ŠIAULIŲ UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

Aurelijus Kirna
Informatikos specialybės magistrantūros II kurso neakivaizdinio skyriaus studentas

KRIPTOSCHEMOS IDENTIFIKACINĖS INFORMACIJOS PAGRINDU

MAGISTRO DARBAS

Darbo vadovė:
Doc. R. Steuding

Recenzentas:
Lekt. V. Giedrimas

Šiauliai, 2005/2006 m.m.

Turinys

Įvadas

Sparčiai besivystant informacinėms technologijoms ir vis labiau besiskverbiant į įvairias gyvenimo sferas, kompiuterių panaudojimas tapo neatsiejama mūsų gyvenimo dalimi. Todėl iškilo natūralus poreikis apsaugoti saugomą ir perduodamą informaciją. Tai paskatino kriptografijos mokslo vystymąsi ir tuo pačiu naujų kriptografinių schemų poreikį ir jų atsiradimą. Sąlyginai neseniai atsiradusi kriptografija identifikacinės informacijos (toliau ID) pagrindu suteikia galimybę naudoti raktus pasinaudojant vartotoją identifikuojančia informacija ir taip išvengiant vartotojų autentifikavimo naudojant sertifikatus.

Šio darbo mokslinis naujumas pasireiškia ne fundamentaliais atradimais, bet esamų schemų gerųjų savybių suliejimu į efektyvią schemą, analizuojant jos dabartines savybes ir numatant tobulinimo bei pritaikymo galimybes.

Darbo tikslas – susipažinti ir aprašyti kriptografijos ID pagrindu koncepciją bei panagrinėti jos taikymo galimybes.

Šiam tikslui pasiekti buvo suformuluoti uždaviniai:

- Svarbiausių ID kriptografinių schemų aprašymas ir analizė.
- Efektyvios schemos kūrimas.
- Atlikti programinio schemos modelio realizaciją.

Atliktas darbas gali būti naudingas pagrindinių ID schemos veikimo principų suvokimui bei konstravimui naujų, pagal konkretų poreikį pritaikomų ir naudojamų schemų.

1. Apie kriptografiją ID pagrindu

1.1. Atsiradimas

Kriptografijos ID pagrindu pradininku laikomas Adis Šamiras (Adi Shamir) [2] vienas iš RSA [18] kūrėjų, kuris 1984 m. pateikė koncepciją. Pasak jo, bet kokia vartotojus identifikuojanti informacija (pvz.: elektroninio pašto ar IP adresas, vardas, pavardė, telefonas arba jų kombinacija) vietoj skaitmeninių sertifikatų gali būti panaudojama kaip viešas raktas užšifravimui ar parašų verifikavimui. Todėl kriptografija ID pagrindu palengvina sistemos sudėtingumą bei raktų autentifikavimo struktūros Public Key Infrastructure (PKI) kūrimą bei palaikymą.

Šamirui, sukūrusiam parašo schemą identifikacinės informacijos pagrindu (*identy based signature* IBS) pasinaudojusiam jau sukurta RSA kriptosistema, nepavyko sukurti užšifravimo identifikacinės informacijos pagrindu (*identy based encryption* IBE) schemos, kurios kūrimo paieškos tapo ilgai besitęsiančia atvira problema. Tik 2001 m. ši Šamiro atviroji problema buvo nepriklausomai išspręsta kartu dirbusių Bonė ir Franklino (Boneh and Franklin) [6] bei atskirai dirbusio Kokso (Cocks) [4]. Kadangi jiems pavyko sėkmingai realizuoti užšifravimą, šia kriptografija pradėjo domėtis įvairios tyrinėtojų grupės.

1.2. Originali Šamiro įdėja

Pasak jo – „tai naujo tipo kriptografinė schema, kuri leidžia bet kokiai vartotojų porai bendrauti saugiai ir patikrinti vienas kito parašus nesikeičiant privačiais ir viešais raktais, nelaikant raktų žinynų ir nesinaudojant trečiųjų asmenų paslaugomis. Pagal schemą, reikalingi patikimi raktų generavimo centrai, kurių viena iš paskirčių duoti kiekvienam vartotojui asmeninę kortelę (*smart card*) kai jis pirmą kartą prisijungia prie tinklo. Kortelėje esanti informacija leidžia vartotojui pasirašyti ir užšifruoti žinutes, kurias jis siunčia, ir iššifruoti bei patikrinti žinutes kurios gaunamos visiškai nepriklausomu keliu, nepaisant kitų asmenų tapatybės. Kortelių nereikia atnaujinti atsiradus naujam vartotojui, o įvairiems centrams nereikia koordinuoti savo veiklos ar net turėti vartotojų sąrašo. Centrai gali būti uždaromi po kortelių išdavimo, o tinklai gali funkcionuoti visiškai decentralizuotu būdu neribotą laiką.

Schemos pagrindas – viešo rakto kriptosistema su viena detale: vietoj to, kad būtų generuojama viešo ir slapto raktų pora, ir publikuojamas vienas jų, vartotojas pasirenka savo vardą ir tinklo adresą kaip savo viešąjį raktą. Bet kokia kombinacija iš vardo, socialinės apsaugos numerio, gatvės adreso, ofiso numerio ar telefono numerio gali būti naudojama

(priklausomai nuo konteksto) išskirti unikalią vartotojo tapatybę, kurios vėliau jis negalėtų paneigti ir kuri lengvai prieinama kitiems. Atitinkamas slaptas raktas sugeneruojamas centre ir kortelės forma išduodamas vartotojui, kai jis pirmą kartą prisijungia prie tinklo. Kortelė turi mikroprocesorių, įėjimo/išėjimo sąsają (I/O port), RAM, ROM su slaptu raktu, programinę įrangą žinučių užšifravimui bei dešifravimui, ir parašo generavimui bei patikrinimui.

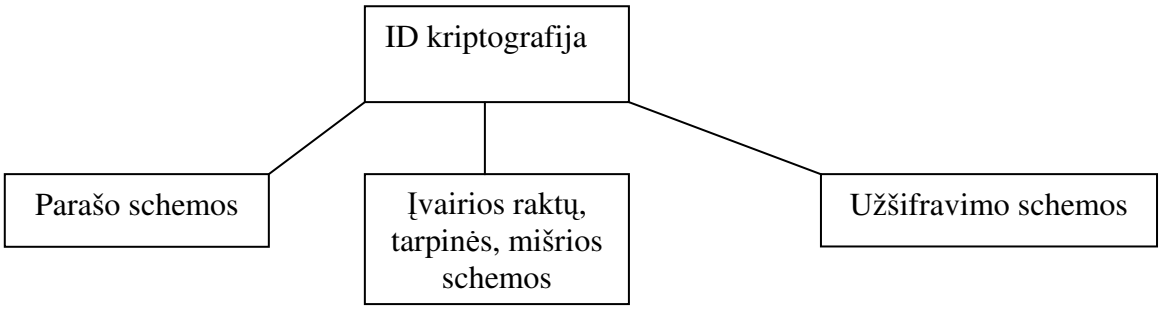
Kai vartotojas A nori pasiųsti žinutę vartotojui B, jis pasirašo slaptuoju raktu esančiu jo kortelėje, užšifruoja rezultatus naudodamas B vardą ir tinklo adresą, prideda savo vardą ir tinklo adresą prie žinutės ir siunčia vartotojui B. Kai B gauna žinutę, iššifruoja ją naudodamasis slaptu raktu savo kortelėje, patikrina parašą naudodamas siuntėjo vardą ir tinklo adresą kaip verifikavimo raktą.

Slapti raktai apskaičiuojami generavimo centrų, o ne vartotojų, nes pačios vartotojų tapatybės nėra kuo nors išsiskiriančios: jei vartotojas A galėtų apskaičiuoti slaptą raktą, kuris atitinka A viešą raktą, jis galėtų apskaičiuoti slaptus raktus, kurie priklauso B, C ir t.t. - schema nebūtų saugi. Raktų generavimo centrai turėtų privilegiją žinoti šiek tiek slaptos informacijos (tokios kaip didelių skaičių faktorizacija), kuri leistų apskaičiuoti visų tinklo vartotojų slaptus raktus.“[…] [2]

1.3. Klasifikacija

ID kriptografiją galima pavaizduoti atvaizduoti schema (1 pav.), kurioje matomos dvi esminės tyrimo kryptys: 1) Parašo schemos, tokios kaip Šamiro IBS [2], 2) Užšifravimo schemos, tokios kaip Bonė ir Franklino IBE schema [6].

Be šių pagrindinių krypčių, įvairios tyrinėtojų grupės yra sukūrusios ir daugybę teorinių mišrių parašo ir užšifravimo arba kitų, itin didelį dėmesį teikiančių raktų suteikimui ir platinimui. Vienas iš aktyvių dabartinių „Pravimo pagrindu“ kriptografijos tyrėjų Paulo Barreto savo tinklalapyje [17] pamėgino sudaryti sąrašą kriptosistemų ID pagrindu. Tačiau jas visas surašyti ir suskirstyti nėra lengva, nes tai sparčiai besivystanti kriptografijos dalis ir pasak jo, „eksponentiškai“ sukuriama naujų schemų.



1 pav. *Abstraktus ID kriptografijos skirstymas.*

1.4. Pagrindinės IBS ir IBE koncepcijos [9]

Kaip minėta, siuntėjas Aldona siunčiamajai žinutei užšifruoti gali naudoti bet kokią gavėjo identifikacinę informaciją, netgi jo skaitmeninę nuotrauką [1]. Tada gavėjas Benas, turi turėti privatų raktą, susietą su jo identifikacine informacija, kuris gaunamas iš patikimos trečios šalies (būtina sąlyga) - "Privačių raktų generatoriaus" centro (Private Key Generator - toliau PKG), ir tik tada gali iššifruoti gautą užšifruotą tekstą.

IBE schema:

- Pradžia: PKG sukuria savo valdančiojo (privataus) ir viešo raktų porą atitinkamai sk_{PKG} ir pk_{PKG}. pk_{PKG} išduodamas visiems vartotojams ir yra nekintantis ilgą laiką.
- Privataus rakto gavimas: Gavėjas Benas praneša apie save ir įrodo savo tapatybę PKG ir gauna privatų raktą $sk_{IDBenas}$ susietą su identifikacine informacija ID_{Benas}.
- Užšifravimas: naudodamas Beno identifikacine informacija ID_{Benas} bei PKG pk_{PKG}, siuntėjas Aldona užšifruoja atviro teksto žinutę M ir gauna šifrą C.
- Iššifravimas: Gautą iš Aldonos šifrą C, Benas iššifruoja naudodamas savo privatų raktą $sk_{IDBenas}$ ir vėl gauna atvirą tekstą M.

IBS schemoje, pasirašančioji Aldona pirmiausia gauna privatų raktą, kuris susiejamas su jos identifikacine informacija iš PKG. Tada ji pasirašo žinutę naudodama privačiu raktu. Tikrintojas Benas dabar naudoja Aldonos identifikacinę informaciją verifikuoti Beno parašą. Taigi Benui gauti Aldonos sertifikato nereikia.

IBS schema:

- Pradžia: PKG – patikima trečioji šalis, sukuria savo valdančiojo ir viešo raktų porą atitinkamai sk_{PKG} ir pk_{PKG}.
- Privataus rakto gavimas: Pasirašančioji Aldona autentifikuoja save PKG ir gauna privatų raktą $sk_{IDAldona}$, susietą su jos identifikacine informacija ID_{Aldona}.

- Parašo generavimas: Naudodama savo privatų raktą $sk_{IDAldona}$, Aldona sukuria parašą σ savai žinutei M.

Parašo verifikavimas: Turėdamas gautą parašą σ ir žinutę M iš Aldonos, tikrinantis Benas tikrina ar σ tikras M parašas naudodamas Aldonos identifikacinę informaciją ID_{Aldona} ir PKG viešą raktą pk_{PKG}. Jei taip, gražinama "*Priimti*", jei ne gražinama "*Atmesti*".

1.5. Bitiesinio poravimo kriptografinės schemos identifikacinės informacijos pagrindu

Bitiesinis poravimas yra matematinė struktūra, be kurios neįsivaizduojama dabartinė ID kriptografija nuo pat Bonė ir Franklino panaudojimo savo IBE schemoje [6].

Bitiesinio poravimo apibrėžimas. Bitiesinis poravimas P apibrėžiamas virš dviejų grupių G ir F su ta pačia pirmine eile q. (G^* pažymime $G \setminus \{O\}$, kur O yra grupės G neutralusis elementas, o $\mathbb{Z}_q^* = \{1,2,3,\ldots,q-1\}$ atitinkamai $\mathbb{Z}_q \setminus \{0\}$.)

Praktikoje grupė G realizuojama naudojant tam tikrų elipsinių kreivių taškų grupę, o grupė F bus realizuota naudojant baigtinio kūno multiplikatyviosios grupės pogrupį. Leistinas bitiesinis atvaizdis išreikštas $P : G \times G \rightarrow F$ turi savybes:

- Bitiesinė: $P(aR_1,bR_2) = P(R_1,R_2)^{ab}$, kur $R_1,R_2 \in \mathbb{Z}_q^*$.
- Neišsigimusi: P neatvaizduoja visų taškų porų $G \times G$ į grupės F neutralųjį elementą. (Vadinasi, jei R yra G generatorius, tai $P(R,R)$ yra F generatorius.)
- Apskaičiuojama: visiems $R_1,R_2 \in G$ atvaizdis $P(R_1,R_2)$ yra efektyviai apskaičiuojamas.

Bitiesinė Diffie-Hellman prielaida. Dėl bitiesinio poravimo atsirado problema, vadinamoji "Bitiesinė Diffie-Hellman (BDH)" problema:

- Duota (G, q, P, R, aR, bR, cR), kur a, b, c yra atsitiktinai parinkti iš $\mathbb{Z}_q^*$, reikia apskaičiuoti $P(R, R)^{abc}$.

BHD prielaida reiškia, kad aukščiau pateikta problema sudėtinga skaičiuojamuoju pažiūriu. Yra pastebėta, kad daugelio kriptosistemų saugumas priklauso nuo BDH apskaičiavimo (arba jos variantų) išsprendžiamumo sudėtingumo.

Bitiesinis poravimas naudojamas konstruojant ir kitas įdomias ne ID kriptografines schemas.

1.6. Šamiro IBS schema [2]

Parašo schema pagrįsta verifikavimo sąlyga

$$s^{e} = i \cdot t^{f(t, m)} \qquad (\text{mod } n)$$

Kur

- m žinutė
- s, t parašai
- i vartotojo identifikacinė informacija
- n dviejų didelių pirminių skaičių sandauga
- e didelis pirminis skaičius, kuris tarpusavyje pirminis $\varphi(n)$
- f vienakryptė funkcija

Parametrai n, e ir funkcija f parenkami PKG, o visi vartotojai turi tuos pačius n, e ir tą pačią f savo kortelėse. Dydžiai gali būti vieši, bet n faktorizacija turi būti žinoma tik PKG. Vienintelis skirtumas tarp vartotojų yra dydis i, o slaptas raktas yra unikalus skaičius

$$g^{e} = i \qquad (\text{mod } n)$$

Šis g gali būti apskaičiuojamas PKG, tačiau niekas negali ištraukti e-osios šaknies mod n.

Kiekviena žinutė m turi didžiulį skaičių galimų (s, t) parašų, bet jų tankis toks mažas, kad atsitiktinai ieškant yra beveik neįtikėtina surasti vieną jų. Bet koks mėginimas vieną iš (s, t) susieti su atsitiktiniu dydžiu ir surasti kitą kintamąjį reikalauja šaknų ištraukimo mod n, kas būtų be galo sudėtinga skaičiavimo užduotis. Tačiau, kai g žinomas, yra labai paprasta generuoti bet kokį skaičių žinučių parašų, net kai n faktorizacija nėra žinoma.

Kad pasirašytų žinutę m, vartotojas pasirenka atsitiktinį skaičių r ir suskaičiuoja

$$t = r^{e} \qquad (\text{mod } n)$$

Verifikavimo sąlyga gali būti užrašyta taip:

$$s^{e} = g^{e} \cdot r^{e f(t, m)} \qquad (\text{mod } n)$$

Kai e tarpusavyje pirminis skaičius su $\varphi(n)$, galima eliminuoti bendrą daugiklį e iš eksponenčių

$$s = g \cdot r^{f(t, m)} \qquad (\text{mod } n)$$

Ir taip s gali būti apskaičiuotas be šaknų traukimo.

Kad apsisaugoti nuo atakų, besiremiančių sąryšių dauginimų tarp vartotojų identifikacinių informacijų, patartina išplėsti eilutę, kuri apibūdina vartotojo identifikacinę informaciją, į ilgą pseudoatsitiktinę eilutę universalios, vienos krypties funkcijos pagalba, ir naudoti šią išplėstą formą verifikavimo sąlygoje vietoje *i*. Kai visi tinkle žino kaip pasinaudoti šia funkcija, schema išlaiko savo *ID* pėdsaką net kai parašo verifikavimo raktas nėra tiksliai lygus vartotojo identifikacinei informacijai.

Schemos saugumas priklauso nuo kriptoanalitiko negalėjimo išskirti *g* analizuojant didelį kiekį turimų žinučių parašų. Jei didžiausias bendras daliklis $DBD(f, e)$ yra $c \neq 1$, įmanoma ištraukti c – ąją šaknį iš i manipuliuojant verifikavimo sąlyga, ir taip svarbu padaryti e pakankamai dideliu pirminiu skaičiumi, o f pakankamai stipria vienos krypties funkcija, kad tokiu būdu šis atvejis praktiškai niekada neįvyktų. Dydis r niekada neturėtų būti pakartotinai naudojamas ar atskleistas, kai g yra bet kokiame konkrečiame paraše apsaugotas savo atsitiktinumu bei slaptumu.

Verifikavimo sąlygų variantai, kuriame vienas iš dviejų t eliminuojamas (pvz.: pakeičiamas konstanta) yra nesaugūs. Todėl svarbu naudoti vienakryptę funkciją, kuri visiškai sumaišytų t ir m (pageidautina ne aritmetinėmis ir ne neigimo operacijomis) ir kuri turėtų didžiulį galimų reikšmių intervalą.

1.7. Bonė ir Franklino IBE schema [6]

Bonė ir Franklinas sukūrė schemą pavadinimu *BasicIdent*, bei šios schemos modifikuotą versiją *FullIdent*, kuri šiame darbe nebus nagrinėjama.

BasicIdent schema susideda iš keturių dalių – Setup, Extract, Encrypt, Decrypt. Tarkime, kad k yra saugumo parametras duotas Setup dalyje. *IG* yra BDH parametro generatorius.

SETUP: Pirmiausia sugeneruojamos dvi grupės G_1 ir G_2. Bitiesinis atvaizdis $e : G_1 \times G_1 \rightarrow G_2$. Tarkime, kad q yra grupių G_1 ir G_2 eilė. Pasirenkamas generatorius $R \in G_1$.

Pasirenkamas bet koks $s \in \mathbb{Z}_q^*$ ir nustatomas $R_{pub} = sP$.

Pasirenkama kriptografinė maišos funkcija $H_1 : \{0,1\}^* \rightarrow G_1^*$. Pasirenkama kita kriptografinė maišos funkcija H_2: $G_2 \rightarrow \{0,1\}^n$ su tam tikru n. Žinutė yra $M \in \{0,1\}^n$. Sistemos parametrai yra $\{ G_1 , G_2 , e, n , R, R_{pub}, H_1, H_2\}$. Valdantysis raktas yra $s \in \mathbb{Z}_q^*$.

EXTRACT: Su turima teksto eilute ID $\in \{0,1\}^*$ suskaičiuojamas $Q_{ID} = H_1(ID) \in G_1^*$ ir surandamas privatus raktas $d_{ID} = s\, Q_{ID}$, kur s yra valdantysis raktas.

ENCRYPT: Kad užšifruoti žinutę M su viešu raktu ID reikia suskaičiuoti $Q_{ID} = H_1(ID) \in G_1^*$ ir pasirinkti atsitiktinį $r \in \mathbb{Z}_q^*$, tada šifras bus $C = \{rR,\ M \oplus H_2(g_{ID}^r)\}$, kur $g_{ID} = e(Q_{ID}, R_{pub},)$.

DECRYPT: Tarkime, kad $C = \{U,V\}$ yra šifras užšifruotas naudojant viešą raktą ID. Norint iššifruoti C naudojamas privatus raktas $d_{ID} \in G_1^*$ ir skaičiuojama $M = V \oplus H_2(e(d_{ID}, U))$.

Bonė ir Franklinas realizuodami šią schemą skaičiavimams naudojo Weilo poravimą.

Yra pastebėta, kad ši schema yra saugi prieš pasirinkto atviro teksto ataką. Tai reiškia, kad BHD problema yra sunkiai išsprendžiama.

1.7.1. IBE įgyvendinimas

Iki šiol yra įgyvendinta keletas projektų, naudojančių IBE.

Adresu http://crypt.stanford.edu/ibe/download.html galima rasti realizaciją Bonė ir Franklino schemos pavadinimu "Stanfordo IBE sistema" Debian GNU/Linux.

Yra sukurta Shamus Software kriptografinė biblioteka pavadinimu "MIRACL", kurioje tai pat yra Bonė ir Franklino IBE schema.

Tiek Stanfordo, tiek Shamus bibliotekos sukurtos naudojant C/C++ kalbas.

Nėra dėmesio vertos Java realizacijos.

Verti pasidomėti realaus pasaulio IBE pritaikymai: Voltage Security sukurti elektroninio pašto sistemos priedai skirti Outlook, pine. Didžiosios Britanijos Bristolyje tyrinėtojai iš Hewlett Packard Lab sukūrė sveikatos priežiūros informacinę sistemą

1.8. Kokso IBE schema [4]

SETUP: PKG naudodama saugumo parametrą k sugeneruoja du pirminius p ir q, tokius, kad $p,q \equiv 3 \bmod 4$. p ir q laikomi paslaptyje, yra žinomas tik $M = pq$.

EXTRACT: Benas siunčia savo maišytą identifikacinę informaciją $a = H(ID)$ į PKG, kur JacobiSymbol(a/M) = 1. PKG grąžina dydį

$$r = a^{\frac{M+5-(P+Q)}{8}} \bmod M$$

Tokiu būdu r tenkina $r^2 = a \bmod M$ arba $r^2 = -a \bmod M$.

ENCRYPT: Aldona sugeneruoja transportavimo raktą ir naudoja duomenims užšifruoti simetriniu šifravimu. Tegu x yra atskiras transportavimo rakto bitas 1 arba -1. Aldona atsitiktinai pasirenka dydį k, tokį, kad JacobiSymbol(k/M) = x. Tada ji Benui siunčia

$$s = (k + \frac{a}{k}) \mod M$$

DECRYPT: Benas, gavęs užšifruotą bitą *s*, turi suskaičiuoti $(\frac{s+2r}{M})$, kad gauti pradinį *x*.

Kadangi

$$s + 2r = k + \frac{a}{k} + 2r = k + \frac{r^2}{k} + 2r = k(1 + \frac{r}{k})^2 \mod M$$

Iš to seka, kad $(\frac{s+2r}{M}) = (\frac{k}{M}) = x$

Schemos saugumas remiasi prielaida, kad *M* faktorizaciją suskaičiuoti yra sudėtinga. Tačiau dėl tokio po bitą skaičiavimo, ši schema nėra efektyvi.

2. Projektinė dalis

2.1. Įrankių ir priemonių pasirinkimo analizė

Šiame darbe bus atliekami matematiniai skaičiavimai. Kartais skaičiavimuose pasitaikius nemažiems skaičiams arba atliekant skaičiavimų rutiną yra nepatogu atlikti be pagalbinių priemonių. Matematiniams skaičiavimams reikalingas koks nors matematinis paketas. Žinomiausi 4 variantai: Mathematica, Maple, Matlab ir MathCAD. Kadangi itin ypatingų reikalavimų skaičiavimams ir paketui nėra, tai ne taip svarbu, kurį iš šių paketų pasirinkti. Martijn Maas [12] realizavo Weilo ir Tate poravimų skaičiavimus Mathematicos pagalba, todėl ji buvo pasirinkta ir šiame darbe. Maas skaičiavimus naudosime šio darbo tarpiniams skaičiavimams tikrinti.

Kadangi vienas iš uždavinių yra programinis schemos modelis, todėl tikslinga pasirinkti programavimo kalbą šiam modeliui realizuoti. Ne paskutinėje vietoje yra programavimo kalbos savybė – naudojimo patogumas t.y. turi būti aukšto lygio, objektinė, turėti gerą palaikymą (*support*), galimybė programos kodą rašyti ne tik paprastame tekstiniame redaktoriuje, būtų galimybė greitai ištaisyti klaidas. Labai svarbu, kad šioje kalboje būtų palaikomi labai ilgi skaičiai. Kriptografija gali būti naudojama ir yra naudojama įvairios architektūros kompiuteriuose. Šiame darbe numatomas modelis bandyti paprastiems, paplitusiems PC architektūros kompiuteriams. Dabar bandymai bus skirti PC kompiuteriams, bet vėliau gali prireikti ir naudoti terminaluose su smartcard tipo kortelėmis. Iš to kyla beveik reikalavimas programavimo kalbai būti kuo mažiau priklausomai nuo kompiuterio architektūros. Šiuos reikalavimus praktiškai atitinka Java kalba. Dėka savo neegoistinių paskatų Sun kompanija šią kalbą išpopuliarino ir dabar Java turi ne tik daugybę naudotojų, bet ir tobulintojų. Šiuo atveju tinkama versija Standart Edition Development Kit 5.0. Taigi Java turi ir reikiamą BigInteger klasę neriboto ilgio sveikiesiems skaičiams, ji yra dėl Java Virtualios Mašinos (JVM *Java Virtual Machine*) praktiškai nepriklausoma nuo kompiuterio architektūros. Žinoma, greitis yra svarbus dalykas ir Java kalba parašyta programa daugeliu atveju veiks lėčiau nei analogiška C++ programa, bet Java patogesnė. Numatoma galimybė ID kriptoschemą realizuoti ir apletų (*applets*) pagalba. Alternatyva - nelabai populiarūs C# apletai.

Keletas Java integruotų kūrimo aplinkų, išbandytų kuriant projektą:

Įrankis	Privalumai	Trūkumai
NetBeans IDE 5.5	Leidžia kurti grafiškai formas, įskiepius. Nemokama.	Reikalauja daug kompiuterio resursų. Neleidžia taisyti sugeneruoto programinio kodo.
JBuilder X	Galima kurti grafiškai formas, įskiepius. Generuoja ir UML vaizdą.	Reikia daug kompiuterio resursų. Mokama.
Eclipse 3.1	Galima naudoti papildomus modulius ir kurti grafiškai. Nemokama.	Reikia daug kompiuterio resursų. Įrankio suderinimui reikia nemažai laiko.
Java GUI Builder 1.3	Galima kurti grafiškai formas, įskiepius. Labai paprasta naudoti. Naudoja mažai kompiuterio resursų. Nemokama.	Skirta tik GUI kūrimui. Negalima pakeisti komponento savybių, išskyrus dydį ir tekstą. Nedidelis komponentų pasirinkimas.

1 lentelė. *Java integruotos kūrimo aplinkos.*

Viena geriausių Java kalbos integruotų kūrimo aplinkų (*IDE Integrated Development Environment*) yra Eclipse. Patogi grafinė vartotojo sąsaja, automatinė klaidų paieška, integruotas kompiliatoriaus valdymas, galimybė ne iš komandinės eilutės valdyti java programėles t.y. generuoti javadoc, jar ir kt. Vertas naudoti nemokamas įrankis.

2.2. Matematinės priemonės

2.2.1 Elipsinės kreivės

Kad schema būtų kuo saugesnė panaudosime elipsines kreives (2 pav.). Yra paskaičiuota [19], kad kriptoschemos, kuriose naudojamos elipsinės kreivės yra saugesnės už analogiškas RSA. Apytikriai 160 bitų elipsinių kreivių viešasis raktas atitinka saugumą, kurį duoda RSA 1024 bitų raktas.

Algebroje ir geometrijoje elipsinės kreivės [5] intensyviai tiriamos jau apie 150 metų, todėl jų teorija tapo gana turtinga. Elipsinės kreivės kriptografijoje naudoti 1985m. pirmiausia pasiūlė nepriklausomai vienas nuo kito, Neal Koblitz iš Vašingtono Universiteto ir Victor Miller iš IBM. Nemažai šiuolaikinių kriptografinių sistemų, tame tarpe ir Bonė ir Franklinas IBE schemoje, naudojamos elipsinės kreivės. Tokį susidomėjimą elipsinėmis kreivėmis nulėmė jų

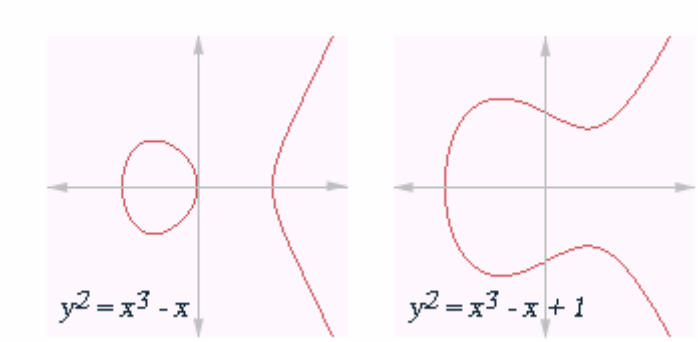
taškų aibės struktūra. Kriptografijoje naudojamos elipsinės kreivės apibrėžtos virš baigtinių kūnų K. Paprastai tai būna $K = F_p$, kur p yra pirminis skaičius, arba $K = F_p^m$.

Elipsinė kreivė virš kūno K, kurio charakteristika nėra lygi 2 arba 3, yra taškų (x,y) aibė, kurie tenkina Vejerštraso (*Weierstrass*) lygybę:

$$y^2 = x^3 + Ax + B,$$

kur taškų koordinatės bei koeficientai A, B priklauso kūnui K bei $4A^3 + 27B^2 \neq 0$.

Algebroje pateikiamame grupės apibrėžime turi būti neutralusis elementas. Šiuo atveju neutralusis elementas yra vadinamasis taškas begalybėje O. Pati grupė yra adityvi, t.y. operacija grupėje yra sudėtis. Kiekvienas taškas P iš elipsinės kreivės turi sau priešingą tašką $-P$ x ašies atžvilgiu, taigi patenkinamas ir kitas reikalavimas grupei – turėti atvirkštinius elementus.



2 pav. *Elipsinių kreivių virš realiųjų skaičių kūno pavyzdžiai.* (Wikipedia)

Tarkime, kad turime du taškus $P = (x_1, y_1)$, $Q = (x_2, y_2)$ ir $P, Q \neq O$.

Kai $P \neq \pm Q$,

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

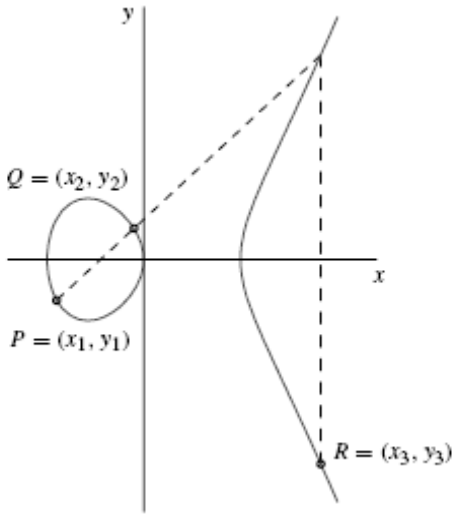
Kai $P = Q$

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

Tada taškas $R = (x_3, y_3) = P + Q$, kurio koordinatės

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda (x_1 - x_3) - y_1$$



3 pav. *Geometrinė taškų sudėtis.*

2.2.2. Supersinguliarios ir ne-supersinguliarios elipsinės kreivės

Pagal Hasse teoremą žinoma, kad grupės eilė t.y. grupės elementų skaičius $\#E(F_q) = q + 1 - t$, kur t yra „*Trace of Frobenius*“ $| t | \leq 2\sqrt{q}$. Kreivė E yra supersinguliari, jei $t^2 = 0; q; 2q; 3q$, ar $4q$; kitu atveju kreivė yra ne-supersinguliari. Plačiau apie šias kreives galima rasti [8].

Įdomiausios kriptografiniu požiūriu supersinguliarios kreivės:

$$y^2 = x^3 + Ax, \text{ kur } p \equiv 3 \bmod 4 \quad (1)$$

$$y^2 = x^3 + B, \text{ kur } p \equiv 2 \bmod 3 \quad (2)$$

Ir ne-supersinguliarios:

$$y^2 = x^3 + Ax, \text{ kur } p \equiv 1 \bmod 4 \quad (3)$$

$$y^2 = x^3 + B, \text{ kur } p \equiv 1 \bmod 3 \quad (4)$$

2.2.3. Bitiesiniai poravimai

Esminė priežastis (neskaitant grupės suformavimo), kodėl kriptografijoje naudojamos elipsinės kreivės yra ta, kad galima atlikti bitiesinius poravimus su taškais, esančiais elipsinėse kreivėse.

Apibrėžimas. Grupių G_1 ir G_2 bitiesinis poravimas yra atvaizdis $e : G_1 \times G_1 \rightarrow G_2$ su savybėmis:

- Bitiesinė: $e(aR_1,bR_2) = e(R_1,R_2)^{ab}$, kur $R_1,R_2 \in \mathbb{Z}_q^*$.

- Neišsigimusi: e neatvaizduoja visų taškų porų $G \times G$ į grupės F neutralųjį elementą. $e(R,R) \neq 1$ (Vadinasi, jei R yra G generatorius, tai $P(R,R)$ yra F generatorius.)
- Apskaičiuojama: atvaizdis $P(R_1,R_2)$ yra efektyviai apskaičiuojamas.

Būtent pirmoji savybė ir yra pagrindinė priežastis, kodėl pradėti naudoti bitiesiniai poravimai kriptografijoje.

Žinomiausi ir dažniausiai sutinkami poravimai yra Weil poravimas ir Tate poravimas.

2.2.4. Tate poravimas

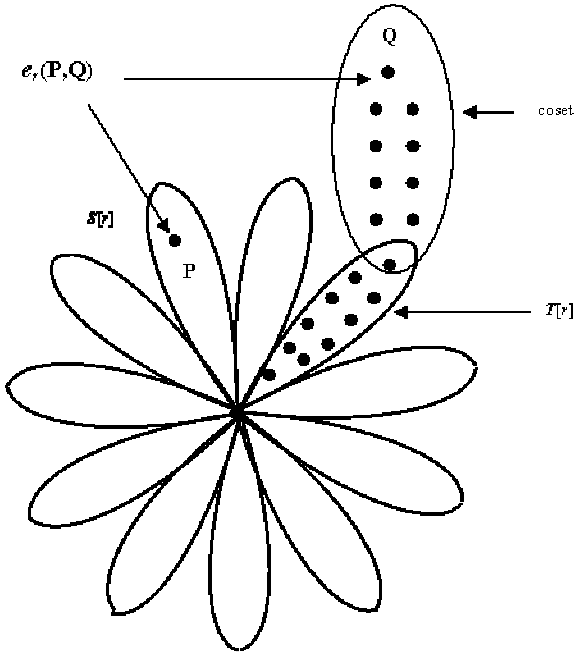
Praktikoje yra pastebėta, kad Tate poravimas [10], [11] yra šiek tiek greičiau apskaičiuojamas ir todėl vertesnis naudoti.

Tarkime elipsinė kreivė E apibrėžta virš baigtinio kūno K. Tarkime, kad egzistuoja toks r, kad $E(K)$ turi tašką, kurio eilė yra r. Skirtingų taškų rinkinys, gautas padauginus kiekvieną $E(K)$ tašką iš r ir žymimas $rE(K)$. Šis rinkinys yra $E(K)$ gretutinė klasė (*coset*). Gretutinės klasės taškų skaičius lygus $\#C = \#E(K)/r$. Tokiu būdu $E(K)$ taškų grupė gali būti išskaidyta į kelias gretutines klases. Tarkime, kad r yra priminis skaičius. Jei $E[r] \not\subseteq E(K)$ tai kiekvienoje gretutinėje klasėje bus lygiai vienas r-tosios eilės taškas. Kitu atveju, jei $E[r] \subseteq E(K)$ tai kiekvienoje gretutinėje klasėje bus r r-tosios eilės taškų.

Tarkime, kad elipsinė kreivė E/F_q apibrėžta virš kūno F_q. Tarkime, kad r yra teigiamas sveikas skaičius, tarpusavyje pirminis su q, toks, kad $E(F_q)$ turi tašką, kurio eilė r. Kriptografijoje r dažniausiai imamas didelis pirminis skaičius, toks kad $r \mid \#E(F_q)$. Tegu k yra mažiausias sveikas skaičius, tenkinantis $r \mid q^k - 1$. Šis dydis yra įtvirtinimo laipsnis (*embedding degree*). Naudosime dydį $k = 2$ t.y. tinkantį ir supersinguliarioms ir ne-supersinguliarioms kreivėms.

Panašiai kaip Weilo poravimas, šis taip pat turi naudingas kriptografijoje savybes:

1. $(O,Q) = 1$ su visais $Q \in E(F_q^k)$, o $(P, Q) \in (F_q^{k*})^r$ visiems $P \in E(F_q^k)[r]$ ir visiems $Q \in rE(F_q^k)$.
2. Kiekvienam taškui $P \in E(F_q^k)[r] \setminus \{O\}$ yra kitas taškas $Q \in E(F_q^k)$, toks kad $(P,Q) \notin (F_q^{k*})^r$ (ne išsigimusi).
3. Visiems $P_1,P_2,P \in E(F_q^k)[r]$ ir $Q_1,Q_2,Q \in E(F_q^k)$ galioja $(P_1+P_2,Q) \equiv (P_1,Q)\,(P_2,Q)$ ir $(P,Q_1+Q_2) \equiv (P,Q_1)\,(P,Q_2)$ (bitiesiškumas).

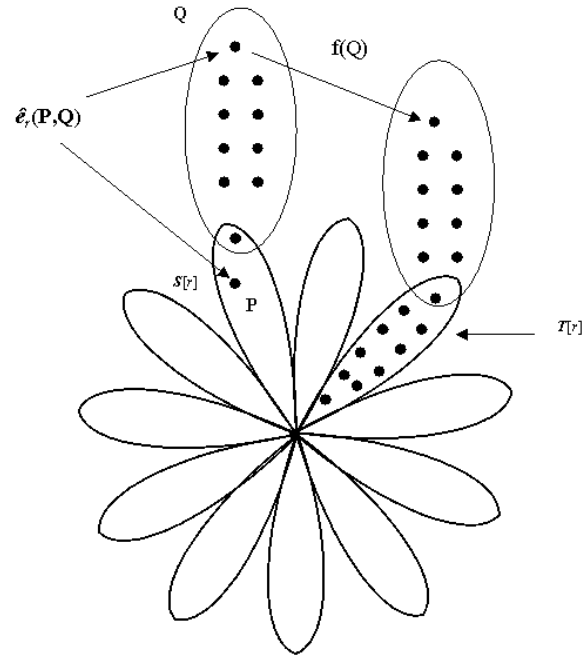


4 pav. *Taškų parinkimas Tate poravimui.*

Taigi Tate poravimas operuoja su taškų pora: *r*-tosios eilės *P* ir *Q*, kuris yra vienas iš gretutinės klasės narių. Tate poravimo reikšmė gali gautis 1, jei *P* yra *Q* kartotinis t.y. jei *Q* parenkamas iš tos pačios gretutinės klasės *S*[*r*]. Kadangi Tate poravimas yra neišsigimęs, tai bet kokiam $P \neq O$ visada galima rasti tokį *Q*, kad $e(P,Q) \neq 1$. Taip pat $e(P,P) = 1$ bet kokiam $P \in S[r]$.

Kitas būdas yra naudoti supersinguliarias kreives, nes tik jos turi vieną itin patogią savybę – egzistuoja vienareikšmė atitiktis (*distortion map*) (apie ją kiek vėliau).

Panaudojus taško *Q* vienareikšmę atitiktį f(*Q*) (5 pav.) galima skaičiuoti Tate poravimą $e(P,Q) = e(P, f(Q))$, kur *P* yra *S*[*r*] narys ir *Q* yra *S* narys.



5 pav. *Taškų parinkimas Tate poravimui supersinguliarių kreivių atveju.*

2.2.5 Algoritmai

Milerio algoritmas. Milerio algoritmas [21] tinka ir Weilo poravimui ir Tate poravimui. Kiekvienai taškų porai U ir $V \in E(F_q^k)$ tegu $g_{U,V} \in F_q^k(E)$ yra racionali funkcija atitinkanti liniją $l_1 y + l_2 x + l_3 = 0$ per taškus U ir V.

$$U = (x_U, y_U),\ V = (x_V, y_V),\ Q = (x, y),$$
$$\lambda_1 = (3x_V^{\,2} + a)/(2y_V),$$
$$\lambda_2 = (y_U - y_V)/(x_U - x_V).$$

Kai

$$g_{U,V}(O) = g_{U,O}(Q) = g_{O,V}(Q) = 1,$$
$$g_{V,V}(Q) = \lambda_1(x - x_V) - y + y_V,\ Q \neq O,$$
$$g_{U,V}(Q) = \lambda_2(x - x_V) - y + y_V,\ Q \neq O,\ U \neq \pm V,$$
$$g_{V,-V}(Q) = x - x_1,\ Q \neq O.$$

Tada Milerio algoritmas:

$f \leftarrow 1$, V $\leftarrow$ P
for i $\leftarrow$ t – 1 **downto** 0 **do**
  $f \leftarrow f^2 \cdot g_{V,V}(Q+R) \cdot g_{2V,-2V}(R) / g_{2V,-2V}(Q+R) \cdot g_{V,V}(R)$, V$\leftarrow$ 2V
  if ri = 1 **then**
    $f \leftarrow f \cdot g_{V,P}(Q+R) \cdot g_{V+P,-V-P}(R) / g_{V+P,-V-P}(Q+R) \cdot g_{V,P}(R)$,
    V $\leftarrow$ V + P
  end if
end for
$z \leftarrow (q^k - 1) / r$
return f^z

BKLS algoritmas. Analogiškas yra BKLS (Barreto, Kim, Lynn, Scott) [16] algoritmas, tik jame yra pašalinti vardikliai. Todėl lyginant su tokiais pat parametrais abu algoritmus, BKLS skaičiuoja greičiau.

BKLS algoritmas:

$f \leftarrow 1$, V $\leftarrow$ P
for i $\leftarrow$ t – 1 **downto** 0 **do**
  $f \leftarrow f^2 \cdot g_{V,V}(Q)$, V $\leftarrow$ 2V
  if ri = 1 **then**
    $f \leftarrow f \cdot g_{V,P}(Q)$, V $\leftarrow$ V + P
  end if
end for
$z \leftarrow (q^k - 1) / r$
return f^z

2.3. Projekto vykdymo planas

Kaip ir kiekvieno darbo pradžioje reikalinga turimų arba prieinamų informacijos šaltinių paieška ir informacijos tyrimas. Visų pirma svarbiausia suprasti pačią ID kriptografijos esmę. Kuo ji skiriasi nuo kitų krypčių? Esminis dalykas būdingas visos ID schemoms yra identifikacinė informacija. Vienokiu pavidalu ar kitokiu, ji išlieka svarbiausiu elementu, kuris atlieka ne tik viešojo rakto vaidmenį, bet ir leidžia nenaudoti skaitmeninių sertifikatų, vartotojui identifikuoti.

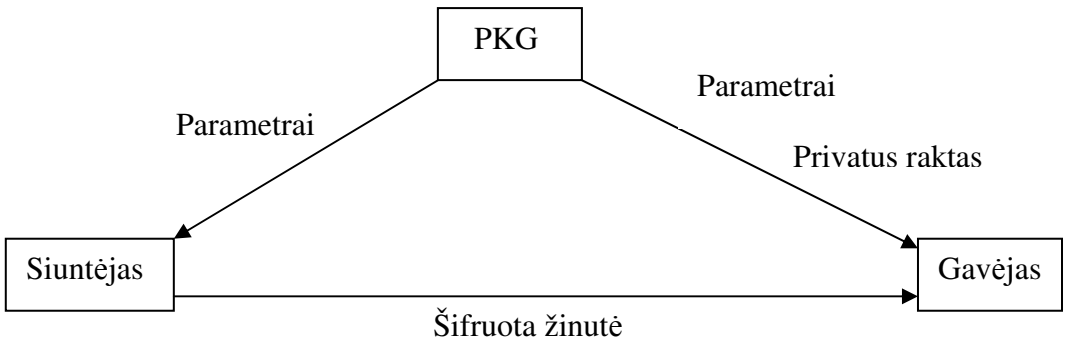
Antra, ko ieškoma šaltiniuose – tai kokios yra dabartinės populiariausios schemos. Kadangi modelis bus užšifravimo schema, tai šiame darbe didesnis dėmesys skiriamas ir literatūroje ieškoma tokių schemų. Pagal rastą informaciją matyti, kad pati populiariausia ir davusi pamatus kitoms schemoms yra Bonė ir Franklino IBE schema.

Kuo ypatinga schema, kaip ji veikia, kas schemą padaro saugią, kas greitą? Į šiuos klausimus turi būti ieškomi atsakymai darbo vykdymo metu.

2.4. Pradinis projekto aprašymas

Daugumoje ID šifravimo schemų dalyvauja trys pagrindinės šalys: PKG, siuntėjas ir gavėjas. Modelis turi parodyti kokius veiksmus atlieka kieviena šalis ir kokia informacijos dalimi disponuojama. Informacija tarp šių šalių keičiamasi nesaugiais kanalais pvz.: elektroniniu paštu. Iš esmės kanalai schemoje nevaidina svarbaus vaidmens. Todėl į juos bus kreipiama nedaug dėmesio. Svarbus pats principas – pradinė informacija patenka iš PKG pas vartotojus – Siuntėją ir Gavėją (6 pav.), o vėliau Siuntėjas gali siųsti Gavėjui neribotą skaičių užšifruotų žinučių.

Pati schema bus IBE tipo.



6 pav. *Informacijos keitimasis tarp šalių.*

Bet kokioje rimtoje schemoje šiuo metu reikia naudoti elipsines kreives. Čia iškyla dilema, kokios elipsinės kreivės turėtų būti naudojamos. Naudojant supersinguliarias kreives

skaičiavimai atliekami greičiau, ypatingai virš baigtinio binarinio kūno F_2^m, jas paprasčiau naudoti. Tačiau saugumas nukenčia dėl greitumo. Ne-supersinguliarių kreivių atveju saugumas yra didesnis, tačiau veikia lėčiau, nebent pritaikytume Skoto (Scott) [13] siūlomą galimybę pasinaudoti efektyviais endomorfizmais ir taip pagreitinti taškų dauginimo procesą. Tačiau yra dar didesnė problema – tinkamas ne-supersinguliarias kreives sunku surasti, o dar sunkiau yra pasirinkti taškus poravimui. Skirtingai nei supersinguliariųjų kreivių atveju, ne-supersinguliarioms kreivėms negalioja ankščiau minėtoji vienareikšmė atitiktis t.y. nėra galimybės greitai ir efektyviai parinkti poravimui taškus, o tai kritinė problema.

Pagal išvardintas savybes geresnis variantas yra naudoti supersinguliarias kreives.

Tate poravimas yra greitesnis [3] nei Weilo, todėl schemoje skaičiavimams naudosime Tate poravimą. Minėtas BKLS algoritmas už Milerio algoritmą yra greitesnis, todėl jis bus naudojamas Tate poravimui.

3. Darbo eigos aprašymas

3.1. Darbo eigos planas

Išnagrinėjus esamą ID kriptografijos literatūrą, buvo sugalvota numatomos schemos struktūra. Tiksliau tariant, nebus kuriama visiškai nauja, bet modifikuojama esama Bonė ir Franklino IBE schema. Kadangi su Java kalba jau susipažinta jau anksčiau, tai atskirai daug laiko skirti nereikėjo. Iškylančių problemų sprendimų paieškai verta naudotis oficialiu Java tinklalapiu [15], pateikiama dokumentacija ir forumu.

Visą darbų eigą galima suskirstyti į tokius etapus:

- Esamų schemų analizė, kokiomis priemonėmis bandoma realizuoti schemas, kurių dauguma yra teorinės.
- Numatomos schemos projektavimas
- Matematinių priemonių analizė ir pasirinkimas
- Įrankių, kuriais bandoma realizuoti schemą pasirinkimas
- Atikti elementarūs veiksmai su supersinguliariosiomis kreivėmis
- Skaičiuotas Tate poravimas BKLS ir Milerio algoritmų pagalba; ir rastas tinkamas variantas.
- Trijų schemoje dalyvaujančių šalių programinis kūrimas, joms skirtų atlikti veiksmų nustatymas
- Atliktas sėkmingas schemos veikimo įgyvendinimas – Siuntėjas perduoda šifrą Gavėjui, ir šis jį iššifruoja.

3.2. Darbo metu kilusios problemos

Kriptoschemų ID pagrindu yra sukurta nemažai, nes tai sparčiai besivystanti sritis. Beveik kiekviena tyrėjų grupė stengiasi kiek modifikavę kokią prieš tai buvusią schemą, paskelbti apie savo sukurtą naują schemą. Tačiau iš tikro sukurti absoliučiai naujovišką, fundamentalią schemą yra itin sudėtinga. Todėl verta pažvelgti labiau ne į pačios schemos naujoviškumą, bet į principinius dalykus. Kas ir mėginta atlikti šiame darbe. Sprendimas būtų nekurti naujos schemos, bet pasinaudoti esamomis schemomis jas modifikuojant, keičiant, tobulinant.

Matematinių priemonių gana platus spektras, paėmus tam tikrą jų rinkinį kaip schemos pagrindą, galima patyrinėti, kokios yra galimybės šiai schemai būti sparčia ir saugia. Tai turbūt vienas didžiausių galvosūkių, būtent kas yra svarbiau: sparta ar saugumas. Tokiu atveju

dažniausiai renkamasi tarpinis, aukso vidurio variantas, nebent turimas išankstinis poreikis turėti itin sparčią arba itin saugią. Ne išimtis ir šis darbas (turima galvoje ne programinę realizaciją).

Saugumo požiūriu ne itin geras variantas – nesirinkti ne-supersinguliarių kreivių. Čia geriausiai galėtų talkinti matematikai, nes iš esmės tai matematinė problema – teisingas reikalingų poravimui taškų parinkimas. Kadangi, kaip jau minėta, ir pačios kreivės tinkančios kriptografijai suradimas ir taškų parinkimas yra sudėtinga problema, vėliau galinti turėti rimtų pasekmių praktinei realizacijai. Geriausias, bent jau šiuo metu variantas yra rinktis supersinguliarias kreives su kiek mažesniu saugumu. Supersinguliari kreivė (1) turi vienareikšmę atitiktį φ: (x,y) → (-*x*,*yi*) t.y. bet kokiam taškui $P(x,y) \in E(F_p)[l]$ egzistuoja taškas $Q(-x,yi) \in E(F_p^2)[l]$, kur *i* nurodo kompleksinį skaičių (menamasis vienetas √-1). Dėl šios atitikties parinkti teisingus taškus bitiesiniam poravimui tampa nesudėtinga. Be to, būtent šiai kreivei (1) taškų skaičius yra $\#E(F_p) = p + 1$ ir nereikia naudoti kitos, sudėtingesnės taškų apskaičiavimo formulės.

Deja, nepavyko greitesnio BKLS algoritmo priversti visada skaičiuoti teisingai. Nedidelis palyginimas realizuotas Mathematica:

Tarkime turime kreivę $y^2 = x^3 - 10x$, kur $p = 67$.

Pasirenkame tašką $P(37,10)[17]$, ir tašką $Q(41,29)[41]$. Gautas pagal vienareikšmę atitiktį taškas $\varphi Q(26,29i)$.

Atlikus skaičiavimus pagal BKLS gaunamas Tate poravimas $e(P, \varphi Q) = 57 + 13i$, $e(2P, \varphi Q) = 30 + 21i$ t.y. $e(P, \varphi Q)^2 \neq e(2P, \varphi Q)$, nes $(57 + 13i)^2 = 65 + 8i \neq 30 + 21i$. Neatitinka bitiesiškumo savybės.

Atlikus skaičiavimus pagal Milerį (Maas [12]), Tate poravimas $e(P, \varphi Q) = 65 + 59i$, $e(2P, \varphi Q) = 7 + 32i$ t.y. $e(P, \varphi Q)^2 = e(2P, \varphi Q)$, nes $(65 + 59i)^2 = 7 + 32i$. Atitinka bitiesiškumo savybę.

Todėl buvo nuspręsta naudoti Milerio algoritmą pagal Maas.

2.3. Galutinė schemos struktūra

Galutinis projekto variantas yra kiek skiriasi nuo pradinio, svarbiausias skirtumas – Tate poravimo apskaičiavimo algoritmas yra Milerio algoritmas.

Ši schema remiasi Bonė ir Franklino IBE schema. Susideda iš keturių etapų:

1) Pasiruošimas(SETUP): PKG parenkami kreivės parametrai *A* ir *p*. *A* gali būti bet koks sveikasis skaičius, išskyrus 0. *p* yra pirminis skaičius, kur $p \equiv 3 \bmod 4$. Iš tiesų dėl didelio saugumo *p* turėtų būti 160 bitų skaičius (prisiminkime elipsinių kreivių ir RSA saugumo palyginimą), tačiau praktiniame pavyzdyje šis skaičius mažesnis - iki ir apie

300 (nors tokio griežto apribojimo ir nėra), nes dėl skaičiavimų sudėtingumo reikia laukti daug laiko.

Taškų parinkimas – bet koks taškas $P \in E(F_p^2) \neq O, \neq (0,0)$, kurio eilė yra l – pirminis skaičius (programiniame pavyzdyje apribojimas $l > 10$) skaičius. Iš tiesų rinktis mažo l neverta, bet geriausia, jei yra tokia galimybė, kad l būtų mažo Hamingo svorio (*low Hamming weight*), nes tai palengvina Tate poravimo skaičiavimus. Šį l galima perduoti vartotojams, tokiu būdu paspartintume skaičiavimus, kad kiekvieną kartą nebereiktų patiems vartotojams apskaičiuoti.

Toliau pasirenkame valdantįjį raktą $0 < s < l$. Jo negalima niekam atskleisti, jis težinomas vieninteliam PKG.

Kad gautume antrąjį reikalingą tašką, reikia suskaičiuoti $Q = [s]P$.

Visi parametrai, išskyrus, minėtąjį s yra vieši.

Visa tai atliekama vieną kartą. Nereikia kaskart, atsiradus naujam vartotojui arba norint išsiusti žinutę, vis inicijuoti pasiruošimo.

Informacija apie kreivę perduodami visiems vartotojams.

2) Naujo vartotojo atsiradimas(EXTRACT): Atsiradus naujam vartotojui PKG, pagal vartotojo identifikacinę informaciją *ID*, suranda kreivėje $E(F_p)$ tašką $Q_{\mathrm{ID}}[l]$, kuris yra vartotojo viešasis raktas.

   Kad gauti vartotojo privatųjį raktą reikia suskaičiuoti $S_{\mathrm{ID}} = [s]Q_{\mathrm{ID}}$.

   Q_{ID} parametras yra viešas, tačiau S_{ID} turi žinoti tik šis vartotojas, nes turint šifrą ir S_{ID} galima gauti atvirą tekstą.

   S_{ID} parametras perduodamas vartotojui – gavėjui.

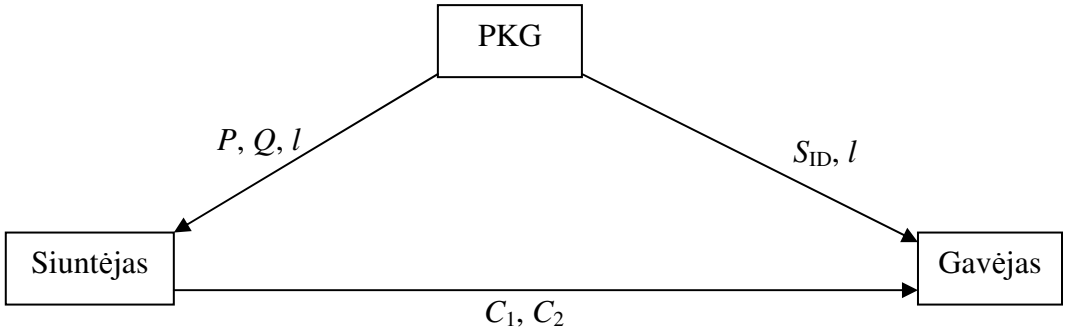
3) Užšifravimas(ENCRYPT): vartotojas – siuntėjas pasirenka atsitiktinį skaičių $0 < r < l$. Žinodamas gavėjo *ID*, siuntėjas turi surasti kreivėje $E(F_p)$ tašką $Q_{\mathrm{ID}}[l]$. Yra galimybė PKG perduoti Q_{ID}, bet tada reikėtų perdavinėti visų naujų vartotojų Q_{ID}, visiems jau esantiems vartotojams, kas būtų didelis nepatogumas. Todėl geriau yra kiekvienam siuntėjui pačiam apskaičiuoti šį tašką atskirai.

   Tada jis suskaičiuoja šifro dalis: tašką $C_1 = [r]P$ ir skaičių $C_2 = m * \mathrm{e}(Q_{\mathrm{ID}}, Q)^r$, kur m yra atviro teksto žinutė išreikšta skaičiumi. Žinutę m geriausia yra skirstyti blokais pvz.: po 10 simbolių (tai įgyvendinta programoje). Visas šifras perduodamas tam tikrais kanalais gavėjui.

4) Iššifravimas(DECRYPT): vartotojui – gavėjui iššifruoti šifrą ir gauti atvirą tekstą yra paprasta - jam tereikia privataus rakto S_{ID}. Atviro teksto žinutė gaunama pagal Tate poravimo bitiesiškumo savybę, suskaičiuojant:

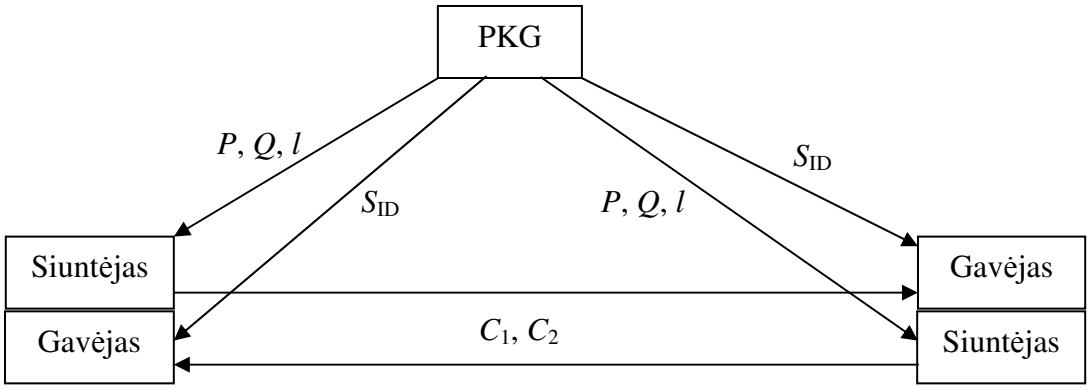
$$C_2 * e(S_{ID},C_1)^{-1} = m * e(Q_{ID},Q)^r * e(S_{ID},C_1)^{-1} = m * e(Q_{ID},[s]P)^r * e([s]Q_{ID},[r]P)^{-1} =$$

$$= m * e(Q_{ID},P)^{rs} * e(Q_{ID},P)^{-rs} = m$$



7 pav. *Informacijos keitimasis tarp šalių schemoje.*

Realiame gyvenime Siuntėjas ir Gavėjas gali būti tas pats asmuo, todėl šis 7 paveikslėlis gali būti papildytas atitinkamais komponentais (8 pav.):



8 pav. *Informacijos keitimasis realiame gyvenime.*

2.4. Darbo rezultatų analizė

Darbo rezultatas – nesudėtingos struktūros schema pagal Bonė ir Franklino schemą bei programinė jos realizacija kartu su konsole, kurioje matomi svarbiausių veiksmų rezultatai ir norimi gauti duomenys. Konkretus šios schemos programinis modelis leidžia skaičiuoti kai *p* yra iki/apie 300. Teoriškai schema yra greita, tačiau praktinis jos įgyvendinimas, dėl tam tikrų

subjektyvių priežasčių yra šiek tiek komplikuotas. Tai nulėmė lėtesnis elipsinių kreivių veiksmų atlikimas. Pati operacija „mod *p*“ taip pat reikalauja nemažų skaičiavimų. Jei būtų naudojami tik *integer* tipo kintamieji, tai skaičiavimai būtų greitesni, šiuo atveju naudojamas kiekvienam kintamajam *BigInteger* objektas, kas taip pat sulėtina programos veikimą. Tačiau, naudoti tik *integer* galima, jei skaičiai yra nedideli. Šiuo atveju, teoriškai skaičiai gali būti labai dideli ir *integer* bei kitų standartinių sveikųjų skaičių saugojimo tipų jiems nepakaktų.

Atviro teksto žinutė yra skaidoma į blokus, todėl jos ilgis teoriškai apie 2^{31}-1 blokų t.y kokią didžiausią reikšmę gali saugoti *integer* tipas; gautųsi apie (2^{31}-1) * 10 simbolių.

Paimkime pavyzdį.

Tarkime naudojame kreivę $y^2 = x^3 - 10x$, kur $p = 67$.

PKG sugeneruotas taškas $P(32,39i)$, $l = 17$, valdantysis raktas $s = 15$, taškas $Q = [s]P = (8,38i)$, Beno viešasis raktas - taškas Q_{ID} (47,21), kai *ID*: benas@su.lt , Beno privatus raktas - taškas $S_{ID} = [s]Q_{ID} = (9,61)$. Sukurti du failai: *Aldona.ini* ir *Benas.ini*, kurie perduoti vartotojams. Šiuose failuose yra visa reikalinga informacija: Aldonai suteikiama informacija apie *P*, *Q*, *l* . Benui tereikia S_{ID} ir *l*.

Aldona, norėdama išsiųsti žinutę Benui, suskaičiuoja Q_{ID} (47,21), kai *ID*: benas@su.lt . Sugalvoja atsitiktinį skaičių $r = 9$. Tokiu būdu pirmoji šifro dalis $C_1 = [r]P = (57,30i)$ ir antroji $C_2 = m * e(Q_{ID},Q)^r$ - skaičių rinkinys:

```
-14962991122579629355264329-34126120104128979231304 61i
30796790426290255724513121+70238293954697074459415 89i
27576188014677605827122387+62893060384352434342559 83i
110409396891+25181090519i
```

Benas, gavo šifrą ir nori iššifruoti. Jam įvedinėti nieko nereikia. Jis tiesiog apskaičiuoja $C_2 * e(S_{ID},C_1)^{-1}$ ir sužino, koks buvo atviras tekstas.

Pavyzdys gautas atlikus skaičiavimus naudojantis programine schemos realizacija.

Esant dabartiniam projekto variantui, geriausias panaudojimo būdas yra elektroninis paštas. Programinė realizacija tai patvirtina.

2.5. Schemos tobulinimas ateityje

Saugumo požiūriu tobulinimas galėtų vykti suradus efektyvų būdą panaudoti ne-supersinguliarias kreives. Tai viena iš galimų schemos vystymo krypčių.

Spartos požiūriu įmanoma veikti keliomis kryptimis:

1) Surasti būdą, kaip tinkamai panaudoti spartesnįjį BKLS algoritmą, nes jis sumažina bendrą skaičiavimų kiekį.

2) Šiame darbe buvo naudojamasi afiniosiomis koordinatėmis t.y. išraiška (x,y), bet galima naudotis ir projekcinėmis koordinatėmis [7] t.y. $(X:Y:Z)$, kur $x = X/Z$ ir $y = Y/Z$. Kiekviena išraiška turi ir savų pliusų ir minusų: skaičiuojant taškų sudėtį bei daugybą afiniosiomis koordinatėmis kiekvieną kartą reikia atlikti operaciją „mod p“, bet reikia mažiau paprastos daugybos (projekcinėms koordinatės taškų sudėtis ir daugyba išreikšta kiek kitaip – naujos taško koordinatės surandamos atliekant daug aritmetinių daugybos veiksmų).
3) Naudojant ne-supersinguliarias kreives reikia panaudoti efektyvius endomorfizmus [13], nes tai itin paspartina skaičiavimus t.y. vietoj brangiai kainuojančios taškų daugybos galima greitai gauti tą patį rezultatą panaudojus endomorfizmą, kuris yra palyginti mažai kainuojanti operacija.

Pati schema galėtų ir turėtų būti pritaikoma smartcard tipo kortelėms. Tik reikalingas itin kruopštus optimizavimas, nes kol kas žvelgiant į programinę realizaciją matyti, kad jai reikia nemažai kompiuterio resursų. Pavykus optimizuoti galima būtų schemą naudoti kortelėse, nes bent kol kas smartcard kortelės nepasižymi dideliais resursais. Plačiau apie poravimų įgyvendinimą kortelėse [14], [20].

Išvados

Norintiems susipažinti su kriptografija ID pagrindu nebereikia ieškoti informacijos daugybėje įvairių šaltinių, nes šiame darbe yra aprašyti svarbiausi ID kriptografijos principai ir keletas esminių schemų, davusių pradžią šiuo metu esamai ir "eksponentiškai" besivystančiai kriptografijai.

Modifikuojant IBE schemą, gautas efektyvios, tačiau nesudėtingos savo sudėtimi teorinis modelis, kuris leidžia ne tik suprasti, kas schemą padaro saugią ir greitą, bet bandyti šią schemą tobulinti, nes dėl savo nesudėtingos struktūros ir plėtimo galimybių tai yra įmanoma padaryti.

Atliktas elementarus programinis šios schemos realizavimas, kas leidžia atlikti bandymus ir geriau suprasti schemos veikimą, nes pateikiami duomenys, kurie paprastai realiame gyvenime nematomi paprastiems vartotojams.

Literatūra

[1] A. Sahai and B. Waters *Fuzzy Identity Based Encryption*, IACR ePrint Archive, Report 2004/086. (http://eprint.iacr.org/).

[2] A. Shamir, *Identity-based Cryptosystems and Signature Schemes*, Proceedings of CRYPTO '84, LNCS 196, p 47 -53, Springer-Verlag, 1984.

[3] Ben Lynn's Homepage http://rooster.stanford.edu/~ben/maths/ep/tate.html [žiūrėta 2006-04-02]

[4] C. Cocks, *An Identity Based Encryption Scheme Based on Quadratic Residues*, Cryptography and Coding - Institute of Mathematics and Its Applications International Conference on Cryptography and Coding - Proceedings of IMA 2001, LNCS 2260, p 360-363, Springer-Verlag, 2001.

[5] Certicom Corp. http://www.certicom.com/ [žiūrėta 2006-03-02]

[6] D. Boneh and M. Franklin, *Identity-Based Encryption from the Weil Pairing*, Proceedings of CRYPTO 2001, LNCS 2139, p 213-229, Springer-Verlag, 2001.

[7] H. Cohen, A. Miyaji, T. Ono, *Efficient Elliptic Curve Exponentiation Using Mixed Coordinates*, ASIACRYPT 1998.

[8] Hankerson, D., Menezes, A., Vanstone, S.: *Guide to Elliptic Curve Cryptography*. p. 75-152 Springer-Verlag (2004)

[9] Joonsang Baek, Reihaneh Safavi-Naini, Willy Susilo, Jan Newmarch 2004, 'A Survey of Identity-Based Cryptography', *AUUG 2004 Who Are You Identification and Authentication Issues in Computing*, AUUG Incorporated, Victoria Australia, p 95-102.

[10] M. Scott. *Computing the Tate pairing*. In CT-RSA, volume 3376 of Lecture Notes in Computer Science, p 293-304. Springer-Verlag, 2005.

[11] Maiklo Skoto tinklalapis apie Tate poravimą http://www.computing.dcu.ie/~mike/tate.html [žiūrėta 2006-02-27]

[12] MASTER'S THESIS *Pairing-Based Cryptography* by Martijn Maas 2004. Appendix A

[13] Michael Scott *Faster Pairings using an Elliptic Curve with an Efficient Endomorphism*, IACR ePrint Archive, Report 2005/252. (http://eprint.iacr.org/).

[14] Michael Scott and Neil Costigan and Wesam Abdulwahab, *Implementing Cryptographic Pairings on Smartcards*, IACR ePrint Archive, Report 2006/144. (http://eprint.iacr.org/).

[15] Oficialus Sun Java kalbos tinklalapis http://java.sun.com/

[16]P.S.L.M. Barreto, H.Y. Kim, B. Lynn and M. Scott. Efficient algorithms for pairing-based cryptosystems. *Advances in Cryptology - CRYPTO 2002*, Lecture Notes in Comput. Sci. 2442, p 354–368, 2002.

[17] Paulo Barreto internetinis tinklalapis http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html [žiūrėta 2006-05-01]

[18] Ronald L. Rivest, Adi Shamir, and Leonard M.Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM 21 (2), p 120-126, 1978.

[19] RSA Laboratories http://www.rsasecurity.com/rsalabs/node.asp?id=2013 [žiūrėta 2006-04- 12]

[20] THE ELLIPTIC CURVE CRYPTOSYSTEM FOR SMART CARDS The seventh in a series of ECC white papers A Certicom White Paper Published: May 1998

[21] V. Miller, *Short programs for functions on curve*, unpublished manuscript, 1986.

Anotacija

Kriptografija identifikacinės informacijos pagrindu yra aktyviai vystoma ir tiriama paskutiniuosius keletą metų, bei kelianti didelį susidomėjimą tyrinėtojų grupėse. Bendruoju atveju, ID kriptografijos esmė – vartotoją identifikuojanti informacija atstojanti viešąjį raktą, leidžia nenaudoti sertifikatų vartotojo autentifikavimui. Šiame darbe apžvelgiamos keletas pamatinių kriptoschemų, kurios daro didžiausią įtaką naujų schemų atsiradimui. Mėginama atsakyti, kas yra reikalinga, norint sukurti efektyvią IBE kriptoschemą. Šiomis dienomis patikimos kriptoschemos kuriamos naudojant elipsines kreives ir bitiesinius poravimus. Todėl šios matematinės priemonės panaudojamos efektyvios schemos kūrimui. Atliekama bandomoji schemos programinė realizacija.

Summary

Identy based cryptography has been, for a few recent years, the most active area of research and currently is of a great interest to the researchers groups. In general case the root of ID based cryptography is that user identifying data is used like a public key and so is no need of certificates for the user authentification. In this work we survey a few basic the most influent cryptoschemes. We attempt to answer what is needed for constructing an efficient IBE scheme. There are trustworthy cryptoschemes that uses elliptic curves and bilinear pairings in nowadays. We use these mathematical implements for efficient cryptoscheme too. And finally, there is a pilot software realization for this scheme.

Kompaktinės plokštelės turinys

Kataloge **dokumentai** saugoma ši informacija:

Titulinis.doc	Magistro darbo titulinis puslapis
Darbo aprasymas.doc	Darbo aprašymas
Vartotojo vadovas.doc	Vartotojo vadovas
Testavimo protokolas.doc	Testavimo protokolas
java docs katalogas	Informacija apie sukurtas klases
CD turinys.doc	Šis dokumentas

Kataloge **projektas** saugomi visų trijų programėlių projektai.

Kataloge **programa naudojimui** saugomos programėlės skirtos naudojimui.

Kataloge **papildoma** saugomos laisvai platinamos programos, kuriomis buvo naudojamasi darbo metu ir sukurtas bandomasis Mathematica failas *BKLS.nb*.