. Taigi nulinio atskleidimo protokolai yra tikimybiniai.

Kaip pavyzdį pateiksime Fiato ir Shamiro (angl. *Fiat-Shamir*) protokolą. Praktikoje patartina naudoti efektyvesnius nulinio atskleidimo protokolus, apsikeičiančius keliomis užklausomis per vieną iteraciją.

Tarkime, kad pareiškėjas $\mathcal{A}$ nori autentifikuotis tikrintojui $\mathcal{B}$, įrodydamas savo paslapties *s* žinojimą ir neatskleisdamas jokios informacijos apie *s*, kurios tikrintojas nežino ir negali apskaičiuoti prieš vykdydamas

protokolą. Protokolo saugumas pagrįstas kvadratinės šaknies iš didelio sudėtinio sveikojo skaičiaus n apskaičiavimo sudėtingumu. Ši problema ekvivalenti sveikojo skaičiaus n faktorizavimui.

Prieš pradėdamas vykdyti protokolą, patikimas trečiasis asmuo $\mathcal{T}$ parenka ir paskelbia skaičių $n=pq$ (panašiai kaip RSA algoritme), bet pirminius skaičius p ir q laiko paslaptyje. Pareiškėjas $\mathcal{A}$ pasirenka paslaptį s, kuri yra tarpusavyje pirminis skaičius su n ($1 \le s \le n–1$). Taip pat apskaičiuoja $v = s^2 \pmod n$ ir pas patikimą trečiąjį asmenį $\mathcal{T}$ užregistruoja v kaip savo viešąjį raktą.

Protokolas vykdomas t kartų, kad autentifikacija būtų kuo patikimesnė. Parametro t konkrečią reikšmę pasirenka tikrintojas $\mathcal{B}$.

Per kiekvieną iš t iteracijų protokolo dalyviai apsikeičia trimis pranešimais:

$$\mathcal{A}\rightarrow\mathcal{B}\text{: } x = r^2 \pmod n$$

$$\mathcal{A}\leftarrow\mathcal{B}\text{: } e \in \{0, 1\}$$

$$\mathcal{A}\rightarrow\mathcal{B}\text{: } y = r \cdot s^e \pmod n$$

Bendruoju atveju subjektas $\mathcal{T}$ turėtų patikrinti sąlygą $DBD(s, n) = 1$ arba ekvivalenčią sąlygą $DBD(v, n) = 1$. Tikrintojas $\mathcal{B}$ turėtų nutraukti protokolo vykdymą, jei $DBD(y, n) \ne 1$. Jeigu kuri nors sąlyga netenkinama, skaičių n galima faktorizuoti, o tai prieštarauja protokolo saugumo prielaidai.

Toliau pateikiami žingsniai yra atliekami t kartų (nuosekliai ir nepriklausomai vienas nuo kito). Tikrintojas patiki pareiškėju, jei visos t iteracijos yra teisingos.

1. Pareiškėjas $\mathcal{A}$ pasirenka atsitiktinį skaičių r (įsipareigojimą), $1 \le r \le n–1$, ir tikrintojui $\mathcal{B}$ siunčia $x = r^2 \pmod n$ (liudininką).
2. Tikrintojas $\mathcal{B}$ atsitiktinai pasirenka bitą (užklausą) $e = 0$ arba $e = 1$ ir siunčia jį pareiškėjui $\mathcal{A}$.
3. Pareiškėjas $\mathcal{A}$ apskaičiuoja y (atsakymą) ir siunčia jį tikrintojui $\mathcal{B}$. Jeigu $e = 0$, tai $y = r$. Priešingu atveju – $y = r \cdot s \pmod n$.
4. Tikrintojas $\mathcal{B}$ atmeta autentiškumo įrodymą, jei $y = 0$, ir priima, patikrinęs lygybę $y^2 \equiv x \cdot v^e \pmod n$. Atsižvelgiant į bito e reikšmę, $y^2 = x$ arba $y^2 = x \cdot v \pmod n$, nes $v = s^2 \pmod n$. Pastebėsime, kad sąlyga $y = 0$ neleidžia susidaryti atvejui, kai $r = 0$.

Kenkėjas, apsimetęs pareiškėju $\mathcal{A}$, gali bandyti sukčiauti: pasirinkti bet kokį r ir nustatyti $x = \dfrac{r^2}{v} \pmod n$. Taip kenkėjas galėtų atsakyti į užklausą $e = 1$, teisingai priskirdamas $y = r$, tačiau negalėtų atsakyti į užklausą $e = 0$, nes reikėtų žinoti kvadratinę šaknį iš x moduliu n. Taigi kenkėjas gali atsakyti į užklausą su tikimybe $\dfrac{1}{2}$. Pareiškėjas $\mathcal{A}$, žinodamas paslaptį s, gali atsakyti į abu klausimus. Siekiant sumažinti sukčiavimo tikimybę iki priimtinai mažo dydžio 2^{-t}, protokolas iteruojamas t kartų. Pareiškėjas sėkmingai autentifikuojasi tikrintojui, kai visos t užklausos yra teisingai atsakomos.

8.4. ATAKOS PRIEŠ AUTENTIFIKACIJOS PROTOKOLUS

Atakos prieš slaptažodžių autentifikavimo sistemas aptartos 8.1.2 skyrelyje. Toliau nagrinėsime bendras atakas prieš autentifikacijos protokolus:

1. Apsimetimo ataka (angl. *impersonation*): vienas subjektas apsimeta kitu.
2. Pakartotinio panaudojimo ataka (angl. *replay attack*): panaudojama vieno prieš tai buvusio protokolo vykdymo informacija, taikoma tam pačiam arba skirtingam tikrintojui.
3. Kombinacijų ataka (angl. *interleaving attack*): panaudojamos tam tikros vieno ar kelių buvusių arba esamų protokolų vykdymo informacijos kombinacijos.
4. Atspindėjimo ataka (angl. *reflection attack*): tai kombinacijų ataka, kai kenkėjas vykdomo protokolo metu persiunčia informaciją ją išsiuntusiam subjektui.

5. Priverstinis vėlavimas“ (angl. *forced delay*): kenkėjas sukelia priverstinį vėlavimą, kai jis perima pranešimą (dažniausiai turintį serijos numerį) ir vėliau pakartotinai jį panaudoja. Tai nėra pakartotinio panaudojimo ataka.
6. Pasirinkto teksto ataka (angl. *chosen-text attack*): ataka prieš užklausos ir reakcijos protokolus, kurios metu kenkėjas strategiškai pasirenka užklausas, taip bandydamas išgauti informaciją apie pareiškėjo paslaptį.

8.1 lentelėje pateikiami būdai, kaip apsisaugoti nuo išvardytų atakų.

8.1 lentelė.

Atakos prieš autentifikacijos protokolus ir jų išvengimo būdai

Ataka	Išvengimo būdai
Pakartotinis panaudojimas	Taikyti užklausos ir reakcijos metodus; naudoti nuo laiko priklausančius parametrus, identifikatorių naudojimas pranešime.
Kombinavimas	Tarpusavyje susieti visus protokolo pranešimus (pvz., naudoti susietus ir nuo laiko priklausančius parametrus).
Atspindėjimas	Į užklausos atsakymą įterpti kito subjekto identifikatorių; sudaryti protokolus, kuriuose kiekvienas pranešimas būtų skirtingo formato; naudoti vienkrypčius raktus.
Pasirinktas tekstas	Taikyti nulinio atskleidimo metodus; į užklausos atsakymą įterpti savo pasirinktą atsitiktinį skaičių.
Priverstinis vėlavimas	Naudoti atsitiktinius skaičius ir trumpus atsakymo laiko intervalus; naudoti laikrodžio rodmenis.

Jei tas pats kriptografinis raktas naudojamas ne tik autentifikacijai, bet ir kitiems tikslams, reikia elgtis labai atsargiai. Pavyzdžiui, naudojant RSA kriptosistemos raktą tapatybės autentifikacijai ir elektroniniam parašui, galima pažeisti šios sistemos saugumą, taikant pasirinkto teksto ataką. Tarkime, vykstant autentifikacijai, tikrintojas $\mathcal{B}$ pateikia užklausas pareiškėjui $\mathcal{A}$, naudodamas atsitiktinius skaičius r_B, užšifruotus subjekto $\mathcal{A}$ RSA viešuoju raktu. Pareiškėjas $\mathcal{A}$ turi pateikti atsakymą su iššifruotu atsitiktiniu skaičiumi. Jeigu tikrintojas $\mathcal{B}$ pateikia užklausą $r_B = H(x)$, tai pareiškėjas $\mathcal{A}$, atsakydamas į ją, gali to net nenorėdamas suteikti subjektui $\mathcal{B}$ savo RSA parašą ant pranešimo x santraukos funkcijos.

Kenkėjas gali įsiterpti į ryšio seansą tarp dviejų autentifikacijos protokolo subjektų $\mathcal{A}$ ir $\mathcal{B}$ bei perduoti pranešimus tarp subjektų, apsimesdamas ryšio kanalu. Dėl to gali būti pažeistas sistemos saugumas. Pavyzdžiui, kenkėjas gali apsimesti subjektu $\mathcal{B}$. Subjektui $\mathcal{A}$ paprašius subjekto $\mathcal{B}$ tapatybės įrodymo, kenkėjas gali (realiuoju laiku ir nepastebimai apsimesdamas subjektu $\mathcal{A}$) perduoti užklausas tikrajam $\mathcal{B}$, gauti tinkamą atsakymą ir jį panaudoti įrodydamas savo apsimestą tapatybę subjektui $\mathcal{A}$. Tokiu atveju reikia imtis papildomų saugumo priemonių.

Autentifikacijos protokolai suteikia tapatybės įrodymą tik duotu laiko momentu. Prireikus tokio įrodymo tęstinumo, taikomi papildomi metodai, siekiant apsisaugoti nuo aktyvių kenkėjų. Pavyzdžiui, autentifikacija buvo atlikta ryšio seanso pradžioje, kad būtų gauta prieiga prie slaptų informacijos išteklių. Kenkėjas gali įsiterpti į ryšio liniją iš karto po sėkmingos tikrojo subjekto autentifikacijos. Apsisaugoti galima šiais būdais:

- Atliekant periodinę autentifikaciją arba autentifikuojantis esant kiekvienam konkrečiam sisteminių išteklių prašymui (pvz., kiekvienai prieinamai rinkmenai). Deja, kenkėjas gali palikti ryšio liniją kiekvieną kartą prieš atliekant pakartotinę autentifikaciją, leisdamas tikrajam vartotojui sėkmingai autentifikuotis prieš vėl grįždamas į ryšio liniją.
- Susiejant autentifikaciją su teikiama paslauga. Šiuo atveju autentifikacija turėtų būti susieta su kriptografinio rakto nustatymo algoritmu, kad sėkmingai autentifikavus būtų gautas raktas, reikalingas tolesniems veiksmams atlikti.

9. PSEUDOATSITIKTINIŲ SKAIČIŲ GENERATORIŲ KŪRIMAS, TYRIMAS IR PRITAIKYMAS

Atsitiktiniai skaičiai yra labai svarbūs įvairiose kriptografinėse sistemose, todėl reikia turėti patikimą atsitiktinių skaičių šaltinį.

Kiekvienas atsitiktinių skaičių sekos narys yra nepriklausomas nuo kitų reikšmių, t. y. nė vienas šios sekos skaičius matematiniais sąryšiais nesusijęs su kitais elementais. Be to, atsitiktinių skaičių seka yra neprognozuojama.

Kriptografijoje yra naudojami pseudoatsitiktinių skaičių generatoriai (PASG). PASG gaunama reikšmių seka nėra visiškai atsitiktinė, tačiau pasižymi atsitiktinės skaičių sekos savybėmis. Bet kuris PASG generuoja baigtinę skaičių seką, t. y. pradedant kažkuria reikšme PASG gaunamos reikšmės pradės kartotis. Generatoriaus ciklo ilgis – viena iš PASG savybių. Tačiau patikimo pseudoatsitiktinių skaičių generatoriaus sugeneruotos skaičių sekos statistiniais metodais neįmanoma atskirti nuo fizikinio triukšmo generatoriaus gautos sekos. Be to, kuriant PASG siekiama, kad gaunamos reikšmės būtų nekoreliuotos ir tolygiai pasiskirsčiusios.

Šiame skyriuje apžvelgsime kai kuriuos PASG bei statistinius pseudoatsitiktinių skaičių testus.

9.1. KONGRUENCINIAI GENERATORIAI

Tiesiniai kongruenciniai generatoriai (TKG) yra vieni seniausių ir geriausiai žinomų pseudoatsitiktinių skaičių generavimo algoritmų. Tačiau šio tipo generatorių savybės nėra idealios. TKG apibrėžiami rekurenčiuoju ryšiu:

$$x_{n+1} = (Ax_n + B) \bmod M. \qquad (9.1)$$

Kita atsitiktinė reikšmė x_{n+1} apskaičiuojama pasitelkus prieš tai buvusią reikšmę x_n, skaitines konstantas A ir B bei skaitinį modulį M. Reikšmė $Ax_n + B$ apskaičiuojama moduliu M ir taip gaunamas naujas atsitiktinis skaičius x_{n+1}.

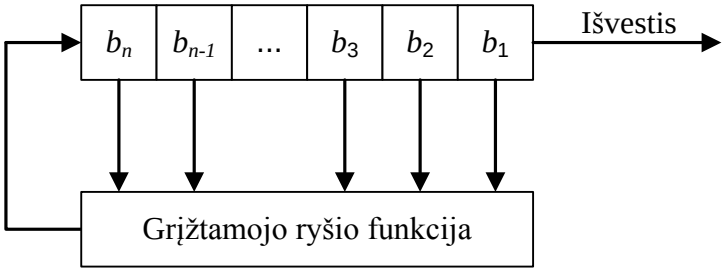
Didžiausias galimas TKG periodas yra M. Suformuluosime tris sąlygas, kurios yra būtinos ir pakankamos norint gauti didžiausią galimą periodo ilgį M:

1. B ir M yra tarpusavyje pirminiai.
2. Kiekvienam pirminiam p, dalijančiam M, A–1 yra p kartotinis.
3. Jei M dalus iš 4, tai A–1 yra 4 kartotinis.

TKG galima apibendrinti taip: $x_{n+1} = P_m(x_n) \bmod M$; čia $P_m(x_n)$ yra m-tosios eilės polinomas. Tačiau kongruenciniai generatoriai nenaudojami kriptografijoje, nes jie lengvai prognozuojami.

9.2. TIESINIAI GRĮŽTAMOJO RYŠIO POSTŪMIO REGISTRAI

Grįžtamojo ryšio postūmio registrai yra sudaryti iš dviejų dalių: postūmio registro ir grįžtamojo ryšio funkcijos. Postūmio registras – bitų seka, o šios sekos ilgis rodo registro ilgį. Postūmio registrai dažniausiai generuoja po vieną bitą, t. y. kita reikšmė gaunama perstūmus registrą per vieną bitą į dešinę. Išvestyje gaunamas mažiausiai reikšmingas postūmio registro bitas, o reikšmingiausią bitą keičia grįžtamojo ryšio funkcijos reikšmė (9.1 pav.)



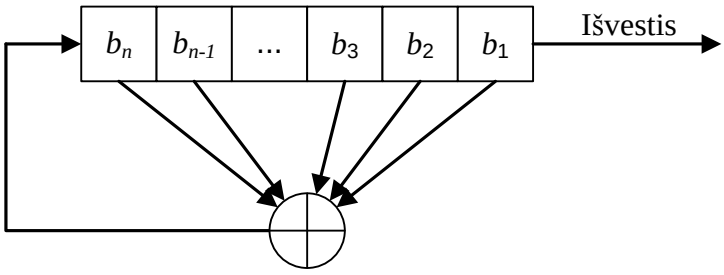
9.1. pav. Grįžtamojo ryšio postūmio registro schema

Grįžtamojo ryšio postūmio registro veiksmai apibrėžiami formulėmis:

$$x_i = b_1,$$
$$b_j = b_j+1,\ j = 1, 2, ..., n–1,$$
$$b_n = f(b_{n–1}, b_{n–2}, ..., b_1, x_i),$$

čia x_i – generatoriaus išvestis.

Paprasčiausias grįžtamojo ryšio postūmio registras gaunamas tada, kai grįžtamojo ryšio funkcija yra kelių bitų sudėtis moduliu 2 (9.2. pav.). Toks generatorius vadinamas tiesinio grįžtamojo ryšio postūmio registru (TGPR, angl. LFSR – *linear feedback shift register*).



9.2. pav. Tiesinio grįžtamojo ryšio postūmio registro schema

Grįžtamojo ryšio funkciją paprasta užrašyti kaip daugianarį $a_nx^n + a_{n–1}x^{n–1} + ... a_1x + 1$ virš lauko GF(2). Pavyzdžiui, grįžtamojo ryšio funkcija aprašoma daugianariu $x^5 + x^2 + 1$. Paskui, skaičiuojant funkcijos reikšmę, sudedami penktoje ir antroje ląstelėse esantys bitai.

TGPR iš viso turi $2^n–1$ vidinių būsenų, o didžiausias įmanomas ciklo ilgis – $2^n–1$. Didžiausias galimas ciklo ilgis gaunamas grįžtamojo ryšio funkciją aprašant pirminiu daugianariu. Todėl, kuriant TGPR, grįžtamojo ryšio funkciją turi aprašyti reikiamos eilės pirminis daugianaris.

9.1. Pavyzdys. Tarkime, kad TGPR aprašo daugianaris $x^3 + x + 1$. Nustatykime šio generatoriaus ciklo ilgį ir užrašykime išvesties reikšmes, kai inicializacijos metu postūmio registras įgyja reikšmę 101.

Kadangi daugianaris $x^3 + x + 1$ yra pirminis, šio TGPR generatoriaus ciklo ilgis turi būti $2^3–1 = 7$. Nagrinėjamu atveju grįžtamojo ryšio funkcija sudeda pirmą ir paskutinį bitus, esančius postūmio registre. Surašykime visas generatoriaus vidines būsenas ciklo metu ir gaunamas išvestis:

$$101 \rightarrow 1$$
$$010 \rightarrow 0$$
$$001 \rightarrow 1$$
$$100 \rightarrow 0$$
$$110 \rightarrow 0$$
$$111 \rightarrow 1$$
$$011 \rightarrow 1$$
$$101 \rightarrow 1$$

Gauname 10100111 bitų seką, o generatoriaus vidinė būsena pasikartoja po septynių iteracijų.

Pastebėkime, kad bet kuriuo metu TGPR generatoriaus vidinė būsena tiksliai nusako kitas n išvestis; kur n yra registro ilgis. Nežinant grįžtamojo ryšio funkcijos, pakanka turėti $2n$ bitų išvesties seką, ir jau galima atskleisti vidinę TGPR generatoriaus struktūrą, t. y. TGPR išvesties duomenys yra stipriai koreliuoti.

9.3. BBS GENERATORIUS

Įrodyta, kad bet kuri vienkryptė funkcija apibūdina atsitiktinių bitų generatorių (Goldwasser, et al., 2001). Tokiu būdu ir RSA funkcija $x^e \bmod n$, $n = pq$, ir modulinė eksponentė $g^x \bmod p$ gali būti naudojamos kuriant atsitiktinių skaičių generatorius.

Panagrinėsime BBS (Blum, Blum, Schub) bitų generatorių, kuris sukonstruotas naudojant vienkryptę funkciją $x^2 \bmod n$, $n = pq$.

Apibrėžiant BBS generatorių, pasirenkami tokie du dideli pirminiai skaičiai p ir q, kad $p \bmod 4 = q \bmod 4 = 3$. Generatorius inicializuojamas reikšme s, $(s, pq) = 1$, t. y. skaičiai s ir pq yra reliatyviai pirminiai. BBS generatoriaus algoritmas:

9.1. Algoritmas. BBS generatorius

(įvestis s, $n = pq$)

$x_0 = s^2 \bmod n$
nuo $i = 1$ iki ∞
  $x_i = x_{i-1}^2 \bmod n$
  $b_i = x_i \bmod 2$

(išvestis $\{b_i\}$)

9.2. Pavyzdys. Tarkime, kad BBS generatoriaus parametrai $p = 43$, $q = 47$, $s = 1035$. Sugeneruokime 10 atsitiktinių bitų.
Įsitikiname, kad parametrai parinkti tinkamai:
$p = 10 \cdot 4 + 3$, $q = 11 \cdot 4 + 3$, $s = 22 \cdot 47 + 1 = 24 \cdot 43 + 3$,
t. y. $p \bmod 4 = q \bmod 4$ ir $(s, pq) = 1$
$n = 43 \cdot 47 = 2021$
$x_0 = 1035^2 \bmod 2021 = 95$
$x_1 = 95^2 \bmod 2021 = 941$, $b_i = 1$
$x_2 = 941^2 \bmod 2021 = 283$, $b_i = 1$
$x_3 = 283^2 \bmod 2021 = 1270$, $b_i = 0$
$x_4 = 1270^2 \bmod 2021 = 142$, $b_i = 0$
$x_5 = 142^2 \bmod 2021 = 1975$, $b_i = 1$
$x_6 = 1975^2 \bmod 2021 = 95$, $b_i = 1$
$x_7 = 95^2 \bmod 2021 = 941$, $b_i = 1$
$x_8 = 941^2 \bmod 2021 = 283$, $b_i = 1$
$x_9 = 283^2 \bmod 2021 = 1270$, $b_i = 0$
$x_{10} = 1270^2 \bmod 2021 = 142$, $b_i = 0$
Gauname 1100111100 bitų eilutę.

Pastebėkime, kad BBS generatoriaus ciklas gali būti labai trumpas.

9.2 pavyzdyje parametrai atitinka visus reikalavimus, tačiau generatoriaus ciklas sudarytas iš šešių reikšmių: $95 \to 941 \to 283 \to 1270 \to 142 \to 1975 \to 95$.

Galima pastebėti, kad žinant visus BBS generatoriaus parametrus, t. y. du pirminius skaičius p ir q, bei inicializavimo reikšmę s, galima nesunkiai nustatyti j-tąjį išvesties sekos bitą b_j, neskaičiuojant $b_1, b_2, ..., b_{j-1}$:

$$x_1 = s^2 \bmod pq,\ x_2 = x_1^2 \bmod pq = s^4 \bmod pq,\ ...,\ x_j = x_{j-1}^2 \bmod pq = s^{2^j} \bmod pq,\ b_i = x_i \bmod 2.$$

Gautąją išraiška galima pertvarkyti. Pažymėkime $\alpha = 2^j \bmod (p-1)(q-1)$, tada pagal Oilerio teorema $s^{2^j} \bmod pq = s^\alpha \bmod pq$.

9.3. Pavyzdys. Tarkime, kad BBS generatoriaus parametrai $p = 7$, $q = 11$, $s = 4$. Raskime septintą išvesties sekos bitą.
I būdas. Atliekame septynias BBS generatoriaus iteracijas ir gauname reikiamą rezultatą.
$x_1 = s^2 \bmod 77 = 4^2 \bmod 77 = 16$, $b_1 = 0$
$x_2 = x_1^2 \bmod 77 = 16^2 \bmod 77 = 25$, $b_2 = 1$
$x_3 = x_2^2 \bmod 77 = 25^2 \bmod 77 = 9$, $b_3 = 1$
$x_4 = x_3^2 \bmod 77 = 9^2 \bmod 77 = 4$, $b_4 = 0$
$x_5 = x_4^2 \bmod 77 = 4^2 \bmod 77 = 16$, $b_5 = 0$
$x_6 = x_5^2 \bmod 77 = 16^2 \bmod 77 = 25$, $b_6 = 1$
$x_7 = x_6^2 \bmod 77 = 25^2 \bmod 77 = 9$, $\boldsymbol{b_7 = 1}$
II būdas.
Pasinaudodami BBD generatoriaus savybe, gauname $x_7 = s^{2^7} \bmod 77 = s^{128 \bmod (7-1)(11-1)} \bmod 77 = s^8 \bmod 77 = 4^8 \bmod 77 = 9$, $\boldsymbol{b_7 = 1}$.

9.4. Apibrėžimas. Sakoma, kad bitų generatorius tenkina paskesnio bito kriterijų, jei neegzistuoja polinominis laiko algoritmas, kuris pagal turimus k bitų prognozuotų $k + 1$ bitą, esant didesnei nei 0,5 tikimybei. Generatoriai, tenkinantys paskesnio bito kriterijų, vadinami kriptografiškai saugiais bitų generatoriais.

Paskesnio bito kriterijus reiškia, kad bitų generatoriaus išvestyje gaunamas 0 arba 1 (tikimybė – 0,5), neatsižvelgiant į generatoriaus vidinę būseną.

Įrodyta, kad BBS bitų generatorius tenkina paskesnio bito kriterijų (Goldwasser, et al., 2001).

9.4. DINAMINIŲ SISTEMŲ GENERATORIAI

Egzistuoja dinaminės sistemos, aprašomos lygtimi $x_{i+1} = \mu\, x_i(1 - x_i)$, $0 < x_i < 1$. Sistemos elgesys priklauso nuo parametro μ reikšmės. Kai parametras $\mu = 4$, gaunama chaotiška dinaminė sistema. Toks sistemos elgesys tinkamas generuojant atsitiktinius skaičius, nes svarbu gauti kuo mažiau koreliuotas skaičių sekas ir kuo didesnį generatoriaus ciklo ilgį.

Taigi vienas galimų netiesinių generatorių aprašomas lygtimi:

$$x_{i+1} = 4x_i(1 - x_i),\ 0 < x_i < 1. \tag{9.2}$$

Šį generatorių galima apibendrinti tokia išraiška:

$$x_{i+1} = (1 + B)\left(1 + \frac{1}{B}\right)^B x_i(1 - x_i)^B,\ 1 \le B \le 4,\ 0 < x_i < 1. \tag{9.3}$$

Šie generatoriai pasižymi maža gaunamų reikšmių koreliacija. Jų ciklai nerasti.

9.5. STATISTINIAI PASG TESTAI

Siekiant nustatyti, ar generatoriaus gauta pseudoatsitiktinių bitų seka pasižymi tikros atsitiktinės sekos savybėmis (Menezes, et al., 2001), sudaryti penki pagrindiniai statistiniai testai.

Dažnumo testu (angl. *frequency test*) tikrinama, ar gautoje bitų sekoje nulių ir vienetų skaičius statistiškai nesiskiria. Tarkime, n_0 ir n_1 yra atitinkamai nulių ir vienetų skaičius nagrinėjamoje sekoje, n – nagrinėjamos sekos ilgis. Tuomet dydis $X_1 = \dfrac{(n_0 - n_1)^2}{n}$ yra pasiskirstęs pagal Chi kvadrato dėsnį su 1 laisvės laipsniu.

Dviejų bitų testu (angl. *two-bit test*) tikrinama, ar nagrinėjamoje bitų sekoje porų 00, 01, 10, 11 skaičius statistiškai nesiskiria. Tarkime, n_0 ir n_1 yra atitinkamai nulių ir vienetų skaičius, n_{00}, n_{01}, n_{10}, n_{11} yra

atitinkamų bitų porų skaičius nagrinėjamoje sekoje, o n – nagrinėjamos sekos ilgis. Tuomet dydis $X_2 = \frac{4}{(n-1)}\left(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2\right) - \frac{2}{n}\left(n_0^2 + n_1^2\right) + 1$ yra pasiskirstęs pagal Chi kvadrato skirstinį su dviem laisvės laipsniais.

Pokerio testu (angl. *poker test*) tikrinama, ar m bitų atkarpos dažnis visoje nagrinėjamoje sekoje atitinka atsitiktinį bitų pasiskirstymą. Sakykime, kad m tenkina sąryšį $\left\lfloor \frac{n}{m} \right\rfloor \geq 5 \cdot 2^m$. Nagrinėjamą bitų seką dalijame į $k = \left\lfloor \frac{n}{m} \right\rfloor$ nepersidengiančių m ilgio blokų. Apskaičiuojame i-tosios dalies dažnį n_i. Tuomet dydis $X_3 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - 5$ yra pasiskirstęs pagal Chi kvadrato dėsnį su $2^m - 1$ laisvės laipsniais.

Serijų testu (angl. *runs test*) tikrinama, ar vienodų reikšmių serijų dažnis atitinka atsitiktinį bitų pasiskirstymą. Tarkime, kad $e_i = (n - i + 3)/2^{i+2}$. Tada k yra didžiausias sveikasis skaičius i, su kuriuo $e_i \geq 5$, o B_j ir G_j – serijų, sudarytų iš j nulių arba vienetų, skaičius nagrinėjamoje sekoje. Dydis $X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i}$ yra pasiskirstęs pagal Chi kvadrato dėsnį su $2k - 2$ laisvės laipsniais.

Autokoreliacijos testu (angl. *autocorelation test*) tikrinama, ar yra autokoreliacinių ryšių nagrinėjamoje bitų sekoje s. Tarkime, kad $1 \leq d \leq \left\lfloor \frac{n}{2} \right\rfloor$. Apskaičiuojama reikšmė $A(d) = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}$. Tuomet dydis $X_5 = \frac{2\left(A(d) - \frac{n-d}{2}\right)}{\sqrt{n-d}}$ yra pasiskirstęs pagal normalųjį skirstinį $N(0, 1)$.

Pagal FIPS-140-2 standartą yra numatyti reikalavimai pseudoatsitiktinių skaičių generatoriams, naudojamiems kriptografinėse sistemose. Šie reikalavimai labai susiję su aukščiau pateiktais penkiais statistiniais testais. Pagal FIPS reikalavimus tikrinamos 20000 bitų ilgio sekos. FIPS-140-2 standarto kriterijai ir jų kritinės reikšmės pateikti 9.1 lentelėje.

9.1. lentelė

FIPS-140-2 standarto reikalavimai

Testas	Kritinė reikšmė
Dažnumo testas	$9654 < n_1 < 10346$
Pokerio testas	$1{,}03 < X_3 < 57{,}4$
Trumpų serijų testas	Turi būti nuo 2267 iki 2733 vieno bito serijų. Turi būti nuo 1079 iki 1421 dviejų bitų serijos. Turi būti nuo 502 iki 748 trijų bitų serijų. Turi būti nuo 223 iki 402 keturių bitų serijų. Turi būti nuo 90 iki 223 penkių bitų serijų. Turi būti nuo 90 iki 223 šešių bitų serijų.
Ilgų serijų testas	Negali būti 34 ir daugiau bitų serijų.

Maurerio universalusis testas (angl. *Maurer's universal test*) apima aukščiau minėtus penkis statistinius testus. Tačiau norint, kad šio testo rezultatai būtų patikimi, reikia nagrinėti kur kas didesnę bitų seką nei kituose testuose.

Pasirenkamas parametras L iš intervalo [6, 16]. Nagrinėjama bitų seka s suskaidoma į $Q+K$ nepersidengiančių L ilgio blokų. Tarkime, kad b_i ($i \leq 1 \leq Q+K$) yra sveikasis skaičius, kurio dvejetainė išraiška sutampa su i-tojo bloko turiniu. Pirmieji Q blokai inicijuoja tarpinių reikšmių lentelę T. Visi kiti K blokai naudojami apskaičiuojant statistiką $X_u = \frac{1}{K}\sum_{j=Q+1}^{Q+K} \ln\left(i - T(b_i)\right)$.

9.2. Algoritmas. Maurerio universalusis testas

(įvestis $\{b_i\}$, $i = \overline{0, n-1}$, L, K, Q)

---

nuo $j = 0$ iki 2^{L-1}
$T[j] = 0$
nuo i = 1 iki Q
    $T[b_i] = i$
suma = 0
nuo $i = Q + 1$ iki $Q + K$
    suma = suma + $\ln(i - T[b_i])$
    $T[b_i] = i$
X_u = suma / K

(išvestis X_u)

Tuomet dydis $Z_u = \dfrac{X_u - \mu_L}{\sigma_L}$ yra pasiskirstęs pagal normalųjį skirstinį $N(0, 1)$. Parametrai μ_L ir σ_L^2 yra statistikos X_u vidurkis ir dispersija. Jų reikšmes kai L yra iš intervalo [6; 16] pateiktos 9.2 lentelėje.

9.2. lentelė

Maurerio universaliojo testo parametrų reikšmės

L	μ_L	σ_L^2	L	μ_L	σ_L^2	L	μ_L	σ_L^2
6	5,2177052	2,954	10	9,1723243	3,356	14	13,167693	3,416
7	6,1962507	3,125	11	10,170032	3,384	15	14,167488	3,419
8	7,1836656	3,238	12	11,168765	3,401	16	15,167379	3,421
9	8,1764248	3,311	13	12,168070	3,410			

Siekiant patikimų rezultatų, Q ir K turi būti pakankamai dideli. Rekomenduojama $Q \geq 10 \cdot 2^L$, $K \geq 1000 \cdot 2^L$, t. y. šiam testui atlikti reikalinga bent $1010L \cdot 2^L$ bitų seka.

10. ELIPSINIŲ KREIVIŲ (EK) ALGEBROS MATEMATINIAI PAGRINDAI

10.1. ELIPSINĖS KREIVĖS

Bendruoju atveju elipsinė kreivė aprašoma trečiosios eilės algebrine lygtimi:

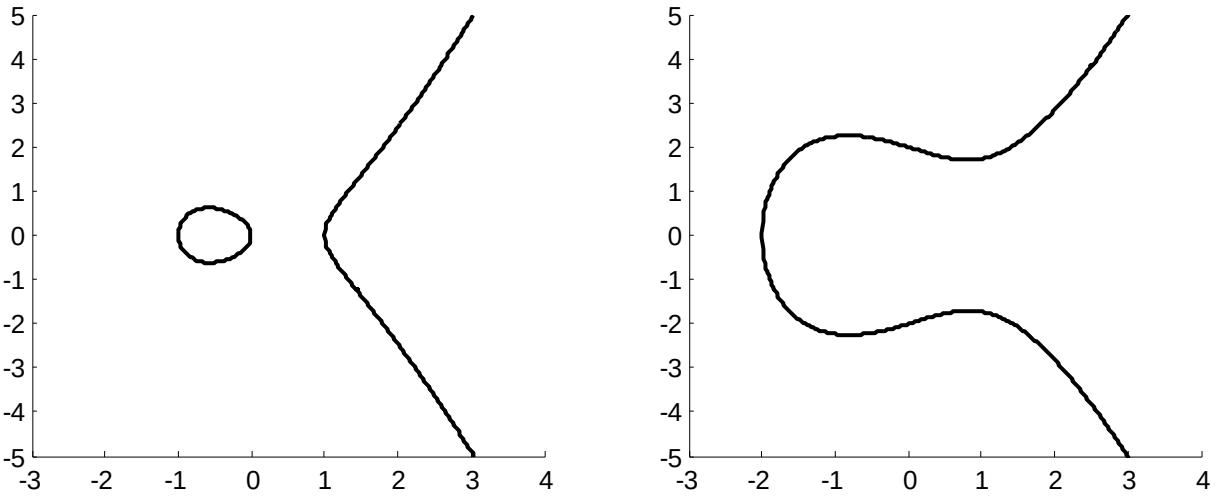
$$y^2 + a_1xy + a_2y = x^3 + a_3x^2 + a_4x + a_5. \quad (10.1)$$

Atlikus pertvarkymus, (10.1) lygtis užrašoma kanoniniu pavidalu E(a, b):

$$y^2 = x^3 + ax + b. \quad (10.2)$$

Parametrams a ir b iškeliama papildoma sąlyga:

$$4a^3 + 27b^2 \neq 0. \quad (10.3)$$



a) $y^2 = x^3 - x$ b) $y^2 = x^3 - 2x + 4$

10.1. pav. Elipsinių kreivių pavyzdžiai

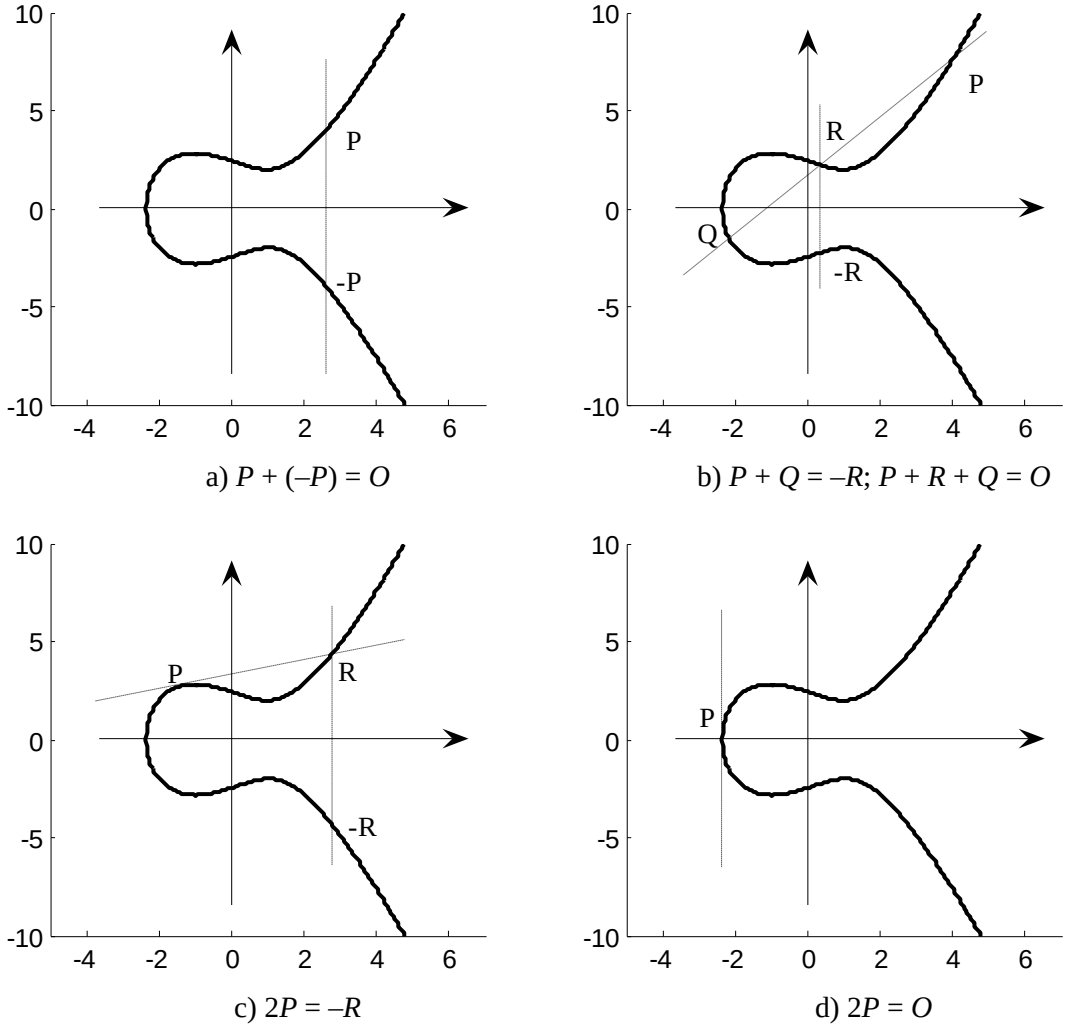
Plokštumos R^2 taškų (x, y), tenkinančių lygtį (10.2) su be galo nutolusiu tašku O, aibėje galima įvesti sumos operaciją. Įvesta sumos operacija pasižymi visomis skaičių sumos savybėmis, t. y. tenkina asociatyvumo ir komutatyvumo dėsnius. Tokia aibė su joje apibrėžta sumos operacija sudaro algebrinę grupę.

Dviejų taškų sumos operacija atliekama vadovaujantis šiomis taisyklėmis:

- Be galo nutolęs taškas yra neutralus elementas O. Bet kuriam elipsinės kreivės taškui P yra teisinga sąlyga $P + O = P$.
- Bet kurio elipsinės kreivės taško $P = (x, y)$ priešingas taškas yra simetriškas Ox ašies atžvilgiu: $-P = (x, -y)$. Be to, vertikali tiesė, einanti per taškus P ir $-P$, taip pat eina ir per be galo nutolusį tašką O (10.2.a pav.).
- Dviejų elipsinės kreivės taškų P ir Q suma nustatoma radus tiesės PQ ir elipsinės kreivės dar vieną bendrąjį tašką R. Tokiu atveju sumos rezultatas yra taškas, priešingas taškui R, t. y. $P + Q = -R$ (10.2.b pav.).
- Norint padvigubinti tašką P, t. y. atlikti veiksmą $P + P$, brėžiama elipsinės kreivės liestinė taške P, randamas liestinės ir elipsinės kreivės sankirtos taškas R, ir operacijos rezultatas yra taškas $-R$ (10.2.c ir 10.2.d pav.).

10.1. Pastaba. Jei tiesė, einanti per du taškus, arba liestinė kažkuriame taške neturi kitų bendrųjų taškų su elipsine kreive, tuomet atliekamos operacijos rezultatas yra be galo nutolęs taškas *O*. (10.2.a ir 10.2.d pav.)

10.2. Pastaba. Trijų elipsinės kreivės taškų, esančių vienoje tiesėje, suma yra be galo nutolęs taškas *O*. (10.2.b pav.)



a) $P + (-P) = O$ b) $P + Q = -R$; $P + R + Q = O$

c) $2P = -R$ d) $2P = O$

10.2. pav. Veiksmai elipsinėje kreivėje

Konkrečios elipsinės kreivės E(a, b) atveju dviejų taškų $P = (x_1, y_1)$ ir $Q = (x_2, y_3)$ suma $R = (x_3, y_3)$ apibrėžiama formulėmis:

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= \lambda(x_1 - x_3) - y_1; \end{aligned} \tag{10.4}$$

čia

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{kai } P \neq Q, \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{kai } P = Q. \end{cases} \tag{10.5}$$

Turint sumos operaciją, galima įvesti taško dauginimo iš teigiamo sveikojo skaičiaus operaciją. Norint padauginti tašką P iš teigiamo sveikojo skaičiaus k, reikia sudėti taško P k kopijų, t. y. $4P = P + P + P + P$.

10.2. ELIPSINĖS KREIVĖS BAIGTINIAME LAUKE

Elipsinių kreivių kriptografijoje yra naudojamos elipsinės kreivės, apibrėžtos baigtiniuose laukuose. Panagrinėkime elipsinės kreivės grupę baigtiniame lauke moduliu p (p – pirminis skaičius).

Elipsinės kreivės grupę baigtiniame lauke moduliu p nusako du tokie sveikieji neneigiami skaičiai a ir b, mažesni už p, kad

$$4a^3 + 27b^2 \bmod p \neq 0. \tag{10.6}$$

Tuomet elipsinės kreivės grupę $E_p(a, b)$ sudaro aibė porų (x, y) kartu su be galo nutolusiu tašku O. Skaičiai x ir y yra sveikieji neneigiami, mažesni už p, ir tenkina lygybę

$$y^2 = x^3 + ax + b \bmod p. \tag{10.7}$$

Elipsinės kreivės grupė baigtiniame lauke taip pat yra baigtinė. Visi šios grupės taškai randami taip:

1. Apskaičiuojami visų sveikųjų teigiamų skaičių, mažesnių už p, kvadratai moduliu p, t. y. kiekvienam t ($1 \le t < p$) apskaičiuojamas $t^2 \bmod p$.
2. Kiekvienam x ($0 \le x < p$) apskaičiuojamas reiškinys $x^3 + ax + b \bmod p$. Jei gauta reikšmė yra ir baigtinio lauko elementų kvadratų lentelėje, tuomet, ištraukus kvadratinę šaknį iš $x^3 + ax + b \bmod p$, taškas (-ai) (x, y) yra grupės elementas. Čia y atitinka kvadratinę šaknį (moduliu p) iš $x^3 + ax + b$.
3. Prie gautos taškų aibės prijungiamas be galo nutolęs taškas O.

Bendruoju atveju tiksliai nusakyti elipsinės kreivės grupės baigtiniame lauke eilę neįmanoma. Be įrodymo suformuluosime teoremą, kuri padės įvertinti elipsinės kreivės grupės eilę.

10.3. Hasse teorema. (York, 1992) $E_p(a, b)$ yra elipsinės kreivės $y^2 = x^3 + ax + b \bmod p$ grupė. Tarkime, kad $t = p + 1 - |E_p(a, b)|$; čia $|G|$ yra ciklinės grupės G eilė. Tuomet teisingas sąryšis:

$$|t| \le 2\sqrt{p}. \tag{10.8}$$

10.4. Išvada. Atlikę nesudėtingus pertvarkymus, Hasse teoremos rezultatą galime perrašyti taip:

$$p + 1 - 2\sqrt{p} \le |E_p(a,b)| \le p + 1 + 2\sqrt{p}. \tag{10.9}$$

10.5. Pavyzdys. Raskite visus grupės $E_{11}(4, 1)$ taškus.

Remdamiesi Hasse teoremos išvada, įvertiname grupės $E_{11}(4, 1)$ eilę $5{,}4 \le |E_{11}(4,1)| \le 18{,}6$. Taigi šioje grupėje turi būti ne mažiau kaip 6 ir ne daugiau kaip 18 elementų.

Apskaičiuojame visų sveikųjų skaičių nuo 1 iki 10 kvadratus moduliu 11:

$1^2 = 1$; $2^2 = 4$; $3^2 = 9$; $4^2 = 5$; $5^2 = 3$; $6^2 = 3$; $7^2 = 5$; $8^2 = 9$; $9^2 = 4$; $10^2 = 1$.

Kiekvienam x ($0 \le x < 11$) apskaičiuojame reiškinį $f(x) = x^3 + 4x + 1 \bmod 11$.

$f(0) = 1$; 1 yra kvadratų lentelėje, todėl taškai (0, 1) ir (0, 10) yra grupės elementai;

$f(1) = 6$; 6 nėra kvadratų lentelėje;

$f(2) = 6$; 6 nėra kvadratų lentelėje;

$f(3) = 7$; 7 nėra kvadratų lentelėje;

$f(4) = 4$; 4 yra kvadratų lentelėje, todėl taškai (4, 2) ir (4, 9) yra grupės elementai;

$f(5) = 3$; 3 yra kvadratų lentelėje, todėl taškai (5, 5) ir (5, 6) yra grupės elementai;

$f(6) = 10$; 10 nėra kvadratų lentelėje;

$f(7) = 9$; 9 yra kvadratų lentelėje, todėl taškai (7, 3) ir (7, 8) yra grupės elementai;

$f(8) = 6$; 6 nėra kvadratų lentelėje;

$f(9) = 7$; 7 nėra kvadratų lentelėje;

$f(10) = 7$; 7 nėra kvadratų lentelėje.

Prie išvardytų taškų prijungiame be galo nutolusį tašką O. Tokiu būdu grupėje $E_{11}(4, 1)$ yra 9 elementai: (0, 1), (0, 10), (4, 2), (4, 9), (5, 5), (5, 6), (7, 3), (7, 8), O.

Elipsinės kreivės grupės baigtiniame lauke taip pat įvedama dviejų taškų sumos operacija. Sumos operacijos rezultatas apskaičiuojamas taip pat kaip ir realiųjų skaičių lauke, t. y. elipsinės kreivės $E_p(a, b)$ grupės dviejų elementų $P = (x_1, y_1)$ ir $Q = (x_2, y_3)$ suma $R = (x_3, y_3)$ apibrėžiama formulėmis:

$$x_3 = \lambda^2 - x_1 - x_2 \bmod p, \qquad (10.10)$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod p;$$

čia

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} \bmod p, & \text{kai } P \neq Q, \\ \dfrac{3x_1^2 + a}{2y_1} \bmod p, & \text{kai } P = Q. \end{cases} \qquad (10.11)$$

Analogiškai įvedama ir elemento daugybos iš skaičiaus operacija, pvz., $4P = P + P + P + P$. Sumos ir daugybos iš skaičiaus operacijos tenkina distributyvumo dėsnį, t. y. bet kuriems dviem taškams P ir Q iš elipsinės kreivės $E_p(a, b)$ grupės ir bet kuriam sveikajam skaičiui k yra teisinga lygybė:

$$k(P + Q) = kP + kQ. \qquad (10.12)$$

Vienas paprasčiausių būdų efektyviai apskaičiuoti operacijos kP rezultatą – pasitelkti algoritmą, besiremiantį skaičiaus k dvejetainiu išdėstymu $k = \sum_{j=0}^{l-1} b_j 2^j, b_j \in \{0,1\}$:

$$kP = \sum_{j=0}^{l-1} b_j 2^j P = 2\left(\cdots 2\left(2b_{l-1}P + b_{l-2}P\right) + \cdots\right) + b_0 P; \qquad (10.13)$$

čia l yra skaičiaus k ilgis dvejetaine išraiška.

Šis algoritmas modifikuojamas remiantis skaičiaus k išdėstymu $k = \sum_{j=0}^{l-1} a_j 2^j$, $a_j \in \{-1,0,1\}$, vadinamu skaičiaus k negretutine forma (angl. NAF – *non-adjacent form*) (Lopez, et al., 2000). Tokiu būdu gaunamas sumos ir skirtumo (angl. *addition – subtraction*) algoritmas (10.2 algoritmas).

10.1. Algoritmas. NAF(k) apskaičiavimas

(įvestis k)

$r \leftarrow k$
$l \leftarrow 0$
Kol $r > 0$, vykdyti:
  jei r nelyginis, tai
    $a_l \leftarrow 2 - (r \bmod 4)$,
    $r \leftarrow r - a_l$;
  priešingai
    $a_l \leftarrow 0$;
  $r \leftarrow r/2$, $l \leftarrow l + 1$;

(išvestis NAF(k) $\leftarrow$ (a_{l-1} a_{l-2} ... a_0))

Kiekvieno skaičiaus k NAF užrašoma vieninteliu būdu. Taip išdėstant nėra dviejų vienodų nelygių nuliui gretimų koeficientų.

10.2. Algoritmas. Sumos ir skirtumo algoritmas

(įvestis k, P)

(a_{l-1} a_{l-2} ... a_0) $\leftarrow$ NAF(k)
$Q \leftarrow O$
Nuo $j = l$-1 iki 0
  $Q \leftarrow 2Q$
  $Q \leftarrow Q + P$, jei $a_j = 1$
  $Q \leftarrow Q - P$, jei $a_j = -1$

(išvestis $Q \leftarrow kP$)

Sumos ir skirtumo algoritmas yra vidutiniškai 14 proc. efektyvesnis už dvejetainį.

10.3. ELIPSINIŲ KREIVIŲ KRIPTOSISTEMOS

1985 metais sukurtos elipsinių kreivių kriptosistemos yra kriptosistemų, pagrįstų diskretinio logaritmo problema, analogas. Šiose sistemose baigtinė multiplikacinė grupė Z_p^* yra pakeista elipsinės kreivės taškų grupe. Analogiškai suformuluota diskretinio logaritmo elipsinėje kreivėje problema.

Elipsinėje kreivėje $E_p(a, b)$ pažymimas grupę generuojantis taškas Q. Pasirinktam taškui P reikia rasti tokį sveikąjį skaičių k, kad būtų teisinga lygybė $P = kQ$.

Manoma, kad diskretinio logaritmo problema elipsinėje kreivėje yra sudėtingesnė nei analogiška problema grupėje Z_p^*.

Trumpai apžvelkime diskretinio logaritmo elipsinėje kreivėje problemos sprendimo metodus.

Pats paprasčiausias būdas yra visų galimų variantų perrinkimas. T. y. ieškomi taško Q kartotiniai taškai kQ (Q, $2Q$, $3Q$, ...), kol kuris nors iš jų bus lygus P. Atitinkama k reikšmė ir bus diskretinio logaritmo elipsinėje kreivėje atsakymas. Vykdant tokią sprendinio paiešką, daugiausiai gali tekti atlikti n žingsnių, čia n yra taško Q ciklišKumo eilė.

Pohligo ir Hellmano algoritmas (angl. *Pohlig-Hellman*) remiasi taško Q ciklišKumo eilės n skaidymu pirminiais daugikliais. Jei n turi pirminius daugiklius p_1, p_2, p_3, ..., tai Pohligo ir Hellmano algoritmas diskretinio logaritmo problemą keičia į skaičių $k \bmod p_1$, $k \bmod p_2$, $k \bmod p_3$, ... suradimo problemą, o pačią diskretinio logaritmo reikšmę galima apskaičiuoti taikant kinų teoremą apie liekanas. Kad šis algoritmas būtų neefektyvus, reikia užtikrinti, jog elipsinės kreivės grupės taško Q ciklišKumo eilė bus pakankamai didelis pirminis skaičius.

„Mažo žingsnio – didelio žingsnio“ algoritmas (angl. *Baby-Step, Giant-Step*). Tai visiško perrinkimo algoritmo išplėtimas, pasiekiantis kompromisą tarp atminties resursų bei laiko sąnaudų. Jis reikalauja saugoti atmintyje apie $\sqrt{n}$ elipsinės kreivės grupės taškų ir atlieka apytiksliai $\sqrt{n}$ žingsnių.

Pollardo *ro* algoritmas (angl. *Pollard's Rho*). Šis algoritmas yra „mažo žingsnio – didelio žingsnio“ algoritmo išplėtimas naudojant atsitiktinę paiešką. Pollardo *ro* algoritmo laiko sąnaudos yra panašios kaip „mažo žingsnio – didelio žingsnio“ algoritmo atveju (blogiausių atveju reikės $\sqrt{\frac{\pi n}{2}}$ žingsnių), tačiau jo privalumas yra menkas atminties resursų išnaudojimas. Taip pat yra parodyta, kad Pollardo *ro* algoritmą galima pagreitinti ir rezultatą gauti per $\frac{\sqrt{\pi n}}{2}$ žingsnių blogiausiu atveju. Pollardo *ro* algoritmą galima lygiagretinti ir gauti rezultatą per $\frac{\sqrt{\pi n}}{2r}$ žingsnių, čia r yra lygiagrečiai sujungtų procesorių skaičius.

Pollardo *lambda* algoritmas (angl. *Pollard's lambda*) yra dar vienas atsitiktinės paieškos algoritmas, kurį, kaip ir Pollardo *ro*, galima lygiagretinti. Bendru atveju Pollardo *lambda* algoritmas yra lėtesnis nei Pollardo *ro*. Pollardo *lambda* metodas yra naudojamas, kai sprendinio paieškos intervalą galima susiaurinti nuo $[1, n–1]$ iki $[1, b]$, $b < 0{,}39n$.

Diskretinio logaritmo problema anomaliose kreivėse. Elipsinė kreivė $E_p(a, b)$ vadinama anomalia, jeigu šios kreivės grupės eilė sutampa su skaičiumi p, t. y. $|E_p(a, b)| = p$. Yra įrodyta, kad tokiose kreivėse diskretinio logaritmo problema yra efektyviai sprendžiama. Tačiau sprendimo radimo metodika nėra perkeliama kitoms elipsinių kreivių klasėms ir todėl būtina, formuojant elipsinės kreivės grupę, patikrinti nelygybę $|E_p(a, b)| \neq p$.

Suformulavus sudėtingą algoritminę problemą, elipsinėms kreivėms nesunkiai pritaikomi žinomi kriptografiniai algoritmai. Kaip pavyzdį pateikiame ElGamalio šifravimo algoritmo apibendrinimą.

Raktų generavimo algoritmas:

1. Pasirenkama elipsinės kreivės grupė $E_p(a, b)$, kurios eilė yra q. Pažymimas grupę generuojantis taškas Q;
2. Pasirenkamas atsitiktinis skaičius x, $2 \le x \le q-2$ ir apskaičiuojamas $P = xQ$;
3. Viešasis raktas VR = P, privatusis raktas PR = x. Elipsinė kreivė $E_p(a, b)$ ir taškas Q yra sisteminiai parametrai.

Užšifravimo algoritmas:

1. $\mathcal{A}$, norėdama nusiųsti $\mathcal{B}$ pranešimą, pasiima jo viešąjį raktą $VR_B = P$.
2. Tekstograma t atvaizduojama į elipsinės kreivės $E_p(a, b)$ tašką t.
3. Pasirenkamas atsitiktinis skaičius k, $2 \le k \le p-2$.
4. Apskaičiuojami dydžiai $\gamma = kQ$ ir $\delta = t + kP$.
5. Šifrograma (γ, δ) siunčiama $\mathcal{B}$.

Iššifravimo algoritmas: naudodamas savo privatųjį raktą $PR_B = x$, $\mathcal{B}$ apskaičiuoja

$$\delta - x\gamma = t + kP - xkQ = t + kxQ - xkQ = t.$$

10.6. Pavyzdys. Pasirenkama elipsinė kreivė $E_{11}(4, 1)$, kurioje yra 9 elementai (žr.10.5 pav.). Pasirenkamas $Q = (0, 1)$.
Pasirenkamas atsitiktinis skaičius $x = 5$ ir apskaičiuojama $P = 5Q = (7, 8)$.
Tuomet sisteminiai parametrai yra $E_{11}(4, 1)$ ir $Q = (0, 1)$, viešasis raktas VR = $P = (7, 8)$, privatusis raktas PR = $x = 5$.
Tarkime, kad $\mathcal{A}$ užšifruoja tekstogramą $t = (4, 9)$, pasirenka atsitiktinį skaičių $k = 3$ ir apskaičiuoja $\gamma = kQ = 3(0, 1) = (5, 6)$ bei $\delta = t + kP = (4, 9) + 3(7, 8) = (4, 9) + (5, 5) = (0, 1)$. Šifrograma $c = ((5, 6), (0, 1))$.
$\mathcal{B}$, iššifruodamas šią šifrogramą, atlieka tokius veiksmus: $t = (0, 1) - 5(5, 6) = (0,1) - (5, 6) = (4, 9)$.

Analogiškai galima suformuluoti rakto apsikeitimo bei e. parašo sistemas elipsinėse kreivėse. Tai bus nagrinėjama kituose skyriuose.

11. RAKTŲ APSIKEITIMO PROTOKOLAI

Šiame skyriuje nagrinėsime grupinį Diffie‘io ir Hellmano raktų apsikeitimo protokolą (RAP) ciklinėje grupėje ir dviejų subjektų RAP elipsinių kreivių grupėje.

11.1. GRUPINIAI DIFFIE‘IO IR HELLMANO (GDH) RAP

Rakto apsikeitimo protokolas yra asimetrinis kriptografinis primityvas, kurio paskirtis – dviem ar didesnei grupei subjektų sukurti bendrą privatųjį raktą. Algoritmo vykdymo metu subjektai keičiasi tarpusavyje tam tikrais apskaičiuotais dydžiais, kuriuos gali stebėti pašaliniai subjektai. Baigę keistis pranešimais, kiekvienas subjektas savarankiškai apskaičiuoja privatųjį raktą, kuris yra vienodas visiems protokolo dalyviams.

Plačiausiai žinomas dviejų subjektų RAP – Diffie‘io ir Hellmano (DH) sistema. Kiekvienas iš subjektų turi viešojo ir privačiojo raktų porą. Kiekvienas subjektas, naudodamasis savo privačiuoju ir savo ar kito subjekto viešuoju raktais, apskaičiuoja bendrą privatųjį raktą. DH RAP yra sudarytas taip, kad abu subjektai, naudodami vienodą algoritmą ir skirtingus įvesties duomenis, gautų vienodus rezultatus – bendrą privatųjį raktą.

DH RAP žingsniai:

- Aldona ir Bronius viešai susitaria dėl bendrų sistemos parametrų – didelio pirminio skaičiaus p ir grupės Z_p^* generatoriaus g.
- Aldona atsitiktinai pasirenka savo privatųjį raktą $PR_A = x$, $1 < x < p$-1, apskaičiuoja savo viešąjį raktą $VR_A = g^x \bmod p$ ir viešąjį VR_A nusiunčia Broniui.
- Bronius atsitiktinai pasirenka savo privatųjį raktą $PR_B = y$, $1 < y < p$-1, apskaičiuoja savo viešąjį raktą $VR_B = g^y \bmod p$ ir viešąjį VR_B nusiunčia Aldonai.
- Aldona, turėdama savo privatųjį raktą $PR_A = x$ ir Broniaus viešąjį $VR_B = g^y \bmod p$, apskaičiuoja bendrą paslaptį $K_A = (VR_B)^{PRA} \bmod p = g^{yx} \bmod p$.
- Bronius, turėdamas savo privatųjį raktą $PR_B = y$ ir Aldonos viešąjį $VR_A = g^x \bmod p$, apskaičiuoja bendrą paslaptį $K_B = (VR_A)^{PRB} \bmod p = g^{xy} \bmod p$.

Šį protokolą galima taikyti grupei asmenų, t. y. gauti grupinį RAP. Grupinio RAP paskirtis – apskaičiuoti grupės asmenų vieną bendrą slaptąjį raktą.

Panagrinėkime tris grupinius RAP, pagrįstus Diffie‘io ir Hellmano protokolo metodika (Steiner, et al., 1996).

Paprasčiausias dviejų subjektų DH RAP praplėtimas daugeliui vartotojų yra GDH.1 protokolas. Tarkime, yra grupė subjektų A_1, A_2, ..., A_n. Vieši protokolo parametrai – didelis pirminis skaičius p ir grupės Z_p^* generatorius g. Kiekvienas iš subjektų turi privatųjį raktą x_1, x_2, ..., x_n, $1 < x_i < p$-1. GDH.1 protokolą sudaro du etapai – užpildymo (angl. *upflow*) ir grįžimo (angl. *downflow*). Užpildymo etapo metu kiekvienas subjektas atlieka vieną kėlimo laipsniu veiksmą ir papildo gautus duomenis savo rezultatu. Tokiu būdu šiame etape tarsi įskaičiuojamas kiekvieno subjekto indėlis į bendrą raktą. Antrajame etape kiekvienas subjektas A_i atlieka i kėlimo laipsniu operacijų. Vienos operacijos rezultatas – bendras privatusis raktas, o kitos operacijos skirtos tarpiniams rezultatams gauti. Tarpiniai rezultatai siunčiami kitam subjektui.

Formaliai GDH.1 protokolą galima aprašyti taip:

I etapas. Subjektas A_1 apskaičiuoja

$$k_1 = g^{x_1} \bmod p$$

ir siunčia subjektui A_2. Subjektas A_2, gavęs k_1, apskaičiuoja

$$k_{12} = g^{x_1 x_2} \bmod p$$

ir siunčia šias reikšmes subjektui A_3.

Lygiai taip pat subjektas A_i (i = 3, 4, ..., n–1) gauna

$$\{ k_1 = g^{x_1} \bmod p,\ k_{12} = g^{x_1 x_2} \bmod p, ...,\ k_{12\ldots i-1} = g^{x_1 x_2 \ldots x_{i-1}} \bmod p \},$$

apskaičiuoja

$$k_{12\ldots i} = g^{x_1 x_2 \ldots x_{i-1} x_i} \bmod p$$

ir siunčia gautą aibę subjektui A_{i+1}.

II etapas. Subjektas A_n gauna

$$\{ k_1 = g^{x_1} \bmod p,\ k_{12} = g^{x_1 x_2} \bmod p, ...,\ k_{12\ldots n-1} = g^{x_1 x_2 \ldots x_{n-1}} \bmod p \},$$

apskaičiuoja

$$k_n = g^{x_n} \bmod p,\ k_{n1} = g^{x_n x_1} \bmod p, ...,\ k_{n12\ldots n-2} = g^{x_n x_1 x_2 \ldots x_{n-2}} \bmod p$$

ir privatųjį raktą

$$K_n = g^{x_1 x_2 \ldots x_{n-1} x_n} \bmod p.$$

A_n siunčia subjektui A_{n-1} aibę

$$\{ k_n = g^{x_n} \bmod p,\ k_{n1} = g^{x_n x_1} \bmod p, ...,\ k_{n12\ldots n-2} = g^{x_n x_1 x_2 \ldots x_{n-2}} \bmod p \}.$$

Subjektas A_i (i =1, 2, ..., n–1) su kiekvienu gautos aibės elementu, kurį žymėsime a, apskaičiuoja $a^{x_i} \bmod p$. Tada A_i, turėdamas elementą $g^{x_1 x_2 \ldots x_{i-1} x_{i+1} \ldots x_n} \bmod p$, apskaičiuoja privatųjį raktą

$$K_i = g^{x_1 x_2 \ldots x_{n-1} x_n} \bmod p,$$

o likusius aibės elementus siunčia subjektui A_{i-1}.

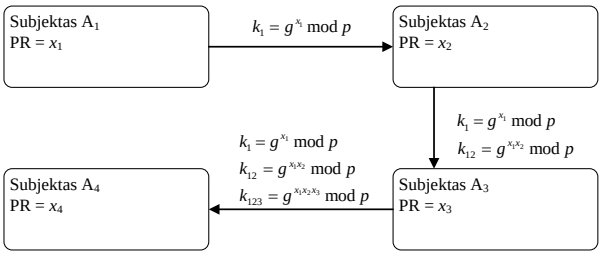
Visi apskaičiuoti raktai bus vienodi, t. y. $K_1 = K_2 = ... = K_n = K$, nes, kaip ir DH algoritmo atveju, galioja komutatyvumo lygybė:

$$g^{x_i x_j} \bmod p = g^{x_j x_i} \bmod p.$$

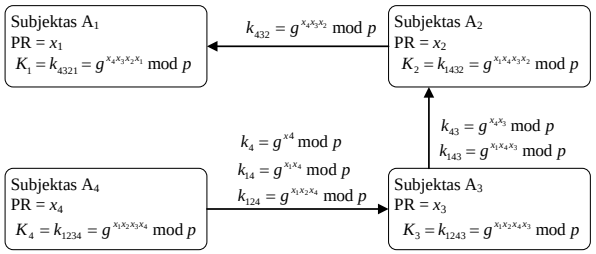
GDH.1 protokolo atliekami veiksmai pavaizduoti 11.1 pav.

Vykdant šį protokolą, atliekama 2(n–1) raundų. Raundu vadinamas ryšio seansas, kai vienas subjektas siunčia pranešimą kitam arba visai grupei. Iš viso, atlikus 2(n–1) raundų, įvykdoma (n^2+3n)/2 – 1 kėlimo laipsniu operacijų. Skirtingi subjektai skaičiavimo veiksmais apkraunami nevienodai, t. y. subjektai A_i (i<n) atlieka i+1 kėlimo laipsniu veiksmą, o A_n – n veiksmų.

GDH.1 algoritmas yra paprastas, tačiau sąlygiškai daug apkrauna tinklą ir netinkamas dinamiškai besikeičiančioms grupėms: prisidėjus naujam grupės nariui ar kuriam nors palikus grupę, tenka algoritmą vykdyti iš naujo.



I etapas



II etapas

11.1 pav. GDH.1 protokolas

11.1. Pavyzdys. Tarkime, keturi subjektai A_1, A_2, A_3, A_4 nori sukurti bendrą privatųjį raktą. Bendri sistemos parametrai yra pirminis skaičius $p = 47$ ir grupės Z_{47}^* generatorius $g = 19$. Parinkime tokius privačiuosius subjektų raktus: $x_1 = 35$, $x_2 = 22$, $x_3 = 34$, $x_4 = 2$.

I etapas:

A_1 apskaičiuoja $k_1 = g^{x_1} \bmod 47 = 19^{35} \bmod 47 = 13$ ir siunčia $k_1 = 13$ subjektui A_2.

A_2 apskaičiuoja $k_{12} = (k_1)^{x_2} \bmod 47 = 13^{22} \bmod 47 = 18$ ir siunčia $\{k_1, k_{12}\} = \{13, 18\}$ subjektui A_3.

A_3 apskaičiuoja $k_{123} = (k_{12})^{x_3} \bmod 47 = 18^{34} \bmod 47 = 9$ ir siunčia $\{k_1, k_{12}, k_{123}\} = \{13, 18, 9\}$ subjektui A_4.

II etapas:

A_4 apskaičiuoja $k_4 = g^{x_4} \bmod 47 = 19^2 \bmod 47 = 32$, $k_{14} = (k_1)^{x_4} \bmod 47 = 13^2 \bmod 47 = 28$, $k_{124} = (k_{12})^{x_4} \bmod 47 = 18^2 \bmod 47 = 42$, $K_4 = (k_{123})^{x_4} \bmod 47 = k_{1234} = 9^2 \bmod 47 = 34$ ir siunčia $\{k_4, k_{14}, k_{124}\} = \{32, 28, 42\}$ subjektui A_3.

A_3 apskaičiuoja $k_{43} = (k_4)^{x_3} \bmod 47 = 32^{34} \bmod 47 = 42$, $k_{143} = (k_{14})^{x_3} \bmod 47 = 28^{34} \bmod 47 = 18$, $K_3 = (k_{124})^{x_3} \bmod 47 = k_{1243} = 42^{34} \bmod 47 = 34$ ir siunčia $\{k_{43}, k_{143}\} = \{42, 18\}$ subjektui A_2.

A_2 apskaičiuoja $k_{432} = (k_{43})^{x_2} \bmod 47 = 42^{22} \bmod 47 = 28$, $K_2 = (k_{143})^{x_2} \bmod 47 = k_{1432} = 18^{22} \bmod 47 = 34$ ir siunčia $k_{432} = 28$ subjektui A_1.

A_1 apskaičiuoja $K_1 = (k_{432})^{x_1} \bmod 47 = k_{4321} = 28^{35} \bmod 47 = 34$.

Matome, kad įvykdžius protokolą kiekvienas subjektas turi bendrą privatųjį raktą $K = 34$.

Dažnai besikeičiančioms grupėms taikomi RAP, dalyvaujant administratoriui. Administratoriaus pareigas atlieka vienas iš subjektų. Vienas tokių protokolų yra GDH.2. Vieši protokolo parametrai – didelis pirminis skaičius $p = 2q+1$, kai q – pirminis, ir grupės Z_p^* generatorius g. Kiekvienas iš subjektų turi privatųjį raktą $x_1, x_2, ..., x_n$, $2 < x_i < p-2$. Be to, kiekvienas privatusis raktas turi būti tarpusavyje pirminis su 2 ir q.

GDH.2 protokolą taip pat sudaro du etapai: užpildymo ir transliavimo (angl. *broadcast*). Nors GDH.2 užpildymo etapo paskirtis yra tokia pat kaip ir GDH.1, tačiau atliekami kiti veiksmai. Šiuo atveju kiekvienas

subjektas A_i atlieka i kėlimo laipsniu veiksmų ir siunčia toliau i tarpinių rezultatų su (i-1) eksponentėmis ir $g^{x_1x_2\ldots x_i}$. Antrajame etape administratorius (apibrėžtumo dėlei tarkime, kad tai yra subjektas A_n) atlieka n kėlimo laipsniu veiksmų ir juos transliuoja, t. y. siunčia kiekvienam grupės nariui.

Formaliai GDH.2 protokolą galima aprašyti taip:

I etapas. Subjektas A_1 apskaičiuoja

$$g^{x_1} \bmod p$$

ir siunčia ją subjektui A_2. Subjektas A_i (i = 2, 3, ..., n–1) gauna aibę reikšmių

$$\{ g^{\prod_{k=1,k\neq j}^{i-1} x_k} \bmod p \},\ j=\overline{1,i-1},\ \text{ir reikšmę}\ g^{x_1x_2\ldots x_{i-1}} \bmod p.$$

Jis apskaičiuoja naują aibę reikšmių

$$\{ g^{\prod_{k=1,k\neq j}^{i} x_k} \bmod p \},\ j=\overline{1,i},\ \text{ir reikšmę}\ g^{x_1x_2\ldots x_i} \bmod p$$

bei siunčia gautą aibę subjektui A_{i+1}.

II etapas. Subjektas A_n gauna

$$\{ g^{\prod_{k=1,k\neq j}^{n-1} x_k} \bmod p \},\ j=\overline{1,n-1},\ \text{ir reikšmę}\ g^{x_1x_2\ldots x_{n-1}} \bmod p,$$

iš kurios apskaičiuoja

$$K_n = g^{x_1x_2\ldots x_{n-1}x_n} \bmod p.$$

A_n transliuoja visai grupei aibę

$$\{ g^{\prod_{k=1,k\neq j}^{n} x_k} \bmod p \},\ j=\overline{1,n}.$$

Kiekvienas subjektas A_i (i =1, 2, ..., n–1) pasiima jam reikalingą reikšmę

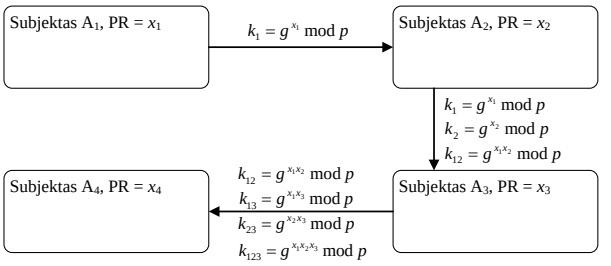
$$g^{\prod_{k=1,k\neq i}^{n} x_k} \bmod p$$

ir, atlikęs kėlimo laipsniu veiksmą, gauna privatųjį raktą

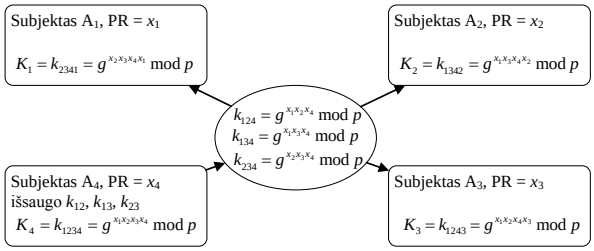
$$K_i = g^{\prod_{k=1}^{n} x_k} \bmod p.$$

Pastebėsime, kad ir šiuo atveju $K_1 = K_2 = ... = K_n$, t. y. visi subjektai turi bendrą privatųjį raktą K.

GDH.2 protokolo atliekami veiksmai pavaizduoti 11.2 pav.



I etapas



II etapas

11.2 pav. GDH.2 protokolas

Vykdant GDH.2 protokolą, atliekama n raundų, siunčiama n pranešimų ir iš viso vykdoma $(n^2+3n)/2 - 1$ kėlimo laipsniu operacijų. Kaip ir GDH.1 protokolo atveju, skirtingi subjektai atlieka nevienodai kėlimo laipsniu operacijų. Subjektai A_i, $i<n$, atlieka $i+1$ kėlimo laipsniu veiksmą, o subjektas A_n – n veiksmų. Šis protokolas efektyviausias tinklo apkrovos atžvilgiu: mažiausiai raundų ir mažiausiai siunčiamų pranešimų.

11.2. Pavyzdys. Tarkime, keturi subjektai A_1, A_2, A_3, A_4 nori pasidalyti bendru privačiuoju raktu. Bendri sistemos parametrai yra tokie pat kaip ir 11.1. pavyzdyje, t. y. pirminis skaičius $p = 47$, grupės Z_{47}^* generatorius $g = 19$. Subjektų privatieji raktai yra $x_1 = 35$, $x_2 = 21$, $x_3 = 33$, $x_4 = 3$.

I etapas:

A_1 apskaičiuoja $k_1 = g^{x_1} \bmod 47 = 19^{35} \bmod 47 = 13$ ir siunčia rezultatą subjektui A_2.

A_2 apskaičiuoja $k_2 = g^{x_2} \bmod 47 = 19^{21} \bmod 47 = 22$, $k_{12} = (k_1)^{x_2} \bmod 47 = 13^{21} \bmod 47 = 5$ ir aibę $\{k_1, k_2, k_{12}\} = \{13, 22, 5\}$ siunčia subjektui A_3.

A_3 apskaičiuoja $k_{13} = g^{x_3} \bmod 47 = 13^{33} \bmod 47 = 45$, $k_{23} = (k_2)^{x_3} \bmod 47 = 22^{33} \bmod 47 = 44$ bei $k_{123} = (k_{12})^{x_3} \bmod 47 = 5^{33} \bmod 47 = 35$ ir aibę $\{k_{12}, k_{13}, k_{23}, k_{123}\} = \{5, 45, 44, 35\}$ siunčia subjektui A_4.

II etapas:

A_4 apskaičiuoja $K_4 = (k_{123})^{x_4} \bmod 47 = k_{1234} = 35^3 \bmod 47 = 11$, $k_{124} = (k_{12})^{x_4} \bmod 47 = 5^3 \bmod 47 = 31$, $k_{134} = (k_{13})^{x_4} \bmod 47 = 45^3 \bmod 47 = 39$, $k_{234} = (k_{23})^{x_4} \bmod 47 = 44^3 \bmod 47 = 20$ ir transliuoja visiems grupės nariams aibę $\{k_{124}, k_{134}, k_{234}\} = \{31, 39, 20\}$.

A_1 pasiima reikšmę $k_{234} = 20$ ir apskaičiuoja $K_1 = (k_{234})^{x_1} \bmod 47 = k_{2341} = 20^{35} \bmod 47 = 11$.

A_2 pasiima reikšmę $k_{134} = 39$ ir apskaičiuoja $K_2 = (k_{134})^{x_2} \bmod 47 = k_{1342} = 39^{21} \bmod 47 = 11$.

A_3 pasiima reikšmę $k_{124} = 31$ ir apskaičiuoja $K_3 = (k_{124})^{x_3} \bmod 47 = k_{1243} = 31^{33} \bmod 47 = 11$.

Įvykdžius protokolą, kiekvienas subjektas turi bendrą privatųjį raktą $K = 11$.

Galima ir kitaip realizuoti grupinį RAP su administratoriumi – pasitelkus GDH.3 protokolą. Vieši protokolo parametrai yra didelis pirminis skaičius $p = 2q+1$ (q – pirminis) ir grupės Z_p^* generatorius g.

Kiekvienas iš subjektų turi privatųjį raktą x_1, x_2, ..., x_n, $2 < x_i < p$-2. Be to, kiekvienas privatusis raktas yra tarpusavyje pirminis su 2 ir q.

GDH.3 protokolas sudarytas iš keturių etapų. Formaliai jį galima aprašyti taip:

I etapas. Subjektas A_1 apskaičiuoja

$$g^{x_1} \bmod p$$

ir siunčia subjektui A_2. Subjektas A_i (i = 2, 3, ..., n–2) gauna

$$g^{x_1 x_2 \ldots x_{i-1}} \bmod p\,,$$

apskaičiuoja

$$g^{x_1 x_2 \ldots x_{i-1} x_i} \bmod p$$

ir siunčia gautą rezultatą subjektui A_{i+1}.

II etapas. Subjektas A_{n-1}, apskaičiavęs

$$g^{x_1 x_2 \ldots x_{n-1}} \bmod p\,,$$

siunčia šį rezultatą visai grupei.

III etapas. Kiekvienas subjektas A_i ($i<n$) apskaičiuoja

$$g^{\frac{x_1 x_2 \ldots x_{n-1}}{x_i}} \bmod p$$

ir rezultatą siunčia administratoriui A_n.

Reikšmė $g^{\frac{x_1 x_2 \ldots x_{n-1}}{x_i}} \bmod p$ apskaičiuojama iš pradžių radus $x_i^{-1} \bmod p-1$. Nors p–1 nėra pirminis skaičius, atvirkštinis visada egzistuos, nes privatieji raktai yra tarpusavyje pirminiai su p–1 = 2q. Vėliau atliekama kėlimo laipsniu operacija laipsniu x_i^{-1}, t. y.

$$g^{(x_1 x_2 \ldots x_{n-1}) x_i^{-1}} \bmod p = g^{\frac{x_1 x_2 \ldots x_{n-1}}{x_i}} \bmod p\,.$$

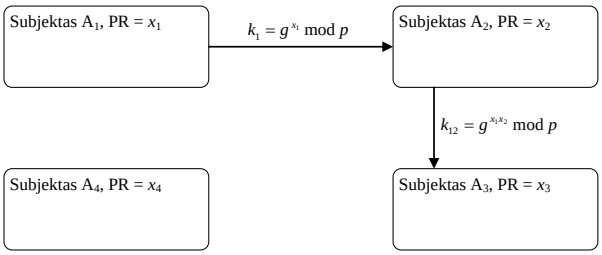
IV etapas. Administratorius apskaičiuoja aibę reikšmių

$$k_i = g^{\frac{x_1 x_2 \ldots x_n}{x_i}} \bmod p\,,\ i < n,$$

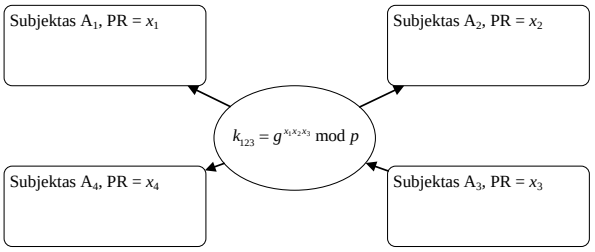
ir jas transliuoja visiems subjektams. Kiekvienas subjektas A_i apskaičiuoja bendrą privatųjį raktą

$$K = \left(g^{\frac{x_1 x_2 \ldots x_n}{x_i}} \right)^{x_i^{-1}} \bmod p = g^{x_1 x_2 \ldots x_n} \bmod p\,.$$

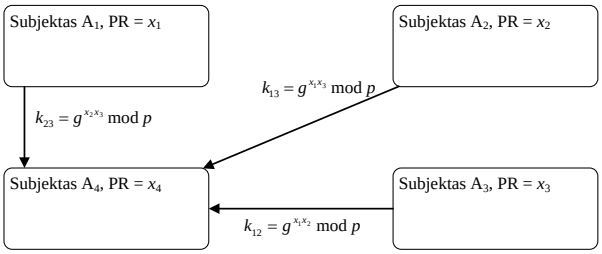
GDH.3 protokolo atliekami veiksmai pavaizduoti 11.3 pav.



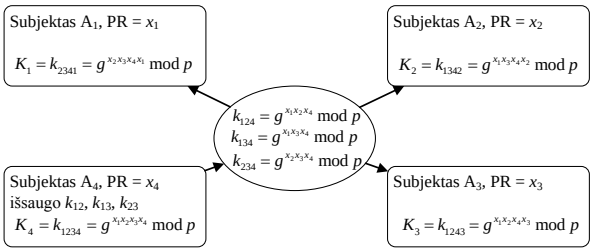
I etapas



II etapas



III etapas



IV etapas

11.3 pav. GDH.3 protokolas

Vienas iš GDH.3 algoritmo privalumų – mažai kėlimo laipsniu operacijų. Kiekvienas subjektas A_i ($i < n–2$) atlieka po keturias operacijas, subjektas $A_{n-1} – 2$, o administratorius $A_n – n$ operacijų. Bendras kėlimo laipsniu operacijų skaičius – $5n–6$. Tai yra kur kas mažiau nei GDH.1 ar GDH.2 protokolo atveju. Tokiu būdu GDH.3 protokolas yra veiksmingas esant didelei subjektų grupei. Be to, vykdant šį protokolą, kiekviename etape (išskyrus paskutinį) siunčiamų pranešimų dydis yra vienodas. Iš viso atliekama $n+1$ raundų ir siunčiama $2n–1$ pranešimų.

11.3. Pavyzdys. Tarkime, keturi subjektai A_1, A_2, A_3, A_4 nori sukurti bendrą privatųjį raktą K. Bendri sistemos parametrai yra tokie pat: pirminis skaičius $p = 47$, grupės Z_{47}^* generatorius $g = 19$. Subjektų privatieji raktai yra $x_1 = 35$, $x_2 = 21$, $x_3 = 33$, $x_4 = 3$.

I etapas:

A_1 apskaičiuoja $k_1 = g^{x_1} \bmod 47 = 19^{35} \bmod 47 = 13$ ir siunčia rezultatą subjektui A_2.

A_2 apskaičiuoja $k_{12} = (k_1)^{x_2} \bmod 47 = 13^{21} \bmod 47 = 5$ ir siunčia rezultatą subjektui A_3.

II etapas:

A_3 apskaičiuoja $k_{123} = (k_{12})^{x_3} \bmod 47 = 5^{33} \bmod 47 = 35$ ir siunčia rezultatą visiems grupės nariams.

III etapas:

A_1 apskaičiuoja $x_1^{-1} = 35^{-1} \bmod 46 = 25$, $k_{23} = (k_{123})^{x_1^{-1}} \bmod 47 = 35^{25} \bmod 47 = 44$ ir rezultatą siunčia subjektui A_4.

A_2 apskaičiuoja $x_2^{-1} = 21^{-1} \bmod 46 = 11$, $k_{13} = (k_{123})^{x_2^{-1}} \bmod 47 = 35^{11} \bmod 47 = 45$ ir rezultatą siunčia subjektui A_4.

A_3 rezultatą $k_{12} = 5$, gautą pirmajame etape, persiunčia subjektui A_4.

IV etapas:

A_4, pasinaudojęs antrajame etape gauta reikšme $k_{123} = 35$, apskaičiuoja $K_4 = (k_{123})^{x_4} \bmod 47 = k_{1234} = 35^3 \bmod 47 = 11$. Jis taip pat apskaičiuoja $k_{124} = (k_{12})^{x_4} \bmod 47 = 5^3 \bmod 47 = 31$, $k_{134} = (k_{13})^{x_4} \bmod 47 = 45^3 \bmod 47 = 39$, $k_{234} = (k_{23})^{x_4} \bmod 47 = 44^3 \bmod 47 = 20$. Reikšmes $\{k_{124}, k_{134}, k_{234}\} = \{31, 39, 20\}$ A_4 transliuoja visiems grupės nariams.

A_1 pasiima reikšmę $k_{234} = 20$ ir apskaičiuoja $K_1 = (k_{234})^{x_1} \bmod 47 = k_{2341} = 20^{35} \bmod 47 = 11$.

A_2 pasiima reikšmę $k_{134} = 39$ ir apskaičiuoja $K_2 = (k_{134})^{x_2} \bmod 47 = k_{1342} = 39^{21} \bmod 47 = 11$.

A_3 pasiima reikšmę $k_{124} = 31$ ir apskaičiuoja $K_3 = (k_{124})^{x_3} \bmod 47 = k_{1243} = 31^{33} \bmod 47 = 11$.

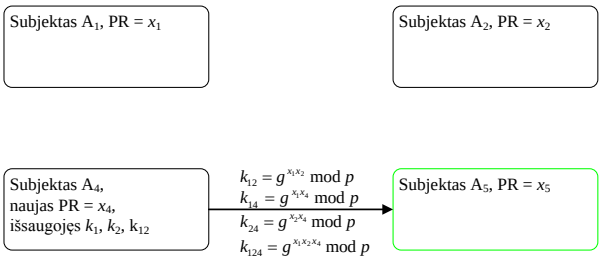
Įvykdžius protokolą, kiekvienas subjektas turi bendrą privatųjį raktą K = 11.

Protokolai GDH.2 ir GDH.3 pasižymi dar viena svarbia savybe: administratoriui išsaugojus tam tikrus tarpinius skaičiavimų rezultatus, paprasčiau į grupę įtraukti ar iš jos pašalinti narį. Šių dviejų protokolų nario įtraukimo ir pašalinimo algoritmai yra vienodi. Ir vienu, ir kitu atveju administratorius turi saugoti aibę reikšmių $\{ g^{\prod_{k=1,k\neq i}^{n-1} x_k} \bmod p \}$, $i = \overline{1, n-1}$, ir reikšmę $g^{x_1 x_2 \ldots x_{n-1}} \bmod p$.

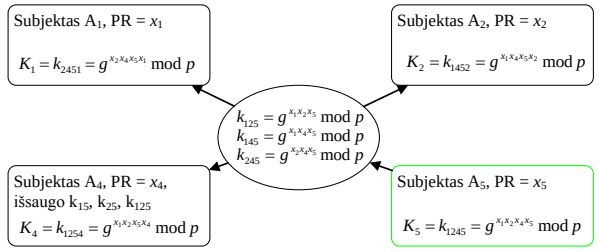
Naujasis narys A_{n+1} įtraukiamas taip (11.4 pav.):

- Administratorius sugeneruoja savo naują privatųjį raktą x_n.
- Apskaičiuoja aibę elementų $\{ g^{\prod_{k=1,k\neq i}^{n} x_k} \bmod p \}$, $i = \overline{1, n}$, ir elementą $g^{x_1 x_2 \ldots x_n} \bmod p$. Šiuos rezultatus siunčia subjektui A_{n+1}.
- Subjektas A_{n+1} sugeneruoja savo privatųjį raktą a_{n+1}, apskaičiuoja aibę elementų $\{ g^{\prod_{k=1,k\neq i}^{n+1} x_k} \bmod p \}$, $i = \overline{1, n+1}$, ir $K = g^{a_1 a_2 \ldots a_n a_{n+1}} \bmod p$. Privatųjį raktą K pasilieka sau, o visus kitus rezultatus siunčia kitiems grupės nariams, kurie, atlikę vieną kėlimo laipsniu veiksmą, turės naują bendrą privatųjį raktą:

$$K = \left(g^{\prod_{k=1,k\neq i}^{n+1} x_k} \right)^{x_i} \bmod p .$$



I etapas



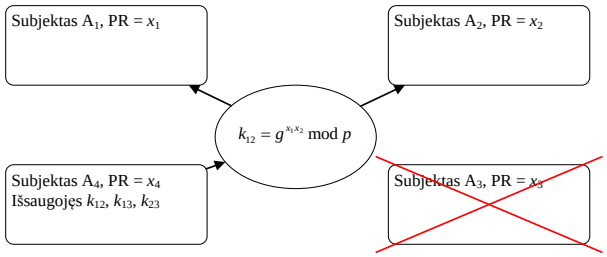
II etapas

11.4 pav. GDH.2 ir GDH.3: naujo nario įtraukimas

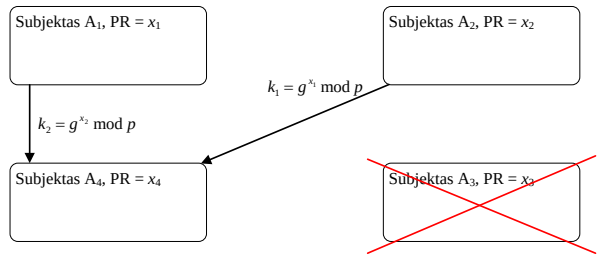
Grupės subjekto A_k ($k < n$) pašalinimas (11.5 pav.):

- Administratorius reikšmę $g^{\prod_{i=1, i\neq k}^{n-1} x_i} \bmod p$ siunčia likusiems grupės nariams A_i, $i < n$, $i \neq k$.
- Kiekvienas subjektas A_i, $i < n$, $i \neq k$, apskaičiuoja modulinę eksponentę laipsniu a_i^{-1} ir gautą rezultatą grąžina administratoriui.
- Administratorius generuoja savo naują privatųjį raktą a_n. Iš visų subjektų jis gauna aibę elementų $\{ g^{\prod_{j=1, j\neq i,k}^{n-1} x_j} \bmod p \}$, $i = \overline{1, n-1}$. Kiekvieną gautos aibės elementą pakelia laipsniu x_n ir rezultatą siunčia visiems grupės nariams. Tuomet kiekvienas subjektas, atlikęs dar vieną kėlimo laipsniu operaciją, apskaičiuoja naują bendrą privatųjį raktą:

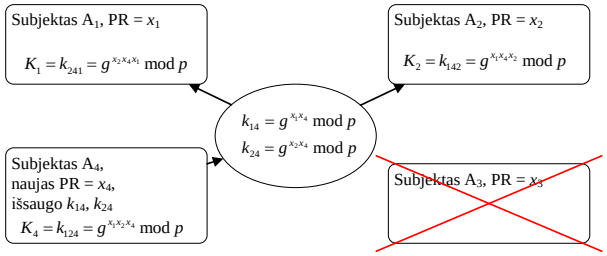
$$K = \left(g^{\prod_{j=1, j\neq k}^{n} x_j} \right)^{x_i} \bmod p .$$



I etapas



II etapas



III etapas

11.5 pav. GDH.2 ir GDH.3: nario pašalinimas

Jei iš grupės pašalinamas administratorius, tai, paskyrus naują, tenka vykdyti GDH protokolą iš naujo.

11.4. Pavyzdys. Tarkime, iš keturių subjektų A_1, A_2, A_3, A_4 grupės pašalinamas subjektas A_3 ir įtraukiamas naujas subjektas A_5. Bendri sistemos parametrai: pirminis skaičius $p = 47$ ir grupės Z_{47}^* generatorius $g = 19$. Subjektų privatieji raktai yra $x_1 = 35$, $x_2 = 21$, $x_3 = 33$, $x_4 = 3$. Imituokime pašalinimo ir įtraukimo protokolus.

A_3 pašalinimas:

Grupės administratorius A_4 turi išsaugojęs tarpinius raktus $k_{12} = 5$, $k_{23} = 44$, $k_{13} = 45$, $k_{123} = 35$. Raktą, kuriame nėra A3 subjekto indėlio k_{12}, jis išsiunčia likusiems subjektams.

A_1 apskaičiuoja $x_1^{-1} \bmod 46 = 25$ ir $k_2 = (k_{12})^{x_1^{-1}} \bmod 47 = 5^{25} \bmod 47 = 22$. Rezultatą siunčia subjektui A_4.

A_2 apskaičiuoja $x_2^{-1} \bmod 46 = 11$ ir $k_1 = (k_{12})^{x_2^{-1}} \bmod 47 = 5^{11} \bmod 47 = 13$. Rezultatą siunčia subjektui A_4.

Administratorius sugeneruoja savo naują privatųjį raktą $a_4 = 15$ ir apskaičiuoja naują bendrą privatųjį $K_4 = (k_{12})^{x_4} \bmod 47 = k_{124} = 5^{15} \bmod 47 = 41$. Taip pat apskaičiuoja $k_{14} = (k_1)^{x_4} \bmod 47 = 13^{15} \bmod 47 = 33$ bei $k_{24} = (k_2)^{x_4} \bmod 47 = 22^{15} \bmod 47 = 11$ ir gautus rezultatus siunčia grupės nariams.

Subjektas A_1 apskaičiuoja $K_1 = k_{241} = (k_{24})^{x_1} \bmod 47 = 11^{35} \bmod 47 = 41$.
Subjektas A_2 apskaičiuoja $K_2 = k_{142} = (k_{14})^{x_2} \bmod 47 = 33^{21} \bmod 47 = 41$.
Subjektai A_1, A_2, A_4 gauna naują bendrą privatųjį raktą $K = 41$.
A_5 įtraukimas:
Grupės administratorius A_4 turi išsaugojęs tarpinius raktus $k_1 = 13$, $k_2 = 22$, $k_{12} = 5$. Jis sugeneruoja savo naują privatųjį raktą $x_4 = 45$, apskaičiuoja $k_{14} = (k_1)^{x_4} \bmod 47 = 13^{45} \bmod 47 = 29$, $k_{24} = (k_2)^{x_4} \bmod 47 = 22^{45} \bmod 47 = 15$, $k_{124} = (k_{12})^{x_4} \bmod 47 = 5^{45} \bmod 47 = 19$ ir $\{k_{12}, k_{14}, k_{24}, k_{124}\} = \{5, 29, 15, 19\}$ siunčia naujam subjektui A_5.
A_5 sugeneruoja savo privatųjį raktą $x_5 = 13$ ir apskaičiuoja $K_5 = (k_{124})^{x_5} \bmod 47 = k_{1245} = 19^{13} \bmod 47 = 35$. Jis taip pat apskaičiuoja $k_{125} = (k_{12})^{x_5} \bmod 47 = 5^{13} \bmod 47 = 43$, $k_{145} = (k_{14})^{x_5} \bmod 47 = 29^{13} \bmod 47 = 45$, $k_{245} = (k_{24})^{x_5} \bmod 47 = 15^{13} \bmod 47 = 44$ ir gautus rezultatus siunčia visiems grupės nariams.
A_1 apskaičiuoja $K_1 = (k_{245})^{x_1} \bmod 47 = k_{2451} = 44^{35} \bmod 47 = 35$. A_2 apskaičiuoja $K_2 = (k_{145})^{x_2} \bmod 47 = k_{1452} = 45^{21} \bmod 47 = 35$. A_4 apskaičiuoja $K_4 = (k_{125})^{x_4} \bmod 47 = k_{1254} = 43^{45} \bmod 47 = 35$. Tokiu būdu visi subjektai turi naują bendrą privatųjį raktą $K = 35$.

Kiekvienas iš pateiktų algoritmų turi savų privalumų ir trūkumų. GDH.1 algoritmo privalumas – paprastumas, tačiau šis algoritmas labiausiai apkrauna tinklą, nes reikalauja atlikti daugiausia raundų ir turi daugiausia siunčiamų pranešimų. Be to, jis reikalauja didelių skaičiavimo sąnaudų, mat vykdoma daugiausia kėlimo laipsniu operacijų. Tačiau didžiausias šio algoritmo trūkumas tas, kad jis netinkamas dinamiškai besikeičiančioms grupėms.

Algoritmai GDH.2 ir GDH.3 yra sudėtingesni, tačiau leidžia efektyviai įtraukti naujus narius arba pašalinti esamus. Be to, GDH.2 yra efektyviausias tinklo apkrovos atžvilgiu, mat jį vykdant atliekama mažiausiai raundų ir siunčiama mažiausiai pranešimų. GDH.3 yra efektyviausias skaičiavimo sąnaudų atžvilgiu, nes jį vykdant reikia atlikti mažiausiai kėlimo laipsniu operacijų. Dar vienas GDH.3 algoritmo privalumas – pastovusis pranešimų dydis. Visuose raunduose, išskyrus paskutinį, siunčiama tik viena modulinė eksponentė.

Visų algoritmų lyginamieji duomenys pateikti 11.1 lentelėje.

11.1. lentelė

GDH algoritmų savybės

	GDH.1	GDH.2	GDH.3
Raundų skaičius	$2(n-1)$	n	$n+1$
Pranešimų skaičius	$2(n-1)$	n	$2n-1$
Bendras kėlimo laipsniu operacijų skaičius	$(n^2+3n)/2 - 1$	$(n^2+3n)/2 - 1$	$5n-6$
Narių įtraukimas ir pašalinimas	–	+	+

11.2. ELIPSINIŲ KREIVIŲ DIFFIE‘IO IR HELLMANO RAP

Tradicinio DH protokolo saugumas yra pagrįstas diskretinio logaritmo problema ciklinėje grupėje. Analogiška problema egzistuoja ir elipsinių kreivių grupėje, todėl joje galima pritaikyti visą DH protokolo metodiką. Tokiu būdu gaunamas elipsinių kreivių DH protokolas (ECDH). ECDH paremtas elipsinių kreivių matematika. (Žr. 10 skyriuje)

ECDH protokolo struktūra (Koblitz, 1999):

- Aldona ($\mathcal{A}$) ir Bronius ($\mathcal{B}$) viešai susitaria dėl bendrų sistemos parametrų – didelio pirminio skaičiaus p, elipsinės kreivės $E_p(a, b)$ ir šios kreivės taško Q.

- $\mathcal{A}$ atsitiktinai pasirenka savo privatųjį raktą $PR_A = x$ ir apskaičiuoja savo viešąjį $VR_A = xQ$. Rezultatą $\mathcal{A}$ išsiunčia $\mathcal{B}$.
- $\mathcal{B}$ atsitiktinai pasirenka savo privatųjį raktą $PR_B = y$, apskaičiuoja savo viešąjį $VR_B = yQ$ ir viešąjį VR_B išsiunčia $\mathcal{A}$.
- $\mathcal{A}$, turėdama savo privatųjį raktą $PR_A = x$ ir $\mathcal{B}$ viešąjį $VR_B = yQ$, apskaičiuoja bendrąjį raktą $K_A = x(yQ)$.
- $\mathcal{B}$, turėdamas savo privatųjį raktą $PR_B = y$ ir $\mathcal{A}$ viešąjį $VR_A = xQ$, apskaičiuoja bendrąjį raktą $K_B = y(xQ)$.
- Gauti $\mathcal{A}$ ir $\mathcal{B}$ raktai yra lygūs, t. y. $K = K_A = K_B$, nes elipsinių kreivių grupėje galioja asociatyvumo dėsnis $x(yQ) = y(xQ)$.

11.5. Pavyzdys. Imituokime ECDH protokolo veikimą.
Vieši sistemos parametrai: pirminis skaičius $p = 23$, elipsinė kreivė $E_{23}(1, 1) = z^3 + z + 1 \bmod 23$ ir elipsinės kreivės taškas $Q = (9, 7)$.
$\mathcal{A}$ pasirenka savo privatųjį raktą $PR_A = 5$, apskaičiuoja savo viešąjį $VR_A = 5Q = (19, 5)$ ir siunčia jį $\mathcal{B}$.
$\mathcal{B}$ pasirenka savo privatųjį raktą $PR_B = 4$, apskaičiuoja savo viešąjį $VR_B = 4Q = (13, 16)$ ir siunčia $\mathcal{A}$.
$\mathcal{A}$ apskaičiuoja bendrąjį raktą $K_A = 5VR_B = (5, 4)$.
$\mathcal{B}$ apskaičiuoja bendrąjį raktą $K_B = 4VR_A = (5, 4)$.
Subjektų gauti raktai sutampa: $K = K_A = K_B$.

Taikant šią metodiką, galima sukurti ir grupinius RAP, panašius į jau minėtus šiame skyriuje.

12. EK E. PARAŠO SISTEMA

Šiuolaikinėje visuomenėje sparčiai plintant interneto galimybėmis, vis dažniau naudojami elektroniniai (virtualūs) dokumentai. Kitaip nei popieriniai, kurie perduodami iš rankų į rankas arba pasiekia adresatą naudojantis tam skirtų tarnybų paslaugomis (paštu, įvairiais kurjeriais), elektroniniai dokumentai perduodami viešomis ryšio linijomis, kurios gali būti lengvai pažeidžiamos. Todėl elektroniniai dokumentai taip pat turi užtikrinti vartotojų pasitikėjimą.

Vienas e. dokumentų saugumo užtikrinimo būdų – e. parašas. Kaip ir įprastas žmogaus parašas dokumento pabaigoje, e. parašas yra tam tikra prie e. dokumento pridedama papildoma informacija, pagal kurią nustatomas dokumento autorius. Patikima e. parašo schema užtikrina apsaugą nuo dokumento klastojimo ir keitimo, taip pat siuntėjo identifikavimą bei parašo nepaneigiamumo savybę. T. y. dokumentą pasirašęs vartotojas negali išsiginti savo parašo, teigdamas, kad jis padirbtas. Be to, e. parašo gavėjas ar trečioji šalis, turėdami e. parašą ir dokumentą, gali patikrinti, ar parašas atitinka dokumento autorių ir ar dokumentas nebuvo pakeistas jį perduodant. E. parašo schemą sudaro du algoritmai – parašo formavimo ir parašo patikros. Parašo formavimo algoritmo įvesties duomenys – pasirašomas dokumentas ir privatusis vartotojo raktas PR. Patikros funkcijos įvesties duomenys – gautas pasirašytas dokumentas ir viešasis vartotojo raktas VR.

Siekiant užtikrinti vartotojo autentifikavimą, e. parašo schema turi tenkinti šias savybes:

- Paraše turi būti siuntėjo tapatybę patvirtinanti informacija. Ši informacija yra susijusi su siuntėjo PR. Tai reiškia, kad e. parašas formuojamas naudojant siuntėjo PR.
- Būtina, kad siuntėjo tapatybę patvirtinančią informaciją būtų galima sudaryti tik žinant siuntėjo PR.
- Gavėjas, nežinodamas siuntėjo PR, turi lengvai patikrinti, ar siuntėjas pasirašė e. dokumentą savo PR. Tikrinama siuntėjo VR, kuris yra matematiškai susijęs su atitinkamu PR.
- Nežinant siuntėjo privačiojo rakto, neįmanoma suformuoti parašo, kurio patikros funkcijos rezultatas būtų teigiamas.

Realizavus šias savybes, gautas e. parašas laiduoja siunčiamo dokumento vientisumą ir vartotojo autentifikavimą. Tačiau dokumentas yra atviras visiems, t. y. e. parašas neužtikrina dokumento konfidencialumo. Norint dokumentą išlaikyti paslaptyje, būtina naudoti šifravimo algoritmus.

Šiuo metu pasaulyje paplitusias e. parašo sistemas galima klasifikuoti pagal algoritmines problemas, kuriomis grindžiamas naudojamų e. parašo sistemų saugumas:

- Skaičių faktorizacijos problema yra pagrįsta RSA e. parašo sistema.
- Diskretinio logaritmo problema – ElGamalio e. parašo sistemos pagrindas.
- Elipsinių kreivių diskretinio logaritmo problema – visų kriptografinių algoritmų, besiremiančių elipsinių kreivių algebra, pagrindas.

Šiame skyriuje nagrinėsime elipsinių kreivių e. parašo sistemą (angl. ECDSA – *elliptic curve digital signature algorithm*).

ECDSA yra DSA e. parašo sistemos analogas, sukurtas 1992 metais ir aprašytas FIPS 186-2 standarte (NIST, 2000).

12.1. ECDSA ALGORITMAS

ECDSA sisteminiai parametrai yra elipsinė kreivė $E_p(a, b)$, šios kreivės taškas Q ir elipsinės kreivės grupės eilė n.

Raktų generavimo algoritmas:

1. Pasirenkamas atsitiktinis skaičius x, $1 < x < n–1$.
2. Apskaičiuojama $P = xQ$.
3. Viešasis raktas VR = P, privatusis raktas PR = x.

E. parašo formavimo algoritmas:

1. Pasirenkamas atsitiktinis skaičius k, $1 \le k \le n–1$.
2. Apskaičiuojami $kQ = (x_1, y_1)$ ir $r = x_1 \bmod n$. Jei $r = 0$, e. parašas formuojamas iš naujo.
3. Apskaičiuojama $k^{-1} \bmod n$.
4. Apskaičiuojama e. dokumento t santrauka $h = H(t)$.
5. Apskaičiuojama $s = k^{-1}(h + xr) \bmod n$. Jei s = 0, e. parašas formuojamas iš naujo.
6. E. dokumento t e. parašą sudaro pora (r, s).

E. parašo tikrinimo algoritmas:

1. Apskaičiuojama gauto e. dokumento t santrauka $h = H(t)$.
2. Apskaičiuojama $w = s^{-1} \bmod n$.
3. Apskaičiuojami $u_1 = hw \bmod n$ ir $u_2 = rw \bmod n$.
4. Apskaičiuojamas $X = u_1Q + u_2P = (x_1, y_1)$. Jei $X = O$, t. y. X yra be galo nutolęs taškas, tai e. parašas – suklastotas. Jei $X \ne O$, apskaičiuojamas $v = x_1 \bmod n$.
5. E. parašas yra tikras tada ir tik tada, kai $v = r$.

Aprašyta e. parašo schema yra korektiška, nes iš lygybės $s = k^{-1}(h + xr) \bmod n$ išreiškę k, gauname:

$$k = s^{-1}(h + xr) \bmod n = s^{-1}h + s^{-1}xr \bmod n = wh + wxr \bmod n = u_1 + u_2x \bmod n.$$

Tuomet

$$kQ = (u_1 + u_2x)Q = u_1Q + u_2xQ = u_1Q + u_2P \text{ ir } v = r.$$

12.1. Pavyzdys.
Raktų generavimas
Pasirenkami sisteminiai ECDSA parametrai: elipsinė kreivė $E_{23}(1, 1)$, grupės generatorius $Q = (0,1)$, grupės eilė $n = 28$.
Pasirenkamas atsitiktinis skaičius $x = 11$ ir apskaičiuojama $P = 11Q = (1, 16)$. Viešasis raktas VR = $P = (1, 16)$, privatusis raktas PR = $x = 11$.
E. parašo formavimas
Tarkime, pasirašomas e. dokumentas t, kurio santrauka $h = 6$.
Pasirenkamas skaičius $k = 3$. Apskaičiuojama $3Q = 3(0, 1) = (3, 13)$. $r = 3 \bmod 28 = 3$.
Apskaičiuojama $k^{-1} = 3^{-1} \bmod 28 = 19$.
Apskaičiuojama $s = 19(6 + 11 \cdot 3) \bmod 28 = 13$.
Gaunamas e. dokumento t e. parašas $(r, s) = (3, 13)$.
E. parašo tikrinimas
Apskaičiuojama $w = 13^{-1} \bmod 28 = 13$.
Apskaičiuojami $u_1 = 6 \cdot 13 \bmod 28 = 22$ ir $u_2 = 3 \cdot 13 \bmod 28 = 11$.
Apskaičiuojama $X = 22Q + 11P = 22(0, 1) + 11(1, 16) = (7, 12) + (19, 18) = (3, 13)$.
Kadangi $v = 3 \bmod 28 = 3 = r$, e. parašas yra tikras.

Siekiant užtikrinti naudojamos e. parašo sistemos saugumą, pagal standartą numatyti reikalavimai sisteminiams parametrams bei šių parametrų generavimo algoritmai.

12.2. REIKALAVIMAI SISTEMINIAMS PARAMETRAMS

Elipsinės kreivės kriptosistemos sisteminius parametrus sudaro p, a, b, Q, n.

Sisteminiai parametrai parenkami taip, kad esami diskretinio logaritmo problemos sprendimo algoritmai būtų nepajėgūs išspręsti problemos per priimtiną laiko tarpą. Parametrai turi tenkinti šiuos reikalavimus:

- Elipsinės kreivės $E_p(a, b)$ modulis p turi būti didelis pirminis skaičius.
- Taškas Q turi generuoti ciklinį pogrupį, kurio eilė n būtų pirminis skaičius.
- Pogrupio eilė n turi tenkinti nelygybes $n > 2^{160}$ bei $n > 4\sqrt{p}$.

FIPS-186-2 nurodyti kreivės ir sisteminiai parametrai, tinkami naudoti kriptosistemose, taip pat pateikiamas algoritmas, generuojantis atsitiktinius sisteminius parametrus.

Atsitiktinės elipsinės kreivės generavimas:

1. Pasirenkamas didelis pirminis skaičius p.
2. Apskaičiuojamos $t = \lceil \log_2 p \rceil$, $s = \left\lfloor \frac{t-1}{160} \right\rfloor$ ir $v = t - 160 \cdot s$.
3. Sugeneruojama atsitiktinių bitų seka *seed*, kurios ilgis $g \geq 160$.
4. Apskaičiuojama *seed* santraukos funkcijos SHA-1 reikšmė H = SHA-1(*seed*). Kintamajam c_0 priskiriami gauto santraukos rezultato v mažiausiai reikšmingi bitai.
5. Kintamojo c_0 pirmasis bitas pakeičiamas į 0.
6. Kintamajam z priskiriamas skaičius, kurio dvejetainė išraiška sutampa su *seed* bitų eilute.
7. Apskaičiuojamos reikšmės $W_i = \text{SHA-1}((z + i) \bmod 2^g)$, kai $i = 1, 2, ..., s$.
8. Kintamajam r priskiriamas skaičius, kurio dvejetainė išraiška sutampa su $W_1|W_2|...|W_s$ bitų eilute.
9. Jei $r = 0$ arba $4r + 27 = 0 \bmod p$, tai elipsinė kreivė pradedama generuoti iš naujo, parenkant kitą *seed* reikšmę.
10. Pasirenkami tokie a ir b parametrai, kad $0 < a, b < p$ ir $r \cdot b^2 = a^3$. Vienas galimų pasirinkimų – $a = b = r \bmod p$ reikšmė.

Šis algoritmas užtikrina, kad generuojama atsitiktinė elipsinė kreivė $E_p(a, b)$, kurią visiškai apibūdina p ir *seed* reikšmių pora.

Q ir n parametrų generavimas:

1. Atsitiktinai sugeneruojama elipsinė kreivė $E_p(a, b)$.
2. Apskaičiuojamas kreivės taškų skaičius $N = |E_p(a, b)|$.
3. Patikrinama, ar N dalijasi iš didelio pirminio skaičiaus n ($n > 2^{160}$ ir $n > 4\sqrt{p}$). Jei nesidalija, grįžtama į pirmąjį žingsnį, t. y. generuojama nauja elipsinė kreivė.
4. Patikrinama, ar n nedalija $p^k - 1$, kai $1 \leq k \leq 20$. Jei dalija, grįžtama į pirmąjį žingsnį.
5. Patikrinama, ar $p \neq n$. Jei $p = n$, grįžtama į pirmąjį žingsnį.
6. Atsitiktinai pasirenkamas taškas Q_1 ir apskaičiuojamas $Q = (N/n)Q_1$. Šis žingsnis kartojamas tol, kol $Q \neq O$.

FIPS-186-2 rekomenduoja e. parašo gavėjui pasitikrinti sisteminius parametrus. Tikrinti galima trejopai:

1. Subjektai sisteminius parametrus gauna iš patikimos trečiosios šalies.
2. Subjektai, sutarę dėl inicializavimo duomenų, generuoja sisteminius parametrus savarankiškai.
3. Sisteminius parametrus generuoja e. parašo siuntėjas, o e. parašo gavėjas juos patikrina.

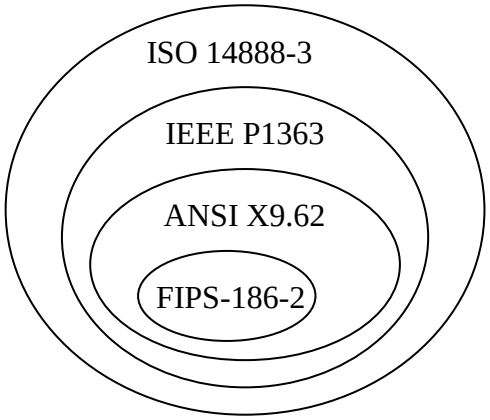
Sisteminių parametrų (p, a, b, Q, n) tikrinimo algoritmas:

1. Tikrinama, ar p yra pirminis skaičius.
2. Tikrinama, ar $4a^3 + 27b^2 \bmod p \neq 0$.
3. Tikrinama, ar parametrai a, b bei taškas Q atitinka elipsinės kreivės grupės parametrų reikalavimus, t. y. ar Q priklauso elipsinei kreivei $E_p(a, b)$ ir ar $Q \neq O$.
4. Jei sugeneruojama atsitiktinė elipsinė kreivė, tikrinama, ar *seed* reikšmė atitinka parametrus a, b. Tai daroma generuojant naują atsitiktinę elipsinę kreivę, pasitelkus *seed* ir p reikšmes.
5. Tikrinama, ar n yra pirminis skaičius, $n \neq p$, $n > 2^{160}$ bei $n > 4\sqrt{p}$.
6. Tikrinama, ar $nQ = O$.

Jei bent vienas patikrinimas yra klaidingas – sisteminiai parametrai buvo pakeisti juos siunčiant.

12.3. ECDSA STANDARTAI

Yra keletas oficialiai patvirtintų standartų, aprašančių ECDSA struktūrą. Tai yra ANSI X9.62, FIPS-186-2, IEEE P1363 ir ISO 14888-3. Trumpai apžvelkime juos.



12.1 pav. ECDSA standartai

ANSI X9.62. Pradėtas kurti 1995 metais, 1999 metais patvirtintas kaip oficialus ANSI standartas. Pagrindiniai šio standarto tikslai buvo pasiekti aukštą saugumo lygį ir užtikrinti didelį algoritmo operatyvumą. Standarte nusakyti reikalavimai baigtiniems laukams, kuriuose yra formuojamos elipsinių kreivių grupės. Elipsinės kreivės gali būti pasirenkamos iš žinomos aibės arba generuojamos atsitiktinai. Elipsinės kreivės grupei keliamas vienas reikalavimas – grupės eilė turi būti ne mažesnė nei 2^{160}.

FIPS-186-2. 1997 metais NIST paskelbė planus peržiūrėti FIPS-186 standartą, įtraukiant į jį RSA ir elipsinių kreivių e. parašo sistemas. 1998 metais paskelbtas FIPS-186-1 standartas, kuriame aprašytos RSA ir DSA e. parašo sistemų specifikacijos. Vėliau buvo paskelbtos 15 rekomenduojamų elipsinių kreivių, kurios yra suderintos su ANSI X9.62 standartu. 2000 metais standartas peržiūrėtas dar kartą, įtraukiant į jį ECDSA e. parašo sistemą, ir pavadintas FIPS-186-2 standartu. Šiame standarte pateikta ECDSA e. parašo sistemos specifikacija yra aptarta aukščiau.

IEEE P1363 standartas oficialiai paskelbtas 2000 metais. Šis standartas apima daug viešojo rakto šifravimo, rakto apsikeitimo ir e. parašo kriptografinių algoritmų, pagrįstų skaičių faktorizavimo, diskretinio logaritmo baigtiniame lauke ir elipsinėje kreivėje problemų neišsprendžiamumu. Pagrindinis šio standarto skirtumas nuo ANSI X9.62 ir FIPS-186-2 standartų yra tas, kad nėra minimalių saugumo reikalavimų sistemos parametrams, t. y. nėra nusakyta minimali elipsinės kreivės grupės eilė ir pan.

ISO 14888-3 standarte yra pateiktos kelių e. parašo schemų, tarp jų ir ECDSA, specifikacijos, kurios detaliau aprašytos ANSI X9.62 standarte.

ECDSA e. parašo sistemos, tenkinančios FIPS-186-2 standartą, taip pat tenkins ir ANSI X9.62, bet atvirkščias teiginys bendru atveju nėra teisingas. ECDSA sistemos, tenkinančios ANSI X9.62 standarto reikalavimus, tenkins ir IEEE P1363 reikalavimus, bet atvirkščias teiginys nėra teisingas. ECDSA sistemos, tenkinančios IEEE P1363 standarto reikalavimus, tenkins ir ISO 14888-3 standartą, bet atvirkščias teiginys nėra teisingas. Sąryšiai tarp standartų yra pavaizduoti 12.1 pav.

13. AKLOJO E. PARAŠO SISTEMA

Šiame skyriuje nagrinėsime dvi sritis, kuriose dažniausiai naudojami aklieji e. parašai, – tai e. pinigų ir e. balsavimo sistemos.

Akluoju e. parašu pasirašius dokumentus, jų informacija neatskleidžiama. Akluosius e. parašus galima lyginti su paprastu dokumentu ir kalke viename voke. Pasirašius ant paties voko, tuo pačiu pasirašoma ir ant voke esančio dokumento. Tokiu būdu parašas lieka dokumente net išėmus jį iš voko. Kriptografijoje naudoti aklojo e. parašo metodą pirmasis 1982 m. pasiūlė D. Chaumas[7].

13.1. AKLASIS E. PARAŠAS

Norint sukurti e. pinigų sistemą, kuri būtų anonimiška vartotojo atžvilgiu, reikia sudaryti aklojo e. parašo protokolą. Šis protokolas iš pirmo žvilgsnio turi labai neįprastą savybę. *Aldona* (*𝒜*) pateikia *Broniui* (*ℬ*) pasirašyti e. dokumentą, kurio turinys jam nežinomas. Prieš tai *𝒜* paruošia įprastai skaitomą e. dokumentą ir užšifruoja jo turinį, kurį iššifruoti gali tik ji pati. Ši operacija vadinama maskavimu. Tarkime, kad *ℬ* aklai pasirašo e. dokumentą savo e. parašu (nors ir negalėdamas perskaityti turinio) ir grąžina jį *𝒜*. *𝒜* demaskuoja (atskleidžia) e. dokumento turinį, tačiau tuo pačiu išsaugo autentišką *ℬ* e. parašą ant savo pirminio jau perskaitomo dokumento. Tokia *ℬ* e. parašo išsaugojimo sąlyga galios tik tuo atveju, jei maskavimo bei demaskavimo operacijos bus tarpusavyje suderintos ir bus komutatyvios e. parašo operacijos atžvilgiu.

Vienas maskavimo operacijų variantų – dauginti e. dokumento turinį iš kokio nors atsitiktinio skaičiaus *r*. Jeigu e. dokumentas yra nedidelis, t. y. jį galima užkoduoti skaičiumi, kurio dydis yra nedidesnis už *n*, tai toks užmaskavimo metodas yra saugus, kai $r \le n$. Daugybos operacija atliekama moduliu *n*, o *r* turi būti „tikrasis“ atsitiktinis skaičius. Tokiu atveju maskavimo algoritmas pasižymi tokiu pat „puikiu“ (angl. *perfect*) saugumu kaip ir, pavyzdžiui, Shamiro paslapties padalijimo schema. E. dokumentas, kuriuo gali būti paremtas e. pinigas, kaip tik ir gali būti užkoduotas vienu pakankamai dideliu skaičiumi, neviršijančiu *n*. Šis skaičius gali būti, pavyzdžiui, 4096 bitų eilės.

Kadangi *ℬ* yra juridiškai atsakingas už savo akląjį e. parašą ant jam nežinomo e. dokumento, šiuo veiksmu jis gali būti apgautas, jeigu *𝒜* yra nesąžiningas protokolo dalyvis. Apie tai, kaip *𝒜* gali apgauti *ℬ* ir kaip *ℬ* gali išvengti apgavystės, pakalbėsime vėliau. Dabar paminėsime dar vieną aklojo e. parašo savybę, labai svarbią e. pinigų sistemoje, dėl kurios šis parašas ir buvo sukurtas. *ℬ* negali susieti pasirašomo dokumento su pačiu pasirašymo aktu. Net saugodamas visus aklai pasirašytus e. dokumentus ir gaudamas atitinkamus jau demaskuotus e. dokumentus su autentiškais savo e. parašais, *ℬ* negalės nustatyti, kokį demaskuotą e. dokumentą atitinka maskuotas. Taip užtikrinamas e. dokumento cirkuliacijos anonimiškumas. *ℬ* gali žinoti (ir dažniausiai žino), kas yra *𝒜*, atsiuntusi maskuotą e. dokumentą, antraip jis tikrai nepasirašytų ant jam nežinomo turinio. Tačiau kai *𝒜* gauna *ℬ* pasirašytą e. dokumentą ir jį demaskuoja, dingsta ryšys tarp *𝒜* tapatybės ir dokumento (be abejo, jeigu užmaskuotame e. dokumento turinyje nėra kokios nors *𝒜* tapatybės užuominos).

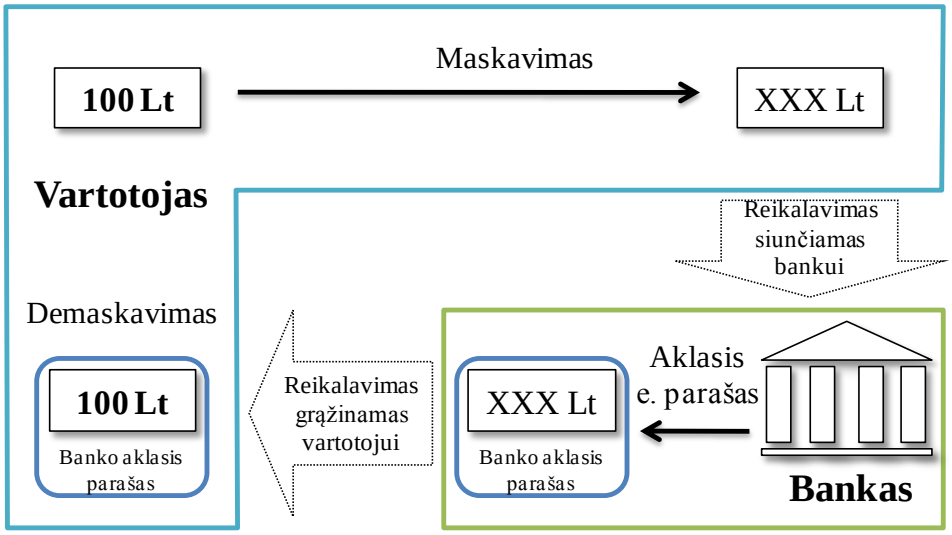
Toliau nagrinėsime e. dokumentus, kurie atitinka e. pinigus, paremtus aklaisiais parašais, ir jų akluosius parašus. Tarkime, subjektas *𝒜* yra vartotojas, subjektas *ℬ* – bankas.

Pateiksime RSA kriptosistemos pagrindu sukurtą Chaumo aklojo e. parašo schemą (Chaum, et al., 1991):

1. Vartotojas sukuria reikalavimą e. pinigui gauti ir jį taip užmaskuoja, kad niekas kitas negalėtų sužinoti reikalaujamo e. pinigo nominalo.
2. Bankas pasirašo užmaskuotą e. pinigą.

[7] David Chaum – JAV kriptografas, Tarptautinės kriptografinių tyrimų asociacijos (IACR) steigėjas.

3. Vartotojas demaskuoja banko pasirašytą e. pinigą, tačiau išsaugo autentišką banko parašą ant e. pinigo nominalo.



13.1 pav. Aklojo e. parašo sistema

Aklojo e. parašo išeities duomenys: banko RSA kriptosistemos raktų pora $PR_B=d$, $VR_B=(n, e)$ ir e. pinigo nominalas Π, pvz., $\Pi = 100$ (turima galvoje 100 Lt).

Aklojo e. parašo protokolo etapai:

1. Vartotojas atsitiktinai parenka skaičių r moduliu n ir užmaskuoja Π, apskaičiuodamas reikšmę Π':

$$\Pi' = \Pi \cdot r^e \bmod n. \tag{13.1}$$

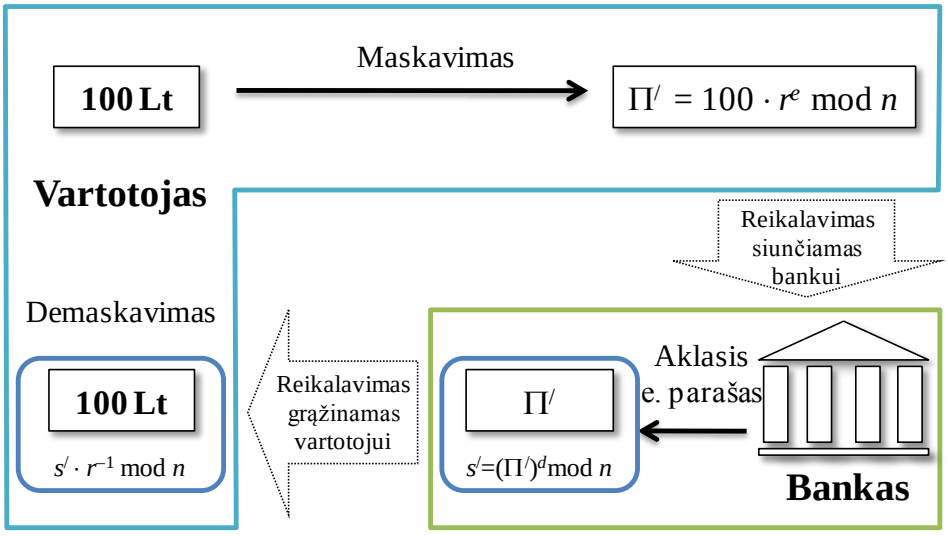
2. Bankas pasirašo Π', apskaičiuodamas atitinkamą e. parašą s'

$$s' = (\Pi')^d \bmod n, \tag{13.2}$$

ir grąžina s' vartotojui.

3. Vartotojas demaskuoja parašą s', apskaičiuodamas s:

$$s = s' \cdot r^{-1} \bmod n. \tag{13.3}$$



13.2 pav. Aklojo e. parašo sistema RSA kriptosistemos pagrindu

Po demaskavimo vartotojas apskaičiuoja autentišką banko parašą s e. pinigui Π. Tai įrodysime remdamiesi aritmetikos taisyklėmis, pateiktomis antrajame skyriuje.

$$s = s^{/} \cdot r^{-1} \bmod n = (\Pi^{/})^{d} \cdot r^{-1} \bmod n = (\Pi \cdot r^{e})^{d} \cdot r^{-1} \bmod n = \Pi^{d} \cdot r^{ed} \cdot r^{-1} \bmod n. \quad (13.4)$$

Priminsime, kad RSA kriptosistemoje n yra sudėtinis skaičius, lygus dviejų pirminių skaičių $p \cdot q$ sandaugai. Skaičius e parenkamas tarpusavyje pirminis su $\varphi(n) = (p - 1)\,(q - 1)$, kur φ yra Oilerio funkcija. PR_B apskaičiuojamas remiantis reikalavimu, kad $e \cdot d = 1 \bmod \varphi(n)$, t. y. $d = e^{-1} \bmod \varphi(n)$. Tada, remiantis liekanų aritmetika, $e \cdot d = 1 + k\,(p - 1)\,(q - 1)$ ir

$$r^{ed} \bmod n = r^{1 + k(p - 1)(q - 1)} \bmod n = r\, r^{k(p - 1)(q - 1)} \bmod n.$$

Sekant Oilerio teorema, $r^{\varphi(n)} = r^{(p - 1)(q - 1)} = 1$. Tuomet

$$r\, r^{k(p - 1)(q - 1)} \bmod n = r\,(r^{(p - 1)(q - 1)})^{k} \bmod n = r \cdot 1^{k} \bmod n = r \bmod n.$$

Įrašę gautą reikšmę į (13.4), gauname:

$$s = \Pi^{d} \cdot r \cdot r^{-1} \bmod n = \Pi^{d} \bmod n. \quad (13.5)$$

Dydis $\Pi^{d} \bmod n$ ir yra banko e. pinigui, kurio nominalas Π, e. parašas.

Taigi pakartosime dar kartą: nors bankas pasirašo ant nežinia kokio (vartotojo užmaskuoto) e. pinigo nominalo, po demaskavimo gaunamas autentiškas banko e. parašas ant vartotojo nustatyto e. pinigo nominalo.

Padarėme prielaidą, kad vartotojas yra labai sąžiningas ir pateiks bankui užmaskuotą e. pinigą $\Pi^{/}$, kurio tikroji vertė neviršys vartotojo pinigų sumos jo sąskaitoje. Tokiu atveju bankas neatseks, kam vartotojas išleis savo e. pinigą, ir bus išsaugotas vartotojo anonimiškumas.

O kas bus, jei bankas pasirašys ant nežinia kokio nominalo e. pinigo esant nesąžiningam vartotojui (o tokių tikrai gali pasitaikyti)? Tada bankas net negalės atsekti, koks vartotojas tai padarė. Be to, vartotojas gali dubliuoti tą patį e. pinigą keletą kartų, įsigydamas už jį atitinkamos vertės prekių. Galima drąsiai teigti, kad tokių bankų, kurie pasirašinėtų nežinia ant ko, tikrai nėra, o jeigu ir buvo, tai visi jau seniai bankrutavo.

Vadinasi, bankas turi sukurti tokį protokolą, kuris su pakankamu patikimumu niekam neleistų jo apgauti. Tam pasitarnaus aklojo e. parašo pagrindu sukurtas protokolas su papildomais saugikliais kovai su sukčiavimu.

13.2. E. PINIGAI

13.2.1. E. PINIGO DUBLIAVIMO PREVENCIJA PASITELKUS PRIJUNGTIES REŽIMĄ

E. pinigo dubliavimo prevenciją galima užtikrinti visas e. pinigo išleidimo operacijas atliekant prijungties režimu su banku, t. y. kiekvienai operacijai gaunant banko patvirtinimą. Tai yra realizuota kreditinių kortelių sistemoje. Tačiau mes turime teisę tikėtis tobulesnės e. pinigų sistemos nei kreditinės kortelės ir tuo pačiu išvengti operacijų vykdymo prijungties režimu su banku. Reikia siekti, kad e. pinigas būtų išleidžiamas atsijungus nuo banko.

13.2.2. E. PINIGO SUMOS FIKSAVIMO PROTOKOLAS

Šis protokolas priklauso (angl. *challenge-response*) stimulo ir reakcijos protokolų klasei ir apsaugo banką nuo vartotojo sukčiavimo, t. y. kad bankas neišduotų vartotojui didesnio nominalo e. pinigo, nei vartotojas skelbiasi norįs paimti. Protokolas vykdomas keturiais etapais, naudojant ryšio seansus tarp vartotojo ir banko:

1. Vartotojas paruošia, pvz., 50 reikalavimų 100 litų e. pinigui gauti. Vartotojas maskuoja visus 50 reikalavimų, parinkęs atitinkamus atsitiktinius skaičius r_1, ..., r_{50}, ir siunčia šiuos reikalavimus bankui.
2. Bankas atsitiktinai parenka vieną reikalavimą ir siunčia vartotojui pranešimą, kad jis demaskuotų likusius 49 reikalavimus.
3. Vartotojas demaskuoja gautus iš banko 49 reikalavimus ir išsiunčia juos bankui.
4. Bankas patikrina, ar e. pinigo nominalai yra vienodi visuose 49 reikalavimuose. Jeigu taip, prideda e. pinigo numerį prie likusio reikalavimo ir aklai pasirašo reikalavimą bei numerį.

Tikimybė, kad vykdant šį protokolą vartotojas apgaus banką, yra 1/50 = 0,02. Tai atitinka tikimybę vartotojui atspėti, kurį reikalavimą iš penkiasdešimties pasirinks bankas. Jeigu vartotojo ir banko sutartyje bus numatytos didelės baudos už tokį sukčiavimą, tada nė vienas vartotojas nerizikuos apgauti banką esant tokiai mažai sėkmės tikimybei.

Šis protokolas yra pagalbinė kitų protokolų, naudojamų e. pinigo cirkuliacijos saugumui užtikrinti, dalis. Prieš pateikdami kitus, įvedame naują sąvoką – *e. pinigo atsitiktinė identiškumo eilutė* (angl. *random identity string* – RIS). Šią eilutę sutrumpintai vadinsime RIS.

13.2.3. ATSITIKTINĖ IDENTIŠKUMO EILUTĖ (RIS)

Šią eilutę formuoja vartotojas, ruošdamas e. pinigo išdavimo reikalavimą. RIS pasižymi šiomis savybėmis:

13.1. Savybė. Ji turi būti skirtinga kiekviename e. pinige.

13.2. Savybė. Tik vartotojas gali suformuoti teisingą RIS ir tik jis yra už tai atsakingas.

13.3. Savybė. Dvi skirtingos RIS, priskiriamos tam pačiam e. pinigui (t. y. e. pinigui su tuo pačiu numeriu), turi leisti bankui nustatyti sukčiaujantį vartotoją.

Vėliau pamatysime, kad 13.3 savybė leidžia bankui nustatyti sukčių, t. y. tą, kuris nori du kartus pirkti už tą patį e. pinigą (dubliuoti).

Išvardytos savybės leidžia sudaryti protokolus, galinčius prižiūrėti e. pinigo cirkuliacijos procesą. Galima sakyti, kad tai yra protokolai, tenkinantys 13.1–13.3 savybes, kuriomis turi pasižymėti e. pinigas. Tačiau šis protokolas yra tik pirmasis bandymas priartėti prie „tikrųjų“ e. pinigų sistemos, nes jis, kaip ir žemiau pateikti e. pinigo protokolai, netenkina tam tikrų papildomų sąlygų. Tai aptarsime šio skyriaus pabaigoje.

Prieš aprašydami e. pinigo cirkuliacijos protokolą, norėtume priminti, kas yra santraukos funkcija, kuri buvo nagrinėjama ankstesniuose skyriuose. Laikysime, kad protokolo dalyviai turi tam tikrą santraukos funkciją, pvz., SHA-1, kuri apibrėžiama taip: $H : \{0,1\}^* \to \{0,1\}^l$, t. y. $H(x) = y$, kai $x \in \{0,1\}^*$ ir $y \in \{0,1\}^l$.

Remiantis apibrėžimu, santraukos funkcija yra vienkryptė (VKF), taigi turint y, algoritmiškai neįmanoma rasti tokio $x' \in \{0,1\}^*$, kad $H(x') = y$.

13.2.4. E. PINIGO PAĖMIMO PROTOKOLAS

Vartotojas ir bankas vykdo e. pinigo paėmimo protokolą tam, kad vartotojas galėtų pateikti bankui e. pinigo paėmimo reikalavimą, o bankas išduotų vartotojui e. pinigą ir užsitikrintų, jog vartotojas negalės sukčiauti nei šio protokolo vykdymo metu, nei vėliau. Protokolas vyksta penkiais etapais, kurių keturi pirmieji turi po vieną ryšio seansą tarp vartotojo ir banko.

1. Vartotojas parengia tam tikrą skaičių, pvz., penkiasdešimt, 100 litų e. pinigo paėmimo reikalavimų Π_1, ..., Π_{50}:

$$\Pi_i = (\pi_i \#123i, y_{i1}, y_{i1}', y_{i2}, y_{i2}', \ldots, y_{iK}, y_{iK}'); \qquad (13.6)$$

čia $\pi_i = 100$; #123*i* yra vartotojo banko sąskaitos numeris #123 su prijungtu indeksu *i*; $i = 1, ..., 50$; $K = 1, ..., 50$; $y_{ij} = H(x_{ij})$; $y_{ij}^{/} = H(x_{ij}^{/})$; x_{ij} ir $x_{ij}^{/}$ yra RIS eilutės, kurios, kaip jau buvo minėta, parenkamos atsitiktinai ir su visomis *i*, *j* reikšmėmis tenkina lygybę:

$$x_{ij} \oplus x_{ij}^{/} = \text{VV}, \qquad (13.7)$$

čia $\oplus$ yra sudėtis moduliu 2 (XOR) pabičiui; VV – vartotojo vardas, užkoduotas kokiu nors skaičiumi.

Vartotojas maskuoja visus Π_i ir apskaičiuoja $\Pi_i^{/}$, naudodamas aukščiau pateiktą maskavimo algoritmą.

Vartotojas siunčia visus $\Pi_i^{/}$, $i = 1, ..., 50$, bankui.

2. Bankas atsitiktinai parenka kokį nors $\Pi_i^{/}$, pvz., $\Pi_{33}^{/}$ ir padeda jį į šalį. Likusius 49 $\Pi_1^{/}$, ..., $\Pi_{32}^{/}$,$\Pi_{34}^{/}$, ..., $\Pi_{50}^{/}$ bankas grąžina vartotojui, reikalaudamas juos demaskuoti.
3. Vartotojas demaskuoja $\Pi_1^{/}$, ..., $\Pi_{32}^{/}$, $\Pi_{34}^{/}$, ..., $\Pi_{50}^{/}$ ir atskleidžia bankui atitinkamas RIS $\{x_{ij}, x_{ij}^{/}\}$, išskyrus tą, kuri atitinka Π'_{33}. Vartotojas siunčia $\Pi_1^{/}$, ..., $\Pi_{32}^{/}$,$\Pi_{34}^{/}$, ..., $\Pi_{50}^{/}$ ir juos atitinkančias RIS $\{x_{ij}, x_{ij}^{/}\}$ bankui.
4. Bankas patikrina, ar visi π_1, ..., π_{32}, π_{34}, ..., π_{50} atitinka 100 Lt e. pinigą. Jis taip pat patikrina, ar $y_{ij} = H(x_{ij})$, $y_{ij}^{/} = H(x_{ij}^{/})$ ir lygybę $x_{ij} \oplus x_{ij}^{/} = \text{VV}$. Jeigu viskas tvarkoje, bankas aklai pasirašo $\Pi_{33}^{/}$. Toliau trumpumo dėlei atmesime indeksą 33 ir priskirsime $\Pi^{/} = \Pi_{33}^{/}$. Tada $\Pi^{/}$ atitinkančią RIS žymėsime x_j, $x_j^{/}$, $j = 1, ..., K = 50$. Banko parašą ant $\Pi^{/} = \Pi_{33}^{/}$ žymėsime $s^{/}$. Bankas siunčia $s^{/}$ vartotojui.
5. Vartotojas, žinodamas savo slaptą maskavimo konstantą *r*, demaskuoja $s^{/}$ ir apskaičiuoja Banko e. parašą e. pinigo reikalavimui $\Pi = \Pi_{33}$. Taip gaunamas e. pinigas, kuris nusakomas skaičių pora (Π, s).

13.2.5. E. PINIGO MOKĖJIMO PROTOKOLAS

Šis protokolas vykdomas tarp vartotojo bei pardavėjo ir aprašomas penkiais etapais. Pirmuosiuose trijuose etapuose pasitelkiami ryšio seansai tarp vartotojo ir pardavėjo.

1. Vartotojas siunčia pardavėjui prekės užsakymą kartu su e. pinigu (Π, s).
2. Pardavėjas patikrina banko parašą *s* e. pinigo komponentei Π. Jeigu e. parašas tikras, siunčia vartotojui atsitiktinę bitų eilutę $b_1, ..., b_K$, kurios ilgis $K = 50$.
3. Vartotojas gauna $b_1, ..., b_K$ ir, jeigu $b_j = 0$, atskleidžia pardavėjui x_j, kitu atveju atskleidžia $x_j^{/}$.
4. Atitinkamus x_j, $x_j^{/}$ vartotojas siunčia pardavėjui, kuriuos žymėsime RIS_P.
5. Pardavėjas tikrina, ar $y_j = H(x_j)$ arba $y_j^{/} = H(x_j^{/})$. Jeigu lygybės galioja, pardavėjas priima e. pinigą ir išsiunčia prekes vartotojui.

13.2.6. E. PINIGO DEPOZITAVIMO PROTOKOLAS

Šis protokolas yra sudarytas iš dviejų etapų ir vykdomas tarp pardavėjo bei banko. Numatytas tik vienas ryšio seansas pirmajame etape.

1. Pardavėjas, gavęs iš vartotojo e. pinigą (Π, s), siunčia jį kartu su atskleista RIS eilutės dalimi RIS_P.
2. Bankas patikrina savo parašą *s* e. pinigui Π ir nustato, ar tas pats e. pinigas nebuvo pateiktas bankui su tuo pačiu numeriu. Jeigu nebuvo, tada bankas depozituoja e. pinigą į pardavėjo sąskaitą, padidindamas ją 100 litų suma.

13.2.7. E. PINIGO CIRKULIACIJOS PROTOKOLŲ SAUGUMO ANALIZĖ

E. pinigo paėmimo, mokėjimo ir depozitavimo protokolai sudaro e. pinigo cirkuliacijos ciklą. Visas ciklas reikalauja 4 + 3 + 1 = 8 ryšio seansų tarp protokolo dalyvių.

Priminsime, kad vykdant e. pinigo paėmimo protokolą vartotojo tikimybė apgauti banką paimant didesnį e. pinigą nei deklaruojama, yra $1/K = 1/50 = 0{,}02$. Kuo didesnis K, tuo ši tikimybė mažesnė. Galima daryti prielaidą, jog šis sukčiavimo būdas vartotojui neperspektyvus, nes jis labai rizikuoja gauti dideles baudas jau pačioje sukčiavimo pradžioje. Tikimybė, kad vartotojas paims e. pinigą ne savo vardu, taip pat yra $1/K$. Tačiau tam numatytos ir kitos apsaugos priemonės banko duomenų bazėje.

Todėl galima teigti, kad e. pinigo paėmimo protokolas yra gana saugus. Jo saugumo lygis nusakomas parametru K.

> ***13.4. Klausimas.*** Kokį K reikėtų parinkti, kad vartotojo sukčiavimo tikimybė būtų 0,005?

E. pinigo mokėjimo protokolo saugumas taip pat priklauso nuo K, ir ši priklausomybė yra gerokai stipresnė. Jeigu pardavėjas atsitiktinai sugeneruoja K ilgio bitų seką $b_1, ..., b_K$, tai tokių sekų skaičius yra 2^K. Vadinasi, jeigu vartotojas ir vėl nori sukčiauti, tikimybė, kad jis tam pačiam e. pinigui pateiks vienodas RIS_P, yra $1/2^K = 2^{-K} = 2^{-50}$. Tokia galimybė yra mažai tikėtina ir atmestina. Todėl vartotojas, bandydamas du kartus mokėti tuo pačiu e. pinigu, pateiks pardavėjui skirtingas RIS_P, t. y. RIS_{P1} ir RIS_{P2}, esant tikimybei, artimai vienetui. Taip apibrėžta RIS tenkina jai keliamus reikalavimus, pateiktus 13.2.3 skyrelyje. Tik vartotojas gali sukurti teisingą RIS, nes tai yra susiję su jo vardu VV, o santraukos funkcija yra vienkryptė. Kita vertus, pardavėjas taip pat negali apsimesti vartotoju ir klastoti jo RIS_P.

Tarkime, kad vartotojas yra rizikuojantis sukčius, bandantis du kartus pirkti už tą patį e. pinigą iš lengvatikio pardavėjo. Kaip jau minėjome, jis pateiks pardavėjui dvi skirtingas $RIS_{P1} \neq RIS_{P2}$. Tada bankas, gavęs iš pardavėjo (Π, s) ir RIS_{P2}, savo duomenų bazėje randa tą patį e. pinigą (Π, s) su RIS_{P1}. Palyginęs RIS_{P1} ir RIS_{P2} bankas mato, kad jos skirtingos. Kadangi teisingą RIS gali suformuoti tik vartotojas, jis iš karto nustato, kad tai – vartotojo sukčiavimas. Bankui lieka nustatyti vartotojo vardą VV ir imtis sankcijų. Tam jis sudeda moduliu 2 pabičiui $RIS_{P1} \oplus RIS_{P2}$ ir ieško skaičių sumos $x_{ij} \oplus x_{ij}^{/}$, kuri lygi VV. Peržiūrėjęs visas sumas, jis tikrai ras bent vieną, lygią VV. Tokiu būdu sukčius vartotojas bus nustatytas.

> ***13.5. Klausimas.*** Kodėl ir kokia tikimybė, kad bent viena RIS_{P1} ir RIS_{P2} komponenčių suma bus lygi VV?

Klaidinga manyti, kad didžiausi sukčiai yra vartotojai – jie veikiau yra sąžiningiausi ir mažiausiai apginti rinkos dalyviai. Panagrinėkime, kokių galimybių sukčiauti turi pardavėjai, juolab kad ir pasitikėjimas jais gana mažas.

Tarkime, pardavėjas, kuriam kilo mintis apgauti banką, turi e. pinigą (Π, s) ir vartotojo RIS. Norėdamas dukart padidinti savo banko sąskaitą, bet neturėdamas pakankamo išsilavinimo, jis du kartus, pvz., kas savaitę, pateikia bankui tuos pačius (Π, s) ir RIS. Bankas iš karto nustato sukčių.

Tačiau labiau išsilavinęs nesąžiningas pardavėjas gali bandyti apgauti banką, visą kaltę suversdamas vartotojui. Tam jis turi pagaminti RIS_{P1} ir RIS_{P2} tam pačiam e. pinigui (Π, s). Tačiau kaip jam tai jeigu žinodamas $y_1, ..., y_K$ jis negali apskaičiuoti x_j, kuris tenkina lygybę $y_i = H(x_j)$. Negana to, net ir žinodamas y'_j, jis turi rasti tokį $x_j^{/}$, kad $y_j^{/} = H(x_j^{/})$, kai x_j ir $x_j^{/}$ turi tenkinti lygybę $x_{ij} \oplus x_{ij}^{/} = VV$. Tai neįmanoma jau vien todėl, kad santraukos funkcija yra vienkryptė. Todėl pardavėjas pigiai pasipelnyti, užkraudamas atsakomybę vartotojui, niekaip negalės.

Čia paminėtos pagrindinės kriptografinės atakos prieš e. pinigo cirkuliacijos protokolus, todėl galima teigti, kad pasirinkus pakankamai didelį K šis protokolas užtikrina saugią e. pinigo cirkuliaciją.

13.2.8. KITOS E. PINIGO SAVYBĖS

Sukurtas e. pinigas su jį atitinkančiais cirkuliacijos protokolais turi vieną gana didelį trūkumą: vienas e. pinigas gali būti panaudotas vienai prekei nusipirkti. Dėl to kyla nemažai problemų ir klausimų:

1. Ką daryti, jeigu e. pinigo vertė didesnė už prekės vertę?

2. Ar kiekvienai prekei reikia imti iš banko skirtingus e. pinigus, kurių nominalai atitinka prekių kainoraštį?
3. Nors daugelio e. pinigų išleidimas tenkins vartotojo anonimiškumo reikalavimą, tačiau reikės daug ryšio seansų su banku, norint pasiimti iš jo reikiamą kiekį e. pinigų.

Vienas šių problemų sprendimo būdų – pasinaudoti trečiąja patikima šalimi (angl. *third trust party* – TTP), tačiau tai nesupaprastina cirkuliacijos protokolo. Mes nenagrinėsime šio klausimo plačiau, nes šis sprendimas dėl daugelio priežasčių yra neperspektyvus. Kriptografijos mokslininkai pastaruoju metu daug dėmesio skiria tam, kad būtų sukurti cirkuliacijos protokolai, užtikrinantys šias e. pinigų savybes:

13.6. Savybė. E. pinigai turi būti dalomi, t. y. vienas ir tas pats banko išduotas e. pinigas turi galimybę būti išleistas dalimis, įsigyjant prekių, kurių suma neviršija e. pinigo nominalo.

13.7. Savybė. E. pinigo dalijimas neturi būti susietas su trečiosios patikimos šalies dalyvavimu.

13.8. Savybė. Keleto e. pinigų paėmimas ir išleidimas turi būti efektyvesnis nei atskiri paėmimo ir išleidimo protokolai, vykdomi atskirai.

Visai neseniai EUROCRYPT 2007 konferencijoje buvo pateiktas e. pinigų cirkuliacijos protokolas, tenkinantis 13.6–13.8 ir 13.1–13.3 savybes. Tai pirmasis pasaulyje tokio tipo protokolas, leidžiantis paimti e. pinigą, kurio nominalas yra 2^L; čia L – natūralusis skaičius. Be abejo, L yra apribotas, nes, pradedant tam tikru skaičiumi, tiek pinigų neturės niekas pasaulyje. Pvz., Lietuvos multimilijonierius gali paimti e. pinigą, lygų 2^{32}. Tai sudarys 4294967296 Lt. E. pinigas, kurio nominalas yra 2^L, gali būti išleidžiamas 2^l dalimis, kai $0 \le l \le L$. Kitaip sakant, smulkiausia e. pinigo dalis yra $2^0 = 1$ Lt, kiti nominalai – $2^1 = 2$ Lt, $2^2 = 4$ Lt ir t. t. Kaip matome, šie piniginiai ženklai neatitinka įprastų pinigų nominalų (1 Lt, 2 Lt, 5 Lt, 10 Lt ir t. t.), tačiau perėjimas prie e. pinigų informacinėje visuomenėje – tik įpratimo reikalas.

Naujų e. pinigų autoriai – S. Carardas ir A. Gouget. Reikia tikėtis, kad šių ir kitų mokslininkų dėka maždaug 2015 m. turėsime visaverčius e. pinigus, pakeisiančius mums įprastus metalinius ir popierinius e. pinigus bei elektronines mokėjimo korteles.

13.3. E. BALSAVIMAS

Kita itin perspektyvi aklojo e. parašo pritaikymo sritis – e. balsavimo sistemos. Pagrindinis e. balsavimo sistemų uždavinys – rinkėjų balsų privatumo ir konfidencialumo išsaugojimas, tuo pačiu užtikrinant jų autentiškumą ir patikimą galutinio balsavimo rezultato apskaičiavimą.

E. balsavimo sistemos turėtų tenkinti šias savybes:

1. Balsuoti gali tik įgaliotieji rinkėjai.
2. Niekas negali balsuoti daugiau kaip du kartus.
3. Turi būti išsaugomas balsų slaptumas.
4. Niekas negali suklastoti kito rinkėjo balso.
5. Rezultatas turi būti apskaičiuojamas teisingai ir vienareikšmiškai.
6. Bet kas gali patikrinti rezultatą.
7. Sistema turi būti atspari trikdžiams, t. y. turi veikti esant ir nesąžiningoms šalims.
8. Rinkėjas, net ir norėdamas, niekam negali formaliai įrodyti, kaip balsavo.

Teoriškai ideali e. balsavimo sistema turi būti pagrįsta kriptografiškai saugiu skaičiavimu, padalytu tarp visų e. balsavimo protokolo dalyvių. Tačiau tokios sistemos yra nepraktiškos, nes reikalauja didelių skaičiavimo sąnaudų ir visų rinkėjų bendradarbiavimo. Todėl praktikoje dažniau taikomas įgaliotos balsavimo institucijos (rinkimų komisija) modelis, taip pat specialiai balsuoti skirti protokolai, kuriais rinkėjas bendrauja su institucijomis naudodamasis įprastais ryšio kanalais (internetu) bei kompiuteriu. Šių balsavimo protokolų tikslas – užtikrinti, kad rinkimų komisijos negalėtų nustatyti, kas ir už ką balsavo, bet galėtų patikrinti balsų tinkamumą ir teisingai apskaičiuoti rinkimų rezultatus.

Sprendžiant saugumo klausimą, neapsiribojama vien vietos, kurioje kaupiami balsai, saugojimu. E. balsavimo sistemos taip pat apima šias saugumo sritis:

- Autentifikavimas. Tai tikrinimas, ar pareiškimas dėl tapatybės arba teisės balsuoti yra teisingas. Šio proceso metu siekiama atsakyti į klausimus „kas tu esi?“ ir „ar tu turi teisę balsuoti?“. E. balsavimo sistemose svarbūs du autentifikavimo aspektai:
  - tapatybės tikrinimo;
  - teisės balsuoti tikrinimo.
- Privatumas/konfidencialumas. Tai užtikrinimas, kad informacija apie balsuotojus ir jų balsus nebus atskleista, išskyrus tuos atvejus, kai tai bus būtina skaičiuojant balsus ir tikrinant, ar teisingi rezultatai. Kitais atvejais neturėtų būti įmanoma sužinoti, kaip balsavo konkretus rinkėjas. Be to, iki rinkimų pabaigos neturėtų būti įmanoma gauti balsų pasiskirstymo rezultatų.
- Vientisumas. Tai užtikrinimas, kad rinkimų balsai yra teisingi ir nesuklastoti. Sudarius kandidatų ir rinkėjų sąrašus, turi būti užtikrinta, kad kiekvienam balsuotojui bus pateikta teisinga balsavimo informacija, t. y. tinkamas balsavimo biuletenis pagal jo rinkimų apygardą. Be to, rinkimų komisijai gavus balsą, svarbu, kad jis būtų tinkamai saugomas tol, kol prasidės balsų skaičiavimas ir tikrinimas.
- Neišsiginamumas. Tai garantija, kad balsuotojas negalės paneigti savo atliktų veiksmų. E. balsavimo sistemose tapatybės nustatymas reikalauja užtikrinti, kad žmonės, prisistatantys kaip teisėti dalyviai (rinkėjai, rinkimų darbuotojai ir kt.), iš tikrųjų yra tie, kuo prisistato. Neišsiginamumas reikalauja užtikrinti, kad identifikuotas dalyvis negalėtų sėkmingai išsiginti savo veiksmų, priskirtų jam identifikavimo mechanizmu, ir teigti, jog iš tikrųjų juos atliko kažkas kitas. Šie du reikalavimai yra susiję tuo, kad sistema, turinti nepriekaištingą tapatybės nustatymo mechanizmą ir neginčytiną visų veiksmų įrodymą, neleidžia išsiginti savo veiksmų.

Skiriami du pagrindiniai metodai elektroninių balsų privatumui išsaugoti:

1. Rinkėjo tapatybės slaptumas.
2. Balso slaptumas.

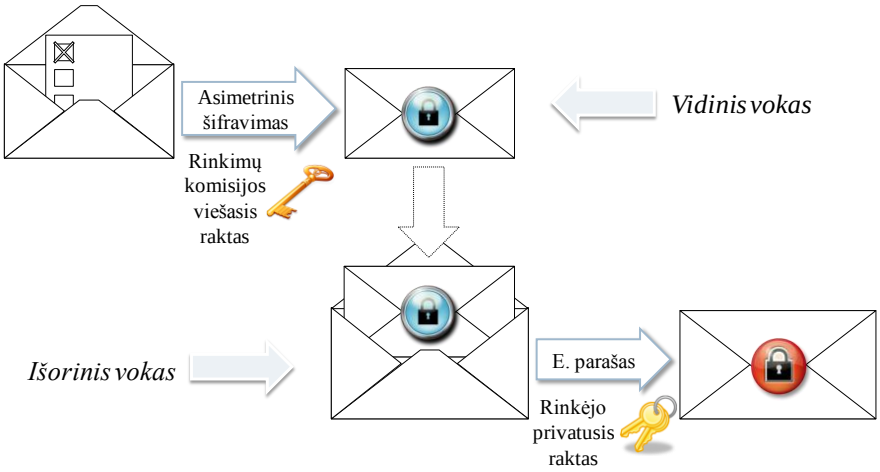
Pirmuoju atveju turi būti naudojamas anoniminis kanalas, kuriuo rinkimų komisija gauna biuletenius, bet negali nustatyti, kas juos siuntė. Šiuo atveju reikia nustatyti gautų anoniminių biuletenių galiojimą (teisėtumą), aptikti visus daugkartinius to paties asmens balsus ir patikrinti, ar visi balsavę rinkėjai turėjo tam teisę. Būtent šiam metodui realizuoti ir naudojamas aklasis e. parašas.

Balso slaptumą užtikrinančioje sistemoje rinkimų komisija gali identifikuoti rinkėją, tačiau jo balsas yra užšifruotas ir jai nežinomas. Idealiu atveju siekiama, kad sistema leistų suskaičiuoti kiekvieną ir visus balsus jų neiššifruojant – iššifruoti tik gautą galutinį rezultatą. Šifrai, pasižymintys tokia savybe, t. y. leidžiantys sudėti užšifruotus duomenis, yra vadinami homomorfiniais. Balsų slaptumui užtikrinti galima taikyti ir sumaišymo procedūrą, kai kiekvienas balsas iššifruojamas atskirai, tačiau tik tada, kai balsai sumaišomi ir neįmanoma jų susieti su bet kokiu rinkėjų parašu. Taigi šiose sistemose reikia saugiai atskirti rinkėjo identifikacinius duomenis nuo jo užšifruoto balso.

Balso slaptumą užtikrinančioje sistemoje paprastai taikomas elektroninio voko principas: e. rinkėjas užpildo vidinio e. voko turinį, t. y. biuletenį. Voko užklijavimą atitinka biuletenio užšifravimas rinkimų komisijos viešuoju raktu. Tada šį voką jis „įdeda“ į išorinį e. voką ir jį pasirašo savo e. parašu (13.3 pav.).

E. voko formavimas:

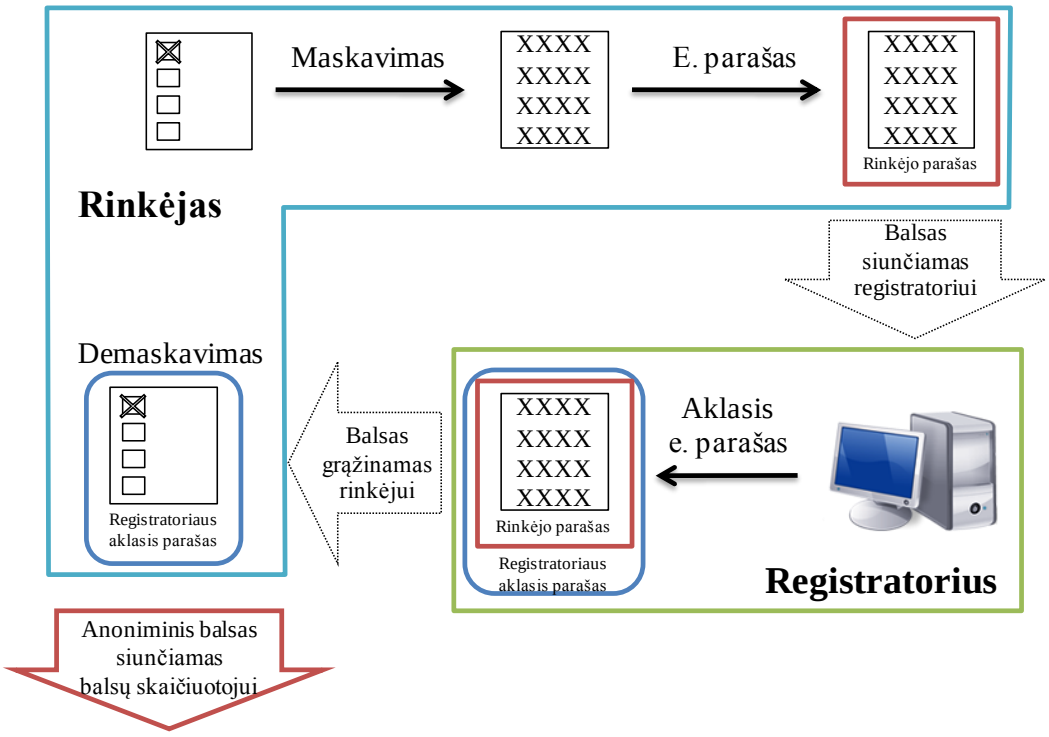
1. Sudaroma rinkmena, kurioje nurodomas balsavimo rezultatas ir prie jo prijungtas atsitiktinis skaičius.
2. Rinkmena užšifruojama viešuoju rinkimų komisijos raktu.
3. Rinkmena pasirašoma e. parašu su privačiuoju rinkėjo raktu.



13.3 pav. Elektroninio voko metodas

Atsitiktinis skaičius, formuojant e. voką, prijungiamas rinkėjo balsui užmaskuoti, t. y. kad skirtųsi vienodi skirtingų rinkėjų balsai. Jei du rinkėjai balsuotų už tą patį kandidatą, be atsitiktinio skaičiaus užšifruoti jų balsai su tuo pačiu rinkimų komisijos viešuoju raktu sutaptų.

Aklieji e. parašai naudojami e. balsavimo sistemose registruojant rinkėjus. Registratorius, gavęs iš balsuotojo užpildytą ir užmaskuotą rinkimų biuletenį, patikrina, ar rinkėjas turi teisę balsuoti, ir pasirašo rinkėjo biuletenį, nežinodamas jo turinio. Tada demaskuotas pasirašytas biuletenis paverčiamas anoniminiu, atskiriant balsuotojo tapatybę, ir siunčiamas balsų skaičiuotojui, kuris patikrina registratoriaus parašą ir suskaičiuoja balsus.



13.4 pav. Rinkimų protokolas pasitelkus akląjį e. parašą

Bendruoju atveju elektroninių rinkimų protokolas pasitelkus akląjį e. parašą vykdomas tokiais etapais:

1. Rinkėjas užmaskuoja savo balsą rinkimų biuletenyje.
2. Biuletenis elektroniniu būdu pasirašomas.
3. Biuletenis perduodamas registratoriui.

4. Registratorius patikrina rinkėjo autentiškumą.
5. Registratorius pasirašo biuletenį akluoju e. parašu.
6. Pasirašytas biuletenis grąžinamas rinkėjui.
7. Rinkėjas demaskuoja savo biuletenį, išsaugodamas savo balso registratoriaus parašą, ir anonimiškai siunčia jį balsų skaičiuotojui.

Nepriklausomai nuo to, ar naudojamas tapatybės slėpimas, ar balso slėpimas, e. balsavimo sistemos turi bendras saugumo problemas. Dažniausiai pasiūlomi rinkimų variantai, kai rinkėjo ir rinkimų pareigūnų bendravimas vyksta aplinkoje, kurią nevisiškai kontroliuoja ir stebi rinkimų pareigūnai ir (ar) stebėtojai (pvz., internetas, privatusis tinklas, telefonai, kabelinių televizijų tinklai ir kt.). Šiose „nuotolinėse“ ir „neprižiūrimose“ aplinkose yra keletas konkrečių saugumo problemų ir klausimų:

- Iš kur aš žinau, kad man pateikta informacija apie kandidatus yra teisinga?
- Iš kur aš žinau, kad mano balsas teisingai įrašytas?
- Iš kur aš žinau, kad nėra „žmogaus viduryje“, kuris pakeis mano atiduotą balsą?
- Iš kur aš žinau, kad esu prijungtas prie tikro e. balsavimo serverio, kuris įrašys mano balsą, o ne prie apsimestinio, kuris iš tikrųjų išmes mano balsą?
- Iš kur aš žinau, kad kuris nors sistemos komponentas neturi kenkėjiškos įrangos, gebančios pakeisti man pateikiamus balsavimo pasirinkimus ar pakeisti mano balsą?

Konkrečių rinkimų tipas lemia, kokios įtakos turi aukščiau išvardytos problemos, taip pat ar jos kelia realią grėsmę balsavimo procesui bei rinkimų rezultatams.

13.3.1. FOO BALSAVIMO PROTOKOLAS

Viena pirmųjų balsavimo sistemų, pasitelkus akląjį e. parašą, buvo pasiūlyta 1993 m. (Fujioka, et al., 1993). Aklieji e. parašai leidžia išspręsti vadinamąją tikrintojo ir skaičiuotojo sąmokslo problemą, kai bendrininkaujant registratoriui (balsavimo teisės tikrintojui) ir balsų skaičiuotojui gali būti pažeistas rinkėjų balsų konfidencialumas. Taip pat užkertamas kelias registratoriui pasinaudoti rinkimuose nedalyvaujančių rinkėjų balsais, klastojant rinkimų rezultatus. Trumpai šią sistemą vadinsime FOO balsavimo protokolu. Įtraukime šiuos kintamuosius:

- b – biuletenis;
- (PR_R, VR_R) – rinkėjo privatusis (užšifravimo) ir viešasis (iššifravimo) raktai;
- r – atsitiktinė balso maskavimo reikšmė;
- (PR_T, VR_T) – registratoriaus privatusis ir viešasis raktai.

FOO balsavimo protokolo etapai:

1. Rinkėjas užpildo biuletenį b, užšifruoja jį privačiuoju raktu $b^{PR_R} = B$ ir jį užmaskuoja ($B * r^{VR_T}$). Užmaskuotą biuletenį pasirašo savo e. parašu su kitu privačiuoju raktu, kurį atitinkantis viešasis raktas yra visiems žinomas ir susietas su rinkėjo tapatybe. Užmaskuotas biuletenis su e. parašu σ, t. y. ($B * r^{VR_T}$, σ), siunčiamas registratoriui.
2. Registratorius patikrina, ar gautas parašas σ priklauso registruotam rinkėjui, kuris dar nebalsavo. Jei balsas galioja, jis pasirašo biuletenį $(B * r^{VR_T})^{PR_T}$ ir grąžina rinkėjui.
3. Rinkėjas demaskuoja pasirašytą biuletenį $(B * r^{VR_T})^{PR_T} / r = B^{PR_T}$, ir gaunamas užšifruotas biuletenis su registratoriaus parašu. Pasirašytas ir užšifruotas biuletenis be rinkėjo identifikacinės informacijos, t. y. be parašo σ, anonimiškai siunčiamas balsų skaičiuotojui.
4. Skaičiuotojas patikrina užšifruoto balso akląjį e. parašą. Jei parašas yra galiojantis, jis pridedamas prie balsų sąrašo, kuris paviešinamas pasibaigus balsavimui.

5. Paskelbus balsų sąrašą, rinkėjai patikrina, ar jų balsai yra sąraše, ir anonimiškai siunčia balsų skaičiuotojui viešuosius raktus VR_R, reikalingus biuleteniams iššifruoti.
6. Balsų skaičiuotojas iššifruoja biuletenius ir prideda balsus prie bendrų rinkimų rezultatų.
7. Pasibaigus rinkimams, balsų skaičiuotojas paviešina visus iššifravimo raktus kartu su biuleteniais (B, b, VR_R), kad kiekvienas norintis galėtų individualiai patikrinti rinkimų rezultatus.

Taikant šį protokolą, naudojama biuletenių skelbimų lenta, kurioje paskelbiami užšifruoti biuleteniai ir jų raktai. Taip pat vartojamas anoniminis kanalas, kuriuo rinkėjai balsų skaičiuotojui perduoda pasirašytus akluoju e. parašu biuletenius ir atitinkamus jų iššifravimo raktus. Šie biuleteniai yra perduodami demaskuoti ir be identifikacinių duomenų.

FOO balsavimo protokolas užtikrina:

- Tikslumą: pažeisti, anuliuoti ir neteisėti balsai gali būti aptikti ir pataisyti. Tačiau registratorius gali pasinaudoti nebalsavusių rinkėjų balsais ir balsuoti vietoj jų. Šį faktą gali nustatyti auditas, bet balsų pataisyti nepavyks.
- Privatumą: nors užtikrinamas rinkėjų anonimiškumas, tačiau jie neapsaugoti nuo balsavimo prieš savo valią (papirkinėjimas, grasinimai ir pan.).
- Patikrą: rinkėjai gali patikrinti, ar balsai buvo teisingai suskaičiuoti, ir anonimiškai protestuoti, pateikdami užšifruotą biuletenį ir atitinkamą raktą tam skirtoje biuletenių skelbimų lentoje.
- Patogumą: šis protokolas buvo realizuotas Vašingtono universitete 1996 m. diegiant „Sensus“ sistemą. Testavimas parodė, kad rinkėjai gali greitai balsuoti be ypatingos techninės įrangos ir specialaus pasirengimo.

Vėliau buvo pateiktas patobulintas FOO balsavimo protokolas, kai rinkėjams nereikia dalyvauti balsų skaičiavimo etape (Ohkubo, et al., 1999).

Pateiktame protokolo aprašyme nesukonkretinti nei e. parašo, nei šifravimo algoritmai: galima naudoti bet kokią asimetrinę kriptografinę sistemą, leidžiančią maskuoti. Jei naudotume RSA kriptosistemą, * operaciją atitiktų įprasta skaičių daugyba moduliu n (viešas parametras), o raktai nurodytų atitinkamus laipsnio rodiklius.

14. ĮMONĖS KRIPTOSISTEMA IR JOS SAUGUMO ANALIZĖ

Kiekvienos įmonės tikslas – pelningos verslo operacijos. Tačiau dėl globalizacijos ir didesnės konkurencijos, taip pat dėl veiklos susiejimo ir sujungimo įmonių darbo aplinka tampa vis sudėtingesnė.

Gerinant našumą, konkurencingumą ir formuojant kompanijos tikslus, vis labiau akcentuojamas saugumas. Kompanija privalo remtis konkrečia veiksmų programa saugumui užtikrinti, ypač kai saugumo įsipareigojimai yra įtraukti į įmonės veiklos koncepciją arba yra socialinės atsakomybės dalis.

Pažeidus saugumą, kyla daugybė pavojų: neigiamas poveikis įmonės vardui, varginančios išlaidos informuojant nukentėjusius klientus, galimas intelektinės nuosavybės atskleidimas ir nesugebėjimas laikytis Vyriausybės nutarimų.

Dėl šių priežasčių gali kilti daug finansinių problemų. JAV atlikti tyrimai parodė, kad vidutiniai vieno saugumo pažeidimo nuostoliai vienai kompanijai, įskaitant pranešimo sąnaudas, klientų praradimą ir problemas pritraukiant naujus klientus, siekia 4,8 mln. JAV dol. Nuolat didėjantis atakų skaičius ir kylančios naujos grėsmės skatina įmonės imtis atsakomųjų veiksmų ir diegti arba stiprinti savo kriptosistemą.

Įmonėms nuolat metami iššūkiai, todėl jos priverstos užtikrinti duomenų saugumą. Kiekviena produktyvumo pažanga verčia plėsti saugumo strategijas. Be to, reikia tai daryti greitai, mat pavėlavus pritaikyti technologijas rizikuojama prarasti konkurencinį pranašumą.

Įmonėse svarbų vaidmenį vaidina personalas, todėl daug dėmesio reikia skirti saugumo nuostatoms, tobulinant darbuotojų supratimą, susipažinimą, išsilavinimą ir praktinį pasirengimą.

14.1. ĮMONĖS KRIPTOSISTEMA

Įmonės saugumo veikla visada turi remtis verslo tikslais. Konkurencingumas ir saugumas nėra priešingybės. Saugumas – pagrindinė konkurencingumo sąlyga ir svarbi kokybiškos verslo veiklos dalis. Įmonės saugumas pagrįstas bendru kompanijos saugumo klausimų sprendimu. Praktinė įmonės apsaugos veikla susideda iš prevencinių veiksmų, kuriais apsaugomas kompanijos personalas, turtas, informacija, aplinka ir reputacija dėl kenkėjiškų nusikalstamų veiksmų ir patirtų nuostolių. Todėl įmonės saugumas yra viena kompanijos rizikos valdymo proceso dalių.

Įmonės kriptosistema apima šias sritis:

- Fizinę apsaugą: įrenginių, raktų ir slaptažodžių apsauga, nenaudojamų dokumentų naikinimas.
- Vartotojo autentifikaciją: vartotojų tapatybės nustatymas.
- Prieigos kontrolę: tinklo išteklių pasiekiamumo apibūdinimas.
- Duomenų šifravimą: naudojamų kriptografinių algoritmų apibūdinimas.

Įmonė, siekdama vientisumo ir prieigos kontrolės, gali naudoti bet kuriuos iš šių elementų. Geriausias pasirinkimas priklauso nuo rizikos laipsnio, diegimo ir priežiūros sąnaudų bei nuostolių dydžio, pažeidus saugumą ar praradus duomenis. įmonė

Įmonės kriptosistema turi:

- Suteikti prieigą prie informacijos įgaliotiems subjektams.
- Naudoti saugią vartotojų autentifikavimo sistemą.
- Atmesti bet kokius neteisėtus ar kenkėjiškus bandymus pasiekti informaciją.
- Apsaugoti informaciją nuo jos gyvavimo pradžios iki pabaigos.

Savanoriškos ir tikslingos investicijos į įmonės saugumą suteikia kompanijos veiklai pridėtinę vertę: gerinamas komercinės veiklos ir darbo aplinkos saugumas, nepertraukiamai tęsiama produkcijos ir paslaugų

gamyba, kontroliuojama kenkėjiškos veiklos rizika. Be to, visi šie dalykai turi teigiamą poveikį kompanijos išteklių valdymui, išoriniam įvaizdžiui ir konkurencingumui.

Kompanijos savininkai, administracija, klientai ir kitos svarbios suinteresuotos grupės siekia savo tikslų ir yra numačiusios savus reikalavimus, susijusius su verslo operacijomis ir įmonės saugumu. Sisteminėje veiklos aplinkoje įmonės saugumo veikla turi vis labiau tenkinti sutartimis pagrįstus ryšius ir kokybės reikalavimus, pavyzdžiui, sutarčių su subrangovais tinkle. Kompanijoje dirbantis personalas taip pat vykdo jiems pavestą saugumo veiklą. Taigi įmonės saugumas yra viso personalo problema. Produktyvus įmonės saugumo principų ir tinkamo personalo mąstymo vystymas bus sėkmingas tik rengiant tikslingus ir sistemingus su saugumu susijusius mokymus.

Įmonės saugumo užtikrinimas – nenutrūkstamas procesas. Tačiau nuolat plėtojamos technologijos lemia ir naujas vidines bei išorines grėsmes. Skiriamos trys įmonės saugumo grėsmės:

- Atsisakymas aptarnauti (angl. *denial of service*). Kenkėjas sutrikdo įprastą sistemos darbą, ir reikiama informacija nepasiekiama.
- Informacijos vagystė. Kenkėjas gauna informaciją, kuri yra įmonės nuosavybė. Informacija gali būti pasisavinta apsimetus legaliu sistemos vartotoju arba tiesiogiai įsilaužus.
- Duomenų iškraipymas (angl. *corruption of data*). Kenkėjas sunaikina arba iškraipo duomenis, saugomus diske ar siunčiamus informaciniais kanalais.

Informacijos prieinamumo, valdymo ir vientisumo grėsmę gali kelti bet kuris iš šių taškų:

- Žmonės, besinaudojantys sistema, t. y. vartotojai (slaptažodžių atskleidimas, raktų, kortelių pametimas ir pan.).
- Vidiniai tinklo ryšiai.
- Tarptinkliniai atskirų padalinių ryšiai intranetu ar internetu.
- Tretieji ryšio paslaugų teikėjai.
- Programinio lygmens apsimetėliai, užpuolikai ir kiti kenkėjai.

Negalima diegti atitinkamos ar net neįveikiamos saugumo sistemos viename iš šių taškų, paliekant neapsaugotą vieną ar daugiau kitų. Profesionalus vagis ar kerštingas darbuotojas išmėgins visus galimus įsilaužimo būdus, ypač jei informacija yra vertinga, o prieiga prie jos – sąlygiškai nesudėtinga. Kaip ir paprasčiausioje grandinėje, įmonės kriptosistemos saugumas priklauso nuo silpniausios grandies saugumo.

14.2. FIZINĖS APSAUGOS PRIEMONĖS

Įmonės kriptosistemos apimamos fizinės apsaugos priemonės:

- Įrenginių apsauga: įrenginių naudojimo taisyklės (vartotojų darbo vietų, nešiojamųjų įrenginių ir pan.).
- Nereikalingų dokumentų naikinimas: nenaudojami duomenys turi būti patikimai sunaikinti (yra daugybė priemonių, kurias pasitelkus iš menkiausių nuotrupų galima atkurti disko informaciją).
- Tinkamas šifravimo raktų saugojimas: raktų saugojimo taisyklės.
- Tinkamas slaptažodžių naudojimas: slaptažodžių naudojimo taisyklės.

Jei naudojama viešųjų raktų infrastruktūra, vartotojų privatieji raktai gali būti saugomi įprastose kompiuterių laikmenose arba specialiose laikmenose su apsaugota atmintine. Paprasčiausios laikmenos yra CD ir DVD diskai, USB laikmenos ir paprasčiausias kompiuteris. Šiose laikmenose paprastai privatieji raktai apsaugomi PIN kodu arba slaptažodžiu.

Raktų saugojimo įrenginiai, kuriuose įrengta apsaugota atmintinė:

- lustinė kortelė;

- speciali elektroninių raktų laikmena (angl. *token*).

Šiuose įrenginiuose, be apsaugotos atmintinės, įrengti ir mikroprocesoriai, atliekantys visus kriptografinius skaičiavimus laikmenoje. Kadangi visi skaičiavimai yra vidiniai, slapti privatieji raktai visada lieka įrenginio viduje. Kai kurie USB laikmenų modeliai turi slaptažodžiu apsaugotą atmintinę, tačiau tokios laikmenos neturi mikroprocesorių, tad priskiriamos paprastoms. Pagal visas kitas savybes šias raktų laikmenas galima laikyti identiškomis. Verta paminėti nebent tai, kad dėl ribotų mikroprocesorių galimybių lustinėse kortelėse ir specialiose elektroninių raktų laikmenose rengiamos ribotos talpos atmintinės (pvz., 32, 64 kB). Tuo tarpu paprastose elektroninėse laikmenose (CD, DVD) galima saugoti gerokai didesnius raktus.

Remiantis ANSI X9.17 standartu, skiriami šių tipų raktai:

- Raktų šifravimo raktai. Šie raktai yra ilgaamžiškiausi ir turi būti saugiausi. Juos generuoti ir paskirstyti – sudėtingiausia ir brangiausia.
- Duomenų šifravimo raktai. Tai vidutinio ilgaamžiškumo bendrojo pobūdžio raktai, dažniausiai skirstomi pagal sudėtingesnę schemą nei Diffie'io ir Hellmano.

Pagal reikalaujamą saugumą visus raktus galima skirstyti į tris lygius:

- Raktų šifravimo raktai. Šis saugumo lygis nustatomas neatsižvelgiant į tai, koks algoritmas bus naudojamas raktams generuoti.
- EKE protokolu sukurti raktai. Šių raktų saugumo lygis nustatomas pagal naudojamą algoritmą. Kadangi EKE remiasi simetrine ir asimetrine kriptografija, šių raktų saugumo lygis yra aukštesnis nei taikant kriptografijas atskirai.
- Diffie'io ir Hellmano protokolu sukurti raktai. Šie raktai pagrįsti tik asimetrine kriptografija, todėl jie yra ne tokie saugūs nei tie, kuriuose taikoma drauge simetrinė ir asimetrinė kriptografijos.

Įmonė, norėdama užtikrinti tinkamą raktų administravimą, gali įrengti raktų paskirstymo centrą, kuris būtų atsakingas už penkis pagrindinius raktų „gyvavimo" etapus:

1. Raktų generavimą. Pirmiausia raktai turi būti „pagaminti", t. y. sugeneruoti.
2. Raktų paskirstymą. Sugeneravus raktus, juos reikia paskirstyti.
3. Raktų atnaujinimą. Pasibaigus raktų galiojimo laikui, juos reikia atnaujinti.
4. Raktų deponavimą. Senų raktų dažnai negalima sunaikinti, antraip nebus galima iššifruoti senų šifrogramų, kurios turi būti saugomos.
5. Raktai sunaikinam tada, kai nėra jais saugomų šifrogramų.

Viena populiariausių ir jautriausių sistemos dalių yra slaptažodžiai. Kad ir kokios geros būtų fizinės apsaugos priemonės, didesnė atsakomybė vis dėlto tenka paprastiems sistemos vartotojams, kuriems paprastai paliekama teisė sudaryti savo slaptažodžius.

Norint užtikrinti saugų ir veiksmingą sistemos darbą, reikia laikytis šių pagrindinių saugaus slaptažodžio sudarymo taisyklių:

- Nuolat keisti slaptažodį: slaptažodžio keitimo dažnumas priklauso nuo saugumo lygio.
- Derinti didžiąsias ir mažąsias raides su skaičiais bei specialiaisiais simboliais: padidėja galimų variantų skaičius.
- Vartoti 8 ir daugiau simbolių: padidėja galimų variantų skaičius.
- Nevartoti egzotiškų ar retai pasitaikančių tarptautinių žodžių: toks slaptažodis gali būti pažeistas žodyno atakos metu.
- Vengti vartoti tą patį slaptažodį keliose vietose: praradus slaptažodį vienoje sistemoje, kiltų pavojus kitoms sistemoms.
- Nevartoti paprastų derinių: sudarant lengvai įsimenamą slaptažodį iš paprastų derinių, palengvinama žodyno ataka.

- Pasirinkti tokį slaptažodį, kurį būtų sunku „nulaužti“, tačiau būtų galima prisiminti jo neužsirašius.
- Nevartoti nė viena kalba taisyklingai parašytų žodžių.

Jei slaptažodžiai sudaromi nesilaikant šių taisyklių, kyla pavojus, kad kenkėjai gali įvykdyti vieną iš slaptažodžių atakų.

Skiriamos šios slaptažodžių atakų rūšys:

- Atspėjimo: remiantis kokia nors informacija, bandoma atspėti slaptažodį.
- Žodyno: tikrinami tam tikri žodyno žodžiai.
- Visiško perrinkimo: perrenkami visi galimi variantai.
- Išankstinio apskaičiavimo: jei naudojama kokia nors slaptažodžio transformacija, tai, panašiai kaip ir žodyno atakos metu, sudaromi atitinkamų transformacijų sąrašai ir jų reikšmės lyginamos su puolamu slaptažodžiu.

14.3. KRIPTOGRAFINIAI PROTOKOLAI

Kuriant įmonės infrastruktūrą, svarbu saugiai bendrauti internetu. Reikia užtikrinti saugų įmonės darbuotojų ryšį, siekiant apsaugoti informaciją nuo pašalinių akių bei pramoninio šnipinėjimo. Taip pat svarbu užtikrinti įmonės bendravimą su klientais, siekiant apsaugoti jų privačią informaciją: klientų asmens duomenis, užsakymus, banko sąskaitų numerius ir t. t.

Saugus bendravimas internetu užtikrinamas taikant įvairius kriptografinius metodus. Kriptografiniai algoritmai, pasitelkus matematines priemones, užtikrina informacijos autentiškumą, vientisumą ir nepaneigiamumą. Tačiau taikant kriptografinius metodus svarbu suvokti protokolų ir algoritmų skirtumus. Pavyzdžiui, galima teigti, kad pagal tam tikrą metodą užšifruoti duomenys bus saugūs. Teoriškai kenkėjui gali prireikti milijono metų iššifruoti informaciją, pasitelkus tobuliausius kompiuterius. Tačiau tai dar nereiškia, kad protokolas, naudojantis šį algoritmą kaip vieną iš sudedamųjų dalių, yra toks pat patikimas. Protokole gali būti kitų spragų, kurios paprasčiausiai leistų apeiti net saugiausius šifravimo algoritmus.

Protokolu išsamiai aprašomi žingsniai, atliekantys kriptografinius veiksmus įvairiose situacijose. Algoritmas yra daug siauresnė procedūra, transformuojanti skaitmeninius duomenis į kitokios formos skaitmeninę informaciją. Kriptografiniai protokolai neišvengiamai naudoja vieną ar daugiau kriptografinių algoritmų, tačiau protokolo saugumą lemia visų jo sudedamųjų dalių saugumas.

Saugus bendravimas internetu įmonėje neatsiejamas nuo kriptografinių protokolų. Kurti nuosavus protokolus – labai sudėtinga ir rizikinga. Palikus bent menkiausią spragą naujajame protokole, įmonė bei jos klientai gali patirti didelių nuostolių, todėl dažnai rekomenduojama naudoti plačiai paplitusius ir laiko patikrintus protokolus.

Vienas tokių protokolų – IPsec (angl. *Secure Internet Protocol*), kuris realizuojamas operacinės sistemos branduolyje. Jis leidžia saugiai perduoti informaciją, naudojant patikimus duomenų perdavimo protokolus (pvz., TCP ar UDP), ir yra suderintas su įvairiais interneto protokolo standartais (IPv4 ir IPv6). Informacijos šifravimas paskutiniame etape (prieš pat siuntimą) leidžia sėkmingai šifruoti beveik visų taikomųjų programų siunčiamus duomenis. Taip pat atskiroms taikomosioms programoms nereikia atskirai realizuoti IPsec protokolo. Protokolas taiko simetrinius duomenų šifravimo metodus ir bendruoju atveju neišskiria nė vieno bendraujančio subjekto. Didžiausias šio protokolo trūkumas – simetrinių šifravimo raktų paskirstymas. Šifravimo raktai gali būti skirstomi rankiniu būdu (tai labai sudėtinga didelėse įmonėse) arba automatiškai. Tačiau automatinis kriptografinių raktų paskirstymas reikalauja atskiro protokolo.

Internetu dažniausiai bendrauja klientas ir serveris, todėl dauguma kriptografinių protokolų orientuoti būtent į tokį bendravimo modelį. Ne išimtis ir S-HTTP (angl. *Secure Hypertext Transfare Protocol*) protokolas, kuris buvo kuriamas atsižvelgiant į duomenų perdavimą HTTP formatu. S-HTTP protokolu šifruojama ne visa ryšio linija, o atskiri pranešimai. Jis yra lankstus ir suderinamas su daug šifravimo metodų. Kiekviena taikomoji programa turi turėti atskirą protokolo realizaciją. Tačiau dėl didelės kitų

protokolų konkurencijos jis plačiai nepaplito ir jau ilgą laiką netobulinamas (paskutinė oficialioji šio protokolo versija 1.4 buvo išleista 1999 metais).

Įmonėse labai svarbus nuotolinis prisijungimas prie kompiuterių ar kitų sistemų. Tam dažnai naudojami tokie protokolai kaip *rlogin, telnet, ftp* ir t. t. Tačiau šie protokolai slaptažodį internetu siunčia neužšifruotą. Šiai saugumo spragai užlopyti sukurtas SSH (angl. *Secure Shell*) protokolas, leidžiantis serveriui, pasitelkus asimetrinį šifravimą ir elektroninį parašą, autentifikuotis klientui bei apsikeisti simetrinio šifravimo raktu. Taip sukuriamas saugus ryšio kanalas, kuriuo klientas gali saugiai siųsti savo slaptažodį. Tačiau protokolo specifikacija yra labai plati ir suteikia galimybę naudoti nepatikimus elektroninio parašo sertifikatus. Protokolas taip pat leidžia vartotojui autentifikuotis serveriui taikant viešojo rakto kriptografiją, tačiau bendruoju atveju serverio ir vartotojo autentifikavimas yra išskaidytas į du atskirus protokolus. Kiekvienoje taikomojoje programoje turi būti atskira protokolo realizacija.

SSL (angl. *Secure Sockets Layer*) protokolas ir jo įpėdinis TLS (angl. *Transpot Layer Security*) – universalūs ir plačiai paplitę kriptografiniai protokolai. Pagrindinė jų paskirtis – sukurti saugų ryšį tarp kliento ir serverio. Įmonėje protokolą galima naudoti darbuotojų ir įmonės klientų bendravimui su įmonės serveriu.

SSL protokolas vienareikšmiškai autentifikuoja subjektus. Užšifruoti duomenys saugiai siunčiami patikimais duomenų perdavimo protokolais (pvz., TCP ar UDP) per interneto protokolus (IPv4 ir IPv6). Naudojant SSL, galima šifruoti visų taikomųjų programų ir aukštesnio lygmens protokolų informaciją. Dažniausiai SSL protokolu šifruojami elektroninio pašto protokolai SMTP ir POP3, pokalbių protokolas IRC, duomenų perdavimo protokolas FTP (toks protokolų derinys kitaip dar vadinamas FTPS), interneto puslapių protokolas HTTP (kitaip dar vadinamas HTTPS) ir daugelis kitų. Tačiau kiekviena taikomoji programa turi turėti atskirą protokolo realizaciją.

Siekiant užtikrinti saugų bendravimą internetu, neužtenka vien šifruoti duomenis. Tiek darbuotojams, tiek klientams (toliau vadinsime vartotojais) reikia patikimai autentifikuotis įmonės serveryje, kad pašaliniai asmenys negalėtų pakenkti įmonės veiklai, apsimetę vartotojais. Kenkėjai gali imituoti įmonės serverį ir taip sužinoti privačius vartotojų duomenis. Akivaizdu, kad tik pateikus vartotojo ar įmonės vardą subjektu pasitikėti negalima. Taikant kriptografinius protokolus, tikslinga žinoti, kada ir kokiais duomenimis galima atlikti patikimą autentifikaciją.

Atsižvelgiant į siekiamą saugumo lygį, parenkami įvairūs parametrai. Dažnai užtenka paprasto slaptažodžio, tačiau nereikėtų pamiršti, kad siunčiant slaptažodį viešuoju tinklu jis yra visiems matomas. Slaptažodžius patartina siųsti saugiais ryšio kanalais, apsaugotais tokiais kriptografiniais protokolais kaip SSL.

Siekiant supaprastinti autentifikacijos procesą, plačiai naudojami slapukai (angl. *cookies*). Tai mažos duomenų rinkmenos, kuriose saugoma informacija apie vartotoją. Pakartotinai jungiantis prie interneto puslapio ar jį naršant, slapukais apsikeičia serveris ir vartotojas, kad nereikėtų kaskart įvesti slaptažodžio. Pavyzdžiui, slapukai gali būti labai naudingi įmonėi kuriant interneto parduotuvę, kad nereikėtų pakartotinai autentifikuotis dedant naują pirkinį į krepšelį. Tačiau nereikėtų pamiršti, kad, norint saugiai naudoti slapukus, juos reikia siųsti kriptografiniu protokolu apsaugotu ryšio kanalu. Slapukai taip pat atvirai saugomi vartotojo kompiuteryje ir gali būti lengvai pasiekiami. Yra daugybė būdų išgauti slapukus, todėl juos naudoti net saugiame ryšio kanale reikia labai atsargiai.

Norint užtikrinti ypatingą saugumą, vien slaptažodžių neužtenka. Tokiu atveju įmonėje patartina naudoti elektroninį parašą, tačiau turėti tik elektroninio parašo sertifikatą taip pat negana, mat pats sertifikatas tik susieja subjekto viešąjį raktą su pačiu subjektu. Bet kas gali gauti jūsų sertifikatą ir, apsimetęs jumis, nusiųsti sertifikatą kitam subjektui. Taigi be duomenų apsaugos algoritmų sertifikatas autentiškumo neužtikrina. Turimus elektroninio parašo sertifikatus reikia naudoti kartu su kriptografiniais protokolais, kurie užtikrina, kad subjektas, išsiuntęs sertifikatą viešuoju raktu, taip pat turi ir privatųjį.

Žinoma, reikia užtikrinti ir paties sertifikato patikimumą, nes bet kas gali pasidaryti savo elektroninio parašo sertifikatą. Tam kuriami patikimi sertifikavimo centrai. Jei sertifikatas naudojamas tik įmonės viduje, įmonė gali įsteigti savo privatų sertifikavimo centrą, kuris susies viešąjį raktą su subjektu. Dažnai serveriui išduodant sertifikatą, jame nurodomas serverio interneto adresas, taip dar labiau susiejant sertifikatą su serveriu.

Įmonė, naudojanti sertifikatus, privalo atkreipti dėmesį į jų galiojimo laiką ir į sertifikatų atšaukimo sąrašą, nes pametus sertifikatą juo naudotis neleidžiama. Darbuotojui nutraukus darbo sutartį ar klientui bendradarbiavimą, šių subjektų sertifikatai turi būti atšaukti, kad jie negalėtų naudotis įmonės sisteminiais ištekliais.

Įmonės serveris privalo turėti patikimų sertifikatų sąrašą, kad tik vartotojai su serveriui žinomais sertifikatais galėtų pasiekti reikiamus sisteminius išteklius. Vartotojai bendruoju atveju patikimų sertifikatų sąrašo nenaudoja, nes neturint serverio sertifikato neįmanoma bendrauti su tokiu serveriu. Vartotojams pakanka turėti tik sertifikavimo centro sertifikatą ir patikrinti visus norimus sertifikatus.

14.4. ĮMONĖS ŠIFRAVIMO SISTEMA

Dar visai neseniai vienintelis būdas naudotis šifravimo galimybe įmonėje buvo įsigyti, išdėstyti ir valdyti atskirą šifravimo programą kiekvienos rūšies duomenims, kuriuos būtina apsaugoti.

Tokiu būdu apsaugomi duomenys, bet kyla nemažai kitų problemų: įmones įklampina papildomi darbai, nesugebama valdyti visų šifravimo programų iš vieno valdymo centro ir greitai išplėsti savo apsaugos strategijos, prisitaikant prie nuolatinės produktyvumo pažangos.

Ši situacija įmonėms – ne naujiena. Tokia pat problema egzistavo kuriant programinę įrangą įmonės ištekliams planuoti ir klientų santykiams valdyti. Tuomet taip pat buvo reikalingas kompleksinis pritaikymas – atskiras išdėstymas ir valdymas, kol įmonės pasirinko platformos principą, kuris centralizavo valdymą, sumažino papildomų darbų skaičių ir leido naujas programas įdiegti sparčiai ir lengvai.

Šiuo metu toks pats metodas prieinamas ir šifruojant. Šifravimo platforma sumažina saugomų verslo duomenų sudėtingumą, leisdama organizacijoms diegti ir valdyti kelias šifravimo programas iš vieno valdymo pulto. Platforma paremtas sprendimas leidžia organizacijoms pagal poreikį sparčiai diegti naujas šifravimo programas. Pavyzdžiui, kompanija gali įdiegti e. pašto šifravimą, pasirinkti įdiegti viso diskinio kaupiklio šifravimo priemones į visus nešiojamuosius kompiuterius.

Vėliau platforma pateikia sprendimus, kaip įdiegti tiesioginį atmintinės šifravimą inžinerinėms, žmogiškųjų išteklių, finansinėms, teisinėms ir kitoms pagrindinėms funkcijoms, kurios naudoja jautrią ar konfidencialią informaciją. Visas diegimo procesas valdomas iš vieno administratoriaus valdymo aplinkos, taikant svarbiausias nustatytas šifravimo strategijas, kad pagal poreikį būtų automatizuojamas šifravimas arba įtraukiami nauji vartotojai ir programos.

14.5. ĮMONĖS SAUGUMO VERTINIMAS

Apsaugos efektyvumo balas (angl. *Security Effectiveness Scores* (SESs)) atspindi asmenų, atsakingų už įmonės saugumą, pasitikėjimo lygį, atsižvelgiant į jų organizacijos bendrą apsaugą ir vidinę kontrolę. Balas paremtas asmeninių atsakymų į 24 požymius, kurie yra laikomi kritiniais sėkmingos įmonės apsaugos atžvilgiu, vidurkiu. Aukščiausias galimas SES respondento organizacijai yra +2, o mažiausias – –2.

Išvardysime 24 požymius, pagal kuriuos apibūdinama efektyvi IT apsauga:

1. Atpažinti daugumą duomenų pažeidimų, susijusių su jautria ar konfidencialia informacija.
2. Nustatyti pagrindines duomenų pažeidimų, susijusių su jautria ar konfidencialia informacija, priežastis.
3. Žinoti fizinę vietą, kurioje yra jautri ar konfidenciali informacija.
4. Apsaugoti nenaudojamą jautrią ar konfidencialią informaciją.
5. Apsaugoti jautrią ar konfidencialią informaciją, naudojamą kasdienėje veikloje.
6. Apsaugoti galutinius tinklo taškus.
7. Identifikuoti ir autentifikuoti galutinius sistemos vartotojus prieš suteikiant jiems teisę pasiekti jautrią ar konfidencialią informaciją.

8. Apsaugoti jautrią ar konfidencialią informaciją, kuria naudojasi pašaliniai asmenys (įskaitant trečiąsias šalis, bendrovių filialus ir verslo partnerius).
9. Išvengti ar mažinti duomenų pažeidimų, susijusių su jautria ar konfidencialia informacija, skaičių.
10. Riboti ar išvengti bandymų piratauti, siekiant gauti jautrią ar konfidencialią informaciją.
11. Riboti ar išvengti paslaugos atmetimo atakų (angl. *denial of service attacks*).
12. Riboti fizinę prieigą prie duomenų laikymo įrenginių, kuriuose yra jautri ar konfidenciali informacija.
13. Parodyti kompanijos kriptosistemos programos ekonominę vertę ar kitą apčiuopiamą naudą.
14. Užtikrinti kuo mažesnę prastovą ar sistemos suardymą kilus saugumo problemų.
15. Laikytis teisėtų reikalavimų ir strategijų (įskaitant privatumo įstatymus ir statutus).
16. Atitikti pagrindinius susireguliavimo reikalavimus, apibrėžiančius duomenų apsaugą (pvz., ISO 17799, PCI ir kt.).
17. Riboti ar išvengti užkrėtimų virusais, kirminais, „Trojos arkliais“ ir šnipinėjimo programomis.
18. Nuolat atnaujinti visas pagrindines apsaugos pataisas.
19. Tikrinti visus svarbius duomenis, naudojamus tobulinant sistemas.
20. Įgyvendinti įmonės strategiją, įskaitant sutarčių su darbuotojais ar rangovais nutraukimą, jei šie keltų rimtą grėsmę įmonės konfidencialiai informacijai.
21. Pritraukti ir išsaugoti aukštos kvalifikacijos IT apsaugos personalą.
22. Mokymų ir supratimo programa – visiems sistemos vartotojams.
23. Vadovauti nepriklausomam sistemos auditui.
24. Nuolat užtikrinti apsaugos programos valdymą.

Šie požymiai atspindi svarbiausius įmonės kriptosistemos taškus, kurių saugumas ir priežiūra lemia bendrą sistemos saugumą.

15. LITERATŪRA

1. **Biham E. ir Shamir A.** Differential cryptanalysis of DES-like cryptosystems. [Žurnalas] // Journal of Cryptology, 4 (1). - 1991 m.. - psl. 3-72.
2. **Biham E., Dunkelman O. ir Keller N.** New results on boomerang and rectangle attacks. [Konferencija] // Proceedings of Fast Software Encryption, Leuven, LNCS, vol. 2365, eds. J. Daemen ir V. Rijmen. - Berlin : Springer, 2002. - psl. 1-16.
3. **Biryukov A. ir De Cannière C.** Block Ciphers and Systems of Quadratic Equations [Konferencija] // Fast Software Encryption – FSE 2003 / mont. Johansson T.. - [s.l.] : Springer, 2003. - T. 2887. - psl. 274-289.
4. **Bulota K. ir Survila P.** Algebra ir skaičių teorija, 2 d. [Knyga]. - Vilnius : Mokslas, 1990.
5. **Chabaud F. ir Vaudenay S.** Links between differential and linear cryptanalysis. [Konferencija] // Advances in Cryptology -EUROCRYPT'94, LNCS, vol. 950. - New York : Springer-Verlag, 1995. - psl. 356-365.
6. **Chaum D., van Heijst E. ir Pfitzmann B.** Cryptographically Strong Undeniable Signatures, Unconditionally Secure for the Signer. [Konferencija] // Advances in Cryptology CRYPTO '91, LNCS, vol. 576. - Berlin / Heidelberg : Springer-Verlag, 1991. - psl. 470-484.
7. **Cheon J.H. ir Lee D.H.** Resistance of S-boxes against Algebraic Attacks. [Konferencija] // Fast Software Encryption, LNCS, vol. 3017. - [s.l.] : Springer-Verlag, 2004. - psl. 83-94.
8. **Cid C. ir Leurent G.** An Analysis of the XSL Algorithm. [Konferencija] // Advances in Cryptology - ASIACRYPT 2005. - Berlin / Heidelberg : Springer, 2005. - T. 3788. - psl. 333-352.
9. **Courtois N.** Is AES a Secure Cipher? [Tinkle] // MinRank Foundation - promoting Multivariate Cryptography. - 2007 m.. - http://www.cryptosystem.net/aes/.
10. **Courtois N., Debraize B. ir Garrido E.** On exact algebraic [non-]immunity of S-boxes based on power functions. [Tinkle] // Cryptology ePrint Archive. - 2005 m.. - http://eprint.iacr.org/2005/203.
11. **Courtois N.T. ir Pieprzyk J.** Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. [Konferencija] // Advances in Cryptology – ASIACRYPT 2002, LNCS / mont. Zheng Y.. - [s.l.] : Springer-Verlag, 2002. - T. 2501. - psl. 267-287.
12. **CRYPTREC** Cryptography Research and Evaluation Committees [Tinkle]. - 2007 m.. - http://www.cryptrec.jp/english/index.html.
13. **Dosinas G. S. ir Navickas Z.** Algebrinės tiesinių operatorių struktūros [Knyga]. - Kaunas : Technologija, 2006.
14. **ECRYPT** ECRYPT STVL Virtual lab [Tinkle] // ECRYPT STVL - SYMMETRIC TECHNIQUES VIRTUAL LAB. - 2005 m..
15. **Fujioka A., Okamoto T. ir Ohta K.** A practical secret voting scheme for large scale elections [Konferencija] // Proceedings of Advances in Cryptology — AUSCRYPT '92, LNCS, vol. 718. - Berlin / Heidelberg : Springer , 1993. - psl. 244-251.
16. **Gentry C., Molnar D. and Ramzan Z.** Efficient Designated Confirmer Signatures Without Random Oracles or General Zero-Knowledge Proofs [Conference] // Advances in Cryptology - ASIACRYPT 2005. - [s.l.] : Springer Berlin / Heidelberg, 2005. - pp. 662-681.
17. **Goldreich O.** Foundations of Cryptography [Knyga]. - Cambridge : Cambridge university Press, 2001.
18. **Goldwasser S. ir Bellare M.** Lecture Notes on Cryptography [Knyga]. - - : MIT, 2001. - http://www-cse.ucsd.edu/~mihir/papers/gb.pdf.
19. **Grigutis R.** Baigtinės algebrinės struktūros [Knyga]. - Vilnius : VU, 1998.

20. **Heys H.M.** A Tutorial on Linear and Differential Cryptanalysis. [Ataskaita] : Technical Report CORR 2001-17, / Centre for Applied Cryptographic Research, Department of Combinatorics and Optimization ; University of Waterloo. - 2001. - http://www.engr.mun.ca/~howard/PAPERS/ldc_tutorial.pdf.
21. **Hong S. [et al.]** Provable security against differential and linear cryptanalysis for the SPN structure. [Konferencija] // Procedings of Fast Software Encryption – FSE 2000, LNCS, vol. 1978, ed. B. Schneier. - Berlin : Springer-Verlag, 2000. - psl. 273-283.
22. **Iao A. C,** Theory and applications of Trapdoor of Computer Science, 23 [Žurnalas]. - 1982 m.. - psl. 80-91.
23. **York E.** Elliptic Curves Over Finite Fields [Tinkle]. - 1992 m.. - http://www.math.rochester.edu/people/grads/jdreibel/ref/yorkECC.pdf.
24. **Kelsey J., Kohno T. ir Schneier B.** Amplified boomerang attacks against reduced-round MARS and Serpent. [Konferencija] // Proceedings of Fast Software Encryption 7, LNCS, vol. 1978, ed. B. Schneier. - Berlin : Springer-Verlag, 2000. - psl. 75-93.
25. **Knudsen L.** Truncated and higher order differentials. [Konferencija] // Proceedings of Fast Software Encryption 2, LNCS, vol. 1008, ed. B. Preneel.. - Berlin : Springer-Verlag, 1995. - psl. 196-211.
26. **Koblitz N.** Algebraic Aspects of Cryptography [Knyga]. - Berlin Heildelberg : Springer-Verlag, 1999.
27. **Lidl R. ir Niederreiter H.** Finite Fields [Knyga]. - London etc. : Addison – Wesley, 1983.
28. **Lopez J. ir Dahap R.** An Overview of Elliptic Curve Cryptography [tinkle] – 2000 - [Tinkle]. - 2000 m.. - http://citeseer.ist.psu.edu/333066.html.
29. **Luby M. ir Rackoff C.** How to Construct Pseudorandom Permutations and Pseudorandom Functions. [Žurnalas] // SIAM Journal Computing, vol. 17. - 1988 m.. - psl. 373-386.
30. **Matsui M.** Linear Cryptanalysis Method for DES Cipher [Konferencija] // Proceedings of Advances in Cryptology — EUROCRYPT'93, LNCS, vol. 765. - Berlin : Springer-Verlag, 1994. - psl. 386-397.
31. **Matsui M.** The First Experimental Cryptanalysis of the Data Encryption Standard [Konferencija] // Proceedings of Advances in Cryptology — CRYPTO ’94, LNCS, vol. 839. - Berlin : Springer-Verlag, 1994. - psl. 1-11.
32. **Matuliauskas A.** Algebra [Knyga]. - Vilnius : Mokslas, 1985.
33. **Menezes A.J., van Oorschot C.P. ir Vanstone S.A.** Handbook of Applied Cryptography [Knyga]. - New Yourk : CRC Press, 2001.
34. **Murphy S. ir Robshaw M.J.B.** Essential Algebraic Structure within the AES. [Konferencija] // Proceedings of CRYPTO 2002, LNCS, vol. 2442. - [s.l.] : Springer-Verlag, 2002. - psl. 1-19.
35. **NESSIE Consortium** Portfolio of recommended cryptographic primitives [Tinkle]. - 2003 m.. - https://www.cosic.esat.kuleuven.be/nessie/.
36. **NIST** Advanced Encryption Standard (AES) [Ataskaita] : FIPS PUB 197. - [s.l.] : National Institute of Standards and Technology, 2001.
37. **NIST** FIPS 186-2 Digital Signature Standard [Ataskaita]. - [s.l.] : NIST, 2000.
38. **Nyberg K. ir Knudsen L.R.** Provable security against a differential attack. [Žurnalas] // Journal of Cryptology, 8 (1). - 1995 m.. - psl. 27-38.
39. **Nyberg K.** Linear approximations of block ciphers. [Konferencija] // Proceeding of Advances in Cryptography – EUROCRYPT'94, LNCS, vol. 950, ed. A. De Santis.. - Berlin : Springer-Verlag, 1994. - psl. 439-444.
40. **Ohkubo M. [et al.]** An Improvement on a Practical Secret Voting Scheme [Konferencija] // Proceedings of Information Security Workshop – ISW'99, LNCS, vol. 1729. - Berlin / Heidelberg : Springer, 1999. - psl. 225-234.

41. **Oppliger R.** Contemporary Cryptography [Knyga]. - Norwood : Artech House Publisher, 2005.

42. **Patarin J.** Luby-Rackoff: 7 Rounds Are Enough for Security. [Konferencija] // Advances in Cryptology - CRYPTO 2003, LNCS, vol. 2729, ed. D. Boneh. - 2003. - psl. 513-529.

43. **Pommerening K.** Fourier Analysis of Boolean Maps – A Tutorial. [Tinkle]. - 2005 m.. - http://www.staff.uni-mainz.de/pommeren/Kryptologie/Bitblock/A_Nonlin/Fourier.pdf.

44. **Preneel B. [et al.]** New European Schemes for Signature, Integrity, and Encryption [Tinkle] // NESSIE, Final Report. - 2004 m.. - https://www.cosic.esat.kuleuven.be/nessie/.

45. **Schaumüller-Bichl I** Cryptanalysis of the Data Encryption Standard by the Method of Formal Coding [Konferencija] // Cryptography: Proceedings of the Workshop on Cryptography, LNCS, vol. 149. - Berlin : Springer, 1983.

46. **Shamir A.** Stream Ciphers: Dead or Alive? [Konferencija] // Advances in Cryptology - ASIACRYPT 2004, LNCS 3329. - [s.l.] : Springer Berlin / Heidelberg, 2004. - psl. 78.

47. **Steiner M., Tsudik G. ir Waidner M.** Diffie–Hellman key distribution extended to groups [Konferencija] // Proceedings of the 3rd ACM conference on Computer and communications security. - New York : ACM Press, 1996. - psl. 31-37.

48. **van der Varden B.L.** Algebra [Knyga]. - Maksva : Nauka, 1976.

49. **van Tilborg H. C. A.** Encyclopedia of Cryptography and Security [Knyga]. - NY : Springer, 2005.

50. **Vaundenay S.** A Classical Introduction to Cryptography: Applications for Comunications Security [Knyga]. - [s.l.] : Springer, 2006. - 0-387-25464-1.

51. **Wagner D.** The boomerang attack. [Konferencija] // Proceedings of Fast Software Encryption, FSE'99, Rome, LNCS, vol. 1636, ed. L. Knudsen.. - Berlin : Springer, 1999. - psl. 156-170.

52. **Гроссман И. ir Магнус В.** Группы и их графы [Knyga]. - Москва : Мир, 1971.

53. **Гэри М. ir Джонсон Д.** Вычислительные машины и труднорешаемые задачи [Knyga]. - М. : Мир.

54. **Кострикин А. И.** Введение в алгебру. Часть III. Оосновные структуры. [Knyga]. - М : физ.-мат. лит., 2001.

55. **Кострикин А.И.** Введение в алгебру : учебник / Александр Иванович Кострикин [Knyga]. - Москва : Наука, 1977.

56. **Логачев А. О., А. Сальников, А. ir Ященко В. В.** Булевы функции в теории колирования и криптологии [Knyga]. - М. : МЦНМО, 2004.

57. **Магнус В., Каррас А. ir Солитэр Д.** Комбинаторная теория групп : представление групп в терминах образующих и соотношений [Knyga]. - Москва : Наука, 1974.

SIMBOLIŲ IR SUTRUMPINIMŲ SĄRAŠAS

Simbolis	**Paaiškinimas**
κ	Paslaptis, kuria turi pasidalyti kriptosubjektai.
$\Rightarrow$	Priežastinis (implikacinis) sąvokų ryšys. Pvz., $A \Rightarrow B$ reiškia, kad iš A seka B.
κ	Paslaptis, kuria turi pasidalyti kriptografiniai subjektai.
$\oplus$	Sudėtis moduliu 2.
$\sigma(PR_A, m_A)$	E. parašo operatorius, suformuojantis e. parašą s_A subjekto $\mathcal{A}$ privačiuoju raktu PER_A duomenų santraukai m.
$\upsilon(PR_A, s_A, m_A)$	$\mathcal{A}$ suformuoto e. parašo s_A patikros funkcija duomenų santraukai m_A.
$(a, b) = 1$ arba $\gcd(a, b) = 1$	Reliatyviai pirminiai sveikieji skaičiai a ir b.
$\{0, 1\}^*$	Baigtinė dvejetainių skaičių aibė arba eilutė.
$\{0, 1\}^l$	l ilgio dvejetainių skaičių aibė arba eilutė.
$\|x\|$	Kintamojo x ilgis, nusakomas bitų, skaitmenų arba simbolių skaičiumi.
$<A; \oplus, \otimes> \tilde{=} <B; \square, \nabla>$	Homomorfinės struktūros.
$<A; \oplus, \otimes> \cong <B; \square, \nabla>$	Izomorfinės struktūros.
$a \equiv b (\mathrm{mod}\ m) \Leftrightarrow m \mid (a - b)$	Kongruencija (lyginys).
$b \nmid a$	b nedalija a; $b \neq 0$.
$b \mid a$	b dalija a, $b \neq 0$.
c	Šifrograma.
c_A	Subjekto $\mathcal{A}$ gauta šifrograma, taikant asimetrinį šifravimą kito subjekto, pvz., $\mathcal{B}$,viešuoju raktu VR_B.
$D(k, c)$	Šifrogramos c iššifravimo funkcija D raktu k.
$D(PR_B, c_A)$	Asimetrinio iššifravimo funkcija, kurią naudoja subjektas $\mathcal{B}$ savo privačiuoju raktu PR_B šifruodamas šifrogramą c_A.
$D_k(c)$	Šifrogramos c iššifravimo funkcija D raktu k.
$E(k, t)$	Tekstogramos t užšifravimo funkcija E raktu k.
$E(VR_B, t_A)$	Asimetrinio užšifravimo funkcija, kurią pasitelkia subjektas $\mathcal{A}$, $\mathcal{B}$ viešuoju raktu apskaičiuodamas šifrogramą c_A.
$E_k(t)$	Tekstogramos t užšifravimo funkcija E raktu k.
$\mathbf{F}[x]/(f)$	Daugianarių žiedo lauke $\mathbf{F}$ faktoržiedas, kurį generuoja f – neskaidomas tame lauke daugianaris.
gcd(a, b) arba *(a, b)*	Sveikųjų skaičių a ir b bendrasis didžiausias daliklis.
GF(p)	Galua laukas; p – pirminis.
$H(t)$	Tekstogramos t santraukos funkcija, apskaičiuojanti santrauką m.
$\mathbf{N}$	Natūraliųjų skaičių aibė: {1, 2, 3, ...}.
Prob[*a*]	Įvykio a tikimybė.
$s_1 \| s_2$	Dviejų bitų eilučių sujungimas.
s_A	Subjekto $\mathcal{A}$ e. parašas duomenų santraukai m_A.
t	Tekstograma.
t_A	Subjekto $\mathcal{A}$ tekstograma.
$x \in \{0, 1\}^*$	Elementas x, priklausantis aibei $\{0, 1\}^*$.
$x \in_R \{0, 1\}^*$	Reikšmė x, atsitiktinai parenkama iš aibės $\{0, 1\}^*$.
$x \| y$	Kintamųjų x ir y simbolių eilučių prijungimas viena prie kitos.
$\mathbf{Z}$	Sveikųjų skaičių aibė: {..., –3, –2, –1, 0, 1, 2, 3, ...}.
$\mathbf{Z}_n$	Liekanų klasių aibė $\{\overline{0}, \overline{1}, \overline{2}, \ldots, \overline{n-1}\}$ moduliu n.

Santraukos (santrumpos)	**Paaiškinimai**
AES	Pažangaus šifravimo standartas (angl. *Advanced Encryption Standard*), specifikuotas FIPS PUB 197.
ANF	Algebrinė normalinė forma.
BBS	*Blum Blum Schubo* generatorius.
CBC	Šifro bloko grandininis šifravimo režimas (angl. *Cipherblock chaining*).
CFB	Grįžtamojo ryšio šifro šifravimo režimas (angl. *Cipher feedback*).
CI	Koreliacinis imunitetas (angl. *Correlation immunity*).
CRHF	Atsparios kolizijoms santraukos funkcijos (angl. CRHF – *Collision Resistant Hash Function*).
DEK	Duomenų šifravimo raktai (angl. *Data Encrypting Keys*)
DH	Mokslininkų W. Diffie'io ir M. Hellmano vardu pavadinti protokolai.
DLP	Diskretinio logaritmo problema (angl. *Discrete Logarithm Problem*).
ECB	Elektroninės šifrų knygos šifravimo režimas (angl. *Electronic codebook – ECB*).
ECDSA	Elipsinių kreivių e. parašo sistemą (angl. *Elliptic curve digital signature algorithm*).
ECDH	Elipsinių kreivių DH RAP.
EK	Elipsinės kreivės (angl. *Elliptic curves*).
EKE	Šifruotų raktų apsikeitimas (angl. *Encrypted Key Exchange*).
IV	Inicijavimo vektorius (angl. *Initiation Vector*).
KEK	Raktų šifravimo raktai (angl. *Key Encrypting Keys*).
MAC	Pranešimo autentifikavimo kodas (angl. MAC – *Message Authentication Code*).
MK	Magistraliniai raktai (angl. *Master Key*).
OFB	Grįžtamojo ryšio išvesties šifravimo režimas (angl. *Output feedback*).
OWHF	Vienkryptės santraukos funkcijos (angl. OWHF – *One-Way Hash Function*).
PASG	Pseudoatsitiktinių skaičių generatorius (angl. *Pseudo-Random Number Generator* (*PRNG*)).
P blokas	Perstatymo blokas.
PPS	Paslapties pasidalijimo schema (angl. *SECRET Sharing Scheme*).
PR	Privatusis raktas (angl. *Private Key*).
PR	Privatusis arba slaptasis raktas.
RAP	Raktų apsikeitimo protokolas (angl. *Key Agreement Protocol* (*RAP*)).
RSA	Viena labiausiai paplitusių viešojo rakto kriptosistemų (RSA). R. Rivest, A. Shamir, L. Adelman – sistemą sukūrę mokslininkai.
SAC	Griežtas lavininis kriterijus (angl. *Strict avalanche criterion* – SAC).
S blokas	Sukeitimo blokas.
SP tinklas	Sukeitimų ir perstatymų tinklas.
TGPR	Tiesinis grįžtamojo ryšio postūmio registras (TGPR, angl. *LFSR – Linear feedback shift register*).
TKG	Tiesinis kongruencinis generatorius.
TTP	Trečioji patikima šalis (angl. *Third Trusted Party*).
VR	Viešasis raktas (angl. *Public Key*).
VRI	Viešojo rakto infrastruktūra (angl. *Public Key Infrastructure* (*PKI*)).

RODYKLĖ

A

B

C

D

E

F

G

H

I

K

L

N

O

P

R

S

Š

T

U

V

Ž

Spausdinti rekomendavo KTU matematikos mokslų krypties kvalifikacijos komisija

KAUNO TECHNOLOGIJOS UNIVERSITEAS

Eligijus Sakalauskas,
Narimantas Listopadskis,
Gediminas Simonas Dosinas

KRIPTOGRAFIJOS TEORIJA

Mokomoji knyga

SL 344. 2008-01-14. 10 leidyb. apsk. l.
Tiražas 60 egz. Užsakymas 3.
UAB „Vitae Litera“, Kurpių g. 5-3, LT-44280 Kaunas