KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

TAIKOMOSIOS MATEMATIKOS KATEDRA

Andrius Gediminskas

ŽIEDINIŲ PARAŠŲ NAUDOJIMAS FORUMUOSE

Magistro darbas

Vadovas

prof. Eligijus Sakalauskas

KAUNAS, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

TAIKOMOSIOS MATEMATIKOS KATEDRA

TVIRTINU

Katedros vedėjas

doc. dr. N. Listopadskis

2012 06 05

ŽIEDINIŲ PARAŠŲ NAUDOJIMAS FORUMUOSE

Taikomosios matematikos magistro baigiamasis darbas

Vadovas

(parašas) prof. Eligijus Sakalauskas

2012 06 01

Recenzentas

(parašas) doc. dr. S. Japertas

2010 06 01

Atliko

FMMM-0 gr. stud.

(parašas)A. Gediminskas

2012 05 30

KAUNAS, 2012

KVALIFIKACINĖ KOMISIJA

Pirmininkas: Rimantas Rudzkis, profesorius (VU MII)

Sekretorius: Eimutis Valakevičius, docentas (KTU)

Nariai: Jonas Valantinas, profesorius (KTU)

Vytautas Janilionis, docentas (KTU)

Vidmantas Povilas Pekarskas, profesorius (KTU)

Zenonas Navickas, profesorius (KTU)

Arūnas Barauskas, dr., vice-prezidentas projektams (UAB „Baltic Amadeus“)

Gediminskas A. Ring Signatures in forums: Master‘s work in applied mathematics / supervisor prof. E. Sakalauskas; Department of Applied mathematics, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2012. – 41 p.

SUMMARY

Ring signature is cryptografic primitive, that enables a user to sign a message anonymously on behalf of a certain group of people, which includes the signer and it is called a ring. From 2001, when Ring Signature was first invented by Rivest, Shamir and Tauman, massive interest has been shown to ring signcryption algorythms and diferent schemes were introduced.

The goal of this work is to find out, whether the ring signatures work well in forums. Suppose, there is a signer who wants to stay anonymous, and there are several group members who would like to identify the signer. In this paper it is shown, that at least one ring signature scheme is created where the signer can be revealed if at least three of the members in the ring start to cooperate with each other, and the whole attack on that particular scheme is described. After it is proven that a particular ring signature is insecure, comparisons to a very similar ring signature scheme are made, and recommendations are given on what should be improved so that this attack will not reveal the identity of the signer.

TURINYS

Paveikslų sąrašas

Lentelių sąrašas

ĮVADAS

Tobulėjant informacinėms technologijoms ne tik lengviau sužinoti, kas vyksta pasaulyje, apsipirkinėti neišeinant iš namų ar bendrauti su kitame pasaulio gale esančiu draugu, tačiau galima ir užsiimti verslu, pervesti pinigus ar net balsuoti nekeliant kojos už namų slenksčio. Didžiąją dalį kasdienių reikalų jau įmanoma sutvarkyti naudojantis kompiuteriu. Tai, žinoma, labai patogu, tačiau vis dažniau yra susiduriama su tapatybės klastojimu, saugumo spragomis. Siekiant išvengti šių nemalonių aspektų, buvo pradėta galvoti apie tai, ar įmanoma nuotoliniu būdu įrodyti savo tapatybę taip, kad dėl to, kas esi, nekiltų jokių dvejonių ir tuo niekas negalėtų pasinaudoti siekdamas naudos sau. Buvo sukurti elektroniniai parašai, kurie šiuo momentu jau yra labai paplitę dėl savo saugumo ir patogumo.

Šiandien yra sukurta daugybė įvairių tipų elektroninių parašų, tobulinama daugybė sričių, kuriose šie parašai yra arba bus taikomi. Ne išimtis yra žiediniai parašai.

Žiedinių parašų koncepcija nėra sena – pirmą kartą tokio tipo elektroninius parašus aprašė Ronald L. Rivest, Adi Shamir ir Yael Tauman 2001 metais straipsnyje „How To Leak a Secret“ („Kaip atskleisti paslaptį“).

Žiedinių parašų esmė yra tokia: pasinaudojant savo viešuoju ir privačiuoju raktais bei kitų asmenų, kuriuos norima įtraukti į parašo sudarymą, viešaisiais raktais, yra sukuriamas parašas taip, kad bet kuriam tikrintojui nėra žinoma, koks konkrečiai asmuo šį parašą sukūrė, aišku tik tai, kad pasirašantysis priklauso tam tikrai asmenų grupei (tų žmonių, kurių viešuosius raktus panaudojo). Žiediniai parašai nuo grupinių parašų skiriasi tuo, kad nėra grupės valdytojo, atsakingo už parašų kūrimą, nereikia tų asmenų, kurių viešieji raktai naudojami, sutikimo. Daugeliu atveju žiedinis parašas suformuojamas taip, kad asmenys, kurie sudaro vadinamąjį žiedą, net nežino, kad jie yra įtraukti į jį. Pasirašantysis gali priklausyti kelioms organizacijoms, keliems forumams, ir visų jų vardu jis gali kurti skirtingus žiedinius parašus.

Pagrindinė žiedinių parašų atsiradimo priežastis yra noras nutekinti informaciją. Tarkime, turime asmenį A, kuris dirba organizacijai X. Asmuo A galvoja, kad labai svarbu asmeniui B pranešti organizacijos X informaciją. Asmuo A tai gali padaryti keturiais skirtingais būdais:

1. parašydamas anoniminį laišką;
2. parašydamas laišką ir pasirašydamas savo elektroniniu parašu;

3. parašydamas laišką ir panaudodamas grupinį parašą;
4. parašydamas laišką ir panaudodamas žiedinį parašą.

Šiuo konkrečiu atveju pirmieji trys būdai nėra geri. Pirmuoju atveju asmuo B neturi jokio pagrindo tikėti, kad gautasis pranešimas yra nuo asmens A, kadangi tokio tipo pranešimą gali atsiųsti bet kas, ir gautoji informacija nėra patikima. Antruoju atveju asmuo B tikrai žino, kad gautasis pranešimas yra nuo asmens A, tačiau pagrįsdamas informacijos patikimumą privalo nurodyti, nuo kurio konkrečiai asmens yra gautasis pranešimas. Jeigu taip atsitinka, organizacija X tiksliai žino, kad informaciją apie jų organizaciją nutekino būtent asmuo A, o to asmuo A nepageidauja. Trečiuoju atveju asmuo B gauna informaciją iš tam tikros grupės G narių (patartina, kad visi asmenys, esantys šioje grupėje G, priklausytų organizacijai X), tačiau nežino, iš kurio asmens tiksliai. Šiuo atveju asmuo B gali pasitikėti gauta informacija, o asmuo A lieka tik kaip vienas iš galimų paslapties atskleidėjų. Tokio parašo problema ta, kad šios grupės G administratorius žino, kas konkrečiai iš grupėje esančių asmenų paviešino informaciją, ir gali savo noru arba verčiamas atskleisti asmens A tapatybę.

Tokiu būdu asmeniui A geriausia rinktis ketvirtąjį variantą – naudoti žiedinius parašus. Asmuo B gaus informaciją, kurią paskleidė žiedo Ž nariai (kaip ir grupės G atveju, patartina, kad visi žiedo Ž nariai būtų iš organizacijos X), tačiau nežino, kuris tiksliai. Šiuo atveju jokio žiedo administratoriaus nėra, todėl asmuo A išlieka tik kaip galimas įtariamasis, atskleidęs informaciją.

Žiediniai parašai gali būti naudingi ne tik norint atskleisti slaptą informaciją ir likti nežinomam. Žiediniai parašai praverčia ir dalijantis slapta informacija. Tarkime, turime dvi kompanijas: A ir B. Šios kompanijos nori susitarti dėl sandėrio, tačiau nenori, kad pašaliniai asmenys išsiaiškintų, ką konkrečiai siūlo kompanija A kompanijai B ir atvirkščiai, tačiau nori būti užtikrintos, kad joks kitas asmuo negalės suklastoti sandėrio. Kompanija A sukuria žiedinį parašą, naudodama savo privatųjį ir viešąjį raktus bei kompanijos B viešąjį raktą, ir išsiunčia pasirašytą pranešimą kompanijai B. Ši atrašo, pasinaudodama kompanijos A viešuoju raktu bei savo viešuoju ir privačiuoju raktais. Tokiu atveju žiede yra tik du nariai (kompanija A ir kompanija B), ir joks pašalinis asmuo negali suprasti, ką konkrečiai rašė kuri kompanija, o mato tik bendrą dialogą. Kita vertus, kompanija A žino, kokius pasiūlymus pateikė ji, ir visi kiti pasiūlymai privalo būti tik nuo kompanijos B, nes daugiau niekas negali įsiterpti į šį dialogą nepakeisdamas žiedinio parašo. Tas pats galioja ir kompanijai B.

Žiediniai parašai gali būti plačiai naudojami įvairiuose balsavimuose, kai tam tikroms grupėms (tarkim, regionams, vartotojų ratams ir panašiai) sudaromi žiediniai parašai leidžia įsitikinti, kiek narių iš žiedo balsavo už tam tikrą žmogų ar įstatymą neatskleidžiant konkretaus žiedinio parašo nario tapatybės ir už ką kiekvienas narys balsavo. Tam labai

praverčia atsekamieji žiediniai parašai – jie yra sukurti būtent tam, kad atsekamo žiedinio parašo narys gali balsuoti visiškai slaptai už jį dominantį žmogų ar įstatymą, tačiau tai jis gali daryti tik vieną kartą – jeigu žiedinio parašo narys pabalsuoja du kartus, įmanoma atskleisti jo tapatybę taip užkertant kelią galimam sukčiavimui.

Šiame darbe yra apžvelgiami elektroniniai parašai, grupiniai parašai ir žiediniai parašai, įvairios šių parašų schemos. Šio darbo tikslas yra išsiaiškinti, ar iš šiuo metu sukurtų žiedinio parašo schemų yra tokių, kurios nėra pakankamai saugios parašo autoriaus anonimiškumo atžvilgiu. Jeigu tokios schemos esama, šiame darbe aprašoma bei išanalizuojama kriptografinė ataka prieš tokią schemą darant prielaidą, kad forumo dalyviai, kurie taip pat yra ir žiedinio parašo nariai, yra linkę tarpusavyje bendradarbiauti su tikslu demaskuoti parašo autorių.

1. ANALITINĖ DALIS

1.1 ELEKTRONINIŲ PARAŠŲ ALGORITMAI

Elektroniniuose parašuose labiausiai paplitę RSA (Rivest, Shamir, Adleman – autorių, sugalvojusių šį algoritmą, pavardžių pirmosios raidės) ir DSA (Digital Signature Algorythm) algoritmai. Abiejų šių algoritmų taikymo esmė yra ta pati: siuntėjas siunčiamam pranešimui pritaiko vienkryptę maišos (hash) funkciją ir sugeneruoja užkoduotą teksto santrauką. Ši užšifruojama siuntėjo privačiuoju raktu (kurį siuntėjas saugo, o jo viešąjį raktą gali matyti visi) ir taip suformuojamas elektroninis parašas. Gavėjas, gavęs pranešimą, atskiria jo tekstą nuo parašo ir tekstui taiko maišos funkciją, taip sugeneruodamas pranešimo santrauką. Tuo tarpu parašas dekoduojamas siuntėjo viešuoju raktu ir taip gaunama teksto antroji santrauka. Gavėjas palygina, ar pirmoji santrauka (gauta panaudojus maišos funkciją) ir antroji santrauka (iššifravus siuntėjo viešuoju raktu) yra identiškos. Jei jos identiškos, vadinasi, siuntėjo pranešimas pasiekė gavėją nepakeistas ir gavėjas yra tikras, kad pranešimą jam siuntė siuntėjas ir niekas kitas.

RSA algoritmas, pateikiamas [28]:

1. Atsitiktinai pasirenkami du dideli pirminiai skaičiai p ir q.
2. $n = pq.$ **(1.1.1)**
3. $\varphi(n) = (p - 1)(q - 1),$ **(1.1.2)**

čia $\varphi(n)$ – Oilerio funkcija.

4. Parenkamas toks skaičius e, kad tenkintų dvi savybes: $1 < e < \varphi(n)$, ir skaičių e ir φ(n) didžiausias bendrasis daliklis yra lygus 1, t.y. skaičiai e ir φ(n) tarpusavyje yra pirminiai.

5. Randamas toks d, kad

$$d = e^{-1} \bmod \varphi(n). \quad \textbf{(1.1.3)}$$

Tokiu būdu yra gaunami viešasis (n, e) ir privatusis (d) raktai.

Pranešimo užšifravimas.

Turime pranešimą M (Message). Jį paveikiame vienkrypte santraukos funkcija: $M \xrightarrow{hash} m$. Gautą santrauką užšifruojame privačiuoju raktu:

$$c = m^d \pmod n. \quad \textbf{(1.1.4)}$$

Pranešimo dešifravimas.

Gautą šifrą c paveikiame siuntėjo viešuoju raktu:

$$m = c^e \ (mod\ n). \qquad \textbf{(1.1.5)}$$

Vienintelis ir esminis skirtumas tarp originaliojo RSA algoritmo ir to RSA algoritmo, kuris yra naudojamas elektroniniuose parašuose, yra tas, kad RSA kodavimo metu bet kas gali išsiųsti pranešimą, o jį matyti gali tik gavėjas (nes tik jis žino savo privatųjį raktą), o elektroninio parašo atveju tik siuntėjas gali išsiųsti pranešimą, o jį matyti gali bet kas. Teoriškai, pasirašyti galima ne santraukos funkciją, o patį pranešimą, tačiau vienkryptės funkcijos naudojimas užtikrina laiško autentiškumą, t.y. garantuoja, kad niekas kitas nepakeitė laiške nei vieno simbolio.

DSA algoritmas, pateikiamas [29]:

1. Pasirenkamas toks N bitų ilgio pirminis skaičius q.
2. Pasirenkamas toks L bitų ilgio pirminis skaičius p, kad (p -1) būtų q kartotinis.
3. Pasirenkamas toks skaičius g, kad

$$g = h^{(p-1)/q} \ (mod\ p), \qquad \textbf{(1.1.6)}$$

skaičiui h galioja $0 < h < p - 1$. Jeigu

$$h^{(p-1)/q} \ (mod\ p) = 1, \qquad \textbf{(1.1.7)}$$

reikia pasirinkti kitą skaičių h. Labai dažnai h = 2.

4. Pasirenkamas toks x, kuriam galioja $0 < x < q$.
5. Apskaičiuojamas $y = g^x \ (mod\ p)$. **(1.1.8)**

Tokiu būdu yra gaunamas viešasis (p, q, g, y) ir privatusis (x) raktai. Pranešimo užšifravimas vyksta taip:

1. Sukuriamas atsitiktinis skaičius k, kuris kiekvienam laiškui yra vis kitoks.
2. Apskaičiuojamas

$$r = (g^k \ (mod\ p)\ mod\ q). \qquad \textbf{(1.1.9)}$$

Jeigu r reikšmė gaunama 0, kas įvyksta gana retai, reikia ieškoti naujo atsitiktinio skaičiaus k ir atlikti procedūrą iš naujo.

3. Apskaičiuojamas

$$s = (k^{-1}(H(m) + x^* r))\ mod\ q), \qquad \textbf{(1.1.10)}$$

čia m – pranešimas, kurį norime pasirašyti, H(x) – santraukos funkcija, H(m) – pranešimo santrauka. Jeigu s reikšmė gaunama 0, kas įvyksta labai retai, reikia pasirinkti kitą atsitiktinį skaičių k ir kartoti visą procedūrą iš naujo.

Tokiu būdu gauname parašą (r, s), kurį siunčiame gavėjui. Gavėjas, gavęs pasirašytą pranešimą bei matydamas siuntėjo viešąjį raktą, dešifruoja pranešimą tokiu būdu:

1. Patikrina, ar galioja sąlygos $0 < r < q$ ir $0 < s < q$. Jei bent viena sąlyga nėra išpildoma, parašas laikomas negaliojančiu (atmetamas).
2. Apskaičiuoja $w = s^{-1} \pmod q$. **(1.1.11)**
3. Apskaičiuoja $u1 = H(m) * w \pmod p$. **(1.1.12)**
4. Apskaičiuoja $u2 = r * w \pmod q$. **(1.1.13)**
5. Apskaičiuoja $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$. **(1.1.14)**

Parašas laikomas galiojančiu, jeigu v = r.

Silpnoji šio algoritmo dalis yra atsitiktinis skaičius k. Šio skaičiaus slaptumas ir unikalumas yra privalomas. Patys DSA kūrėjai teigia, kad to pačio k naudojimas daugiau nei vieną kartą arba k skaičiaus „nuspėjama“ reikšmė gali būti užtektinai atskleistos informacijos, kad DSA algoritmas būtų nulaužtas. Tas pats galioja ir net keletui atskleistų bitų k reikšmės skirtinguose parašuose – to gali užtekti, kad algoritmas taptų nulaužiamas.

1.2 GRUPINIAI PARAŠAI

Nagrinėjant žiedinius parašus, privaloma suprasti ne tik elektroninio parašo veikimo principą bei algoritmus, bet ir grupinius parašus, kaip pateikiama [7]. Būtent iš pastarųjų ir yra susiformavę žiediniai parašai. Grupinių ir žiedinių parašų principas panašus, pritaikymo sritys praktiškai identiškos.

Grupės administratorius paskelbia viešuosius grupinio parašo parametrus

$$PK = (q, P, P_{pub}, Q_{IDi}, H_1, H_2, H_3, \hat{e}), \quad \textbf{(1.2.1)}$$

tuo tarpu slaptasis raktas yra

$$SK = s. \quad \textbf{(1.2.2)}$$

Tarkime, vartotojas U_i nori įstoti į grupę ir tapti šios grupės nariu. Būtina sąlyga – slaptas ir neviešinamas tarpininkavimas tarp bet kurio grupės vartotojo ir grupės administratoriaus. Norėdamas gauti grupės nario sertifikatą, vartotojas U_i privalo įvykdyti šį protokolą:

- Vartotojas U_i siunčia ID_i grupės administratoriui,
- Grupės administratorius apskaičiuoja

$$S_i = s * Q_{IDi} \quad \textbf{(1.2.3)}$$

ir persiunčia gautąjį S_i vartotojui U_i.

Pasirašymas.

ID_i yra viešasis parametras šio RSA parašo tipo. Viešieji/privatieji raktai žymimi (ID_i, d_i). Pirmiausia vartotojas U_i pasirašo žinutę m ϵ {0, 1)* su savo RSA privačiuoju raktu di, t.y.

$$SigRSA = m^{d_i} (mod\ n), \quad \textbf{(1.2.4)}$$

kur n yra RSA-tipo modulis. Tada grupės narys Ui gali sugeneruoti anoniminį ir nesusekamą grupinį parašą prie žinutės m ϵ {0, 1}* tokiu būdu:

- Pasirinkti atsitiktinį k ϵ $\mathbb{Z}_q^*$.

- Apskaičiuoti

$$(R, S) \epsilon G_1 \times G_2, \quad \textbf{(1.2.5)}$$

kur

$$R = k * P, \quad \textbf{(1.2.6)}$$

$$S = k^{-1} (H_2 (m) * P + H_3 (R) * S_i), \quad \textbf{(1.2.7)}$$

kur k^{-1} yra atvirkščias dydžiui k.

- Grupinis parašas Sig yra sąryšis anksčiau sugeneruotų parašų SigRSA ir (R, S) su nario U_i viešuoju raktu ID_i

$$Sig = m^{d_i}(mod\ n)||x_R||x_S||ID_i, \quad \textbf{(1.2.8)}$$

kur x_R yra x-toji R koordinatė, o x_S yra S x-toji koordinatė.

Patikrinimas.

Pirma, tikrintojas patikrina, ar parašas yra sugeneruotas iš grupės. Tam jis tikrina, ar pora (R, S) yra teisingas parašas žinutei m ϵ {0, 1}* su atitinkamu viešuoju raktu U_{IDi}:

- Apskaičiuoja ê (U, V), kur (U, V) yra tariamas žinutės m parašas.
- Patikrina, ar

$$\hat{e}(U,V) = \hat{e}(P,P)^{H_2(m)} * \hat{e}(P_{pub}, Q_{IDi})^{H_3(R)}. \quad \textbf{(1.2.9)}$$

- Parašas priimamas, jei G_2 reikšmės sutampa, ir atmetamas visais kitais atvejais.

Jeigu (R, S) yra teisingas parašas žinutei m, tikrintojas tikrina, ar narys U_i yra priimtas į grupę tokiu būdu:

$$\hat{e}(R,S) = \hat{e}\left(k * P, k^{-1}(H_2 m * P + H_3(R) * S_i)\right) = \hat{e}(P, H_2(m) * P + H_3(R) * S_i) = \hat{e}(P,P)^{H_2(m)} * \hat{e}(P_{pub}, Q_{IDi})^{H_3(R)}$$

. **(1.2.10)**

Antra, tikrintojas tikrina, ar parašas buvo sugeneruotas kažkokio grupės nario, bet ne grupės administratoriaus, ar pasirašančiojo U_i viešasis raktas ID_i atitinka RSA parašą ir įsitikina, kad SigRSA yra teisingas:

$$m = SigRSA^{IDi}\ (mod\ n). \quad \textbf{(1.2.11)}$$

Kadangi grupės administratorius nežino privataus rakto d_i, jis nėra pajėgus sugeneruoti teisingą SigRSA.

Atidarymas.

Grupės administratorius žino kiekvienam ID_j atitinkantį narį U_j. Jam tai yra žinoma dar iš tos fazės, kai narys prisijungia prie grupės, todėl grupės administratoriui, turint žinutę m ir galiojantį grupinį parašą Sig, nėra sunku atsekti, kuris narys pasirašė žinutę.

1.3 ŽIEDINIŲ PARAŠŲ APŽVALGA

1.3.1 KLASIKINIS ŽIEDINIS PARAŠAS

Pirmą kartą žiedinio parašo terminas pavartotas 2001 metais [1]. Straipsnyje aprašomos žiedinio parašo generavimo procedūros (6 žingsniai) ir tikrinimo procedūros (4 žingsniai):

Žiedinio parašo generavimas:

Duoti laiškas m, kuris yra pasirašomas, pasirašančiojo privatus raktas S_s ir vieši viso žiedo narių parašai $P_1, P_2, ..., P_r$:

1. Pasirašantysis apskaičiuoja simetrinį raktą k pasinaudodamas maišos funkcija:

$$k = h(m) \qquad \textbf{(1.3.1.1)}$$

2. Atsitiktinai pasirenkama iniciacijos reikšmė v.
3. Pasirašantysis atsitiktinai pasirenka x_i iš visų žiedo narių $1 \leq i \leq r$, $i \neq s$ ir apskaičiuoja:

$$y_i = g_i(x_i). \qquad \textbf{(1.3.1.2)}$$

4. Pasirašantysis išsprendžia duotąją lygtį apskaičiuotajam y_i:

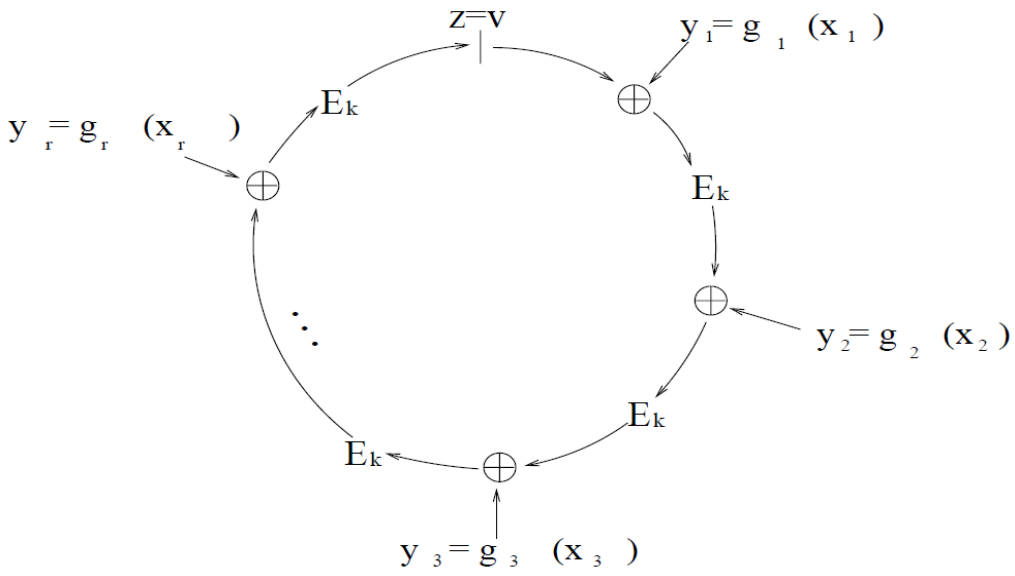
$$C_{k,v}(y_1, y_2, ..., y_r) = v. \qquad \textbf{(1.3.1.3)}$$

5. Pasirašantysis apskaičiuoja x_s:

$$x_s = g_s^{-1}(y_s). \qquad \textbf{(1.3.1.4)}$$

6. Pateikiamas pasirašytas m pranešimas yra (2r + 1) eilės:

$$(P_1, P_2, ..., P_r; v; x_1, x_2, ..., x_r). \qquad \textbf{(1.3.1.5)}$$



1.3.1.1 pav. Klasikinio žiedinio parašo generavimo schema

Žiedinio parašo tikrinimas.

Duoti parametrai (P_1, P_2, ..., P_r; v; x_1, x_2, ..., x_r). Tikrintojas gali patikrinti, ar parašas teisingas, atlikdamas šiuos žingsnius:

1. Kiekvienam i = 1, 2, ..., r tikrintojas apskaičiuoja:

$$y_i = g_i(x_i). \qquad \textbf{(1.3.1.6)}$$

2. Tikrintojas apskaičiuoja simetrinį raktą k pasinaudodamas maišos funkcija:

$$k = h(m). \qquad \textbf{(1.3.1.7)}$$

3. Tikrintojas išsprendžia lygtį:

$$C_{k,v}(y_1, y_2, ..., y_r) = v. \qquad \textbf{(1.3.1.8)}$$

Jeigu lygtis teisinga, parašas priimamas kaip geras ir galiojantis. Visais kitais atvejais parašas laikomas blogu.

1.3.2 SLENKSTINIS ŽIEDINIS PARAŠAS

Po šio klasikinio žiedinio parašo paskelbimo atsirado didelis susidomėjimas žiediniais parašais: pamatyta jų pritaikymo sritis, pradėti kurti nauji žiedinio parašo modeliai. Sukurta tokių žiedinio parašo modifikacijų, kurios priartėja prie grupinių parašų, pvz., slenkstiniai žiediniai parašai [2]. Yra keletas šio tipo žiedinių parašų modifikacijų, tačiau bendra slenkstinio žiedinio parašo schema yra tokia:

Slenkstinio žiedinio parašo generavimas:

1. Žiedinio parašo administratorius, kurio pareiga išlaikyti visų žiedinio parašo narių anonimiškumą, paruošia slenkstinį žiedinį parašą, naudodamas tokias procedūras: kiekvienam i ϵ {t + 1, ..., n} parenka x_i ir h_i ϵ_R $\mathbb{Z}_q^*$, apskaičiuoja

$$U_i = x_i P - h_i P_{pub} \quad \textbf{(1.3.2.1)}$$

$$V_i = x_i Q_{IDi}. \quad \textbf{(1.3.2.2)}$$

2. Kiekvienam j ϵ {1, ..., t} kiekvienas pasirašantysis ID_j atsitiktinai parenka r_j ϵ_R $\mathbb{Z}_q^*$ ir apskaičiuoja

$$U_j = r_j P. \quad \textbf{(1.3.2.3)}$$

3. Bet kuris pasirašantysis iš t žiede esančių pasirašančiųjų apskaičiuoja

$$h_0 = H_0(L,t,m,U_{k=1}^n\{U_k\}) \quad \textbf{(1.3.2.4)}$$

ir sukonstruoja n-t eilės daugianarį f tokį, kad kiekvienam t + 1 ≤ i ≤ n

$$f(0) = h_0 \quad \textbf{(1.3.2.5)}$$

ir

$$f(i) = h_i. \quad \textbf{(1.3.2.6)}$$

4. Kiekvienam j ϵ {1, ..., t} kiekvienas pasirašantysis ID_j apskaičiuoja

$$h_j = f(j) \quad \textbf{(1.3.2.7)}$$

ir

$$V_j = r_j Q_{IDj} + h_j S_{IDj}. \quad \textbf{(1.3.2.8)}$$

5. Bet kas iš t pasirašančiųjų apskaičiuoja

$$V = \sum_{k=1}^n V_k. \quad \textbf{(1.3.2.9)}$$

6. Pateikiamas parašas m ir L kaip

$$\sigma = (U_{k=1}^n\{U_k\},V,f). \quad \textbf{(1.3.2.10)}$$

Slenkstinio žiedinio parašo tikrinimas:

1. Tikrintojas tikrina, ar polinomas f yra n-t eilės ir ar $H_0(L,t,m,U_{k=1}^n\{U_k\})$ yra pastovus f dydis. Tik jeigu abi sąlygos tenkinamos, tikrinama toliau.

2. Kiekvienam k ϵ (1, ..., n) apskaičiuojamas

$$h_k = f(k). \quad \textbf{(1.3.2.11)}$$

3. Patikrinama, ar

$$\prod_{k=1}^n \hat{e}\left(Q_{ID_k}, U_k + h_k P_{pub}\right) = \hat{e}(P,V). \quad \textbf{(1.3.2.12)}$$

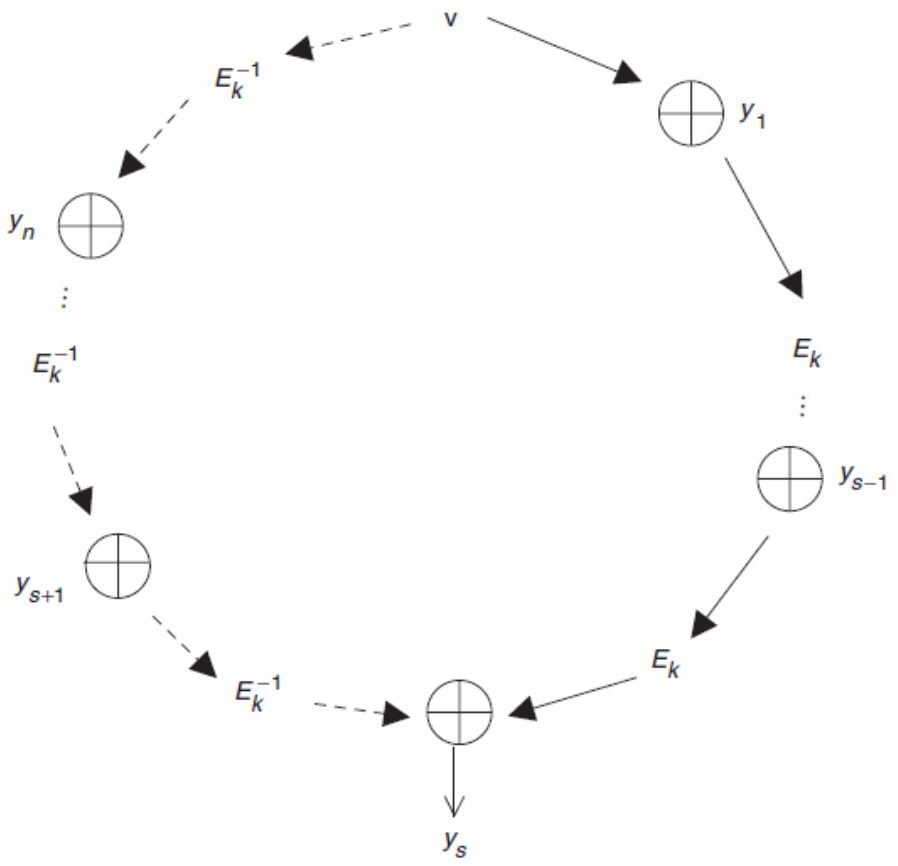
Jei lygybė teisinga, parašas laikomas geru. Jeigu ne, parašas atmetamas.

1.3.3 KONVERTUOJAMAS ŽIEDINIS PARAŠAS

Kartais egzistuoja tokios situacijos, kai pasirašančiajam verta atskleisti savo tapatybę. Tokiais atvejais neužtenka vien prisipažinimo, kad žiedinį parašą suformavo būtent tas žmogus, nes bet kas iš žiedinio parašo narių gali prisiimti atsakomybę. Atsižvelgiant į tai, buvo sukurti konvertuojami žiediniai parašai [3], t.y. tokie žiediniai parašai, kuriuose pasirašantysis yra nežinomas, tačiau, jeigu būtina, pats pasirašantysis gali įrodyti, kad parašas yra sukurtas jo, o ne kurio nors kito iš žiedinio parašo narių. Skirtingai nei anksčiau minėtų modelių, šio tipo žiedinių parašų schema susideda iš 4 etapų: žiedinio parašo generavimo, žiedinio parašo tikrinimo, žiedinio parašo konvertavimo ir žiedinio konvertuoto parašo tikrinimo.

Konvertuojamo žiedinio parašo generavimas.

Duota žinia m, kurią reikia pasirašyti, n – 1 viešųjų raktų, pasirašantysis A_s naudoja savo privatųjį raktą d_s ir viešąjį raktą A_s.



1.3.3.1 pav. Konvertuojamo žiedinio parašo schema

Pasirašantysis A_s atsitiktinai parenka visų narių padėtis žiede pagal atitinkamai tų narių viešuosius raktus A_1, A_2, ..., A_n. Tada pasirašantysis A_s pasinaudoja savo privačiu raktu d_s, visų narių (įskaitant ir savo) viešaisiais raktais ir generuoja žiedinį parašą S‘ žinutei m tokiu būdu:

1. Kiekvienam i = 1, 2, ..., n, i ≠ s, pasirašantysis A_s atsitinktinai pažymi

$$x_i = Q_i N_i + c_i \qquad \textbf{(1.3.3.1)}$$

ir apskaičiuoja

$$y_i = g_i(x_i) = \begin{cases} Q_i N_i + f_i(c_i), & jei\ (Q_i + 1)N_i \le 2^l \\ x_i & ,kitu\ atveju \end{cases} \qquad \textbf{(1.3.3.2)}$$

2. Pasirašantysis A_s atsitiktinai pasirenka reikšmę r ϵ {0, 1}[1], šią reikšmę saugo ir neatskleidžia jos niekam bei apskaičiuoja

$$t = h\ (x_1, ..., x_{s-1}, x_{s+1}, ..., x_n, r). \qquad \textbf{(1.3.3.3)}$$

3. Pasirašantysis As apskaičiuoja

$$h = h(m \,||\, t), \qquad \textbf{(1.3.3.4)}$$

kur m || t reiškia m ir t sąryšį. Panaudotas k yra simetrinis raktas, panaudotas simetriniam šifravimui $E_k()$ ir atvirkštiniam šifravimui E_k^{-1}.

4. Pasirašantysis A_s atsitiktinai parenka inicializacijos reikšmę v ir apskaičiuoja y_s, kur:

$$\alpha = E_k^{-1}\ (y_{s+1} \oplus E_k^{-1}\ (y_{s+2} \oplus E_k^{-1}\ (... \oplus E_k^{-1}(v)))) \qquad \textbf{(1.3.3.5)}$$

$$\beta = E_k\ (y_{s-1} ... \oplus E_k\ (y_2 \oplus E_k\ (y_1 \oplus (v)))\ ) \qquad \textbf{(1.3.3.6)}$$

$$y_s = \alpha \oplus \beta \qquad \textbf{(1.3.3.7)}$$

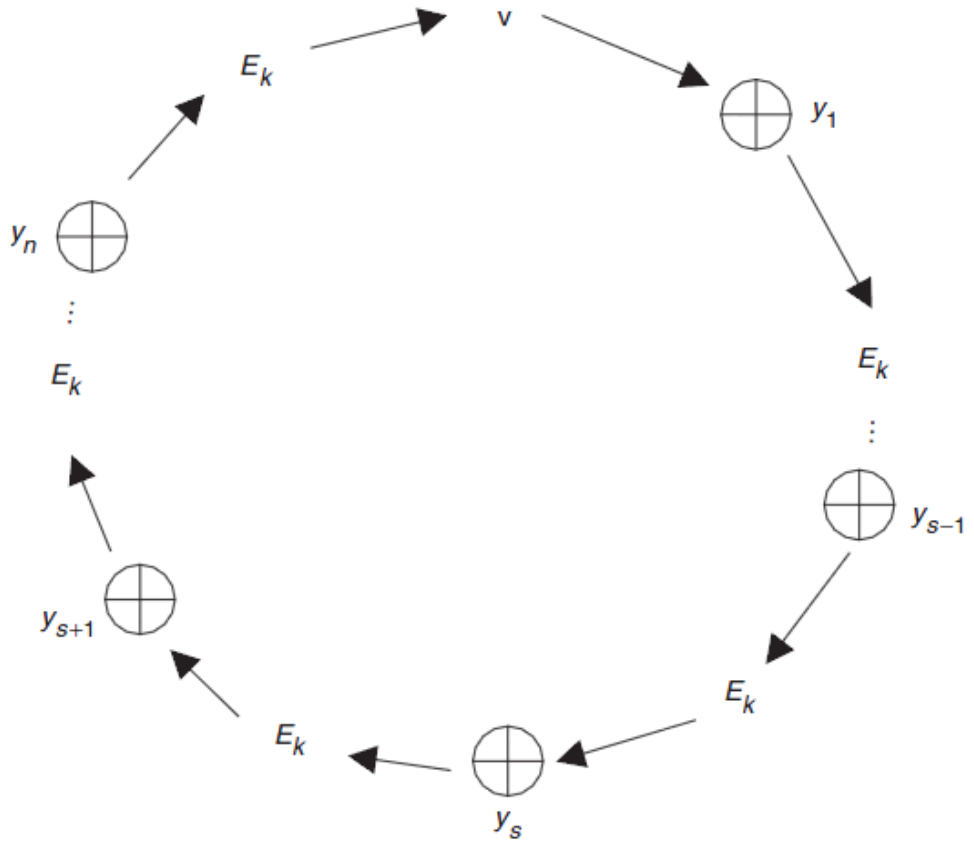
Pažymėtina, kad α yra apskaičiuojama prieš laikrodžio rodyklę, β apskaičiuojama pagal laikrodžio rodyklę, o y_s yra α ⊕ β.

5. Pasirašantysis A_s panaudoja savo privatųjį raktą d_s pasinaudodamas y_s ir vienpuse funkcija g_s (kurios atvirkštinę funkciją gali rasti tik jis) ir randa x_s:

$$x_s = g_s^{-1}(y_s) = \begin{cases} Q_s N_s + f_s^{-1}(y_s - Q_s N_s), & jei\ (Q_s + 1)N_s \le 2^l \\ x_s, & kitu\ atveju \end{cases} \qquad \textbf{(1.3.3.8)}$$

ir baigia generuoti konvertuojamą žiedinį parašą

$$S‘ = (e_1, ..., e_n, v, x_1, ..., x_n, t). \qquad \textbf{(1.3.3.9)}$$



1.3.3.2 pav. Konvertuojamo žiedinio parašo pasirašymo schema

Kovertuojamo žiedinio parašo tikrinimas:

1. Tikrintojas apskaičiuoja

$$y_i = g_i(x_i) = \begin{cases} Q_i N_i + f_i(c_i), & ,jei\ (Q_i + 1)N_i \leq 2^l \\ x_i & ,kitu\ atveju \end{cases} \quad \textbf{(1.3.3.10)}$$

2. Tikrintojas pasinaudoja maišos funkcija h ir apskaičiuoja m || t, norėdamas gauti simetrinį raktą

$$k = h\ (m\ ||\ t). \quad \textbf{(1.3.3.11)}$$

3. Tikrintojas patikrina, ar lygybė teisinga kryptimi „pagal laikrodžio rodyklę“:

$$v = E_k\ (y_n \oplus \ldots \oplus E_k\ (y_2 \oplus E_k\ (y_1 \oplus (v)))). \quad \textbf{(1.3.3.12)}$$

Jeigu lygybė teisinga, parašas laikomas geru, ir tikrintojas žino, kad jį pasirašė vienas iš žiedo narių A_1, A_2, ..., A_n, tačiau nežino, kuris tiksliai.

Konvertuojamo žiedinio parašo konvertavimas.

Jei pasirašantysis As nori atskleisti, kad tai jis iš žiedo narių pasirašė konvertuojamą žiedinį parašą S‘, jam reikia konvertuoti žiedinį parašą į įprastą elektroninį parašą, atskliedžiant (x_1, ..., x_{s-1}, x_{s+1}, ..., x_n, r).

Konvertuoto žiedinio parašo tikrinimas.

Su turima informacija (x_1, ..., x_{s-1}, x_{s+1}, ..., x_n, r), bet kas iš žiedo narių gali patikrinti, ar pasirašantysis yra A_s. Pirmiausia bet kuris tikrintojas turi patikrinti, ar

$$(x_1, ..., x_{s-1}, x_{s+1}, ..., x_n) \subset S'. \qquad \textbf{(1.3.3.13)}$$

Jei

$$(x_1, ..., x_{s-1}, x_{s+1}, ..., x_n) \not\subset S', \qquad \textbf{(1.3.3.14)}$$

tikrintojas nustoja tikrinti parašą ir priima jį kaip neteisingą. Priešingu atveju, tikrintojas žiūri, ar teisinga lygybė

$$t = h\,(x_1, ..., x_{s-1}, x_{s+1}, ..., x_n, r). \qquad \textbf{(1.3.3.15)}$$

Jeigu lygybė teisinga, tikrintojas gali atsekti pasirašiusįjį A_s lygindamas (x_1, ..., x_{s-1}, x_{s+1}, ..., x_n) su (x_1, ..., x_n). Trūkstamoji dalis x_s parodo, kad A_s yra parašo autorius. Jeigu A_s atskleistų reikšmę r bet kuriam žiedinio parašo nariui A_l, $l \neq s$, A_l, vis tiek negalėtų įrodyti, kad parašo autorius yra jis.

1.3.4 ATSEKAMAS ŽIEDINIS PARAŠAS

Jeigu žiedinis parašas pasirenkamas naudoti tokiose srityse, kaip balsavimas, pravartu naudoti atsekamąjį žiedinį parašą [4]. Šio tipo žiedinio parašo esmė: kiekvienas žiedo narys gali naudoti tokį parašą ir likti anoniminis, tačiau jis tai gali daryti tik kartą. Panaudojęs šį parašą daugiau nei vieną kartą per tam tikrą laiko intervalą (arba tag‘ą), šis narys yra identifikuojamas kaip pasirašiusysis. Šis principas sukurtas būtent tokiems atvejams, kaip balsavimas: bet kuris žiedo narys gali balsuoti ir nebijoti, kad kažkas seka ir žino, už ką būtent balsavo konkretus žiedo narys. Jeigu balsuoja visi žiedo nariai, kurių kiekis yra n, ir yra tik 2 balsavimo variantai, tai iš išorės yra matoma, kad k žiedo narių balsavo už opciją A, o n – k žiedo narių balsavo už opciją B, tačiau niekas nežino, už ką balsavo i-tasis žiedo narys. Jeigu atsiranda nesąžiningų žiedo narių, kurie nori balsuoti daugiau nei kartą, atsekamo žiedinio parašo sistema sudaro sąlygas identifikuoti nesąžiningą žiedo narį.

Tegul G yra multiplikatyvinė didelio pirminio skaičiaus q eilės grupė, o g yra grupės G generatorius. Tegul

$$H: \{0, 1\}^* \rightarrow G, \quad \textbf{(1.3.4.1)}$$

$$H': \{0, 1\}^* \rightarrow G \quad \textbf{(1.3.4.2)}$$

$$H'': \{0, 1\} \rightarrow \mathbb{Z}_q \quad \textbf{(1.3.4.3)}$$

yra atskiros maišos funkcijos. Tai yra viešieji modelio parametrai.

Rakto generavimas nariui i yra toks: narys i atsitiktinai pasirenka skaičių x_i iš $\mathbb{Z}_q$ ir apskaičiuoja

$$y_i = g^{xi}. \quad \textbf{(1.3.4.4)}$$

Viešasis i nario raktas yra $pk_i = \{g, y_i, G\}$, o atitinkantis privatusis raktas yra

$$sk_i = \{pk_i, x_i\}. \quad \textbf{(1.3.4.5)}$$

Narys i užregistruoja savo viešąjį raktą atsekamo žiedinio parašo raktų registre.

Pažymėkime $N = \{1, ..., n\}$ eiliškas sąrašas n narių. Atitinkamai $pk_N = (pk_1, ..., pk_n)$ – eiliškas n narių viešųjų raktų sąrašas. Pasirenkama parametro issue reikšmė sutartinai iš $\{0, 1\}^*$.

Atsekamojo žiedinio parašo generavimas.

Norint pasirašyti žinutę $m \in \{0, 1\}^*$, atsižvelgiant į tagą

$$L = (issue, pk_N), \quad \textbf{(1.3.4.6)}$$

naudojamas pasirašančiojo slaptas raktas sk_i tokiu būdu:

1. Pasirašantysis apskaičiuoja

$$h = H(L) \quad \textbf{(1.3.4.7)}$$

$$\sigma_i = h^{x_i}, \quad \textbf{(1.3.4.8)}$$

naudodamas $x_i \in \mathbb{Z}_q$.

2. Pasirašantysis nustato

$$A_0 = H'(L, m) \quad \textbf{(1.3.4.9)}$$

$$A_1 = (\sigma_i / A_0)^{1/i}. \quad \textbf{(1.3.4.10)}$$

3. Kiekvienam $j \neq i$ pasirašantysis apskaičiuoja

$$\sigma_j = A_0 A_1^j \in G. \quad \textbf{(1.3.4.11)}$$

4. Pasirašantysis generuoja parašą (c_N, z_N) ant (L, m) tokiu būdu:

$\Upsilon = \{(L, h, \sigma_N)) \mid$ taip, kad egzistuoja toks $i' \in N$, kad $\log_g (y_{i'}) = \log_h (\sigma_{i'})\}$. **(1.3.4.12)**

Čia $\sigma_N = (\sigma_1, ..., \sigma_n)$ randami tokiu būdu:

- Pasirašantysis pasirenka atsitiktinį $w_i \leftarrow \mathbb{Z}_q$ ir apskaičiuoja

$$a_i = g^{wi}, \quad \textbf{(1.3.4.13)}$$

$$b_i = h^{wi.} \quad \textbf{(1.3.4.14)}$$

- Pasirašantysis pasirenka atsitiktinius zj, cj ← $\mathbb{Z}_q$ ir apskaičiuoja

$$a_j = g^{z_j} y_i^{c_j}, \quad \textbf{(1.3.4.15)}$$

$$b_j = h^{z_j} \sigma_j^{c_j} \quad \textbf{(1.3.4.16)}$$

kiekvienam j ≠ i.

- Pasirašantysis apskaičiuoja

$$c = H''(L, A_0, A_1, a_N, b_N), \quad \textbf{(1.3.4.17)}$$

kur

$$a_N = (a_1, ..., a_n) \quad \textbf{(1.3.4.18)}$$

$$b_N = (b_1, ..., b_n). \quad \textbf{(1.3.4.19)}$$

- Pasirašantysis apskaičiuoja

$$c_i = c - \sum j \neq i \, c_j (mod\ q) \quad \textbf{(1.3.4.20)}$$

$$z_i = w_i - c_i x_i \ (mod\ q). \quad \textbf{(1.3.4.21)}$$

Grąžinama reikšmė (c_N, z_N), kur

$$c_N = (c_1, ..., c_n) \quad \textbf{(1.3.4.22)}$$

$$z_N = (z_1, ..., z_n) \quad \textbf{(1.3.4.23)}$$

yra įrodymas ϒ.

5. Pasirašantysis pateikia σ = (A_1, c_N, z_N) kaip parašą ant (L, m).

Atsekamojo žiedinio parašo tikrinimas:

Norėdamas patikrinti, ar parašas σ = (A_1, c_N, z_N) su žinute m ir tagu L yra teisingas, tikrintojas atlieka šiuos veiksmus:

1. Tikrintojas išnagrinėja L kaip (issue, pk_N). Patikrina g, A_1 ϵG, c_i, z_i ϵ $\mathbb{Z}_q$ ir y_i ϵ G kiekvienam i ≠ j. Apskaičiuoja

$$h = H(L) \quad \textbf{(1.3.4.24)}$$

$$A_0 = H'(L, m), \quad \textbf{(1.3.4.25)}$$

ir apskaičiuoja

$$\sigma_i = A_0 A_1^{\,i} \epsilon G \quad \textbf{(1.3.4.26)}$$

kiekvienam i ϵ N.

2. Tikrintojas apskaičiuoja

$$a_i = g^{z_i} y_i^{c_i} \quad \textbf{(1.3.4.27)}$$

$$b_i = h^{z_i}\sigma_i^{c_i} \quad \textbf{(1.3.4.28)}$$

kiekvienam i ϵ N.

3. Tikrintojas patikrina, ar

$$H''(L, m, A_0, A_1, a_N, b_N) \equiv \sum i \in N\, c_i (mod\ q), \quad \textbf{(1.3.4.29)}$$

kur

$$a_N = (a_1, ..., a_n) \quad \textbf{(1.3.4.30)}$$

$$b_N = (b_1, ..., b_n). \quad \textbf{(1.3.4.31)}$$

4. Jei visos anksčiau minėtos sąlygos tenkinamos, pasirašantysis parašą priima kaip teisingą, priešingu atveju pasirašantysis parašą atmeta.

Žiedinio parašo atsekimas.

Norint patikrinti sąryšį tarp (m, σ) ir (m‘, σ‘) su tagu L, kur

$$\sigma = (A_1, c_N, z_N) \quad \textbf{(1.3.4.32)}$$

$$\sigma' = (A_1', c_N', z_N') \quad \textbf{(1.3.4.33)}$$

reikia atlikti šiuos veiksmus:

1. Išnagrinėti L kaip (issue, pk_N). Apskaičiuoti

$$h = H(L), \quad \textbf{(1.3.4.34)}$$

$$A_0 = H'(L, m) \quad \textbf{(1.3.4.35)}$$

$$\sigma_i = A_0 A_1^{\,i} \in G \quad \textbf{(1.3.4.36)}$$

kiekvienam i ϵ N. Tą pačią procedūrą atlikti su σ‘ ir gauti σ_i‘ kiekvienam i ϵ N.

2. Kiekvienam i ϵ N, jei $\sigma_i = \sigma_i$‘, patalpinti pk_i reikšmę į Tlist sąrašą, kuris initializavimo momentu yra tuščias.

3. Išvesti pk reikšmę, jeigu sąraše Tlist yra ši reikšmė.

1.4 PORAVIMO FUNKCIJOS

Matematikoje poravimo funkcijos (pairing fuctions) naudojamos turint tikslą iš dviejų naturaliųjų skaičių užkodavimo metodu gauti vieną. Visuose aprašytuose žiediniuose parašuose yra naudojamos poravimo funkcijos. Daugeliu atveju žiediniuose parašuose naudojamos poravimo funkcijos yra neskelbiamos, o tiesiog aprašomos šių funkcijų savybės, kaip buvo padaryta [1], [2], [3], [4], [5], [6], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18].

Tegul G_1 yra adityvinė grupė su dideliu pirminiu skaičiumi q, o G_2 – multiplikatyvinė grupė su tuo pačiu skaičiumi q. Poravimo funkcija žymima ê: G_1 x $G_1 \rightarrow G_2$, taip, kad tenkintų šias sąlygas:

1. Dvilinijiškumas (bilinearity):
   - $\hat{e}(P + Q, R) = \hat{e}(P, R)\,\hat{e}(Q, R)$. **(1.4.1)**
   - $\hat{e}(P, Q + R) = \hat{e}(P, Q)\,\hat{e}(P, R)$. **(1.4.2)**
   - $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ **(1.4.3)**
2. Atsparumas:

   Egzistuoja tokie P, Q ϵ G1, kad

   $$\hat{e}(P,Q) \neq I_{G_2}\,, \qquad \textbf{(1.4.4)}$$

   kur I_{G_2} yra grupės G_2 tapatybės elementas.

3. Apskaičiuojamumas:

   Egzistuoja efektyvus algoritmas apskaičiuoti ê (P, Q) kiekvienam P, Q ϵ G_1.

Pateiksime paprasčiausią poravimo funkcijos pavyzdį [30].

Kantoro poravimo funkcija.

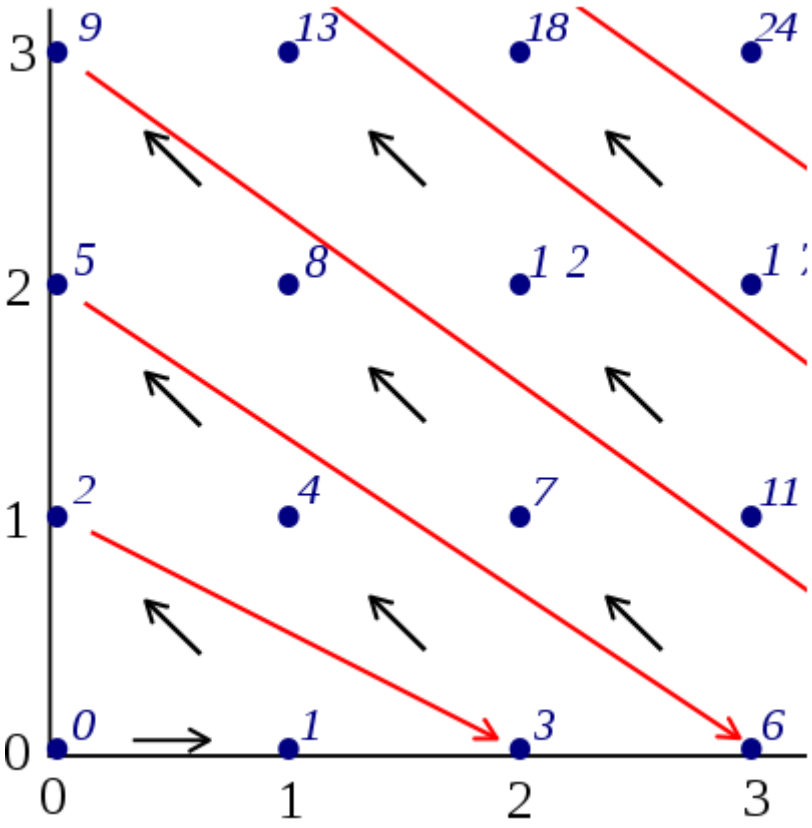
Kantoro poravimo funkcija žymima raide π :

$$\pi: N \times N \rightarrow N \qquad \textbf{(1.4.5)}$$

ir yra apibrėžiama kaip

$$\pi(k_1, k_2) = 1/2\,(k_1 + k_2)(k_1 + k_2 + 1) + k_2. \qquad \textbf{(1.4.6)}$$

Įstačius reikšmes k_1 ir k_2 į Kantoro poravimo funkciją, gaunamas rezultatas ‹k_1, k_2›.



1.4 pav. Kantoro poravimo funkcijos elementų porai priskiriamo elemento schema

Kantoro poravimo funkcija yra rekursinė funkcija:

$$\pi^{(n)} : N^n \rightarrow N \qquad \textbf{(1.4.7)}$$

yra lygu

$$\pi(n)\ (k_1, \ldots, k_{n-1}, k_n) = \pi\ (\pi^{(n-1)}\ (k_1, \ldots, k_{n-1}), k_n). \qquad \textbf{(1.4.8)}$$

Atvirkštinė Kantoro poravimo funkcija.

Tarkim

$$z = \langle x, y \rangle = \frac{(x+y)(x+y+1)}{2} + y \qquad \textbf{(1.4.9)}$$

ir reikia rasti x ir y reikšmes.

$$w = x + y. \qquad \textbf{(1.4.10)}$$

$$t = \frac{w\,(w+1)}{2} = \frac{w^2+w}{2}. \qquad \textbf{(1.4.11)}$$

$$z = t + y. \qquad \textbf{(1.4.12)}$$

Norint rasti w reikšmes, reikia išspręsti lygybę

$$w^2 + w - 2t = 0. \qquad \textbf{(1.4.13)}$$

Iš čia gauname

$$w = \frac{\sqrt{8t+1}-1}{2} \quad \textbf{(1.4.14)}$$

didėjančia ir tolygią funkciją. Kadangi

$$t \leq z = t + y < t + (w + 1) = \frac{(w+1)^2 + (w+1)}{2}, \quad \textbf{(1.4.15)}$$

gauname

$$w \leq \frac{\sqrt{8z+1}-1}{2} < w + 1 \quad \textbf{(1.4.16)}$$

$$w = \left\lfloor \frac{\sqrt{8z+1}-1}{2} \right\rfloor \quad \textbf{(1.4.17)}$$

Taigi, norint apskaičiuoti x ir y vertes, kai turime z = ‹x, y›, reikia apskaičiuoti:

$$w = \left\lfloor \frac{\sqrt{8z+1}-1}{2} \right\rfloor \quad \textbf{(1.4.18)}$$

$$t = \frac{w^2+w}{2} \quad \textbf{(1.4.19)}$$

$$y = z - t \quad \textbf{(1.4.20)}$$

$$x = w - y. \quad \textbf{(1.4.21)}$$

1.4 paveiksle parodyta, kaip Kantoro poravimo funkcija vienam natūraliajam skaičiui priskiria kiekvieną natūraliųjų skaičių porą.

2. TIRIAMOJI DALIS

2.1 BENDROS PROBLEMOS

Diskretaus logaritmo problema.

Matematikoje diskretieji logaritmai yra teorinių grupių analogai paprastiems logaritmams. Pvz., įprasto logaritmas $\log_a b$ yra lygties

$$a^x = b \qquad \textbf{(2.1.1)}$$

sprendinys. Panašiai ir su diskrečiaisiais logaritmais – jeigu g ir h priklauso baigtinei grupei G, tai sprendinys x lygčiai

$$g^x = h \qquad \textbf{(2.2.2)}$$

yra vadinamas diskretaus logaritmo uždaviniu grupėje G.

Kriptografijoje dažnai naudojamos grupės G yra multiplikatyvinės grupės, kurias sudaro tik sveikieji skaičiai su moduliu n. Remiantis kriptografijos taisyklėmis, skaičius n privalo būti pirminis skaičius. Jeigu skaičius n nėra didelis pirminis skaičius, diskrečiojo logaritmo uždavinys tampa išsprendžiamas. Kadangi technologijos labai greitai tobulėja, uždavinys, kuris buvo neišspręndžiamas prieš 20 metų, gali būti be jokių problemų išspręndžiamas dabar. 1999m. buvo įrodyta, kad įmanoma praktiškai nulaužti 512 bitų ilgio RSA algoritmu paremtą raktą. Šią dieną tokio ilgio raktui dešifruoti prireiktų poros savaičių. 2010 metai didžiausias dešifruotas skaičius buvo 768 bitų ilgio. Jeigu įmanoma dešifruoti raktą, t.y. turint n rasti tokius p ir q, kad

$$n = (p - 1)(q - 1), \qquad \textbf{(2.2.3)}$$

galima išspręsti diskrečiojo logaritmo problemą. 1024 bitų ilgio skaičius n atrodo taip:

n =

119294134840169509055527211331255649644606569661527638012067481954943056851
150333806315957037715620297305000118628770846689969112892212245457118060574
995989517080042105263427376322274266393116193517839570773505632231596681121
927337473973220312512599061231322250945506260066557538238517575390621262940
383913963.

Jį atitinkantys p ir q atrodo šitaip:

p =

109337661836325758176115170347306682871557999846322234541387456711212734562876700082908433028755212749702453145932229461290645383585810186155398284791464469.

q = 109106169673491102317237340786149226453370608821417489682098342251389760111799933942998101597369044685540217082898243965534121805148279964448454381760 99727.

Jeigu rakto ilgis yra vos kelių bitų ilgio, ir jo ieškant sugaištama keletą minučių, tai joks žiedinis parašas nėra saugus ir parašo autorius labai lengvai gali būti surastas. Jeigu rakto ilgis yra bent 1024 bitų ilgio, tai šiuo metu žiedinio parašo autorių surasti „brutal force“ (grubios jėgos) metodu yra neįmanoma.

2005 metais buvo pranešta, kad teisingai išspręstas diskretaus logaritmo uždavinys, kai rakto ilgis buvo imamas 431 bitų ilgio. Tokiam skaičiavimui prireikė 3 savaičių, naudojant 1,15 GHz 16 procesorių HP AlphaServer GS1280 kompiuterį. 2009 buvo išspręstas diskretusis logaritmas elipsinėms kreivėms, kai raktas buvo parinktas 109 bitų ilgio. Skaičiavimas užtruko 549 dienas, skaičiavime dalyvavo 10308 namų lygio kompiuteriai.

Šie pavyzdžiai įrodo, kad norint išaiškinti, kas yra žiedinio parašo autorius, reikia, pasinaudojant turimais duomenimis, apeiti diskretaus logaritmo uždavinio sprendimą ir spragų ieškoti pačiose žiedinio parašo formulėse.

Siekiant išsiaiškinti, kokioms žiedinių parašų schemoms galima taikyti ataką, kurios metu forumo dalyviai ir žiedinio parašo nariai tarpusavyje bendradarbiaudami sugeba išaiškinti, kas yra tikrasis parašo autorius, buvo išnagrinėtos [1], [2], [3], [4], [5], [6], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] aprašytos schemos, tačiau žymesnių saugumo spragų buvo rasta tik [5] ir [6] schemose. Jos yra išanalizuojamos bei pateikiamos žemiau. Taip pat pateikiamas ir visas atakos aprašas.

2.2 NAGRINĖJAMA SCHEMA

Po Rivest, Shamir ir Tauman pristatytos žiedinio parašo koncepcijos atsirado daugybė skirtingų žiedinių parašų versijų, leidžiančių optimizuoti šifravimo bei dešifravimo greičius, naudoti papildomas funkcijas. Nemažai skirtingų žiedinių parašų versijų yra skirtos būtent saugumo stiprinimui ir užtikrinimui.

Nagrinėsime žiedinio parašo schemą, [5]

Šio žiedinio parašo schema susideda iš 4 protokolų: parengties (setup), raktų generavimo (keygen), pasirašymo (signcryption) ir tikrinimo (unsigncryption).

Parengties protokolas.

Parengties algoritmas vykdomas privačiųjų raktų generavimo centre. Įvedus saugumo parametrus k ir l, algoritmas veikia tokiu principu:

pasirenkama adityvinė grupė G_1 ir multiplikatyvinė grupė G_2 taip, kad abidvi būtų to paties didelio priminio skaičiaus q eilės. Poravimo funkcija ê parenkama tokia, kad

$$ê: G_1 \times G_1 \rightarrow G_2. \qquad \textbf{(2.2.1)}$$

Naudojamos trys skirtingos maišos funkcijos H_1, H_2 ir H_3. Pasirenkamas privačiųjų raktų generavimo centro slaptasis raktas $s_{prgc} \in_R \mathbb{Z}_q^*$ ir parenkamas viešasis privačiųjų raktų generavimo centro raktas

$$P_{pub} = s_{prgc}P, \qquad \textbf{(2.2.2)}$$

kur P yra grupės G_1 generatorius. Sistemos parametrai yra (G_1, G_2, ê, q, P, P_{pub}, H_1, H_2, H_3).

Raktų generavimo protokolas.

Privačiųjų raktų generavimo centras suteikia nariui u_i, kurio tapatybė yra ID_i, viešąjį bei privatųjį raktus ‹Q_i, D_i› tokiu būdu:

- Viešasis raktas Q_i apskaičiuojamas

$$Q_i = H_1 (ID_i) \in G_1. \qquad \textbf{(2.2.3)}$$

- Privatusis raktas D_i apskaičiuojamas

$$D_i = s_{prgc}Q_i. \qquad \textbf{(2.2.4)}$$

- Raktai tarp privačiųjų raktų generavimo centro ir vartotojo u_i yra perduodami slaptais bei patikimais kanalais.

Pasirašymo protokolas.

Tegul pasirašantysis būna žiedinio parašo narys, indeksuojamas s indeksu. Pasirašantysis ID_s turi žinoti žinutę m, žiedinio parašo narius $L = \{u_1, u_2, ..., u_n\}$, privalo pats priklausyti šiam žiedui, t.y. $u_s \in L$, žinoti savo privatųjį raktą D_i ir asmens, kuriam skirtas žiedinis parašas, tapatybę ID_B. Pasirašymas vyksta tokiu būdu:

- Pasirašantysis atsitiktinai pasirenka $r_0 \in_R \mathbb{Z}_q^*$, apskaičiuoja

$$R_0 = r_0 P, \quad \textbf{(2.2.5)}$$

$$R'_0 = \hat{e}\,(P_{pub}, Q_B)^{r0}, \quad \textbf{(2.2.6)}$$

$$k = H_2\,(R'_0) \quad \textbf{(2.2.7)}$$

$$c = m \oplus k\,. \quad \textbf{(2.2.8)}$$

- Visiems $i \neq s$, pasirašantysis atsitiktinai parenka $U_i \in G_1$ ir apskaičiuoja

$$h_i = H_3\,(m, k, L, U_i). \quad \textbf{(2.2.9)}$$

- Pasirašantysis atsitiktinai pasirenka $r_s \in_R \mathbb{Z}_q^*$, apskaičiuoja

$$U_s = X - \sum_{i \neq s}\{U_i + h_i Q_i\}, \quad \textbf{(2.2.10)}$$

kur

$$X = r_s Q_s. \quad \textbf{(2.2.11)}$$

- Pasirašantysis apskaičiuoja

$$h_s = H_3\,(m, k, L, U_s) \quad \textbf{(2.2.12)}$$

$$V = (h_s + r_s)\,D_s. \quad \textbf{(2.2.13)}$$

- Pasirašantysis siunčia žiedinį parašą

$$C = (L, c, R_0, h_1, h_2, ..., h_n, U_1, U_2, ..., U_n, V, X) \quad \textbf{(2.2.14)}$$

tikrintojui ID_B.

Tikrinimo protokolas.

Turimas žiedinis parašas (2.2.14) pasirašiusiojo ID_s tikrintojui ID_B. Tikrintojas turi žinoti savo slaptą raktą D_B. Tikrinimo algoritmas vykdomas tokiu būdu:

- Tikrintojas apskaičiuoja

$$k' = H_2\,(\hat{e}\,(R_0, D_B)). \quad \textbf{(2.2.15)}$$

- Tikrintojas atstato žinutę

$$m' = c \oplus k. \quad \textbf{(2.2.16)}$$

- Tikrintojas apskaičiuoja

$$h_i = H_3\,(m', k', L, U_i) \quad \textbf{(2.2.17)}$$

kiekvienam $i \in \{1, 2, ..., n\}$.

- Tikrintojas tikrina, ar teisinga lygybė

$$\hat{e}\,(P_{pub}, \sum_{i=1}^{n}(U_i + h_i Q_i)) = \hat{e}\,(P, V). \qquad \textbf{(2.2.18)}$$

- Jeigu kiekvienam i ϵ {1, 2, ..., n}, kiekvienam (2.2.17) yra teisinga lygybė (2.2.18) tikrintojas priima žinutę m kaip teisingą. Visais kitais atvejais parašas laikomas blogu ir žinutė m laikoma negaliojančia.

2.3 SAUGUMO ANALIZĖ

Tarkime, kad aukščiau aprašyta žiedinio parašo schema naudojama forume. Šiame forume lankosi daugybė įvairių žmonių iš skirtingų kompanijų. Tarkime, kompanijai X dirbantys forumo naudotojai turi galimybę naudoti žiedinius parašus, taip patvirtindami, kad iš tiesų dirba kompanijai X, t.y. kompanijos X serveriuose veikianti privačiųjų raktų generavimo programa yra išdavusi visiems kompanijos X nariams privačiuosius bei viešuosius raktus.

Nagrinėjama situacija tokia, kai vienas iš kompanijai X dirbančių asmenų pradeda naudoti žiedinius parašus skelbdamas melagingus teiginius, šmeižiančius kompaniją X. Kadangi žiediniuose parašuose nėra administratorių, kaip grupiniuose parašuose, dalis kompanijos X atstovų, tuo pačiu ir žiedinio parašo narių, nusprendžia, kad reikia demaskuoti melagingą žiedinio parašo narį.

Kadangi šio tipo žiedinis parašas yra skirtas konkrečiam asmeniui ir bet kas negali patvirtinti, ar parašas yra teisingas ar ne, būtina sąlyga, kad asmuo ID_B, kuriam yra adresuotas šis žiedinis parašas, paskelbtų viešai, ar parašas teisingas. Dar viena būtina sąlyga yra ta, kad iš žiedinio parašo narių būtų bent 2 nariai, kurie norėtų demaskuoti kompanijos X šmeižiką. Tačiau jeigu susidarytų tokia situacija, kad yra tik vienas žiedinio parašo narys, kuris nori demaskuoti šmeižiką, o kitas norintis jį demaskuoti būtų pats šmeižikas (apsimetantis norinčiu padėti, tačiau specialiai teikiantis klaidingą informaciją, ir šitaip bandantis likti neišaiškintas), yra rekomenduojama, kad bent 3 žiedinio parašo nariai norėtų išaiškinti šmeižiką.

Jeigu žiede atsiranda daugiau nei 3 asmenys, kurie nori išaiškinti parašo autorių, jokių papildomų privalumų nepastebima. Nuo to nesumažėja skaičiavimams skirtas laikas, nes surasti privačiųjų raktų generavimo centro privačiajam raktui visiškai pakanka 3 žiedo narių, kaip parodyta (2.3.4).

Tarkime, tikrasis parašo autorius ir ieškomas šmeižikas yra u_s, ir turime tris žiedinio parašo narius u_l, u_m ir u_n. Viešieji parametrai yra visų žiedinio parašo narių viešieji raktai (Q_1, Q_2, ..., Q_n), maišos funkcijos H_1, H_2 ir H_3, poravimo funkcija ê, privačiųjų raktų generavimo centro viešasis parametras P_{pub} ir žiedinis parašas $C = (L, c, R_0, h_1, h_2, ..., h_n, U_1, U_2, ..., U_n, V, X)$.

Norint išaiškinti, kuris narys yra žiedinio parašo autorius, reikia surasti tokius dydžius h_i ir Q_i, su kuriais būtų teisinga lygybė

$$\hat{e}(V, P) = \hat{e}(h_i Q_i + X, P_{pub}), \qquad \textbf{(2.3.1)}$$

čia $X = r_s Q_s$. h_i yra randamas taip:

$$h_i = H_3\,(m, k, L, U_i). \qquad \textbf{(2.3.2)}$$

Jeigu iš žiede esančių narių tik vienas nori surasti parašo autorių, jam nepavyks, nes jis nežino nei P, nei mokės apskaičiuoti h_i, kadangi nemoka rasti k.

Jeigu nariai u_l, u_m ir u_n nori išaiškinti parašo autorių, jie turi vienas kitam atskleisti savo privačiuosius raktus D_l, D_m ir D_n. Kadangi visiems žiedo nariams privačiųjų raktų generavimo centras privačiojo ir viešojo raktų poras generuoja su tuo pačiu privačiuoju raktu s:

$$D_i = s_{prgc}Q_i, \qquad \textbf{(2.3.3)}$$

tai nariai u_l, u_m ir u_n randa privatų privačiųjų raktų generavimo centro raktą s_{prgc}:

$$\frac{D_l}{Q_l} = \frac{D_m}{Q_m} = \frac{D_n}{Q_n} = s_{prgc}$$

(2.3.4)

Kaip jau minėta, rekomenduojama šį veiksmą atlikti bent 3 žiedo nariams, kadangi bent 2 nariai gaus tą patį s_{prgc}; iš esmės šį veiksmą būtų galima atlikti ir vienam žiedo nariui, tačiau jis negali būti tikras, ar jo gautas s_{prgc} yra teisingas.

Dabar nariai a_l, a_m ir a_n jau gali apskaičiuoti h_i, nes m, L ir U_i, kai i ϵ {1, 2, ..., n} reikšmės yra žinomos, o reikšmė k apskaičiuojama:

$$k = H_2\,(R_0\text{‘}). \qquad \textbf{(2.3.5)}$$

Nors R_0‘ yra nežinomas, tačiau

$$R_0\text{‘} = \hat{e}\,(P_{pub}, Q_B)^{r0}. \qquad \textbf{(2.3.6)}$$

Čia nežinomas yra tik r_0, tačiau jis yra apskaičiuojamas iš lygybės

$$R_0 = r_0P, \qquad \textbf{(2.3.7)}$$

o čia nežinomasis P yra apskaičiuojamas pasitelkus privačiųjų raktų generavimo centro privatųjį raktą s_{prgc}:

$$P = \frac{P_{pub}}{s_{prgc}}. \qquad \textbf{(2.3.8)}$$

Tada

$$r_0 = \frac{R_0}{P}, \qquad \textbf{(2.3.9)}$$

Galėdami suskaičiuoti h_i, a_l, a_m ir a_n galim išaiškinti žiedinio parašo autorių iš (3.1) lygybės, nes lygybė yra teisinga tik tuo atveju, jeigu ieškomasis Q_i yra parašo autoriaus viešasis raktas. Tai yra įrodoma taip:

Tegul

$$F_s = X + h_sQ_s, \qquad \textbf{(2.3.10)}$$

kai u_s yra tikrasis žiedinio parašo autorius. Tegul

$$T = \hat{e}\,(V, P). \qquad \textbf{(2.3.11)}$$

Tada

$T = \hat{e}\ (V, P) = \hat{e}\ ((r_s + h_s)\ D_s, P) = \hat{e}\ ((r_s + h_s)\ Q_s, P_{pub}) = \hat{e}(r_sQ_s + h_sQ_s, P_{pub}) = \hat{e}\ (X + h_sQ_s, P_{pub}) = \hat{e}\ (F_s, P_{pub})$. **(2.3.12)**

Tarkim

$$F_i = X + h_iQ_i, \quad \textbf{(2.3.13)}$$

kai i ≠ s, ir u_i nėra tikrasis parašo autorius. Tegul

$$T = \hat{e}\ (V, P). \quad \textbf{(2.3.14)}$$

Tada

$T = \hat{e}\ (V, P) = \hat{e}\ ((r_i + h_i)\ D_i, P) = \hat{e}\ ((r_i + h_i)\ Q_i, P_{pub}) \neq \hat{e}(r_sQ_s + h_iQ_i, P_{pub}) = \hat{e}\ (X + h_iQ_i, P_{pub}) = \hat{e}\ (F_i, P_{pub})$. **(2.3.15)**

Iš (2.3.13) ir (2.3.15) išplaukia, kad (2.3.1) lygybė teisinga tik tada, jeigu i = s, o tai reiškia, kad $u_i = u_s$.

Iš (2.3.1) lygybės radus Q_s, taip pat galima rasti ir D_s, kadangi

$$D_s = s_{prgc}Q_s. \quad \textbf{(2.3.16)}$$

Tokiu būdu, bent trims žiedinio parašo nariams atskleidus vienas kitam savo privačiuosius raktus, galima sužinoti parašo autoriaus privatųjį bei viešąjį raktus. Taigi, žiedinio parašo schema, pateikta [5], yra nepatikima. Tai paaiškinama tuo, kad ši schema buvo sukurta dirbti kaip galima greičiau:

2.3 lentelė

Žiedinio parašo schemų apskaičiavimo greičių palyginimas.

Schemos	G1 Add	G1 Mul	G2 Mul	Poravimas
Schema, pateikta [6]	3n - 1	2n + 2	n	n + 5
Schema, pateikta [5]	4n - 3	2n + 2	1	3

čia n – žiedinio parašo narių skaičius.

2.4 LYGINAMOJI ANALIZĖ

[5] yra greitesnis schemos variantas nei pateikta [6]. Siekiant įrodyti, kad greitumas sąlygoja saugumo spragas, atliksime lyginamąją šių dviejų schemų analizę.

Nagrinėsime žiedinio parašo schemą [6].

Šio žiedinio parašo schema susideda iš 3 protokolų: raktų generavimo (key generation), pasirašymo (signcryption) ir tikrinimo (unsigncryption).

Raktų generavimo protokolas.

Šis protokolas vykdomas privačiųjų raktų generavimo centre. Įvedus saugumo parametrus k ir l, algoritmas veikia tokiu principu:

pasirenkama adityvinė grupė G_1 ir multiplikatyvinė grupė G_2 taip, kad abidvi būtų to paties didelio priminio skaičiaus $q > 2^k$ eilės. Poravimo funkcija ê parenkama tokia, kad

$$\hat{e}: G_1 \times G_1 \rightarrow G_2. \qquad \textbf{(2.4.1)}$$

Naudojamos keturios skirtingos maišos funkcijos H_1, H_2, H_3 ir H_4. Pasirenkamas privačiųjų raktų generavimo centro slaptasis raktas $s \in_R \mathbb{Z}_q^*$ ir parenkamas viešasis privačiųjų raktų generavimo centro raktas

$$P_{pub} = s_{prgc}P, \qquad \textbf{(2.4.2)}$$

kur P yra grupės G_1 generatorius. Sistemos parametrai yra (G_1, G_2, ê, q, P, P_{pub}, H_1, H_2, H_3, H_4).

Privačiųjų raktų generavimo centras suteikia nariui u_i, kurio tapatybė yra ID_i, viešąjį bei privatųjį raktus $\langle Q_i, D_i \rangle$ tokiu būdu:

- Viešasis raktas Q_i apskaičiuojamas

$$Q_i = H_1(ID_i) \in G_1. \qquad \textbf{(2.4.3)}$$

- Privatusis raktas D_i apskaičiuojamas

$$D_i = s_{prgc}Q_i. \qquad \textbf{(2.4.4)}$$

- Raktai tarp privačiųjų raktų generavimo centro ir vartotojo u_i yra perduodami slaptais bei patikimais kanalais.

Pasirašymo protokolas.

Tegul pasirašantysis būna žiedinio parašo narys, indeksuojamas s indeksu. Pasirašantysis ID_s turi žinoti žinutę m, žiedinio parašo narius $L = \{u_1, u_2, ..., u_n\}$, privalo pats priklausyti šiam žiedui, t.y. $u_s \in L$, žinoti savo privatųjį raktą D_s ir asmens, kuriam skirtas žiedinis parašas, tapatybę ID_B. Pasirašymas vyksta tokiu būdu:

- Pasirašantysis atsitiktinai pasirenka $a_0 \in_R \mathbb{Z}_q^*$, $m_r \in_R M$ apskaičiuoja

$$R_0 = a_0 P, \quad \textbf{(2.4.5)}$$

$$R'_0 = \hat{e}\,(P_{pub}, Q_B), \quad \textbf{(2.4.6)}$$

$$k = H_2\,(R'_0) \quad \textbf{(2.4.7)}$$

$$c_1 = m_r \oplus k \quad \textbf{(2.4.8)}$$

$$c_1 = m \oplus H_3(m_r)\,. \quad \textbf{(2.4.9)}$$

- Visiems $i \neq s$, pasirašantysis atsitiktinai parenka $a_i \in_R \mathbb{Z}_q^*$. Apskaičiuoja

$$A_i = a_i P, \quad \textbf{(2.4.10)}$$

$$R_i = \hat{e}\,(A_i, P), \quad \textbf{(2.4.11)}$$

$$h_i = H_4\,(m, k, U, R_i). \quad \textbf{(2.4.12)}$$

- Pasirašantysis atsitiktinai pasirenka $a_s \in_R \mathbb{Z}_q^*$, apskaičiuoja

$$A_s = a_s P, \quad \textbf{(2.4.13)}$$

$$R_s = \hat{e}(A_s, P)\ \ \hat{e}\,(-P_{pub}, \textstyle\sum_{i \neq s} h_i Q_i). \quad \textbf{(2.4.14)}$$

Jei $R_s = 1_{G_2}$ arba $R_s = R_i$, kai $i \neq s$, tada reikia pakartoti 3 žingsnį, kol bus gautas R_s, tenkinantis (2.4.13) ir (2.4.14) sąlygas.

- Pasirašantysis apskaičiuoja

$$h_s = H_4\,(m, k, U, R_s), \quad \textbf{(2.4.15)}$$

$$\sigma = h_s D_s + \textstyle\sum_{i=1}^{n} A_i. \quad \textbf{(2.4.16)}$$

- Pasirašantysis siunčia žiedinį parašą

$$C = (U, c_1, c_2, R_0, R_1, \ldots, R_n, h_2, \ldots, h_n) \quad \textbf{(2.4.17)}$$

tikrintojui ID_B.

Tikrinimo protokolas.

Turimas žiedinis parašas C (2.4.17) nuo pasirašiusiojo ID_s tikrintojui ID_B. Tikrintojas turi žinoti savo slaptą raktą D_B. Tikrinimo algoritmas vykdomas tokiu būdu:

- Tikrintojas apskaičiuoja

$$k' = H_2\,(\hat{e}\,(R_0, D_B)). \quad \textbf{(2.4.18)}$$

- Tikrintojas atstato žinutes

$$m'_r = c_1 \oplus k', \quad \textbf{(2.4.19)}$$

$$m' = c_2 \oplus H_3\,(m'_r). \quad \textbf{(2.4.20)}$$

- Tikrintojas apskaičiuoja

$$h_i = H_4\,(m', k', U, R_i) \quad \textbf{(2.4.21)}$$

kiekvienam $i \in \{1, 2, \ldots, n\}$.

- Tikrintojas tikrina, ar teisinga lygybė

$$R_1 * R_2 * ... * R_n * \hat{e}\,(P_{pub}, \sum_{i=1}^{n} h_i Q_i) = \hat{e}\,(P, \sigma). \qquad \textbf{(2.4.22)}$$

- Jeigu kiekvienam i ϵ {1, 2, ..., n}, kiekvienam (2.4.21) lygybė (2.4.22) yra teisinga, tikrintojas priima žinutę m kaip teisingą. Visais kitais atvejais parašas laikomas blogu ir žinutė m laikoma negaliojančia.

Kaip ir [5] atveju, nagrinėjamas toks atvejis: pasirašantysis u_s forume šmeižia kompaniją X, kurios visus darbuotojus įtraukė į savo žiedinį parašą. Žiedinio parašo nariai ul, um ir un, kurie dirba kompanijai X ir yra forumo dalyviai, nori išaiškinti parašo autorių. Kadangi [5] ir [6] žiedinių parašų schemų privačiųjų raktų generavimo centro protokolai yra identiški, tai nariai u_l, u_m ir u_n randa privatų privačiųjų raktų generavimo centro raktą s_{prgc}:

$$\frac{D_l}{Q_l} = \frac{D_m}{Q_m} = \frac{D_n}{Q_n} = s_{prgc} \qquad \textbf{(2.4.23)}$$

Abiejų žiedinių parašų atvejais ši saugumo spraga yra išnaudojama, kad būtų galima rasti bet kurio žiedinio parašo nario privatųjį raktą, kai yra žinomas jo viešasis raktas. Tačiau, priešingai nei anksčiau minėtoje schemoje, šio žiedinio parašo atveju neįmanoma rasti, kuris narys yra parašo autorius, nes vietoje 3 maišos funkcijų kaip [5], čia yra naudojamos 4 maišos funkcijos. Šiuo atveju, nors skaičiavimams atlikti reikalingas laikas yra gerokai didesnis nei [5], tačiau užtikrinamas saugumas yra didesnis. Nepaisant to, kad neįmanoma rasti žiedinio parašo autoriaus, šios schemos naudojimas forumuose yra nerekomenduojamas.

2.4 lentelė

Žiedinio parašo schemų saugumo palyginimas.

Schemos:	[5]	[6]
Galimybė nustatyti privačiųjų raktų generavimo centro slaptąjį raktą s÷	Taip	Taip
Galimybė nustatyti bet kurio žiedinio parašo dalyvio slaptąjį raktą, kai yra žinomas jo viešasis raktas÷	Taip	Taip
Galimybė nustatyti, kuris žiedinio parašo narys yra parašo autorius	Taip	Ne
Rekomenduojama naudoti žiedinį parašą forumuose	Ne	Ne

IŠVADOS

Atlikus analizę, galima daryti tokias išvadas:

1. Sukurta žiedinio parašo narių bendradarbiavimo ataka yra sėkminga, kadangi įrodyta, jog parašo autorius gali būti demaskuotas.
2. Žiedinio parašo autorius gali būti demaskuotas tarpusavyje bendradarbiaujant bent trim žiedinio parašo nariams. Didesnis skaičius tarpusavyje norinčių bendradarbiauti žiedinio parašo narių nėra reikalingas.
3. Visos žiedinio parašo schemos, kurių privačiųjų raktų generavimo centre naudojamas vienas privatusis raktas visiems žiedinio parašo narių privatiesiems raktams generuoti, nėra visiškai saugios, tačiau tai nebūtinai reiškia, kad parašo autorius gali būti išaiškintas. Nepaisant to, tokių schemų forumuose naudoti nepatartina.

REKOMENDACIJOS

Atlikus žiedinių parašų saugumo analizę, galima teikti tokias rekomendacijas:

1. Visos žiedinių parašų schemos, kuriose privačiųjų raktų generavimo centras visiems žiedo nariams generuoja privačiuosius raktus naudojantis vienu privačiuoju raktu, nėra saugios žiedo narių bendradarbiavimo atakoms, todėl rekomenduojama nenaudoti vieno privačiojo rakto visų žiedo narių privatiesiems raktams generuoti.

2. Kaip parodyta 2.3 ir 2.4 lentelėse, nerekomenduojama didinti skaičiavimo greičio mažinant maišos funkcijų kiekį skaičiavimuose. Nors skaičiavimo greitis [5] yra gerokai didesnis lyginant su [6], tačiau [6] schemos atveju parašo autorius surastas nebuvo.

LITERATŪRA

1. R. L. Rivest, A. Shamir, Y. Tauman / How to leak a secret, 2001.
2. E. Bresson, J. Stern, M. Szydlo / Threshold ring signatures and applications to ad-hoc groups, 2002.
3. K. C. Lee, H. A. Wen, T Hwang / Convertible ring signature, 2005.
4. E. Fujisaki, K. Suzuki / Traceable ring signature, 2006.
5. Y. Yong, L. Fagen, X. Chunxiang, S. Ying / An efficient identity-based anonymous signcryption scheme, 2008.
6. X. Huang, W. Susilo, Y. Mu, F. Zhang / Identity-based ring signcryption schemes: cryptographic primitives for preserving privacy and authenticity in ubiquitous world, 2005.
7. C. Popescu / An efficient id-based group signature scheme, 2002.
8. N. Chandran, J. Groth, A. Sahai / Ring signatures of sub-linear size without random oracles.
9. S. M. E. Y. Alaoui, O. Dagdelen, P. Veron, D. Galindo, P. L. Cayrel / Extended security arguments for ring signature schemes.
10. S. Meiklejohn / An exploration of group and ring signatures, 2011.
11. W. Gao, G. Wang, X. Wang, D. Xie / Controllable ring signatures.
12. J. K. Liu, T. H. Yuen, J. Zhou / Forward secure ring signature without random oracles.
13. C. Y. Lin, T. C. Wu / An identity-based ring signature scheme from bilinear pairings.
14. Y. F. Chung, Z. Y. Wu, F. Lai, T. S. Chen / Anonymous signcryption in ring signature scheme over elliptic curve cryptosystem.
15. J. Xu, Z. Zhang, D. Feng / A ring signature scheme using bilinear pairings, 2004.
16. H. Shacham, B. Water / Efficient ring signature without random oracles.
17. A. Bender, J. Katz, R. Morselli / Ring signatures: Stronger definitions, and constructions without random oracles, 2006.
18. S. S. M. Chow, L. C. K. Hui, S. M. Yiu / Identity based threshold ring signature, 2005.
19. X. Boyen / Mesh signatures: how to leak a secret with unwitting and unwilling participants, 2007.
20. J. Groth / Non-interactive zero-knowledge arguments for voting, 2005.

21. J. Herranz, G. Saez / Forking lemmas for ring signature schemes, 2003.
22. R. Merkle / Secure communications over insecure channels, 1978.
23. A. Shamir / Identity-based cryptosystems and signature schemes, 1984.
24. A. Sahai / Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security, 1999.
25. R. Rivest, A. Shamir, L. Adleman / A method for obtaining sigital signatures and public key cryptosystems, 1978.
26. R. Merkle / A digital signature based on a conventional encryption function.
27. T. Nakanishi, N. Funabiki / Verifier-local revocation group signature schemes with backward unlinkability from bilinear pairings, 2005.
28. M. Bellare, P. Rogaway / The exact security of Digital signatures – how to sign with RSA and Rabin, 1996.
29. D. Naccache, D. M'Raihi, S. Vaudenay, D. Raphaeli / Can D.S.A. be improved, 1995.
30. http://en.wikipedia.org/wiki/Pairing_function.