

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Homomorfinio šifravimo taikymas elektroninio balsavimo sistemoje

Application of homomorphic encryption in a digital voting system

Bakalauro baigiamasis darbas

Atliko: Deivis Zolba

Darbo vadovas: Doc., Dr. Vilius Stakėnas

Darbo recenzentas:

Vilnius – 2024

Santrauka

„Išanalizuotos kriptografinės struktūros, RSA aklašis parašas, Paillier homomorfinė schema su homomorfinė savybė, žinių, nesuteikiantys įrodymai. Visas jas pritaikyta kuriant elektroninio balsavimo puslapį.

Pasinaudojant RSA aklojo parašo schema buvo pasiektas anoniminis prisijungimas į balsavimo procesą. Sistema žino, kad šį parašą galėjo susikurti tik autentifikuotas, leistinas balsuotojas.

Balsavimo procese buvo sėkmingai pritaikyta žinių, nesuteikiančių įrodymų, struktūra, kuri leidžia balsuotojui apskaičiuoti savo balsą. Balsuotojas turi pilną balso kontrolę – pilnai žino jog jo balso šifras atitinka jo pasirinkimą. Pasinaudojant žinių, nesuteikiančiu įrodymu, sistema apie balso žinutę žino tik tai, kad ji yra viena iš leistinių. To pilnai pakanka priimti balsą.

Balsų sumavime buvo pasinaudota Paillier homomorfinė savybė, kuri leido viešai atlikti balsų šifrų sumavimus. Kadangi balsai yra viešai matomi, balsuotojai gali patikrinti sistemos veiklą. Tokiu būdu galima sužinoti, jog visų asmenų balsai tikrai buvo įskaičiuoti.

Balsavimo rezultato paskelbimas pasinaudojo Damgård–Jurik Paillier kriptosistemos plėtinium, kuris leidžia surasti privatų raktą r . Šią reikšmę paskelbus naudotojai gali patikrinti, jog viešai homomorfiškai susumuotų šifrų rezultatas tikrai šifruoja tai ką sistema skelbia.“.

Raktiniai žodžiai: Homomorfinis šifravimas, Paillier kriptosistema, Žinių nesuteikiantys įrodymai, elektroninis balsavimas, RSA aklašis parašas

Summary

“Analyzed cryptographic structures, RSA blind signature, Paillier homomorphic scheme with homomorphic property and ZKP – zero knowledge proofs. All of them have been applied to the development of an digital voting website.

Using RSA blind signature scheme for anonymous login to the voting process was achieved. Because system knows that this signature could only be created by an authenticated voter.

The voting process has successfully applied zero knowledge proof, that allows the voter to calculate his vote. The voter has full control over his vote – he is fully aware that his vote cipher corresponds to his choice. Using a zero knowledge proof, the system only knows about that the vote message is one of the allowed ones and this is fully sufficient to accept the voice.

The vote aggregation made use of Paillier’s homomorphic property, which enables public aggregation of vote ciphers. Since the votes are publicly visible, voters can double check works of the system. In this way, it is possible to know that the votes of all individuals were indeed counted.

The publication of the voting result, made use of an extension to the Damgård-Jurik Paillier cryptosystem which allows the private key r to be found. Once this value is published, users can verify that the result of the publicly homomorphically aggregated ciphers is indeed the cipher that the system publishes.”.

Keywords: Homomorphic encryption, Paillier cryptosystem, zero knowledge proof, digital voting, RSA blind signature

Turinys

ĮVADAS	6
1. BALSAVIMO SISTEMOS DALYS	8
1.1. RSA akklasis parašas	8
1.2. Balsavimas	8
1.3. Balsų sumavimas	8
1.4. Rezultato paskelbimas	8
2. HOMOMORFIZMAS	10
2.1. Homomorfizmas matematikoje	10
2.2. Homomorfizmas kriptografijoje	10
2.3. Homomorfizmo schemų klasifikavimas	10
2.3.1. Dalinis homomorfizmas	10
2.3.2. Nevisiškas homomorfizmas	11
2.3.3. Visiškas homomorfizmas	11
2.4. Modulinė aritmetika	11
2.4.1. Atvirkštinis elementas	11
2.4.2. Greitasis kelimas laipsniu	11
3. RSA	12
3.1. RSA schema	12
3.1.1. RSA raktų kūrimo žingsniai	12
3.1.2. Šifravimas	12
3.1.3. Dešifravimas	12
3.2. RSA akklasis parašas	12
3.2.1. RSA aklojo parašo pavyzdys	13
3.2.2. Aklujų parašų taikymas	14
3.2.3. Saugumo aspektai	14
4. AUTENTIFIKAVIMAS KŪRIAMOJE BALSAVIMO SISTEMOJE	15
4.1. Kriptografinių raktų ilgai	15
4.2. Žinutės apribojimai	16
4.3. Sukurtos RSA aklojo parašo svetainės dalis	16
4.3.1. Parašo susėjimas su naudotoju	18
4.4. Tos pačios žinutės problema	18
4.4.1. Gimtadienių ataka	18
5. PAILLIER KRIPTOSHEMA	20
5.1. Paillier rakto kūrimo žingsniai	20
5.2. Šifravimas	20
5.3. Dešifravimas	20
5.4. Algoritmo pavyzdys su skaičiais	20
5.5. Damgård–Jurik plėtinys	21
5.6. Žinutės formatas balsavimo sistemai	22
6. ĮRODYMAI, NESUTEIKIANTYS ŽINIŲ	24
6.1. Žinių nesuteikiantis įrodymas diskretui logaritmui	25
6.1.1. Galimi bandymai sukčiauti	26
6.1.2. Pavyzdys su realiais skaičiais	26
6.1.3. Rezultatas	27

6.2. Interaktyvūs įrodymai	27
6.3. Neinteraktyvūs įrodymai	27
6.3.1. Kaip generuoti iššūkį neinteraktyviuose įrodymuose	28
6.3.2. Hash funkcija	28
6.4. Elektroninis balsavimas	28
6.4.1. Balsavimo proceso reikalavimai	28
6.4.1.1. Žinių nesuteikiantis įrodymas, balso validumas	28
7. BALSAVIMO PROCESAS SISTEMOJE	32
7.1. Balsavimas, neskaičiuojant balso	32
7.2. Balsavimas, skaičiuojant balsą	33
8. BALSAVIMO PROCESO PABAIGA	38
8.1. Balsų lentelė	38
8.2. Balsų sumavimas	38
8.3. Rezultato paskelbimas	38
8.4. Internetinis puslapis	39
IŠVADOS	40
ŠALTINIAI	41

Išvadas

Šiuolaikiniame skaitmeniniame amžiuje rinkimų procesas internetu gali tapti realybe. Elektroninio balsavimo sistemos žada efektyvesnę, prieinamesnę ir saugesnę balsų surinkimo ir skaičiavimo mechanizmą vietos, nacionaliniuose ir organizaciniuose rinkimuose. Tačiau šios sistemos susiduria su sudėtingais iššūkiais, susijusiais su rinkėjų privatumu, balsų patikrinamumu ir balsavimo proceso saugumu. Tarp kriptografinių technologijų, galinčių padėti spręsti šiuos iššūkius, homomorfinis šifravimas, žinių nesuteikiantys įrodymai, išsiskiria savo gebėjimu užtikrinti ir privatumą, vientisumą skaitmeninio balsavimo sistemose.

Šiame bakalure bus tyrinėjamos elektroninio balsavimo problemos bei jų kriptografiniai sprendimai.

- Prisijungimas į balsavimo sistemą, RSA aklašis parašas,
- Balsavimas, žinių nesuteikiantis įrodižymas pateikiant balsą,
- Balsų suskaičiavimas, Paillier kriptosistemos homomorfizmas,
- Balsų rezultato atskleidimas.

Pirmasis iššūkis su kuriuo susidurtų elektroninis balsavimas, būtų rinkėjų autentiškumo patvirtinimas – autentifikacijos problema balsavimo sistemoje. Šią problemą galime išspręsti pasinaudojant RSA akluoju parašu. Šis kriptografinis metodas teiktų dvejopą naudą: išsaugotų rinkėjo anonimiškumą bei patvirtintų jo teisę dalyvauti rinkimų procese. Pasitelkus RSA akluosius parašus, sistema gali saugiai patikrinti rinkėjo įgaliojimus neatskleisdžiant jo tapatybės ir taip išlaikyti pagrindinį slapto balsavimo principą. Nagrinėjant aklojo RSA parašo ypatumus, bus įvertintas:

- Veiksmingumas saugant rinkėjo tapatybę,
- Neleistinos prieigos užkirtimas,
- Vienodų parašų tikimybė.

Balsavimo procesui iškyla balso priėmimo problema, norint suteikti galimybę balsuotojui turėti savo balso pilną kontrolę. Sistema turi priimti tik leistino formato balsus, balso turinį turime patikrinti jo neatsifravus. Šiai problemai yra skirti žinių nesuteikiantys įrodymų protokolai. Šių protokolų tikslas yra įrodyti žinių turėjimą, jų neatskleidus. Šiame darbe bus nagrinėjama kaip užtikrintai priimti balsą, jei balsas apskaičiuotas balsuotojo.

Surinkus balsus jie turi būti suskaičiuoti išsaugant rinkėjų privatumą. Paillier kriptosistemos homomorfinės savybės suteikia būtent tokia galimybę, nes leidžia susumuoti ir suskaičiuoti užšifruotus balsus neiššifruojant kiekvieno balso individualiai. Šiame darbe bus aptariama, kaip homomorfinis šifravimas leidžia saugiai apskaičiuoti rinkimų rezultatus, aptariamais Paillier kriptosistemos matematiniais pagrindais ir praktinis taikymas. Šis metodas užtikrina, kad kiekvieno balso slaptumas nebus pažeistas, nes per visą balsų skaičiavimo procesą yra išlaikomas šifravimas.

Šio darbo tikslai – išnagrinėti pristatytas kriptografines dalis ir visas jas išanalizavus jas pritaikyti praktiškai, sukuriant internetinę svetainę, kurioje jos būtų apjungtos norint sukurti elektroninio balsavimo sistemą.

Praktiniai uždaviniai siekiant išsiaiškinti ir suprogramuoti kriptografinius sistemos įrankius:

- RSA aklaįų parašą,
- Paillier kriptosistemos įgyvendinimas,
- Źinių, nesuteikiančių įrodymų, sukūrimas,
- Homomorfizmo įgyvendinimas,
- Sukūrimas internetinės balsavimo sistemos.

1. Balsavimo sistemos dalys

Kuriamos elektroninės balsavimo tikslas yra užtikrinti kuo skaidresnį balsavimo procesą, nesukeliant grėsmės saugumui. 1 pav. schema tai pristato. Kiekviename žingsnyje yra naudojami išnagrinėti kriptografiniai įrankiai.

1.1. RSA aklašis parašas

RSA aklojo parašo struktūra įgalina asmens nuasmeninimą. Ja pasinaudojus, asmuo gali būti autentifikuotas į balsavimo procesą. Sistema, gavus validų RSA parašą, turi jį priimti. Naudotojo su parašu niekaip negalima susieti, tad naudotojas tampa anonimu.

1.2. Balsavimas

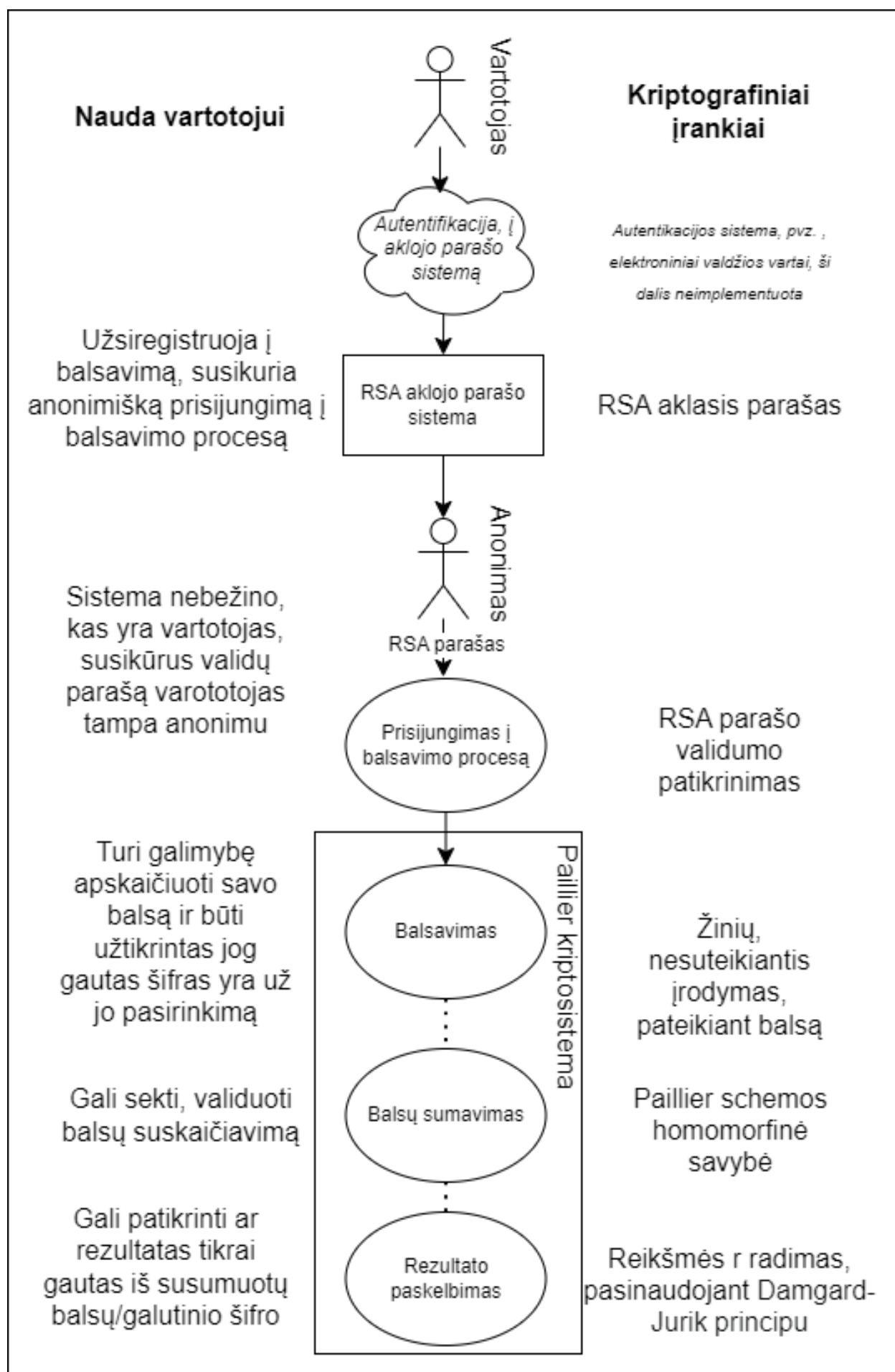
Autentifikuotas naudotojas turi galimybę apskaičiuoti savo balsą. Tokiu būdu jis pilnai yra užtikrintas jo turiniu, bei tuo, kad sistema nežino jo pasirinkimo.

1.3. Balsų sumavimas

Balsavimo sistema naudoja Paillier kriptografinę schemą, kuri įgalina veiksmus su užšifruotais duomenimis, homomorfine savybę. Balsų sumavimas yra atliekamas viešai. Įprasti naudotojai turi galimybę patikrinti sistemos darbą. Sistemai taikant homomorfine savybę galima dešifruoti rezultatą ir gauti galutinį rezultatą per vieną dešifravimo operaciją. Kitos schemos reikalauja kiekvieno šifro dešifravimo.

1.4. Rezultato paskelbimas

Gavus rezultatą sistema pateikia privatųjį raktą r , pagal kurį naudotojai turi galimybę patikrinti rezultato validumą. Tokiu būdu jie yra užtikrinti rezultato validumu.



1 pav. Sistemos procesų eiga

2. Homomorfizmas

2.1. Homomorfizmas matematikoje

Homomorfizmas – atvaizdis iš vieno algebrinio objekto į kitą, išsaugantis kompozicijos dėsnius. A, B yra algebrinės grupės, kuriose operacijos žymimos sudėties ar daugybos ženklais. Atvaizdis $f : A \rightarrow B$ yra vadinamas homomorfizmu, jei bet kuriems $x, y \in A$:

1. $f(x + y) = f(x) + f(y)$
2. $f(x * y) = f(x) * f(y)$

[DM13]

2.2. Homomorfizmas kriptografijoje

Kriptografijoje homomorfizmas galimas per daugybos ar sumos savybę.

E – šifravimo algoritmas.

N – visų galimų žinučių aibė, x, y būtų tos N aibės elementai.

$$E(x * y) = E(x) * E(y)$$

$$E(x + y) = E(x) + E(y)$$

$\forall x, y \in N$ – visos galimos žinutės

Homomorfizmas pasireiškia tada, kai prieš šifravimą x ir y žinutes sudėdami, gauname tą patį šifrą. Jeigu mes jas šifruotume atskirai ir po to sudėtume šifrus, tai būtų tas pats, jei šifruotume kartu. Ši savybė leidžia mums dirbti su šifrais jų neiššifravus – galime turėti daug šifruotų žinučių, kai mums aktualus tik rezultatas, visus šifrus sudėti, klasikinė schema reikalautų kiekvieną atšifruoti ir sudėti/padauginti jau dešifruotas žinutes, bet naudojant homomorfinę sistemą mums to daryti nereikia, mes galime dirbti su pačiais šifrais, juos sudėti/dauginti, tai padarius dešifravimą reikia atlikti tik vieną sykį.

Ši savybė yra galinga, kadangi neturime dešifruoti kiekvieno šifro individualiai, taip neatskleisdžiamė kiekvieno šifro ir galime turėti jau galutinį rezultatą. Pvz., balsų skaičiavime, dešifravus kiekvieną individualiai, būtų galima specialiai išmesti tą balsą žinant, kad jis yra už konkurentus, jeigu mes matome tik šifrus, nebūtų įmanoma ranka išrinkti balsus už konkurentus, o gavus brendrą vaizdą po visų šifrų sudėjimo rezultatų neiškraipytume.

2.3. Homomorfizmo schemų klasifikavimas

2.3.1. Dalinis homomorfizmas

Dalinio homomorfizmo požymį kriptosistema turi tada, kai palaiko tik vieną iš standartinių homomorfizmo operacijų, daugybos arba sudėties. Vadinasi, schema turi palaikyti daugybos ar sumos operacijas šifruose, neprarandant informacijos.

Tik daugybos homomorfizmo požymį turinčios schemos:

- RSA
- Elgamal

Tik sumos homomorfizmo požymį turinčios schemos:

- Pallier

2.3.2. Nevisiškas homomorfizmas

Nevisiško homomorfizmo požymį kriptosistema turi tada, kai palaiko tik limituotą kiekį standartinių homomorfizmo operacijų, daugybos arba sudėties. Šio tipo schemos dažniausiai būna užvaldomos „triukšmo“ po didelio kiekio pritaikymo homomorfiškų operacijų. Egzistuojančios schemos:

- BGN(Benah-Goh-Nissim) [MA18]

2.3.3. Visiškas homomorfizmas

Visiško homomorfizmo požymį kriptosistema turi tada, kai palaiko tiek daugybos, tiek sumos operacijas. Tai reiškia, kad schema turi palaikyti daugybos ir sumos operacijas šifruose, neprarandant informacijos. Visiško homomorfizmo schemos:

- Gentry [AAZS20], [MA18]

2.4. Modulinė aritmetika

Modulinė aritmetika, matematikos šaka, nagrinėjanti skaičių savybes ir operacijas su jais pagal nustatytą modulį. Kriptografijoje labai dažnai yra sutinkama liekanų aritmetika, ypač šifravimo schemose. Neįprastas skaičiavimas būna, kai reikia rasti priešingą elementą $a^{-1} \pmod{n}$.

2.4.1. Atvirkštinis elementas

Jeigu turime skaičių a ir modulį n , norėdami rasti a^{-1} inversiją, reikia rasti tokį skaičių b , kad $a \times b \equiv 1 \pmod{n}$. Šiam elementui rasti naudojamas išplėstinis Euklido algoritmas.

2.4.2. Greitasis kelimas laipsniu

Greitasis kelimas laipsniu – algoritmas kuris palengvina sveikojo skaičiaus kelimo laipsniu modulyje skaičiavimus. Jis dažnai naudojamas kriptografijoje, kad būtų galima efektyviai apskaičiuoti didelius modulinės aritmetikos laipsnius.

3. RSA

RSA schema sukurta 1977 m. trijų matematikų, pagal kurių pavardžių inicialus ir yra pavadinimas, Rono Rivesto, Adi Shamiro ir Leonardo Adlemano. Ši schema pirmoji, kuri turėjo homomorfinę savybę, nors dėl tokio tikslo jos nekūrė. RSA yra viešojo rakto sistema, kuria galima pakankamai saugiai perduoti duomenis, ji ir šiomis dienomis yra naudojama. Schema remiasi pirminiais skaičiais ir jų neturėjimo kelių daliklių savybe. [RSA78]

3.1. RSA schema

3.1.1. RSA raktų kūrimo žingsniai

1. Dviejų pirminių skaičių parinkimas p ir q , kuo didesnis, tuo saugiau
2. Viešojo rakto radimas – $n = p \times q$
3. Antrojo viešojo rakto radimas: e – natūralusis skaičius, kuris yra pirminis, $\varphi(n) = (p-1)(q-1)$, ir e būtų skaičius iš nelygybės $1 < e < \varphi(n)$ ir e būtų tarpusavyje pirminis su $\varphi(n)$.
4. Privataus rakto radimas: $d = e^{-1} \pmod{\varphi(n)}$.
5. Privatus raktas d , viešūs raktai n , e . Atlikę šiuos žingsnius, turime du viešuosius raktus n ir e , ir vieną privatų raktą d .

Skaičiai p ir q neturi būti niekada atskleisti, nes turint juos galima surasti privatųjį raktą.

3.1.2. Šifravimas

C – šifras, m – žinutė, E – šifravimo funkcija:

$$C = E(m) = m^e \pmod{n}$$

Pasinaudojus viešaisiais raktais e ir n norimoje žinutėje gauname šifruotą žinutę.

3.1.3. Dešifravimas

C – šifras, m – žinutė, D – dešifravimo funkcija:

$$m = D(C) = C^d \pmod{n}$$

Norint dešifruoti reikia turėti šifrą, viešąjį raktą n ir privatųjį raktą d .

3.2. RSA aklašis parašas

RSA aklašis parašas – tai skaitmeninio parašo forma, kurioje taikomi RSA šifravimo algoritmo principai siekiant užtikrinti ir autentiškumą ir privatumą. Šį metodą pristatė 1983 David Chaum kaip RSA šifravimo algoritmo išplėtimą [Dav83].

RSA aklas parašas leidžia pasirašančiam asmeniui pasirašyti pranešimą, nežinant jo turinio. Taip autentifikuotas asmuo gali sukurti validų parašą, nuo kurio jis yra nuasmenintas.

Čia žingsnis po žingsnio paaiškinama, kaip veikia RSA aklieji parašai:

Asmuo, kuriam reikia validaus parašo – prašytojas. Pasirašyti galintis asmuo, kuris pateikia viešiuosius raktus – pasirašantis asmuo. Prašytojas pasirenka maskavimo koeficientą r . Užmaskuoja savo žinutę m

$$m' = m \times r^e \pmod n \quad (1)$$

Kur e ir n yra viešieji raktai pateikti pasirašančio asmens. Prašytojas siunčia m' pasirašančiajam asmeniui.

Užmaskuoto pranešimo pasirašymas: pasirašantis asmuo gauna paslėptą pranešimą m' ir pasirašo jį naudodamas savo privatųjį raktą d

$$s' = (m')^d \pmod n \quad (2)$$

Ši operacija prilygsta RSA dešifravimui, svarbu nenaudoti tų pačių raktų duomenų saugojimui.

Pasirašymo atmaskavimas: prašytojas gauna pasirašytą maskuotą pranešimą s' ir jį atskleidžia, kad gautų tikrąjį parašą s , apskaičiuodamas

$$s = s' * r^{-1} \pmod n, \text{ kur } r^{-1} \text{ yra atvirkštinis modulis } r \text{ moduliui } n. \quad (3)$$

Patikrinimas: Prašytojas patikrina parašą s , naudodamas pasirašytojo viešąjį raktą (e, n) . Jis apskaičiuoja m

$$m = s^e \pmod n \quad (4)$$

Jei m sutampa su pradine žinute, žinutė yra sėkmingai pasirašyta. Ši operacija prilygsta RSA dešifravimui, ją galima atlikti be privačių raktų.

Rezultate turime pasirašytą reikšmę, kurios sistema negali susieti su konkrečiu naudotoju. Šią reikšmę ji gali validuoti ir patvirtinti, jog ją pasirašė tikrinančioji sistema [Dav83]. Balsavimo sistemos kontekste naudotojas, kuris autentifikavosi sistemai, susikūrė anonimišką prisijungimą į balsavimo dalį.

3.2.1. RSA aklojo parašo pavyzdys

Užmaskuojame žinutę – 4 su apakinimo reikšme $r = 2$, $n = 33$ ir $e = 7$. Skaičiai n ir e duoti, žinutė ir r pasirenkamieji.

$$\text{užmaskuotaŽinutė} \equiv \text{žinutė} \times r^e \pmod n \equiv 4 \times 2^7 \pmod{33} \equiv 29$$

Sistema pasirašo žinutę. Naudotojas pateikia užmaskuotos žinutės reikšmę – užmaskuotaŽinutė, kurią sistema pasirašo panaudojant privatų raktą $d = 3$.

$$\text{apakintasParašas} \equiv \text{užmaskuotaŽinutė}^d \pmod n \equiv 29^3 \pmod{33} \equiv 24$$

Parašo atskleidimas. Randame atvirkštinį modulį skaičiui r , kad galėtume atišti parašą.

$$\text{parašas} \equiv \text{apakintasParašas} \times r^{-1} \pmod{n} \equiv 24 \times 17 \pmod{33} \equiv 15$$

$$r^{-1} \equiv 17, \text{ nes } 2 \times 17 \pmod{33} \equiv 1$$

Parašo patikrinimas. Parašą galime validuoti pasinaudojant viešuoju raktu e .

$$\text{žinutė} \equiv \text{parašas}^e \pmod{n} \equiv 15^7 \pmod{33} \equiv 4$$

3.2.2. Aklujų parašų taikymas

Šiuo metu dažniausia naudojama skaitmeninių valiutų ir mokėjimo sistemų srityse. [RSA78][Dav83] Skaitmeninių valiutų sistemose aklieji parašai naudojami siekiant užtikrinti sandorių konfidencialumą. Kartu suteikti visuomenei galimybę patikrinti sandorių autentiškumą neatskleidžiant sandorio dalyvių tapatybės. Šis metodas padeda užkirsti kelią sukčiavimui ir užtikrina, kad sandorio nebūtų galima susieti su jame dalyvaujančiomis šalimis [BGENM16]. Šios sistemos kaip ir šaltinyje pristatyta naudojami kitomis aklujų parašų schemomis, ne RSA, bet šių schemų mintis, idėja išlieka ta pati kaip ir pristatyme RSA aklojo parašo veikime.

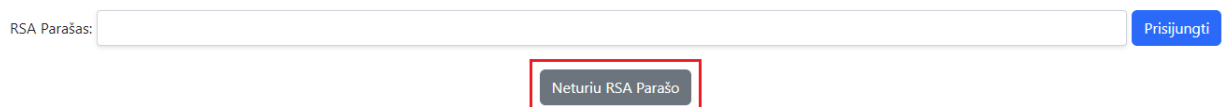
3.2.3. Saugumo aspektai

Nors RSA aklieji parašai užtikrina patikimą saugumą, jie nėra be spragų. RSA saugumas priklauso nuo rakto ilgio. Jei raktai per trumpi arba privatūs raktai nesaugiai valdomi, sistemos saugumas gali būti pažeistas. Be to, labai svarbus yra aklinimo veiksmo vientisumas, jį atskleidus galima susieti parašą su naudotoju. Įgyvendinant sistemą taip pat reikia atsižvelgti į galimas atakas, pavyzdžiui, pakartotines atakas, kai piktavališkai pakartotinai naudojamas senas parašas. [MOV01]

4. Autentifikavimas kūriamoje balsavimo sistemoje

Autentifikavimas: Tai procesas, kurio metu patikrinama, kas yra naudotojas. Jis apima asmens ar įrenginio, bandančio prisijungti prie sistemos, tapatybės patvirtinimą. Dažniausiai naudojami slaptažodžiai, biometriniai nuskaitymai ir saugumo žetonai. Autentiškumo nustatymas užtikrina, kad asmuo, teigiantis, jog turi prieigą, iš tiesų yra tas, kuo jis teigia esąs.

Pirmasis autentifikavimas bus prisijungiant į RSA aklojo parašo sistemą. Realiau atveju naudotojas prisijungtų per saugią sistemą, pvz., elektroninius valdžios vartus. Šiame darbe ši dalis realizuota primityviai. Prisijungimas į RSA aklojo parašo sistemą yra tiesiog mygtukas „Neturiu RSA Parašo“ 2 pav. Antrasis autentifikavimas bus pasinaudojant sukurtu RSA parašu, jau jungiantis į balsavimo dalį, kaip parodyta paveikslėlio viršuje.



2 pav. Prisijungimas į RSA aklojo parašo sistemą, balsuoti.deivis.dev

4.1. Kriptografinių raktų ilgiai

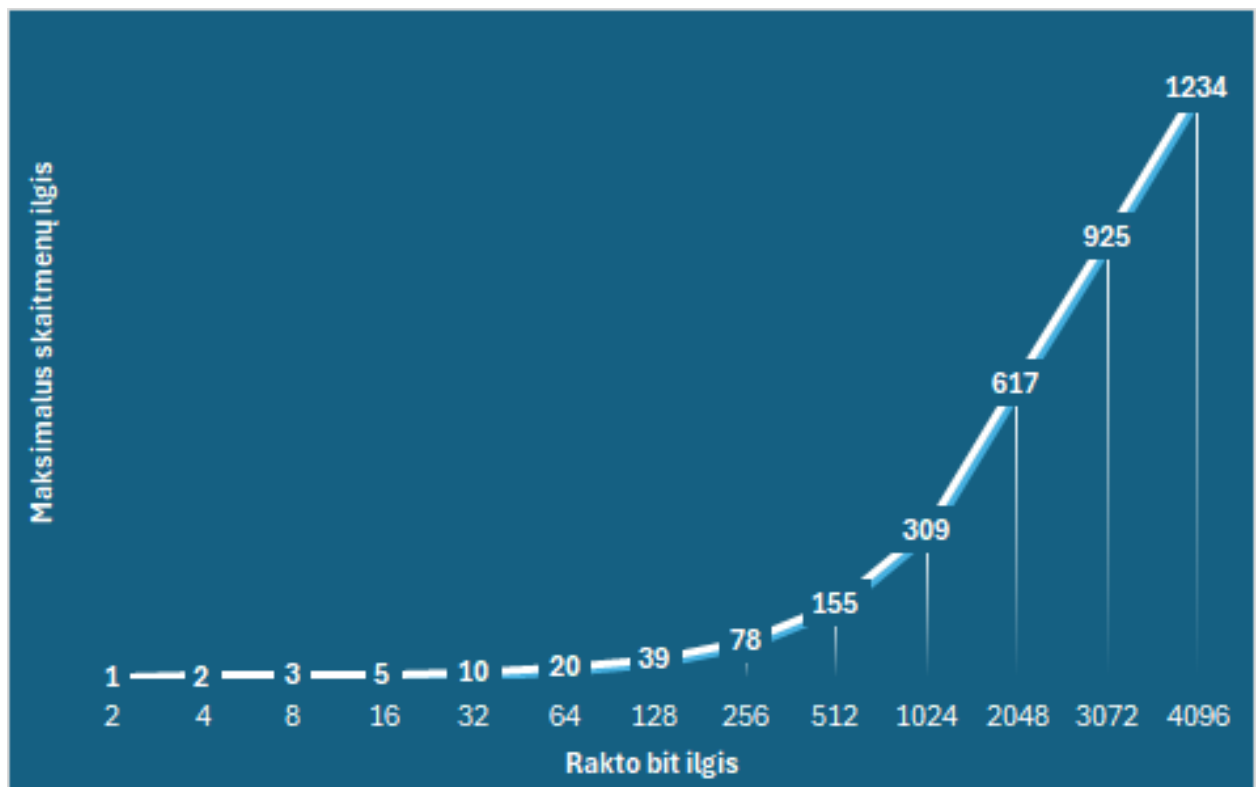
Rakto n ilgis yra tikrai nepakankamas norint turėti saugią implementaciją. Jis specialiai parinktas nedidelis su tikslu palengvinti skaičiavimus, jei asmuo juos atliktų individualiai – ne su sistemos pagalba.

Raktų ilgis kontroliuojamas programiškai skaičiaus, kuris prilygsta rakto bitų ilgiui, pagal konstantą kode. Demonstratinėje svetainėje šio parametro reikšmė 98, raktų p ir q ilgiai atitinkamai perpus mažesni 48 bitų, kadangi $n = p * q$.

Rekomenduojamas RSA raktų bitų ilgis. Naujausiose gairėse pateikiamos kelios įprastos rekomendacijos [BR19]:

1. 2048 bitų: Tai mažiausias rakto ilgis, kurį šiandien rekomenduojama naudoti daugeliu atvejų. Jis užtikrina gerą pusiausvyrą tarp saugumo ir našumo. Dauguma ekspertų mano, kad jo pakanka apsaugoti jautrius duomenis nuo dabartinių kriptografinių atakų.
2. 3072 bitai: Šis ilgis užtikrina stipresnį saugumo lygį ir dažnai rekomenduojamas aplinkoje, kurioje yra norima didesnio saugumo.
3. 4096 bitai: Kai reikia itin didelio saugumo bei našumo, reikalavimai nėra tokie svarbūs, rekomenduojama naudoti šį ilgį.

Nacionalinis standartų ir technologijų institutas (NIST)[BR19] pateikė gaires, kuriose nurodoma, kad 2048 bitų RSA raktai turėtų būti saugūs bent iki 2030 m. Tačiau, atsižvelgiant į kvantinės kompiuterijos pažangą, gali būti, kad ateityje RSA saugumas gali būti pažeistas. Aplinkoms, kurioms reikalingas ilgalaikis saugumas po 2030 m., vertėtų apsvarstyti alternatyvius kriptografijos metodus arba pasiruošti postkvantinei kriptografijai.



3 pav. Rakto skaitmenų priklausomybė nuo rakto ilgio

Turint skaičiaus ilgį bitais (dvejetainė sistema) n , šio skaičiaus skaitmenų ilgis dešimtainėje sistemoje bus d , šių skaičių sąryšis - $d = \lceil n \cdot \log_{10}(2) \rceil$, 2, 10 atspindi skaičių sistemą, operacija lubų, $\lceil \cdot \rceil$, pvz., $\lceil 1.01 \rceil = 2$. 3 pav. šio paveikslėlio rezultatai gauti pagal prieš tai pateiktą formulę.

Viso tai tikslas buvo pademonstruoti niuansą tarp saugumo ir praktiškumo. Tikrai būtų nepraktiška atlikti skaičiavimus su 500+ simbolių skaičiais paprastam žmogui, ypač kelti laipsniu tokiais masyviais skaičiais.

4.2. Žinutės apribojimai

Parašas gali būti bet kuris atsitiktinis skaičius, tad svarbu nustatyti leistinų žinučių rėžį. Neparinke režio visi sveikieji skaičiai būtų validūs parašai. Žinučių aibė būtų n , kaip praeitame skyrelyje minėta šis skaičius turėtų būti pakankamai didelis norint turėti užtikrintą saugumą bei patikimumą. Tarkime $n = 149250455940414997539164166942$, iš galimų parašų leistini tik 899999, žinutės intervalas $[100000 - 999999]$, tikimybė, jog atsitiktinis sveikasis skaičius tiks - $\frac{899999}{149250455940414997539164166942} \approx 6.02922284 \times 10^{-15}$. Padidinus rakto n ilgį tikimybė sparčiai mažėtų.

4.3. Sukurtos RSA aklojo parašo svetainės dalis

Paspaudus „Neturiu RSA Parašo“ 2 pav. mygtuką atsiveria RSA aklojo parašo dalis 4 pav.

RSA Parašas: Prisijungti

Neturiu RSA Parašo

Koks tikslas RSA aklojo parašo?
RSA aklojo parašo schema su pavyzdžiu
Kas yra maskavimo koeficientas?
Kurių skaičių negaliu atlikti be sistemos pagalbos?

RSA aklojo parašo autentifikacijai sukūrimo forma

RSA viešasis raktas n
RSA viešasis raktas e

Pasirinkite žinutę m, iš intervalo [100000 ; 999999]:

Pasirinkite apakinimo koeficientą r:

$\text{maskuotaŽinutė} = \text{Žinutė} * r^e \pmod{n} = \text{Žinutė} * r^{123776043302206128138451647791} \pmod{113263162018876249623280797017}$

$\text{apakintasParašas} = \text{apakintaŽinutė}^d \pmod{n} = \text{apakintaŽinutė}^d \pmod{113263162018876249623280797017}$

$r^{-1} = r^{-1} \pmod{113263162018876249623280797017}$

$\text{parašas} = \text{apakintasParašas} * r^{-1} \pmod{n} = \text{apakintasParašas} * r^{-1} \pmod{113263162018876249623280797017}$

4 pav. RSA aklojo parašo schema, balsuoti.deivis.dev

Šiame lange užmaskuota Žinutė naudotojas gali susipažinti su RSA aklojo parašo schema. Kad naudotojui šis procesas būtų lengvesnis, yra paruošti paaiškinimai. Visi skaičiavimai atvaizduoti vizualiai, keičiasi dinamiškai nuo įvesčių.

RSA Parašas: Prisijungti

Neturiu RSA Parašo

Koks tikslas RSA aklojo parašo?
RSA aklojo parašo schema su pavyzdžiu
Kas yra maskavimo koeficientas?
Kurių skaičių negaliu atlikti be sistemos pagalbos?

RSA aklojo parašo autentifikacijai sukūrimo forma

RSA viešasis raktas n
RSA viešasis raktas e

Pasirinkite žinutę m, iš intervalo [100000 ; 999999]:

Pasirinkite apakinimo koeficientą r:

$\text{maskuotaŽinutė} = \text{Žinutė} * r^e \pmod{n} = 202400 * 1357123^{123776043302206128138451647791} \pmod{113263162018876249623280797017}$

$\text{apakintasParašas} = \text{apakintaŽinutė}^d \pmod{n} = 36507986841746989317149238756^d \pmod{113263162018876249623280797017}$

$r^{-1} = 1357123^{-1} \pmod{113263162018876249623280797017}$

$\text{nes } 1357123 * 57018618819469005290143794808 \pmod{113263162018876249623280797017} = 1$

$\text{parašas} = \text{apakintasParašas} * r^{-1} \pmod{n} = 88228678153589511037875056854 * 57018618819469005290143794808 \pmod{113263162018876249623280797017}$

5 pav. RSA aklojo parašo sukūrimas, balsuoti.deivis.dev

5 pav. pavaizduoja įvykdyta RSA aklojo parašo sukūrimą. Galime pravaliduoti rezultatą ar jis tikrai teisingas, $7735766357993329299625455745^{97899735921113979898417932263} \pmod{149250455940414997539164166943} \equiv 202400$, kas sutampa su pirmine žinute.

Pasiektas rezultatas – sukurtas validus RSA parašas. Viso to dėka yra užtikrintas naudotojo anonimiškumas RSA aklojo parašo pagalba.

4.3.1. Parašo susėjimas su naudotoju

Sistema šio parašo paprastai niekaip nesusieks su konkrečiu naudotoju, nes jai reiktų išspręsti šią lygtį. $r^{-1} = \text{apakintasParašas}^{-1} \times \text{parašas} \pmod{n}$, $r^{-1} = \text{apakintasParašas}^{-1} \times 7735766357993329299625455745 \pmod{149250455940414997539164166943}$

apakintasParašas, bus kažkuris iš visų išduotų parašų sistemos, jei naudotojų 1000, tai kažkuris iš jų yra teisingas šiai lygčiai. Jei žinotume kuris, nebūtų prasmės spręsti lygties, tai turime pusiau du kintamuosius lygtyje. Net žinodami kuris parašas tinka rasti r^{-1} iš šios lygties su pakankamai dideliu n , reikalautų nerealaus laiko. Dar viena problema iškiltų, jei keli naudotojai pasirinko tą pačią žinutę, tokiu atveju niekaip nebūtų įmanoma nustatyti originalios kilmės.

4.4. Tos pačios žinutės problema

Dviem naudotojams turint vienodus parašus iškyla problema. Kadangi šis parašas yra vienkartinis, su juo sistemoje galima prabalsuoti vieną kartą. Todėl antras naudotojas, kuris naudosis šiuo parašu, tiesiog negaus galimybės balsuoti. Šios problemos sprendimo nenagrinėsime.

Galime pabandyti įvertinti tokios problemos iškilimo tikimybę. Imame skaičius iš puslapio, leistinos žinutės iš $[100000; 999999]$. Leistas žinučių intervalą sudaro $999999 - 100000 = 899999$ skirtingos žinutės.

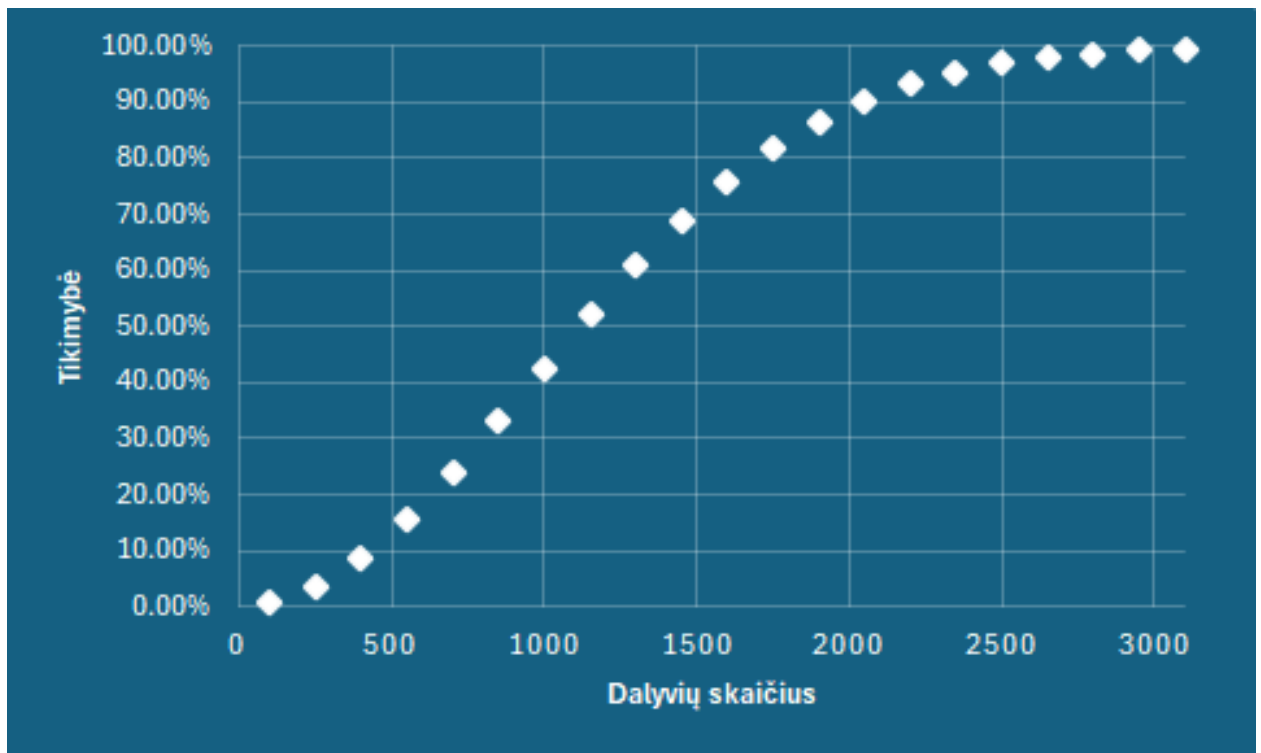
4.4.1. Gimtadienių ataka

Nors atrodo elementaru būtų apskaičiuoti, iš tiesų tikimybę yra sukurta ir apima gimtadienių ataką. „Taip yra todėl, kad ataka siejama su žinomu elementarios tikimybių teorijos uždaviniu: kiek mažiausiai žmonių turi susirinkti, kad įvykio, jog atsiras gimusių tą pačią metų dieną, tikimybė būtų didesnė už $1/2$? Beveik visi, bandantys atsakyti vadovaudamiesi „sveiku protu“, apsigauja. Atsakymas – 23“, puikiai pristato šią problemą Vilius Stakėnas „Kodai ir šifrai“ [Sta07].

$P(n)$, kad bent du žmonės iš n parinks tą pačią žinutę, iš leistinių žinučių N , galima užrašyti taip:

$$P(n) = 1 - \prod_{k=0}^{n-1} \left(1 - \frac{k}{N}\right)$$

Ateimame iš vieneto, nes skaičiuojame priešingą tikimybę. O $1 - \frac{k}{N}$ reiškia tai, kad $k + 1$ asmuo neturės bendro parašo su prieš tai buvusiais dalyviais k . Pagal šią formulę galime sudaryti diagramą 6 pav.



6 pav. Gimtadienių ataka, kai galimų reikšmių 899999.

Pagal gautus duomenis galime padaryti išvadas:

- Maža tikimybė, kai dalyvių skaičius yra mažas, 10 dalyvių – 0,005%, 100 – 0,55%. Jei balsuotojų skaičius apribotas šiame intervale, tikimybė, jog du balsuotojai susikurs tą patį parašą yra gan maža.
- Vidutinė vienodų parašų tikimybė, nuo 1000 dalyvių pasiekia 42,60%, pakilus iki 1600 dalyvių – 75,88%, 2800 – 98,72%. Daryti balsavimą su tiek dalyvių būtų neprotinga. Reikėtų padidinti galimų reikšmių aibę ir atitinkamai prailginti RSA rakto n ilgį.

Visa tai būtų tiesa, jei balsuotojai būtų robotai. Realybėje tikėtis, jog dalyviai pasirinks žinutes atsitiktinai negalime. Šiuos sunkumus galime sumažinti paskelbę jau panaudotų raktų sąrašą, bet to sprendimu vadinti negalime. Visi jau panaudoti raktai kaip ir balsai bus viešai matomi balsavimo metu, atskleidimas neįtakoja saugumo.

5. Paillier kriptoschema

Šis skyrelis nagrinės Paillier kriptoschemą, kaip ją pasinaudoti. Paillier kritosistema leidžia surengti balsavimą. Šiame balsavime visi užšifruoti balsai gali būti viešūs. Balsų viešumas leidžia patikrinti balsavimo rezultatą, nes jis yra gautas „sudėjus“ visus balsus, tad reikalaus vienos dešifravimo operacijos. Balsų sudėjimas yra vykdomas pagal Paillier šifro homomorfinę savybę, šifrų daugyba – prilygsta sudėčiai žinučių prieš užšifravimą.[MA18] [AAZS20]

5.1. Paillier rakto kūrimo žingsniai

- Dviejų pirminių skaičių parinkimas p ir q , kuo didesnius naudosime, tuo saugiau;
- Viešojo rakto radimas – $n = p \cdot q$;
- Lambda radimas, $\lambda = \text{MKD}(p-1, q-1)$
- Skaičiaus g radimas, g yra multiplikatyvinės grupės $\mathbb{F}_{n^2}^*$ elementas bei išpildo sąlygą $\text{DBD}(L(g^\lambda \bmod n^2)) = 1$, kur funkcija $L(u) = \frac{u-1}{n}$ ir $\text{DBD}(L(g^\lambda \bmod n^2), n) = 1$, g priklauso multiplikatyvinei grupei $\mathbb{F}_{n^2}^*$, kai $\text{DBD}(g, n^2) = 1$;
- Privatus rakto μ radimas $(L(g^\lambda \bmod n^2))^{-1} \bmod n$.

Atlikę šiuos žingsnius turime du viešuosius raktus – n ir g , ir vieną privatų raktą μ . Skaičių p , q arba λ šykštu negalima atskleisti, nes turint juos galima surasti privatųjį raktą.

5.2. Šifravimas

C – šifras, m – žinutė, E – šifravimo funkcija, r – privatus raktas tik šiam šifrui. Pirmiausiai reikia pasirinkti raktą r taip, kad r priklausytų multiplikatyvinei grupei – $r \in \mathbb{F}_n^*$, šio rakto patartina nepernaudoti, visada geriausia rinktis naują r .

$$C = E(m)$$
$$E = g^m \cdot r^n \bmod n^2$$

Pasinaudojus viešaisiais raktais n ir g bei pasirinkus r , galime šifruoti žinutę m ir gauti šifrą C .

5.3. Dešifravimas

C – šifras, m – žinutė, D – dešifravimo funkcija, μ – privatus raktas

$$m = D(C)$$
$$D = L(C^\lambda \bmod n^2) \cdot \mu \bmod n$$

5.4. Algoritmo pavyzdys su skaičiais

Pasirenkame pirminius skaičius p ir q – $p = 42793$, $q = 60937$

Paskaičiuojame $n = p \cdot q = 42793 \cdot 60937 = 2607677041$

Randomame λ

$$\lambda = \text{MBK}(p-1, q-1) = \text{MBK}(42792, 60936) = 108648888$$

Pasirenkame $g = 20720$ atitinka abi salygas.

Randame μ

$$\mu = (L(g^\lambda \bmod n^2) - 1 \bmod n) = (L(20720^{108648888} \bmod 2607677041^2) - 1 \bmod 2607677041) = 2151111039$$

Turime sukūrę viešuosius raktus $n = 2607677041$, $g = 20720$ bei privatųjį raktą $\mu = 2151111039$.

Galime užšifruoti žinutę $m = 1000001$

Mums reikia pasirinkti r , kad priklausytų multiplikatyvinei grupei \mathbb{F}_n^* , $r = 21817$ Galime pradėti šifruoti

$$E = g^m r^n \bmod n^2 = 20720^{1000001} \cdot 21817^{2607677041} \bmod 2607677041^2 = 3921753424997215$$

Dešifruojame

$$D(C) = (L(C^\lambda \bmod n^2) \cdot \mu \bmod n) = (L(3921753424997215^{108648888} \bmod 2607677041^2) \cdot 2151111039 \bmod 2607677041) = 1000001$$

užšifruojame dar vieną žinutę $m = 1010000$, $r = 44653$

$$E = g^m r^n \bmod n^2 = 20720^{1010000} \cdot 44653^{2607677041} \bmod 2607677041^2 = 4559655622723153676$$

Patikriname homomorfizmą.

$$m_1 + m_2 = D(E_1 * E_2)$$

$$1000001 + 1010000 = D(4559655622723153676 * 3921753424997215)$$

$$2010001 = L(4563577376148150891^{108648888} \bmod 2607677041^2) \cdot 2151111039 \bmod 2607677041$$

$$2010001 = 2010001$$

5.5. Damgård–Jurik plėtinys

Damgård–Juriko kriptosistema yra Paillier kriptosistemos plėtinys, siūlantis lankstesnę šifravimo schemą su reguliuojamais saugumo lygiais. Šis plėtinys suteikia galimybę rasti r žinant dešifravimo raktą. Norint galėti išskaičiuoti r , privalome parinkti $g = n + 1$.

Norint rasti žinutės m , privatųjį raktą r iš šifro C .

$$C = (1 + n)^m \cdot r^n \bmod n^2$$

Žinome privatųjį dešifravimo raktą $\lambda(n)$. Visa tai galime atlikti rasdami C' užšifruotą žinutę su reikšme 0. m – balsavimo rezultatas.

$$C' = C \cdot (1 - m \cdot n) \bmod n^2$$

$$M = n^{-1} \mod \lambda(n)$$

$$r = C'^M \mod n$$

Visa tai veikia nes,

$$C'^M = r^{n \cdot M} = r^{1+k \cdot \lambda(n)} = r \cdot (r^{\lambda(n)})^k = r \mod n, \text{ nes } \mathbb{Z}_n^* \text{ yra } \lambda(n)$$

[DJ00]

5.6. Žinutės formatas balsavimo sistemai

Norint sėkmingai pasinaudoti Paillier kriptosistema svarbu gerai apgalvoti koks bus žinutės formatas. Pasirinkus netinkamą formatą gali iškilti problemos. Norint jų išvengti turime apgalvoti kelis dalykus:

- Homomorfinės savybės panaudojimas, ši savybė būtina. Akivaizdu, jog balsavimui pasirinkus Paillier kriptosistemą, kuri dešifravimą atliks vieną kartą. Svarbu atsisžvelgti, jog balsai yra tinkami homomorfizmo pritaikymui.
- Dalyvių skaičius, plečiamumas. Pasirinktas formatas turi galėti palaikyti numatytą balsuotojų skaičių. Visi formatai turi limitacijas, kadangi dirbame su apibrėžto dydžio skaičiais.
- Naudojamumas, aiškumas. Formatas turi būti kuo paprastesnis, suprantamesnis dalyviui. Kaip vėliau bus pristatyta, balsuotojas galės pats apskaičiuoti savo balsą, tad svarbu, jog tikrai suprastų, ką jis pasirenka.

Primityviausias efektyviausias būdas būtų pasinaudoti skaičių sistemos ypatybėmis. Skaičiavimo sistema gali atspindėti balsuotojų skaičių, pvz., jei balsavimo procese dalyvautų 9 balsuotojai, galime pasinaudoti dešimtaine skaičiavimo sistema, balsuotojų skaičius + 1. Jei pasirinkimų yra 5, tai mūsų formatą sudarytų 5 skaičiai, kiekvienas skaičius atspindėtų pasirinkimą. Kaip tai atrodytų atspindi 7 pav., kiekviena balso dalis būtų dešimtainis skaičius (0-9), kuris atspindėtų pasirinkimą.



7 pav. Balso žinutės formatas, 5 pasirinkimai A-E

Apribojame tai, jog galime balsuoti tik už vieną kandidatą, tad visi balsai būtų atitinkamai tokie A - 10000, B - 01000, C - 00100, D - 00010, E - 00001. Jei 3 balsai už A, 2 už B, 0 už C, 4 už D, 1 už E, galutinis rezultatas būtų 32041 nes $10000 \cdot 3 + 01000 \cdot 1 + 00100 \cdot 0 + 00010 \cdot 4 + 00001 \cdot 1 = 31041$. Jei būtume leidę balsuoti daugiau dalyvių, nei leidžia formatas, iškiltų problema, pvz., leidžiame balsuoti 13 dalyvių, balsai tokie A - 1, B - 10, C - 1, D - 1, E - 0, $10000 \cdot 0 + 01000 \cdot 10 + 00100 \cdot 1 + 00010 \cdot 1 + 00001 \cdot 0 = 20110$, balsavimą laimėjo A?

Įvyko skaičiaus persokimas, kadangi formatas negalėjo palaikyti tiek balsuotojų. Jei būtume šį balsavimą atlikę su šešioliktaine sistema, rezultatas atrodytų taip 1A110, A šešioliktainėje sistemoje reiškia – 10 dešimtainėje. Balso žinutės užtat irgi keičiasi, jei žinutė už A buvo 10000, dabar ji bus 65536 dešimtainėje sistemoje, nebent visus skaičiavimus atliksime šešioliktainėje sistemoje, tada galėtume palikti 10000. Šis formatas tikriausiai būtų optimaliausias matematiškai, bet jis tiesiog nepraktiškas, gali būti sunkiai suprantamas.

Apibendrinant šį formatą, jis tikrai yra tinkamas homomorfizmui, jei dydis yra parinktas teisingai balsuotojų atžvilgiu. Jis yra lengvai plečiamas irgi, jei 999 balsuotojų galime naudoti 1000 skaičių sistemą, aišku iškiltų simbolių parinkimo problema kiekvienam skaičiui, bet tai yra teoriškai įmanoma ir rezultatą galime pateikti po konvertacijos. Naudojamumas su šiuo formatu būtų sunkus, balsuotojo atžvilgiu sistema turėtų būti kuo paprastesnė, suprantamesnė, šitas formatas tikrai nėra elementarus, paprastam žmogui. Apžvelkime kitokį formatą, šis formatas labai panašus į prieš

Balsų skaitliukas	A	B	C	D	E	Balso žinutė
1	01	00	00	00	00	10100000000 – už A
1	00	01	00	00	00	10001000000 – už B
1	00	00	01	00	00	10000010000 – už C
1	00	00	00	01	00	10000000100 – už D
1	00	00	00	00	01	10000000001 – už E

1 lentelė. Balso formatas

tai aptartą, tik jis nežaidžia su skaičių sistemomis, o išlaiko dešimtainę sistemą. Jo idėja visada naudoti žmonėms įprastą dešimtainę sistemą, tik plėsti pasirinkimo skiltį, kad ji atitiktų balsuotojų skaičių. Jei balsuotojų yra iki 99, tai ir kiekvieno pasirinkimo maksimali reikšmė turi būti 99, kas reikalauja dviejų skaičių tai ir matome lentelėje 1. Jei balsuotojų 145, reikalaus 3 simbolių ir palaikys maksimalų 999 balsų skaičių už vieną kandidatą. Balsų skaitliukas yra kaip kontrolinė suma, pagal kurią matosi, kiek iš viso balsų buvo priimta. Jis pradžioje gali užimti tik vieną skaičių, nes jo augimas nėra problema, žinutės maksimali reikšmė yra $n - 1$, jei n sudaro 500+ simbolių, tai problemos tikrai nesukels.

Sukurtas formatas palaiko homomorfizmą. Lengvai plečiamas. Jei rakto bit ilgis yra 4096, n ilgis bus iki 1234 ilgio. Dėl paprastumo tegul rinkėjai renkasi iš 9 kandidatų, tai maksimalus balsuotojų skaičius būtų $\frac{1234}{9+1(\text{balso skaitliukas})} = 123$ simbolių dalyvių, jei pasaulyje gyvena 8 milijardai gyventojų, kas yra 7 simbolių skaičius, visi jie laisvai galėtų prabalsuoti. Naudojamumas, viskas lieka dešimtainėje, tad nereikia papildomo konteksto, žinutės lengvai atpažįstamos, jų sumavimas irgi turėtų būti aiškus.

6. Įrodymai, nesuteikiantys žinių

Įrodymai, nesuteikiantys žinių (angl. „Zero-knowledge proof“), leidžia įrodyti apie žinių turėjimą jų neatskleidus. Protokole dalyvauja įrodinėtojas, kuris siekia kažką įrodyti tikrintojui. Tikrintojas kaip asmens gali ir nebūti – tokiu atveju protokolas tampa neinteraktyviu ir būtų naudojama hash funkcija arba kitos prilygstančios struktūros.[Jah19]

Standartinė žinių nesuteikiančio įrodymo eiga:

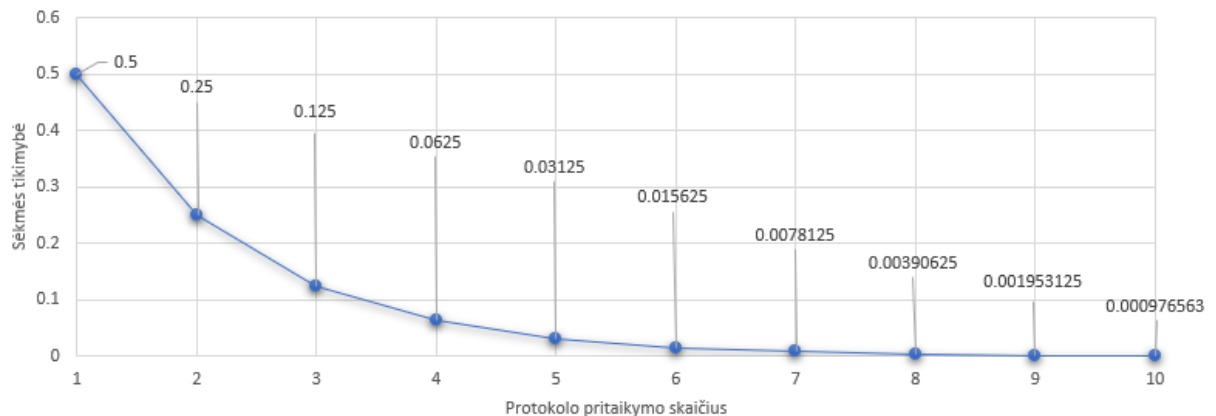
1. Įrodinėtojas nori įrodyti tikrintojui savo teiginio teisingumą neatskleidus jokių žinių;
2. Pasižadėjimo žingsnis. Įrodinėtojas sukuria pasižadėjimą, kuris yra susijęs su įrodoma žinia. Nežinant žinios įrodinėtojas nesugebės sukurti tokio pasižadėjimo, kuris atitiktų visas galimas tikrinimo sąlygas vienu metu. Šiame žingsnyje nesąžiningui įrodinėtojui geriausia strategija būtų atspėti koks iššūkis bus pateiktas ir pagal tai kurti pasižadėjimą;
3. Iššūkio perdavimo žingsnis. Tikrintojas parenka iššūkį, jei protokolas interaktyvus. Iššūkiai priklauso nuo vykdomo protokolo. Pvz., jei protokolas turi du galimus iššūkius, tikrintojas gali vienos iteracijos metu iškelti tik vieną iš jų. Jei tikrintojas paprašytų abiejų rezultatų su skirtingais iššūkiais vienoje iteracijoje, labai tikėtina, kad būtų atskleista informacija, ko mes vengiame. Jei galimi tik du iššūkiai, sukčiaujantis įrodinėtojas tikriausiai galės lengvai sukurti teisingą įrodymą vienam iš jų, bet ne abiem vienu metu, kitaip protokolas būtų beprasmis, kaip tai nėra problema nagrinėsime vėlesniame skyrelyje. Iššūkis gali būti atsitiktinis skaičius, kurį įrodinėtojas turės naudoti protokolo skaičiavimuose;
4. Pasižadėjimo apgynimo žingsnis. Įrodinėtojas, gavęs iššūkį, apskaičiuoja ir pateikia gautą rezultatą tikrintojui. Jei 2 žingsnelyje pasižadėjimas formuojamas nežinant teisingo teiginio, kitaip tariant – „neteisingas“ ir nebuvo teisingai nuspėtas tikrintojas iššūkis – įrodinėtojas niekaip nesugebės pateikti teisingo rezultato, kuris atitiktų abu – pasižadėjimą ir iššūkį;
5. Įrodymo priėmimo žingsnis. Tikrintojas patikrina, ar rezultatas gautas naudojant jo pateiktą iššūkį bei ar 2 žingsnyje pateiktas pasižadėjimas irgi dera su 4 žingsnelyje gautu rezultatu. Protokolas kartojamas tol, kol bus pasiektas protokolo rekomendacinis iteracijų kiekis. Tokiu atveju tikrintojas priims įrodymą. Jei įrodinėtojas pateiks bent kartą klaidingą įrodymą – įrodymas negali būti priimtas ir tai reikštų, jog įrodinėtojas neteisingai laikėsi protokolo, nors iš tiesų žino teisingą teiginį, arba jis bandė sukčiauti ir įrodyti nežinant teisingo teiginio.

Žinių nesuteikiantis įrodymas yra labai galingas įrankis kriptografijoje. Kuris leidžia įrodyti paslapties žinojimą jos neatskleidžiant. Įrodymai, nesuteikiantys žinių, pasižymi šiomis savybėmis pagal [Jah19] darbą:

- Pilnumas: jei įrodomas teiginys yra tiesa bei protokolas, kuriuo bandoma įrodyti taip pat teisingas, tikrintojas turi sutikti su rezultatu.
- Pagrįstumas: jei teiginys yra klaidingas, įrodinėtojas, besilaikantis protokolo, nesugebės suteikti sėkmingų įrodymų tikrintojui su realia tikimybe tai atliekant.

- Nesuteikimas žinių: tikrintojas po įrodymo nesužino nieko apie patį teiginį, tik sužino, kad įrodinėtojas žino teisingą teiginį.

Žinių nesuteikiantis įrodymas niekada neužtikrina, jog įrodinėtojas šimtu procentu turėjo teisingą teiginį pagal [ZFZS20], [Jah19] darbus bei pačią žinių nesuteikiančio įrodymo idėją. Ši tikimybė niekada nepasieks šimto procentų užtikrinimo, kadangi visada bus galima „atspėti“. Pvz., turime interaktyvų protokolą, kuris susideda iš dviejų dalių – A ir B – iš kurių tikrintojas per iteraciją gali tikrinti tik vieną. Darom prielaidą, jog įrodinėtojas lengvai gali sukurti teisingą įrodymą kažkuriai iš dalių, bet ne abiem kartu. Tikrintojas kiekvieną kartą renkasi kurią dalį tikrins, nes protokolas neleidžia patikrinti abiejų vienoje iteracijoje. Vadinasi, įrodinėtojas turi vieną iš dviejų tikimybę sėkmingai pereiti vieną protokolo iteraciją atspėdamas, ar bus tikrinama A ar B dalis. 8 pav. demonstruoja, kaip sparčiai mažėja tikimybė atspėti daug kartų iš eilės kurią dalį rinksis tikrintojas, akivaizdu, kokia maža tikimybė yra jau po 10 kartų. Tikrintojas gali prašyti įrodymo 100 ar 1000 kartų, tačiau ir po tiek kartų tikimybė išliks ne nulinė, jog įrodinėtoju pasisekė įrodyti nežinant teisingo teiginio. Todėl šiais įrodymais niekada nepasieksime šimto procentų tikimybės, jog tikrai žinome teiginį.



8 pav. Teisingo spėjimo ir tikimybės grafikas

6.1. Žinių nesuteikiantis įrodymas diskretui logaritmui

Šis skyrelis buvo rašytas remiantis [Sta07] bei [CPSS11] darbais. Jonas nori įrodyti Tomui, jog žino diskretaus logaritmo sprendinį x .

$$y \equiv g^x \pmod{n} \quad (5)$$

Skaičiai y , g , n yra žinomi abiems. Su dideliais skaičiais išspręsti diskretų logaritmą yra labai sunku, nes tai reikalauja polinominio laiko.

Protokolas:

1. Jonas pasirenka atsitiktinį skaičių r ir apskaičiuoja C , kurį siųs Tomui:

$$C = g^r \pmod{n} \quad (6)$$

Šis žingsnis yra pasižadėjimo žingsnis. Jei šiame žingsnyje Jonas nežino x ir nori sėkmingai įrodyti Tomui, jam teks nuspėti Tomo iššūkį. Žinant iššūkį, galės parinkti palankų C .

2. Tomas pateikia iššūkį z . Nagrinėsime paprastesnę protokolo implementaciją, kurioje z gali būti 1 arba 0. Leidžiant z būti iš didesnės skaičių aibės nei du (0 ir 1), protokolas tampa sudėtingesnis, keičiasi skaičiavimai, pasiekiamas tas pats rezultatas.
3. Jonas apskaičiuoja s ir perduoda Tomui,

$$s = r + xz \pmod{p-1} \quad (7)$$

z yra Tomo 2 žingsnelyje pateiktas skaičius.

4. Validacijos žingsnis. Tomas šiame žingsnyje žino visus kintamuosius y , g , n iš pradinės sąlygos, C ir s pateikti Jono, z sugalvojo pats antrame žingsnyje.

$$g^s \equiv C \cdot y^z \pmod{n} \quad (8)$$

Tomas gali apskaičiuoti abi lygybės puses ir palyginti ar jos sutampa. Jei jos sutampa, reiškia, jog Jonas žino sprendinį x arba sėkmingai nuspėjo skaičių z ir taip pasirinko tinkamus C bei s . Jei lygybės pusės nesutampa, reiškia, jog Jonas neteisingai laikėsi protokolo arba nežino x .

6.1.1. Galimi bandymai sukčiauti

1. Jonas spėja, kad $z = 0$. Jonui užtenka pateikti $s = r$. Tai tinka, nes (5) sąlyga tampa, $g^s \equiv g^r \pmod{n}$, nes $C \cdot y^0 \pmod{n} = g^r \cdot 1 \pmod{n}$.
2. Jonas spėja, kad bus $z = 1$. Jonui pateikus neprotingai parinktą r ir nežinant x , tektų išspręsti diskretų logaritmą $g^s \equiv C \cdot y \pmod{n}$. Jei tai atlikti būtų paprasta, jam nereiktų sukčiauti. Todėl Jonas gali pradėti nuo kito gal – pradžia pasirinkti atsitiktinį s ir apskaičiuoti $C \equiv g^s \cdot y^{-1} \pmod{n}$. Įrodymas bus tinkamas, nes jis turės derančius s ir C . Viskas gerai iki tol, kol Tomas nepateikia $z = 0$. Tuomet Jonas turi išspręsti diskretų logaritmą norint rasti tinkamą r , $C^r \equiv y \pmod{n}$.

Sėkmingai apgauti pavyks su tikimybe $1/2$. Jei protokolą taikysime daug kartų, tikimybė, jog sukčiaujantis Jonas bus nepagautas, eksponenetiškai greitai mažėja, kaip matome 8 pav.

6.1.2. Pavyzdys su realiais skaičiais

Jonas nori įrodyti Tomui, jog žino diskretaus logaritmo sprendinį, neatskleidus pačio sprendinio.

$$n = 101 \cdot 103 = 10403, g = 2023, x = 2024 \text{ (nėra žinomas Tomui)}, y = 3774$$

$$y \equiv g^x \pmod{n} \quad 3774 \equiv 2023^{2024} \pmod{10403}$$

1. Jonas pasirenka $r = 10$ ir apskaičiuoja C pagal (2) lygtį. $C = g^r \bmod n = 2023^{10} \bmod 10403 = 6832$ gautas C perduodamas Tomui;
2. Tomas gavęs C jį išsisaugo ir nusiunčia $z = 1$;
3. Jonas apskaičiuoja C pagal (3) lygtį. $s = r + xz \bmod (p - 1) = 10 + 2024 \cdot 1 \bmod (10403 - 1) = 2035$, gautas s perduodamas Tomui;
4. Tomas atliks validaciją pagal (4) lygtį,

$$\begin{aligned} g^s \bmod n &= C \cdot y^z \bmod n \\ 2023^{2035} \bmod 10403 &= 6832 \cdot 3774^1 \bmod 10403 \\ 5334 &= 5334 \end{aligned}$$

Jei Tomas būtų pateikęs $z = 0$, Jonas pateiktų $s = r + xz \bmod (p - 1) = 10 + x \cdot 0 \bmod 10402 = 10$.

Patikrinimas

$$\begin{aligned} g^s \bmod n &= C \cdot y^z \bmod n \\ 2023^{10} \bmod 10403 &= 6832 \cdot 3774^1 \bmod 10403 \\ 6832 &= 6832 \end{aligned}$$

Tomas negali prašyti abiejų s reikšmių su $z = 0$ ir $z = 1$ tam pačiam C . Tokiu atveju įrodymas suteiktų žinių nes Tomas galėtų surasti x . Tuo pačiu Tomas žinotų, jog Jonas tikrai žino x . Tai pamatėme iš sukčiavimo pavyzdžių, jog sukurti įrodymą abiem atvejams, kai $z = 0$ ir $z = 1$ yra beveik neįmanoma nežinant x .

6.1.3. Rezultatas

Jei tikrintojas ir įrodinėtojas pilnai laikėsi protokolo bei buvo taikytas pakankamai kartų, tikrintojas turi priimti įrodymą. Tikrintojas žinos, jog įrodinėtojas žino x , tuo pačiu tikrintojas neįgijo jokių naujų žinių, kurios padėtų nustatyti x reikšmę.

6.2. Interaktyvūs įrodymai

Interaktyvių įrodymų kriptografiniai protokolai pasižymi tuo, kad įrodyme dalyvauja dvi šalys – tikrintojas ir įrodinėtojas. Pavadinimas atskleidžia, kad šie dalyviai dalyvauja interaktyviame procese, jų tikslas nustatyti teiginio teisingumą. Prieš tai aptariamas protokolai diskrečio logaritmo sprendinio įrodymui buvo interaktyvūs.

6.3. Neinteraktyvūs įrodymai

Neinteraktyvių įrodymų kriptografiniai protokolai pasižymi tuo, kad įrodymas gali būti sukurtas iš anksto be tikrintojo pagalbos. Būtinasis žingsnis interaktyviame protokole – iššūkio sukūrimas – gali būti pakeistas veiksmu, kurį gali atlikti pats įrodinėtojas.

6.3.1. Kaip generuoti iššūkį neinteraktyviuose įrodymuose

Akivaizdu, kad negalime leisti ar pasitikėti, jog įrodinėtojas pasirinktą iššūkį ar nebandys sukčiauti. Todėl iššūkio generacijai galime naudoti hash funkciją. Iššūkį generuojame pasinaudojant hash funkcija, į kurią paduodame pasižadėjimą bei kitus parametrus. Gautas funkcijos rezultatas bus iššūkis. Didžiąją dalį interaktyvių protokolų galime paversti į neinteraktyvius. Dažnu atveju protokolo skaičiavimai yra šiek tiek koreguojami, kad būtų užtikrintas protokolo saugumas bei patikimumas. Hash funkcijos panaudojimas verčiant interaktyvų protokolą į neinteraktyvų yra atrastas bei vadinamas Fiat–Shamir euristika nuo Izraelio matematikų Amos Fiat bei Adi Shamir.

6.3.2. Hash funkcija

Hash funkcija yra tokia funkcija, kuri paima duomenis ir paverčia į fiksuoto dydžio simbolių eilutę. Hash funkcijos rezultatas vadinamas hash kodu. Funkcija yra vienakryptė bei rezultatas yra nenuspėjamas. Vienakryptė reiškia, kad nors ir žnodami rezultatą bei naudotą funkciją, užtruksime begalę daug laiko ieškodami originalios reikšmės. Pavyzdžiui, naudojant SHA-256 hash funkciją ir žinant rezultatą, optimaliausia strategija norint rasti reikšmę, kuri buvo praleista pro šią funkciją, būtų grubios jėgos ataka (angl. brute force attack), kuri eitų per visas galimas reikšmes. Tokių reikšmių būtų 2^{256} , tad sudėtingumas irgi prilygtų $O(2^{256})$. [SG09] [Rja08]

Hash funkcijos pasižymi šiomis savybėmis pagal [Rja08] darbą:

- Determinuotumas: paduodant tą pačią reikšmę visada grąžins tą patį hash kodą.
- Vienakryptiškumas: iš hash kodo negalime atkurti originalios reikšmės.
- Nenuspėjamumas: mažas pokytis reikšmėje sugeneruoja visiškai kitokią hash kodo reikšmę. Pvz., turime žinutę pagrindu 2 (bitų pagrindas) ir pakeitę vieną bitą iš 0 į 1 gausime visiškai kitokį hash kodą, neturintį jokių panašumų tarp pradinio ir naujo hash kodo.

6.4. Elektroninis balsavimas

6.4.1. Balsavimo proceso reikalavimai

Kuriant elektroninio balsavimo sistemą pagrindinis tikslas yra skaidrumas. Norint turėti naudotojo pasitikėjimą sistema, iškyla problema. Paprasčiausias būdas įrodyti naudotojui, kad balso šifras tikrai atitinka jo pasirinkimą, būtų leisti apskaičiuoti savo balsą.

Dešifruoti kiekvieno balso negalime, dešifravimo raktas turi būti ypač saugomas ir panaudotas vieną kartą norint gauti galutinį rezultatą. Sistema negali pasitikėti naudotoju, jog jis nebandys pateikti neleistino balso. Kaip patikrinti balso validumą jo nedešifravus? Šiai problemai spręsti puikiai tinka prieš tai pristatytas kriptografinis įrankis, žinių nesuteikiantys įrodymai.

6.4.1.1. Žinių nesuteikiantis įrodymas, balso validumas

Svarbu prieš priimant balsą žinoti, jog balsuotojas skaičiuodamas šifrą naudojo vieną iš leistinų žinučių. To neatlikus ir priėmus tokį balsą, šis balsas gali „sugadinti“ ar iškraipyti rezultatą. Jei ir visi kiti balsai yra iš leistinų žinučių aibės.

Pvz. leistinų žinučių aibė (01000001, 01000100, 01010000). Turime 3 žinutes - C_1, C_2, C_3 . Dešifravus jas atskirai, gauname m_1, m_2, m_3 . Tarkime, $m_1 = 01000001$ - leistina žinutė, $m_2 = 01010000$ - leistina žinutė, $m_3 = 124127481$ - nevalidi žinutė. Šioms žinutėmis pritaikę Paillier homomorfinę savybę - $D(C_1 \cdot C_2 \cdot C_3) = m_1 + m_2 + m_3 = 01000001 + 01010000 + 124127481 = 126137482$ gautume neteisingą rezultatą. Viena nevalidi žinutė sugadino visą balsavimą.

Atpažinti, ar balsas yra gautas naudojantis vieną iš leistinų žinučių, galime per žinių nesuteikiantį įrodymą. Šis įrodymas patikrina, ar pateiktas šifras yra vienas iš leistinų, neatskleidžiant kuri žinutė buvo naudota. [Dah16] [Ibr16] [DF01]

Elektroninio balsavimo organizacija paskelbia pradines sąlygas:

- Leistinų žinučių aibę M . h - elementų kiekis aibėje M ;
- Paillier viešuosius raktus $g, n, N = n^2$.

Protokolo eiga:

1. Pasižadėjimo žingsnis. Šiame žingsnyje balsuotojas sukuria pasižadėjimą. Balsuotojo skaičiavimai:

- C - balso šifras, $C = g^{m_k} \cdot r^n \pmod{n^2}$, k - šifruojamos žinutės numeris sąrašo M ;
- r - privatus raktas, naudotas šifruojant C ;
- U - sąrašas tarpinių reikšmių, kurios leidžia sukurti pasižadėjimą A . i - sąrašo elemento numeris;

$$U_i = \frac{C}{g^{m_i}} \pmod{N} \quad (9)$$

Pvz. $M = [101; 110]$, $U = [\frac{C}{g^{m_0}} \pmod{n}; \frac{C}{g^{m_1}} \pmod{n}] = [\frac{C}{g^{101}} \pmod{n}; \frac{C}{g^{110}} \pmod{n}]$

- e - sąrašas atsitiktinių skaičių. Sąrašo ilgis yra h (lygus leistinų žinučių kiekiui). $e \in [1; n]$;
- z - sąrašas atsitiktinių skaičių. Sąrašo ilgis yra h (lygus leistinų žinučių kiekiui). $z \in [1; n]$;
- j - atsitiktinis skaičius. $j \in [1; n]$;
- A - sąrašas pasižadėjimų, kiekvienai leistinai žinutei per U sąrašo įrašą.

$$A_i = \frac{z_i^n}{U_i^{e_i}} \pmod{N}, A_k = j^n \pmod{N} \quad (10)$$

- Balsuotojas perduoda sąrašus A ir U bei šifrą C tikrintojui.

2. Iššukio žingsnis.

- Jei protokolas interaktyvus. Tikrintojas gavęs sąrašą A , paruošia iššukį e_{-1} . e_{-1} - atsitiktinis skaičius, $e_{-1} \in [1; n]$. Šį skaičių perduoda įrodinėtoji.
- Jei protokolas neinteraktyvus. Balsuotojas praleidžia sąrašą A pro hash funkciją. Jei hash kodas nėra skaičius, jį pasiverčia į skaičių. Hash kodas bus iššukis $e_{-1} = HashFunkcija(A) \pmod{n}$. Balsuotojas e_{-1} bei hash funkcijos informaciją perduoda tikrintojui.

3. Pasižadėjimo apgynimo žingsnis.

- (a) Įrodinėtojas pakeičia e_k reikšmę pagal gautą iššūkį e_{-1} . $e_k = (e_{-1} - \text{sum}(e)) \pmod{n}$, $\text{sum}(e) = (\sum_{i=0}^h e_i) - e_k$;

- (b) Užpildo sąrašą Z:

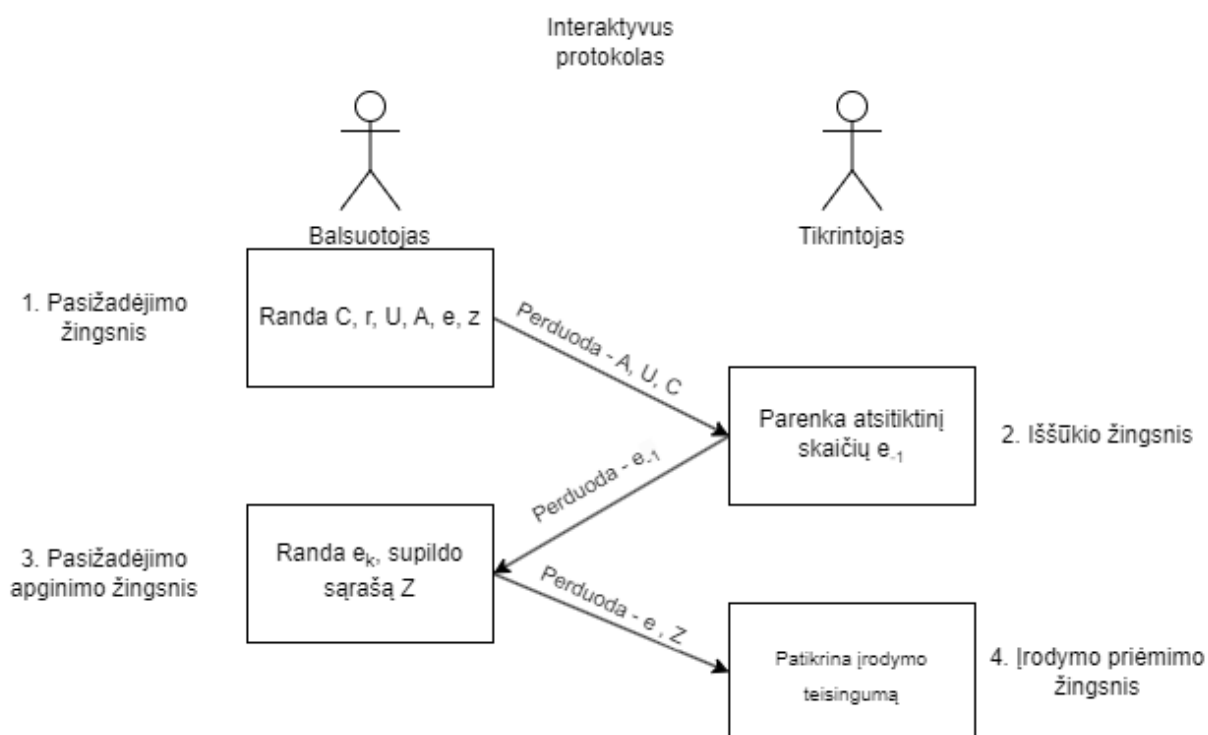
$$Z_i = (r^{e_i} \cdot j) \pmod{N} \quad (11)$$

- (c) Balsuotojas perduoda sąrašus e bei Z tikrintojui.

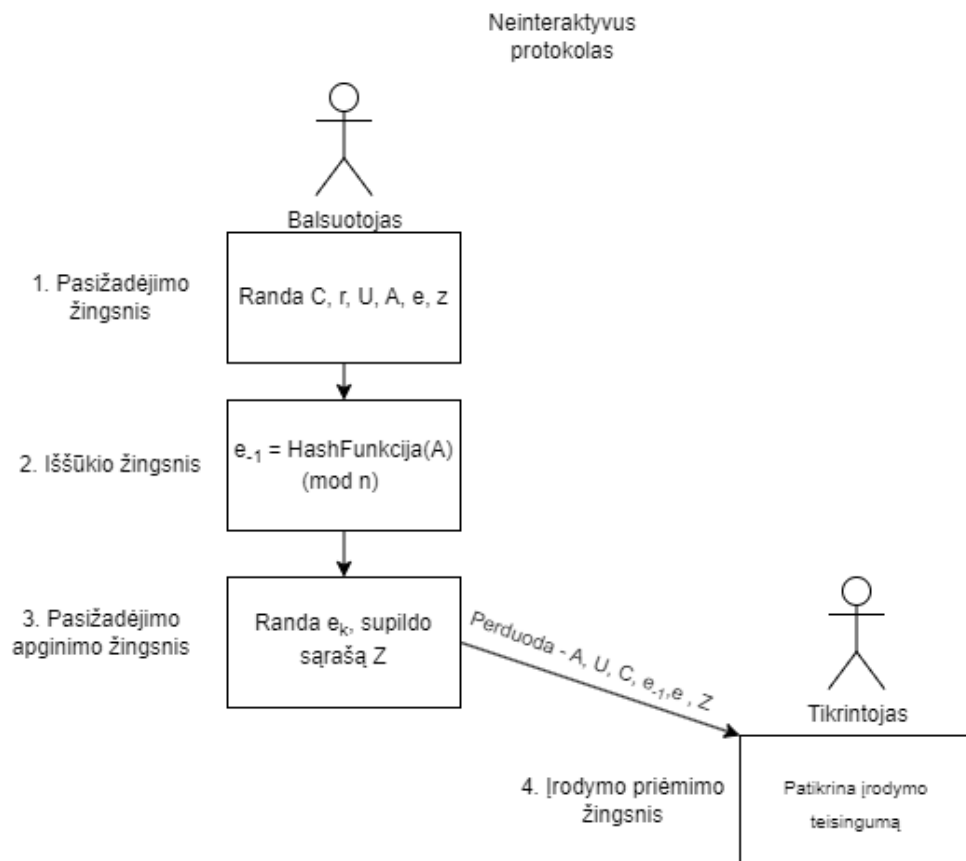
4. Įrodymo priėmimo žingsnis.

- Patikrina, ar įrodinėtojas tikrai panaudojo pateiktą iššūkį $e_{-1} \equiv \sum_{i=0}^h e_i \pmod{n}$
- Patikrina, ar įrodinėtojo kiekviena pasižadėjimo reikšmė yra tinkamai apginta $Z_i^n \equiv A_i \cdot U_i^{e_i} \pmod{N}$

9 pav., 10 pav. vizualiai perteikia protokolų žingsnių eigą.



9 pav. Interaktyvus protokolus.



10 pav. Neinteraktyvus protokolai.

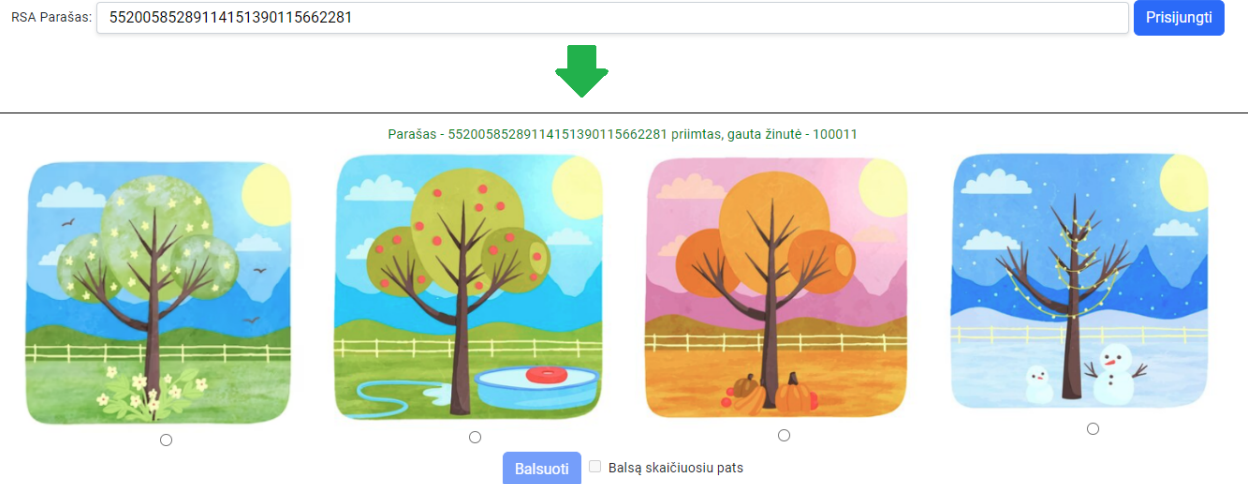
Kadangi šis protokolai yra nedaug išnagrinėtas moksliniuose šaltiniuose, todėl jo veikimą reikia išanalizuoti. Todėl jį realizavau programiškai – programavimo kalba java. Protokolo kodinis pritaikymas leidžia efektyviau tyrinėti bei analizuoti protokolą. Protokolo scenarijams sukūriau testus bei atlikau eksperimentus. Nagrinėti scenarijai:

1. Pagrindinis panaudojimo scenarijus, įrodinėtojas bei tikrintojas sąžiningai laikosi protokolo. Testų rezultatai – sėkmė, eksperimentuota plečiant galimų žinučių kiekį, didinant skaičių p ir q dydį.
2. Įrodinėtojas nesąžiningai laikosi protokolo. Pvz.:
 - Įrodinėtojas pasirinkimo žingsnyje šifruoja neleistiną žinutę. Testų rezultatai – sėkmė. Įrodymas nebuvo priimtas, nes pasirinkimas nederėjo su pasirinkimo apgynimo reikšmėmis, tad įrodymas nebuvo priimtas.
 - Įrodinėtojas pasirinkimo žingsnyje užmiršo r , vėlesniuose skaičiavimuose naudoja kitą reikšmę. Testų rezultatai – sėkmė. Įrodymas nebuvo priimtas nes pasirinkimas nederėjo su pasirinkimo apgynimo reikšmėmis, tad įrodymas nebuvo priimtas.
 - Įrodinėtojas šifravimo žingsnyje naudojo neleistiną g . Eksperimentų rezultatai – sėkmė. Įrodymas nebuvo priimtas, nes pasirinkimo reikšmė U gauta neteisingai, tikrintojas jas gali pats apskaičiuoti žinant C .

Šis žinių nesuteikiantis įrodymas leistų balsuotojui balsą skaičiuoti nepriklausomai nuo sistemos ir būti užtikrintas jog jo balsas tikrai šifruoja pasirinktą žinutę.

7. Balsavimo procesas sistemoje

Balsavimas puslapyje turi du kelius: balsuoti paprastai neskaičiuojant savo balso, arba turint pilną kontrolę ir skaičiuojant savo balsą. Visa tai galima atlikti prisijungus pasinaudojant validžiu RSA parašu 11 pav.



11 pav. Prisijungimas į balsavimo dalį su RSA parašu, balsuoti.deivis.dev

Naudotojas yra informuojamas, jog jo parašas yra validus bei perkeliamas į balsavimo dalį, ką galime pamatyti paveikslėlio apačioje.

Paspaudus, ant norimo balso nuotraukos atsirakina mygtukas „Balsuoti“ ir žymimasis langelis „Balsą skaičiuosiu pats“.

7.1. Balsavimas, neskaičiuojant balso

Mygtukas „Balsuoti“ suformuos balsą už pasirinkimą. Pasirinkus norimą dalyvį, galima šį mygtuką paspausti 12 pav.



12 pav. Balsavimas, neskaičiuojant balso, balsuoti.deivis.dev

Paspaudus „Balsuoti“, naudotojas yra informuojamas, jog jo balsas buvo sėkmingai priimtas

bei jį gali pamatyti balsų lentelėje, kuri yra matoma iš visų sistemos langų, ji bus pristatyta tolimesniajame skyriuje. Visa tai atspindi 13 pav.

Balsas sėkmingai priimtas, matomas lentelėje paryškintas

Neturiu RSA Parašo


Balsų lentelė	
RSA parašas, žinutė	Balsas
30033453330637930474814988448 / 583358	54743234018970307301445259058353950515
66626661627510263977832228349 / 381294	28789403553165441548349900593760783557
104713888110339726042601586619 / 736546	12573995418619261455539154793281056046
49539945686236562527355033809 / 262684	41056361592300004192852483887801665457
9145698155344180793295322557 / 509624	2611250586368344456576509443410137722
59522606749372425601441531117 / 620463	46683898228942219987478135492174616083
74162741311335864182773685686 / 121356	30088705338463885059687121245793195723
9362144272727669736373187661 / 339337	16464278486329016281681558832311147247
6531250255234163092115113614 / 922760	48230972893768231889912026563908850642
77491226505346331251979073496 / 956068	33017026920132080681564517833114691580
55200585289114151390115662281 / 100011	55248871202654881149123284410850405074
101343894127781664644019401363 / 100018	63957594299874604286296993558509137479
74110613256656625104838122929 / 100123	51771318285604102370159428232375291491

13 pav. Prabalsavus įprastai, balsuoti.deivis.dev

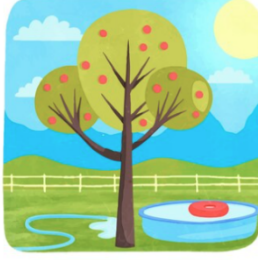
7.2. Balsavimas, skaičiuojant balsą

Skaičiuojant balsą pačiam atsiveria daugiau informacijos, čia naudotojas gali susipažinti su viešaisiais raktais, n , n^2 , g, žinučių reikšmės už pasirinkimus galima matyti po paveikslėliais. Balsų pasirinkimai paryškinti spalvomis, to tikslas pagerinti atsekamumą, supaprastinti ir taip sunkų turinį. Visa tai galime pamatyti 14 pav.


Parašas - 101343894127781664644019401363 priimtas, gauta žinutė - 100018



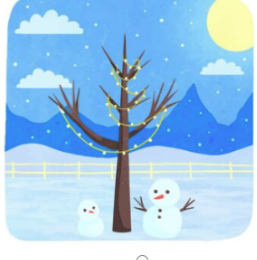
○
100000001



●
100000100



○
100010000



○
101000000

Balsuoti ☒ Balsą skaičiuosiu pats

Skaičius n

Skaičius n²

Skaičius g

Skaičius r

Pasirinkite norima žinutę, galimos yra po paveikslėliais. m :


$C = g^m \cdot r^n \pmod{n^2} = 8077789064635723092^{100000100} \cdot 8077789064635723091 \pmod{8077789064635723091^2}$ =

$U = \frac{C}{m_1} ; \frac{C}{m_2} ; \frac{C}{m_3} ; \frac{C}{m_4} \pmod{n^2} = \frac{C}{100000001} ; \frac{C}{100000100} ; \frac{C}{100010000} ; \frac{C}{101000000} \pmod{n^2} =$

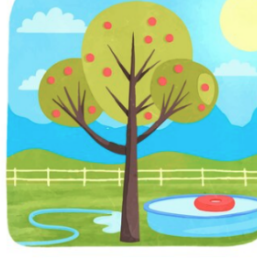
Balsų lentelė ^

14 pav. Balso skaičiavimas 1, balsuoti.deivis.dev


Suvedus privatų raktą r kuris laisvai pasirenkamas, ir žinutę m, galime paspausti mygtuką „=“ norint rasti šifrą, 15 pav.



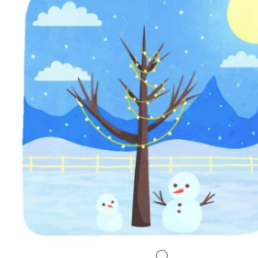
○
100000001



●
100000100



○
100010000



○
101000000

Balsuoti ☒ Balsą skaičiuosiu pats

Skaičius n

Skaičius n²

Skaičius g

Skaičius r

Pasirinkite norima žinutę, galimos yra po paveikslėliais. m :

$C = g^m \cdot r^n \pmod{n^2} = 8077789064635723092^{100000100} \cdot 20257131232131268978493^{8077789064635723091} \pmod{8077789064635723091^2}$ =

$63957594299874604286296993558509137479$

$U = \frac{C}{m_1} ; \frac{C}{m_2} ; \frac{C}{m_3} ; \frac{C}{m_4} \pmod{n^2} = \frac{63957594299874604286296993558509137479}{100000001} ; \frac{63957594299874604286296993558509137479}{100000100} ; \frac{63957594299874604286296993558509137479}{100010000} ; \frac{63957594299874604286296993558509137479}{101000000} \pmod{n^2} =$

15 pav. Balso skaičiavimas 2, balsuoti.deivis.dev

Gavus balso šifrą galime pradėti, žinių nesuteikiantį įrodymą norint pateikti balsą. Jis

reikalingas, nes sistema neseka naudotojo skaičiavimų, nesaugo jokios informacijos šiame žingsnyje. Naudotojas, kuris nepasitiki sistemos įsaugojimu, visus skaičiavimus gali atlikti savomis programomis.

Galime apskaičiuoti U ir užpildyti Z, E atsitiktiniais skaičiais, pasirinkti skaičių j ir paprašyti iššūkio, visa tai matoma 16 pav.

$$U = \frac{C}{m_1} + \frac{C}{m_2} + \frac{C}{m_3} + \frac{C}{m_4} \pmod{n^2} = \frac{63957594299874604286296993558509137479}{100000001} + \frac{63957594299874604286296993558509137479}{100000100} + \frac{63957594299874604286296993558509137479}{100010000} + \frac{63957594299874604286296993558509137479}{101000000} \pmod{n^2} =$$

U₁ =
 U₂ =
 U₃ =
 U₄ =

Z sąrašas atsitiktinių skaičių z_i. Užpildykite z reikšmes. z_i ∈ [1; n²], išskyrus reikšmę su balso indeksu 2.

E sąrašas atsitiktinių skaičių e_i. Užpildykite e reikšmes. e_i ∈ [1; n], išskyrus reikšmę su balso indeksu 2.

z₁ = e₁ =
 z₂ = e₂ =
 z₃ = e₃ =
 z₄ = e₄ =

Įveskite A - a_j = $\frac{z_j^n}{U_j} \pmod{n^2}$, o A_n, kur h yra žinutės numeris, o j atsitiktinis skaičius, A_n = jⁿ mod n, j = balso žinutės numeris - 2.

a₁ = $\frac{4182954498782419040441895862477996533^{8077789064635723091}}{44821644891809988059715743133966642189^{527482679715650138}} \pmod{(8077789064635723091)^2} =$
 a₂ = $\frac{56354312313^{8077789064635723091}}{3641192773863204056633722009397851305^{602659539919058788}} \pmod{(8077789064635723091)^2} =$
 a₃ = $\frac{1342081265861288921652701668971868830^{8077789064635723091}}{38968933749232321587294106286338291672^{846173920194985535}} \pmod{(8077789064635723091)^2} =$
 a₄ = $\frac{9986884773235108203347386928585258905^{8077789064635723091}}{3641192773863204056633722009397851305^{602659539919058788}} \pmod{(8077789064635723091)^2} =$

Iššūkis s. Paspaudus „Gauti iššūkį“ bus nuskaitomi C - balso laukas bei U ir A reikšmių sąrašai. Jų nekeiskite, nes sistema naudos tik tuos skaičius, kurie buvo mygtuko paspaudimo metu.

16 pav. Balso skaičiavimas 3, balsuoti.deivis.dev

Gavus iššūkį galime apskaičiuoti s, rasti $e_{balsoNumeris}$ reikšmę bei $z_{balsoNumeris}$, visa tai atlikę esame pasiruošę įrodymo pridavimui.

Z sąrašas atsitiktinių skaičių z_i . Užpildykite z reikšmes. $z_i \in [1; n^2]$, išskyrus reikšmę su balso indeksu 2. [Užpildyti z atsitiktinėmis reikšmėmis](#)

E sąrašas atsitiktinių skaičių e_i . Užpildykite e reikšmes. $e_i \in [1; n]$, išskyrus reikšmę su balso indeksu 2. [Užpildyti e atsitiktinėmis reikšmėmis](#)

$z_1 =$

$z_2 =$

$z_3 =$

$z_4 =$

$e_1 =$

$e_2 =$

$e_3 =$

$e_4 =$

Iveskite $A - a_i = \frac{z_i^n}{u_i^n} \pmod{(n^2)}$, o A_n kur h yra žinutės numeris o j atsitiktinis skaičius, $A_n = j^n \pmod{n}$, j = balso žinutės numeris - 2 [Apskaičiuoti visus a](#)

$a_1 =$
 $\pmod{(8077789064635723091^2)} =$

$a_2 =$ $\pmod{(8077789064635723091)} =$

$a_3 =$
 $\pmod{(8077789064635723091^2)} =$

$a_4 =$
 $\pmod{(8077789064635723091^2)} =$

Iššukis s. Paspaudus Gauti iššukį bus nuskaitomi C - balso laukas, bei U ir A reikšmių sąrašai. Jų nekeiskite, nes sistema naudos tik tuos skaičius kurie buvo mygtuko paspaudimo metu.

[Gauti iššukį](#)

Random e $z = s - \text{sum}(e) \pmod{n} = 87066624279991722 - (935477593412875568 + 734992860913553085 + 298199281343580351) \pmod{(8077789064635723091)}$ [=](#)

Random e $z = (r^{e_2} * j) \pmod{n^2} = (20257131232131268978493^{6979785953245705809} * 56354312313) \pmod{(8077789064635723091^2)}$ [=](#)

Pasiekime pabaigą, patikriname žinių nesuteikiantį įrodymą. Jau buvo užfiksuotos balso - C reikšmės, sąrašų A, U reikšmės, paspaudus "Gauti iššukį" mygtuką. Paspaudus "Pateikti įrodymą" bus nuskaitytos sąrašų E ir Z reikšmės, bei patikrintas įrodymo validumas. [Pateikti įrodymą](#)

17 pav. Balso skaičiavimas 4, [balsuoti.deivis.dev](#)

Galime priduoti įrodymą, paspaudę mygtuką „Pateikti įrodymą“, atsiveria įrodymo patikrinimas 18 pav. Forma yra užrakinama ir leidžiama arba keisti pateiktį, arba priduoti balsą. Norint keisti pateiktį, reikės vėl paspausti mygtuką „Pateikti įrodymą“, nes jis dings paspaudus „Noriu keisti pateiktį“ mygtuką.

Random e $z = (r^{e_2} * j) \pmod{n^2} = (20257131232131268978493^{6483172218539749529} * 56354312313) \pmod{(8077789064635723091^2)}$ [=](#)

Pasiekime pabaigą, patikriname žinių nesuteikiantį įrodymą. Jau buvo užfiksuotos balso - C reikšmės, sąrašų A, U reikšmės, paspaudus „Gauti iššukį“ mygtuką. Paspaudus „Pateikti įrodymą“ bus nuskaitytos sąrašų E ir Z reikšmės bei patikrintas įrodymo validumas. [Pateikti įrodymą](#)

1. $s = \text{sum}(e) \pmod{n}$; $381699293733720899 = 8459488358369443990 \pmod{(8077789064635723091)}$;
 $381699293733720899 = 381699293733720899$

2. $Z_1^n = A^1 * U_1^{e_1} \pmod{n}$; $4182954498782419040441895862477996533 \pmod{(8077789064635723091)} = 35744291153163242281896474592101540781 * 44821644891809988059715743133966642189 \pmod{(8077789064635723091^2)}$;
 $5718914321841234937449353962348041600 = 5718914321841234937449353962348041600$

$Z_2^n = A^2 * U_2^{e_2} \pmod{n}$; $27590660655770708585522556471432440267 \pmod{(8077789064635723091)} = 36028963314743122760176650659535317583 * 34426956476893619851342729304357554476 \pmod{(8077789064635723091^2)}$;
 $4322967764524468064590953124140075903 = 4322967764524468064590953124140075903$

$Z_3^n = A^3 * U_3^{e_3} \pmod{n}$; $1342081265861288921652701668971868830 \pmod{(8077789064635723091)} = 15208214309471016031429864635084312559 * 38968933749232321587294106286338291672 \pmod{(8077789064635723091^2)}$;
 $6652920293655155828155008672961410583 = 6652920293655155828155008672961410583$

$Z_4^n = A^4 * U_4^{e_4} \pmod{n}$; $9986884773235108203347386928585258905 \pmod{(8077789064635723091)} = 54877217018667763448088358918350987393 * 3641192773863204056633722009397851305 \pmod{(8077789064635723091^2)}$;
 $59656375324433470096329203554598071549 = 59656375324433470096329203554598071549$

Balsas tinkamas

[Noriu keisti pateiktį](#) [Pateikti](#)

18 pav. Balso skaičiavimas 5, [balsuoti.deivis.dev](#)

Jei įrodymas neteisingas, tai irgi yra atvaizduojama. 19 pav.

$\text{Randame } z_2 = (r^{e_2} * j) \pmod{n^2} = (20257131232131268978493^{6483172218539749529} * 56354312313) \pmod{8077789064635723091^2}$

Pasiekėme pabaigą, patikriname žinių nesuteikiantį įrodymą. Jau buvo užfiksuotos balso - C reikšmės, sąrašą A, U reikšmės, paspaudus „Gauti iššūkį“ mygutką. Paspaudus „Pateikti įrodymą“ bus nuskaitytos sąrašų E ir Z reikšmės bei patikrintas įrodymo validumas.

1. $s = \text{sum}(e) \pmod{n}$; $381699293733720899 = 8459488358369443990 \pmod{8077789064635723091}$;
 $381699293733720899 = 381699293733720899$

2. $Z_1^n = A^1 * U_1^{e_1} \pmod{n}$; $4182954498782419040441895862477996533^{8077789064635723091} = 35744291153163242281896474592101540781 * 44821644891809988059715743133966642189^{527482679715650138} \pmod{8077789064635723091^2}$;
 $5718914321841234937449353962348041600 = 5718914321841234937449353962348041600$

$Z_2^n = A^2 * U_2^{e_2} \pmod{n}$; $27590660655770708585522556471432440267^{8077789064635723091} = 360289633147431227601766506595353117583 * 34426956476893619851342729304357554476^{6483172218539749529} \pmod{8077789064635723091^2}$;
 $43229677645244468064590953124140075903 = 46619857574232487410021192304246309607$

$Z_3^n = A^3 * U_3^{e_3} \pmod{n}$; $1342081265861288921652701668971868830^{8077789064635723091} = 15208214309471016031429864635084312559 * 38968933749232321587294106286338291672^{846173920194985535} \pmod{8077789064635723091^2}$;
 $6652920293655155828155008672961410583 = 6652920293655155828155008672961410583$

$Z_4^n = A^4 * U_4^{e_4} \pmod{n}$; $9986884773235108203347386928585258905^{8077789064635723091} = 54877217018667763448088358918350987393 * 36411927773863204056633722009397851305^{602659539919058788} \pmod{8077789064635723091^2}$;
 $59656375324433470096329203554598071549 = 59656375324433470096329203554598071549$

Balsas netinkamas

19 pav. Balso skaičiavimas 6, balsuoti.deivis.dev

Galime pateikti balsą, paspaudę „Pateikti“, naudotojas yra informuojamas, jog balsas sėkmingai priimtas ir jį gali pamatyti balsų lentelėje 13 pav.

Rezultatas, naudotojas gali apskaičiuoti savo balsą, būti užtikrintas, jog užšifruote balse tikrai slypi jo pasirinkimas. Sistema apie naudotoją nieko nežino, kadangi jis prisijungė anonimiškai pasinaudojant RSA parašu bei apie jo pasirinkimą irgi nieko nežino, tik tai, kad jis yra vienas iš leistinių.

8. Balsavimo proceso pabaiga

8.1. Balsų lentelė

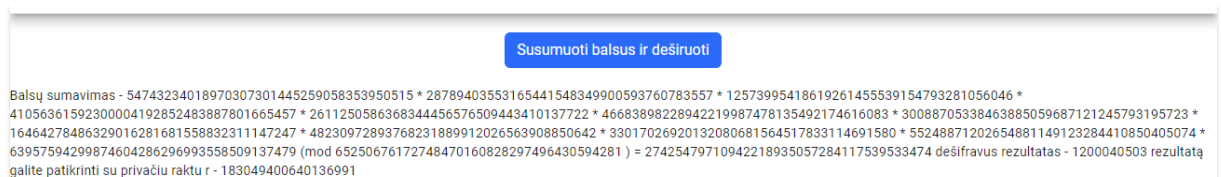
Balsų lentelėje yra du stulpeliai, RSA parašas bei jų žinutės, balsai. Naudotojai viso balsavimo metu gali sekti bei patikrinti ar jų balsai nesikeičia, nedingsta. RSA parašo žinutes irgi gali pažiūrėti, kad nešifruotų tos pačios, ši problema buvo pristatyta ankstesniame skyriuje.

8.2. Balsų sumavimas

Balsų sumavimą naudotojas gali sekti, patikrinti ar sistema tikrai gerai viską atlieka. Pagal Paillier homomorfinę savybę, šifrai, tuo pačiu jų žinutės, gali būti sudėtos, sudauginant šifrus, kad rezultatas nebūtų nereikalingai didelis, dar pritaikoma $mod n^2$. Sudaugintą šifrų rezultatą sistema pateikia, jį gali patikrinti naudotojas.

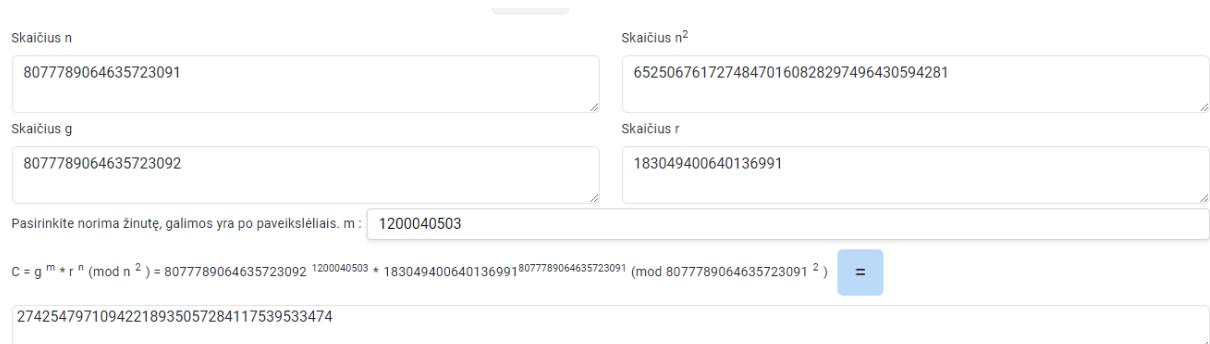
8.3. Rezultato paskelbimas

Rezultato paskelbiamas bei tuo pačiu yra pateikiamas privatus raktas r . Turint raktą r ir žinutę naudotojas irgi gali patikrinti sistemos veiksmus užšifruojant šią žinutę su visa turima informacija, 20 pav.



20 pav. Balsavimo pabaiga, balsuoti.deivis.dev

Galime patikrinti rezultatą, „galutinis šifras – 27425479710942218935057284117539533474 dešifravus rezultatas – 1200040503 rezultatą galite patikrinti su privačiu raktu r – 183049400640136991“. Visa tai įsistačius į balsavimo langą kintamuosius gauname teisingą informaciją, tai viskas teisingai veikia, 21 pav.



21 pav. Rezultato patikrinimas, balsuoti.deivis.dev

Pasiektame rezultate turime viešą sumavimą. Naudotojai gali patikrinti ar sistema viską

atlieka tinkamai. Balsavimo rezultatą dalyviai irgi gali patikrinti, visas balsavimo procesas įvyko viešai.

8.4. Internetinis puslapis

Internetinis puslapis – balsuoti.deivis.dev buvo sukurtas su Java, Spring Boot, Angular, TypeScript. Front-end paleistas per GitHub Pages, back-end paleistas ant išnuomoto serverio per Docker aplinką. Pirminė versija buvo paleista per Azure Cloud. Komunikacija tarp front-end ir back-end dalių vyksta per REST API POST ir GET užklausas.

Išvados

Išanalizuotos kriptografinės struktūros buvo praktiškai pritaikytos kuriant elektroninio balsavimo puslapį, kuris yra patalpintas balsuoti.deivis.dev.

Pasinaudojant RSA aklojo parašo schema buvo pasiektas anoniminis prisijungimas į balsavimo procesą. Sistema žino, kad šį parašą galėjo susikurti tik autentifikuotas, leistinas balsuotojas.

Balsavimo procese buvo sėkmingai pritaikyta žinių, nesuteikiančių įrodymų, struktūra, kuri leidžia balsuotojui apskaičiuoti savo balsą. Balsuotojas turi pilną balso kontrolę – pilnai žino jog jo balso šifras atitinka jo pasirinkimą. Pasinaudojant žinių, nesuteikiančiu įrodymu, sistema apie balso žinutę žino tik tai, kad ji yra viena iš leistinių. To pilnai pakanka priimti balsą.

Balsų sumavime buvo pasinaudota Paillier homomorfine savybė, kuri leido viešai atlikti balsų šifrų sumavimus. Kadangi balsai yra viešai matomi, balsuotojai gali patikrinti sistemos veiklą. Tokiu būdu galima sužinoti, jog visų asmenų balsai tikrai buvo įskaičiuoti.

Balsavimo rezultato paskelbimas pasinaudojo Damgård–Jurik Paillier kriptosistemos plėtinium, kuris leidžia surasti privatų raktą r . Šią reikšmę paskelbus naudotojai gali patikrinti, jog viešai homomorfiškai susumuotų šifrų rezultatas tikrai šifruoja tai ką sistema skelbia.

Šaltiniai

- [Sta07] Vilius Stakėnas, Kodai ir šifrai. Vilnius: Vaistų žinios, 2007
- [Dah16] PAILLIER ZERO - KNOWLAGE PROOF, Taylor Fox Dahlin, 2016.
- [Ibr16] Robust Electronic Voting System using Homomorphic Encryption Protocol and Zero-Knowledge Proof, Dr. Mahmood Khalel Ibrahim 2016.
- [DF01] A Zero Knowledge proof for Subset Selection from a Family of Sets with applications to Multiparty/Multicandidate Electronic Elections. T. Dimitriou, D. Foteinakis, 2001.
- [Jah19] Overview and Applications of Zero Knowledge Proof (ZKP) Jahid Hasan, 2019.
- [ZFZS20] Zero Knowledge Proofs for Decision Tree Predictions and Accuracy. Jiaheng Zhang, Zhiyong Fang, Yupeng Zhang, Dawn Song, 2020.
- [CPSS11] Elliptic Curve Based Zero Knowledge Proofs and Their Applicability on Resource Constrained Devices Ioannis Chatzigiannakis, Apostolos Pyrgelis, Paul G. Spirakis and Yannis C. Stamatiou, 2011.
- [SG09] The Evaluation Report of SHA-256 Crypt Analysis Hash Function. A.Arul Lawrence Selvakumar, C.Suresh Ganandhas, 2009.
- [Rja08] Properties of Cryptographic Hash Functions. Michal Rjaško, 2008.
- [Dav83] Blind Signatures for Untraceable Payments, Chaum David, 1983.
- [RSA78] A Method for Obtaining Digital, Signatures and Public-Key Cryptosystems, R.L. Rivest, A. Shamir, and L. Adleman, 1978.
- [MOV01] Handbook of Applied Cryptography, Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, 2001.
- [DJ00] A Generalisation, a Simplification and some Applications of Paillier's Probabilistic Public-Key System Ivan B. Damgård, Mads J. Jurik, 2000
- [BGENM16] Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, 2016.
- [BR19] Transitioning the Use of Cryptographic Algorithms and Key Lengths, Elaine B. Barker, Allen L. Roginsky, 2019.
- [AAZS20] "Survey on homomorphic encryption and address of new trend." International Journal of Advanced Computer Science and Applications 11.7. Alharbi, Ayman, Haneen Zamzami, and Eman Samkri, 2020.
- [MA18] "Survey on homomorphic encryption." International Conference for Phoenixes on Emerging Current Trends in Engineering and Management (PECTEAM 2018). Sirajudeen, Y. Mohamed, and R. Anitha. Atlantis Press, 2018.
- [DM13] P.Drungilas, H.Markšaitis. Algebra I dalis, Vilniaus universiteto leidykla, Vilnius, 2013.