KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Donatas Vilijošius

Sertifikatų sistema GRID tinkle

Magistro darbas

Darbo vadovas
doc. dr. Gytis Vilutis

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Donatas Vilijošius

Sertifikatų sistema GRID tinkle

Magistro darbas

Recenzentas
dr. Agnius Liutkevičius
2010-05-

Vadovas
Doc. dr. Gytis Vilutis
2010-05-

Atliko
IFN-8/3 gr. stud.
Donatas Vilijošius
2010-05-26

Kaunas, 2010

TURINYS

Certificates system in GRID network

SUMMARY

Both in the persons and in the country's life security - the underlying value, which in these days is an integral and very important in information technology world. It is therefore natural that information security is becoming increasingly urgent problem. To ensure the safety of organizations processes the various security tools are being developed. They should ensure the identity of users or services (authentication), to protect communication integrity, privacy, to define who is allowed to carry out all activities and information resources to use (authorization) and the other.

GRID system provides ability to use the WAS (Web Services) and non-WS based authentication and authorization to ensure the GSI (GRID Security infrastructure). Both methods are based on the same basis - X.509 end entity certificates and proxy certificates standard, which is used to identify end entities such as users and services, in addition, allows to assign privileges to other temporary end entities.

The **aim** of the work – investigate a system of certificates, components of these system, which is introduced into GRID systems and to offer the methods or models how to eliminate security weaknesses in existing systems.

The **tasks**:

1. To analyze the GRID security infrastructure elements and to formulate the existing safety problems.
2. To analyze formulated problem-solving techniques.
3. On the basis of problem-solving techniques to design certificate center model and proxy certificates management mechanism.
4. To realize a certificate center model and proxy certificates management mechanism.
5. To perform implemented systems experiments and testing.

After the analysis a certificate center model and proxy certificates management mechanism was realized. Realized certificate center is protected from possible compromise and new proxy certificates management mechanism allows to control the issue of these certificates and allows to protect their from illegal use. The results of experiments showed that additional security measures worsened labor productivity, but these elements increased security.

ĮVADAS

Tiek atskiro žmogaus, tiek ir šalies gyvenime saugumas – pamatinė vertybė, kuri yra neatsiejama bei labai svarbi ir šių dienų informacinių technologijų pasaulyje, nes nutekėjus informacijai, bet kuri organizacija gali patirti didelius finansinius nuostolius ar visiškai žlugti. Todėl, natūralu, kad informacijos saugumas tampa vis aktualesne problema. Organizacijose vykstančių procesų saugumui užtikrinti kuriami įvairūs saugumo įrankiai.

Saugumo įrankiai turi užtikrinti vartotojų ar paslaugų identiškumą (autentifikaciją), apsaugoti bendravimo vientisumą ir privatumą (žinučių apsaugą), apibrėžti kam yra leidžiama kokius veiksmus vykdyti ir informacijos išteklius naudoti (autorizacija), ir teikti (saugius) įrašus, kurie patvirtintų apie esamos politikos vykdymą (apskaita leidžia atlikti politikos laikymosi auditą). Jie taip pat apima pagalbines funkcijas, tokias kaip vartotojų klasifikavimą, grupės narių informacijos aptarnavimą, administravimo teisių suteikimą ir kita [18].

Šiame darbe bus nagrinėjamos reikalingos priemonės saugumo užtikrinimui GRID tinkle, t.y. sertifikatų centras, sertifikatai, proxy sertifikatai ir kita.

GRID sistemos suteikia galimybę naudotis WS (interneto paslaugų) ir ne WS pagrindu paremta autentifikacija ir autorizacija GSI (GRID saugumo infrastuktūrai) užtikrinti. Abu metodai yra pagrįsti tokia pačia baze, t.y. paremti X.509 galinės esybės ir proxy sertifikatų standartu, kuris yra naudojamas identifikuojant pastovias esybes, tokias kaip vartotojai ir paslaugos, be to, leidžia priskirti laikinas privilegijas kitoms esybėms [18].

Šio darbo **tikslas** – ištirti sertifikatų sistemą, šios sistemos komponentus, kurie yra diegiami į GRID sistemas bei pasiūlyti savo metodus ar modelius, kaip būtų galima pašalinti esamų sistemų saugumo trūkumus.

Darbo **uždaviniai**:

1. Išanalizuoti GRID saugumo infrastruktūros elementus ir suformuluoti egzistuojančias saugumo problemas.
2. Išanalizuoti suformuluotų problemų sprendimo būdus.
3. Remiantis problemų sprendimo būdais, suprojektuoti sertifikatų centro modelį ir proxy sertifikatų valdymo mechanizmą.
4. Realizuoti sertifikatų centro modelį ir proxy sertifikatų valdymo mechanizmą.
5. Atlikti realizuotų sistemų eksperimentus ir testavimą.

Darbo **objektas** – sertifikatų sistema GRID tinkle.

Metodai: mokslinės ir periodinės literatūros bei internetinių šaltinių analizė, eksperimentai.

Informacijos šia tema lietuvių kalba pateikiama labai mažai, todėl rašant darbą pagrinde teko remtis informacija pateikiama moksliniuose straipsniuose, internetiniuose šaltiniuose anglų kalba.

Darbo struktūrą sudaro penkios dalys. Pirmame skyriuje analizuojami GRID saugumo infrastruktūros elementai: saugumo raktų koncepcijos, autentifikacijos ir autorizacijos vykdymo aplinkos, saugumą užtikrinančios paslaugos ir sertifikatų centras. Atlikus analizę suformuluojamos pagrindinės saugumo problemos. Antrame skyriuje nagrinėjami kitų autorių siūlomi suformuluotų saugumo problemų sprendimo būdai. Kitame skyriuje aprašomas atliktas siūlomų sprendimo būdų projektavimas. Ketvirtame šio darbo skyriuje pateikiama suprojektuoto sertifikatų centro ir naujo proxy sertifikatų valdymo mechanizmo realizacija. Paskutiniame skyriuje analizuojami atliktų eksperimentų rezultatai ir nurodoma kokie tolimesni darbai turėtų būti vykdomi remiantis gautais rezultatais ateityje.

1. GRID SAUGUMO INFRASTRUKTŪROS ANALIZĖ

Pirmiausia, norint suprasti kas tai yra sertifikavimo sistema, sertifikatai, kaip visa tai veikia, iš kokių elementų tai susideda, reikia susipažinti su pagrindinėmis GSI (GRID saugumo infrastuktūros) saugumo raktų koncepcijomis. Sąvoka koncepcija apibrėžiama kaip pažiūrų į kuriuos nors reiškinius sistema arba kurių nors reiškinių nagrinėjimo būdas, samprata [24]. Šios saugumo raktų koncepcijos suteikia supratimą apie GSI naudojamų elementų prasmę, jų panaudojimo galimybes bei jų veikimą įvairiuose mechanizmuose ir pan.

1.1. Saugumo raktų koncepcijos

GRID saugumo infrastruktūra, kaip bazę savo funkcionalumui, naudoja viešo rakto kriptografiją (taip pat žinomą kaip asimetrinė kriptografija). Dauguma terminų ir koncepcijų naudojamų apibūdinant GSI yra perimti iš viešo rakto kriptografijos. Smulkus šios infraktustūros aiškinamas yra pateiktas rfc2510 dokumente, kurio pagrindu ir bus analizuojami tolimesni faktai [1].

Svarbiausia GSI motyvacija:

- Reikalingas saugus bendravimas (autentifikacija ir konfidencialumas) tarp GRID elementų.
- Reikia užtikrinti saugumą organizacijos ribose, taigi uždrausti centrinio valdymo saugumo sistemą.
- Reikia palaikyti „vieną prisijungimą“ GRID vartotojams, įskaitant įgaliojimus skaičiavimams, kuriems reikia kelių šaltinių ar svetainių resursų [15].

1.1.1. Viešo rakto kriptografija

Viešo rakto kriptografija yra saugaus bendravimo tarp dviejų šalių metodas, kuris nereikalauja pirminio saugaus rakto apsikeitimo. Jis taip pat gali būti naudojamas skaitmeniniam parašui sukurti. Viešo rakto kriptografija yra fundamentali ir plačiai pasaulyje naudojama technologija, kuri leidžia saugiai apsikeisti informacija internete [30].

Daugelyje informacijos šaltinių akcentuojama Diffie ir Hellman sukurto metodo svarba kriptografijos moksle. Esminis dalykas ką reikia žinoti apie viešo rakto kriptografiją yra tai, jog

skirtingai nei ankstesnėse kriptografinėse sistemose, šifravimas priklauso ne nuo vieno rakto (slaptažodžio ar slapto „kodo“), bet nuo dviejų raktų. Šie raktai yra skaičiai, kurie matematiškai susiję taip, kad jei vienas raktas yra naudojamas žinutei užšifruoti, tai kitas turi būti naudojamas žinutei iššifruoti. Taip pat svarbu, jog neįmanoma (su šiuo metu žinoma matematika ir skaičiavimų galimybėmis) išgauti antro rakto iš pirmojo ir neįmanoma iššifruoti jokios žinutės su pirmuoju raktu. Vienas iš raktų yra prieinamas viešai (viešas raktas), o kitas laikomas privačiai (privatus raktas). Asmuo užšifravęs žinutę gali patvirtinti, kad jis ar ji yra privataus rakto šeimininkas, kadangi žinutė gali būti iššifruota tik viešu raktu. Taigi užšifruodamas žinutę asmuo (t. y. jis ar ji) turėjo naudoti savo privatų raktą. Svarbu, kad privatus raktas būtų išlaikomas privačiai, nes kiekvienas žinantis privatų raktą gali lengvai apsimesti jo savininku [36, 20].

1.1.2. Skaitmeniniai parašai

Kitas koncepcijos elementas yra skaitmeninis arba dar vadinamas elektroninis parašas. Naudojant viešo rakto kriptografiją skaitmeniniu būdu galima „pasirašyti“ informacijos dalį. Informacijos pasirašymas iš esmės reiškia patvirtinimą jos gavėjui, kad informacija nebuvo suklastota kol ji pasiekė jo rankas. Norint pasirašyti informacijos dalį, pirmiausia reikia paskaičiuoti matematiškai informacijos santrauką (hash). (Santrauka tai yra sutrumpinta informacijos versija. Algoritmas naudojamas santraukos skaičiavimui turi būti žinomas ir informacijos gavėjui, šis algoritmas nėra paslaptis, tačiau iš algoritmo apskaičiuotos santraukos neturi būti įmanoma atkurti pradinės žinutės). Naudojant privatų raktą reikia užšifruoti santrauką ir pridėti prie žinutės. Informacijos gavėjas norėdamas patikrinti žinutės autentiškumą, turi paskaičiuoti žinutės santrauką naudodamas tą pati algoritmą ir, iššifravęs gautą santrauką, siuntėjo viešu raktu palyginti jas. Jei gautos žinutės apskaičiuota ir iššifruota santraukos sutampa (matematiškai), tai vadinasi, kad žinutė po pasirašymo nebuvo pakeista [10].

Kitame informacijos šaltinyje skaitmeninio pasirašymo schema aprašoma trimis algortimais:

1. Raktų generavimo algoritmas sugeneruoja privatų ir atitinkamą viešą raktą.
2. Pasirašymo algoritmas iš privataus rakto ir žinutės sukuria parašą.
3. Parašo patikrinimo algoritmas gautą žinutę, viešą raktą ir parašą arba priima arba atmeta [9].

1.1.3. Sertifikatai

Svarbiausia GSI autentifikacijos koncepcija yra sertifikatai. Tai elektroniniai dokumentai, kuriais kiekvienas vartotojas ir paslauga yra identifikuojami GRID sistemose. Jie savyje talpina esminę informaciją reikalingą identifikuojant ir autentifikuojant vartotoją ar paslaugą.

GSI sertifikatus sudaro keturios informacijos dalys:

- Subjekto vardas, kuris identifikuoja asmenį ar objektą.
- Viešas raktas, kuris priklauso subjektui.
- Sertifikatų centro identifikatorius, t. y. kas išdavė sertifikatą ir patvirtino, kad viešas raktas ir identifikacija priklauso tam subjektui.
- Skaitmeninis sertifikatų centro parašas [4].

Kitame šaltinyje [29] be šių keturių sertifikatus sudarančių dalių dar minima, kad sertifikatuose taip pat nurodomas ir jų galiojimo laikas. Sertifikatai gali būti išduodami įvairiam laikotarpiui ar net kokiai tai operacijai atlikti.

Reikia pažymėti, kad trečioji šalis (sertifikatų centras) yra naudojamas, kad patvirtintų ryšį tarp viešo rakto ir subjekto vardo įrašyto sertifikate. Kad būtų galima pasitikėti sertifikatu ir jo turiniu, turi būti patikimas pačio sertifikatų centro sertifikatas. Ryšys tarp sertifikatų centro ir jo išduoto sertifikato turi būti žinomas kokiomis nors nekriptografinėmis priemonėmis, nes kitu atveju sistema būtų nepatikima.

Kaip teigiama, GSI sertifikatai yra šifruojami X.509 sertifikatų formatu, tai standartinis duomenų formatas sertifikatams, kuris yra nustatytas IETF (Internet Engineering Task Force). Šie sertifikatai gali būti naudojami ir su kitomis viešo rakto bazę naudojančiomis programomis ar sistemomis, pavyzdžiui, interneto naršyklėmis Microsoft, Netscape ar kitomis [4].

1.1.4. Abipusė autentifikacija

Abipusė autentifikacija – tai procesas, kuris yra glaudžiai susijęs su ankščiau nagrinėtomis koncepcijomis ir jų panaudojimu. Todėl visą procesą pamėginsime išsiaiškinti išsamiau.

Jei dvi šalys turi sertifikatus ir jei jos pasitiki sertifikatų centrais, kurie išdavė abiejų šalių sertifikatus, tada abi šalys viena kitai gali patvirtinti, kad jos yra tuo, kuo ir sako. Tai ir vadinama abipuse autentifikacija. Šiam procesui GSI naudoja SSL protokolą (SSL taip pat žinomas nauju IETF standarto vardu: Transportinio sluoksnio saugumas arba TLS).

Prieš vykdant abipusę autentifikaciją, abi šalys pirmiausia turi pasitikėti sertifikatų centrais, kurie kiekvienai iš šalių pasirašė sertifikatus. Praktikoje tai reiškia, kad abi šalys turi turėti sertifikatų centrų sertifikatų kopijas, kuriuose yra sertifikatų centrų viešieji raktai, ir taip pat abi šalys turi tikėti, kad šie sertifikatai tikrai priklauso tiems sertifikatų centrams.

Daugelyje šaltinių yra nagrinėjama abipusė autentifikacija tarp kliento ir serverio, tačiau šiame darbe bus aprašytas bendras šio proceso veikimas.

Išskiriami tokie abipusės autentifikacijos proceso žingsniai:

- Asmuo (A) jungiasi su antruoju asmeniu (B).
- A duoda asmeniui B savo sertifikatą, taip A pradeda autentifikacijos procesą.
- Gautas sertifikatas asmeniui B pasako kas yra asmuo A (identifikuoja), koks A asmens viešas raktas ir koks sertifikatų centras išdavė jo sertifikatą.
- Asmuo B pirmiausia įsitikina, kad sertifikatas yra tikras. Jis patikrina sertifikatų centro skaitmeninį parašą norėdamas įsitikinti, kad sertifikatų centras iš tikro pasirašė sertifikatą ir, kad šis sertifikatas nėra suklastotas. (Šioje vietoje asmuo B turi pasitikėti sertifikatų centru, kuris išdavė sertifikatą).
- Kai asmuo B patikrina A sertifikatą, tada jis turi įsitikinti, kad asmuo A tikrai yra tas asmuo, kurį identifikuoja sertifikatas.
- Asmuo B sugeneruoja bet kokią žinutę ir nusiunčia ją asmeniui A prašydamas jo tą žinutę užšifruoti.
- Asmuo A užšifruoja žinutę naudodamas savo privatų raktą ir nusiunčia ją atgal asmeniui B.
- Asmuo B žinutę iššifruoja naudodamas asmens A viešą raktą. Ir jei gaunama ta pati sugeneruota žinutė, tada asmuo B žino, kad asmuo A yra tuo, kuo jis ir sako.
- Kai B asmuo įsitikina asmens A identiškumu, tos pačios operacijos vykdomos priešingai asmenų atžvilgiu. Asmuo B siunčia asmeniui A savo sertifikatą, A jį patikrina ir nusiunčia žinutę prašydamas ją užšifruoti. Asmuo B užšifruoja ir nusiunčia atgal asmeniui A, A iššifruoja ir palygina gautą žinutę su originalia. Jei jos sutampa, tai asmuo A įsitikina asmens B identiškumu.
- Po visų šių veiksmų, asmenys A ir B užmezga ryšį vienas su kitu ir tiki vienas kito identiškumu [26].

1.1.5. Konfidencialus (slaptas) bendravimas

Konfidencialus bendravimas yra informacijos apsikeitimo būdas tarp dviejų žmonių, kai bendravimas turi išlikti privatus ir informacija turi būti neatskleidžiama kitiems, t. y. ji turi būti saugi [27].

Pagal numatytus nustatymus, GSI nevykdo konfidencialaus (šifruoto) bendravimo tarp atskirų dalių. Kai yra įvykdoma abipusė autentifikacija, GSI „išeina iš kelio“, taigi tada bendravimas gali būti vykdomas ne pagal numatytąsias šifravimo ir iššifravimo sąlygas. Jei yra nusprendžiama vykdyti konfidencialų bendravimą, GSI šifravimui gali lengvai naudoti pasidalintą raktą. Kaip saugumo ypatybę GSI vykdo bendravimo integralumą. Integralumas reiškia, kad slaptas pasiklausymas tarp dviejų šalių yra įmanomas, bet ryšio modifikavimas neįmanomas. Ryšio integralumas GSI yra nustatomas pagal numatymą (norint, tai gali būti išjungta). Ryšio integralumas suteikia bendravimui saugumo, tačiau ne tiek daug, kaip ryšio šifravimas [5].

1.1.6. Privačių raktų apsauga

Kitas koncepcijos elementas yra privačių raktų apsauga. Pagrindinė GSI programinė įranga (pvz., Globus Toolkit) numato, kad privačius raktus vartotojai laikytų failo pavidalu kompiuterio atmintyje. Siekiant apsaugoti nuo privataus rakto pavogimo, failas, kuris talpina privatų raktą, yra užšifruojamas slaptažodžiu. Norint naudoti GSI, vartotojui pirmiausia reikia įvesti slaptažodį, kuris reikalingas failo iššifravimui, kuriame ir yra privatus raktas.

Teigiama, kad taip pat yra galimybė vartotojams privačius raktus saugoti kriptografinėje intelektualioje kortelėje ir taip padidinti apsaugą nuo galimo privataus rakto pavogimo. Tokių kortelių naudojimas dar labiau pasunkina priėjimą pašaliniams asmenims prie slapto privataus rakto [32].

Šis koncepcinis elementas yra pats svarbiausias GSI ir šio rakto apsaugai turi būti skiriamas pats didžiausias dėmesys. Atsiradę pažeidimai šiame lygmenyje reiškia, kad pasitikėjimo grandinė nutrūksta, nes privatus raktas yra naudojamas identiškumui įrodyti. Jei identiškumo nelieka, tai nebegalima įrodyti, kad tu esi savimi ir taip pat, kad ką tu pasirašei buvo pasirašyta teisėtai, todėl visa pasitikėjimo grandinė subyrą. Kuo aukštesnės instancijos privaktus raktas pažeidžiamas tuo šis pažeidimas gali atnešti daugiau žalos. Visą žalos dydį galėtume išreikšti

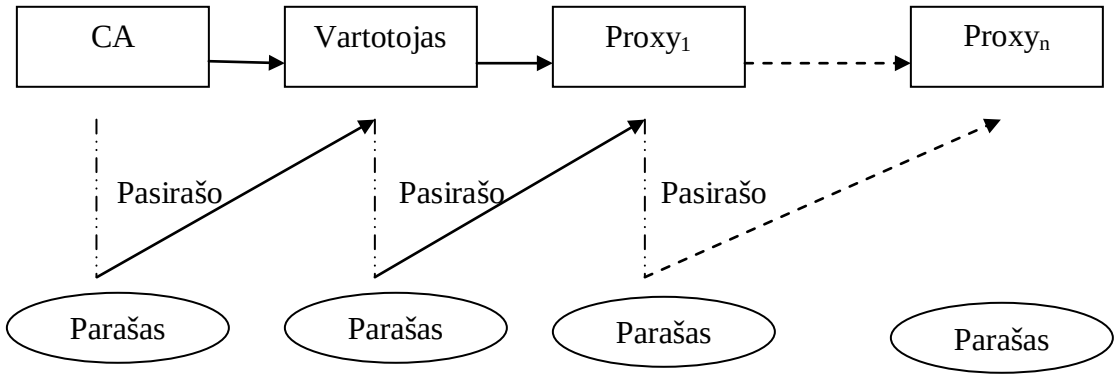
eksponentiniu dydžiu, kuris priklauso nuo to kiek lygių po tavęs kiti sertifikatai dar buvo pasirašinėjami.

1.1.7. Teisių perdavimas, vienas prisijungimas ir Proxy sertifikatai

Vykstant bendravimui tiesiogiai su nutolusiomis paslaugomis galima naudoti savo sertifikatą, taip įrodant savo identiškumą. Tačiau, GRID sistemose dažnai norima, kad nutolusios paslaugos veiktų vartotojo vardu. Pavyzdžiui, užduočiai veikiančiai kokioje nors nutolusioje svetainėje reikia susikalbėti su kitais serveriais ir perduoti failus, o tada šiai svetainei reikia įrodyti, kad jai yra duota teisė naudotis to vartotojo tapatybe. Kitaip sakant, kadangi privatus raktas yra labai svarbus, tai nenorima jo siųsti nutolusiai mašinai, kuri gali būti nesaugi [2]. Kad kiti serveriai vykdytų norimas užduotis, tapatybę įrodyti būtina, o tai šiuo atveju įmanoma padaryti tik patvirtinant savo tapatybę pačiam. Be to, prisimenant tai, jog privatų raktą reikia labai saugoti ir tai, kad jis yra apsaugotas slaptažodžiu, reiškia, jog kiekvieną kartą patvirtinant tapatybę reikės įvedinėti slaptažodį. Dėl šios priežasties buvo pradėta ieškoti alternatyvų kaip išvengti to ir saugiai perduoti savo teises kitam.

GSI suteikia teisių perdavimo galimybę, tai yra standartinio SSL protokolo praplėtimas leidžia suteikti savo teises kitam bei išvengti privataus rakto slaptažodio įvedinėjimo kas kartą, kai atliekama autentifikacija. Pavyzdžiui, jeigu GRID skaičiavimams reikia, kad būtų panaudojama keletas jo resursų (prie kiekvieno resurso reikia abipusės autentifikacijos) arba jei yra poreikis turėti agentus (vietinius ar nutolusius), kurie vartotojo vardu prašo paslaugų, vartotojo slaptažodis turetų būti įvedinėjamas kas kartą. Tačiau įvedinėjimo poreikis gali būti išvengiamas sukuriant įgaliotinius (proxy).

Proxy susideda iš naujo sertifikato ir privataus rakto. Raktų pora, kurią naudoja proxy, susideda iš sertifikate įdėto viešo rakto ir naujo privataus rakto, kurie yra sugeneruojami iš naujo kiekvienam proxy arba gaunami kitais būdais. Naujas sertifikatas savyje turi savininko identifikatorių, kuris yra šiek tiek pakeistas, kad būtų galima identifikuoti, jog tai yra proxy. Naujasis sertifikatas yra pasirašomas vartotojo, o ne sertifikatų centro (žr. 1 pav.). Taip pat sertifikatas turi laiko žymą, kuriai pasibaigus proxy sertifikato kiti turi nebepriimti. Įgaliotiniai (proxies) turi ribotą galiojimą, kuris paprastai būna nuo 12 valandų iki 7 dienų [8].



1 pav. Proxy sertifikatų išdavimo procedūra [8]

Proxy sertifikato privatus raktas turi būti laikomas saugiai, tačiau kadangi proxy negalioja labai ilgai, todėl jis nėra saugomas taip stipriai kaip savininko privatus raktas. Proxy privatus raktas laikomas neužšifruotas vietinėje duomenų laikymo sistemoje, nes jis naudojamas daug kartų identifikacijai patvirtinti. Kitaip sakant šis raktas yra apsaugotas tik tiek, kiek failų sistemų teisės apsaugo jį nuo to, kad niekas negalėtų lengvai jo peržiūrėti. Kai proxy yra sukuriamas, vartotojas abipusei autentifikacijai gali naudoti proxy sertifikatą ir jo privatų raktą, taip išvengdamas slaptažodžių įvedinėjimų.

Teigiama, jog abipusės autentifikacijos procesas šiek tiek skiriasi, kai yra naudojami įgaliotiniai (proxies). Nutolusi šalis gauna ne tik proxy sertifikatą (pasirašytą vartotojo), bet ir savininko tikrąjį sertifikatą. Abipusės autentifikacijos procese savininko viešas raktas (gautas iš jo sertifikato) yra naudojamas proxy sertifikato parašui patikrinti. O tada sertifikatų centro viešas raktas yra naudojamas vartotojo sertifikato parašui patikrinti. Tai atitinka pasitikėjimo grandinę, kuri prasideda nuo sertifikatų centro ir tęsiasi iki proxy sertifikato, o jų tarpe yra vartotojas [35].

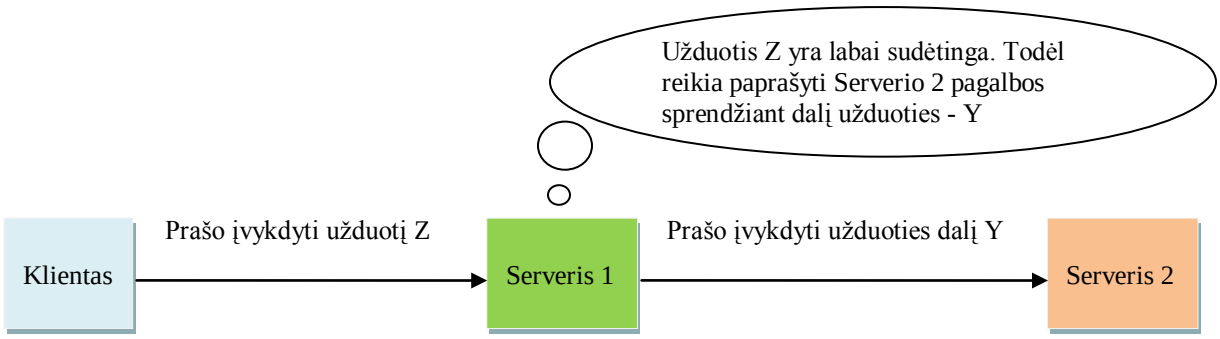
GSI ir ja paremta programinė įranga (ypač Globus Toolkit, GSI-SSH ir GridFTP), šiuo metu yra vienintelė programinė įranga, kuri palaiko teisių perdavimo praplėtimą naudojant TLS (SSL).

Tolimesnis žingsnis, kurį reikia apžvelgti, yra pačio proxy sertifikato išsamus sukūrimo procesas, kurį apibrėžia rfc 3820 dokumentas. Panagrinėsime anksčiau minėtą pavyzdį, kai GRID sistemose reikalingas teisių perdavimas kitai paslaugai ar vartotojui, o tai leidžia privataus rakto slaptažodį įvesti tik vieną kartą.

Tarkime Klientas prašo Serverio 1 atlikti užduotį ir kadangi Serveris 1 pasitiki Klientu, tai jis priima užduotį. Tačiau sakykime, jog užduotis Z yra labai sudėtinga ir, kad viena šios užduoties dalis Y turi būti perduota trečiai organizacijai – Serveriui 2. Šiuo atveju, Serveris 1 prašo Serverio 2 įvykdyti dalinę užduotį Y, bet Serveris 2 pasitiki tik Klientu (žr. 2 pav.) ir kol

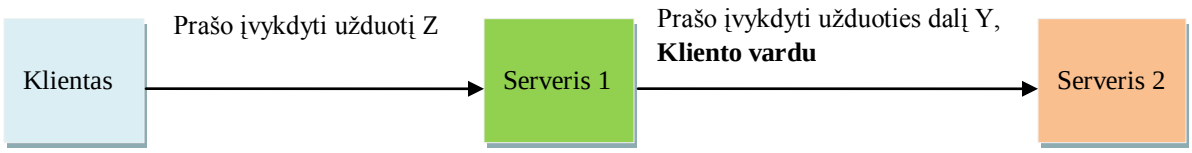
nėra naudojami proxy sertifaktai, tokiu atveju Serveris 2 turi dvi galimybes, kaip pasielgti su Serverio 1 prašymu:

- **Nepriimti Serverio 1 prašymo,** nes Serveris 2 nepasitiki Serveriu 1.
- **Priimti Serverio 1 prašymą**. Vis dėlto, pradinis prašymas buvo Kliento taigi, nors Serveris 2 atsakinėja į Serverio 1 prašymą, tačiau iš tikro užduotis atėjo iš Kliento.



2 pav. Užduoties vykdymo pasidalinimas tarp kelių šalių

Šioje situacijoje atrodo logiška, kad Serveris 2 turėtų priimti Serverio 1 prašymą. Tačiau Serveris 2 turi žinoti, kad prašymas ateinantis iš Serverio 1 yra Kliento vardu (žr. 3 pav.).

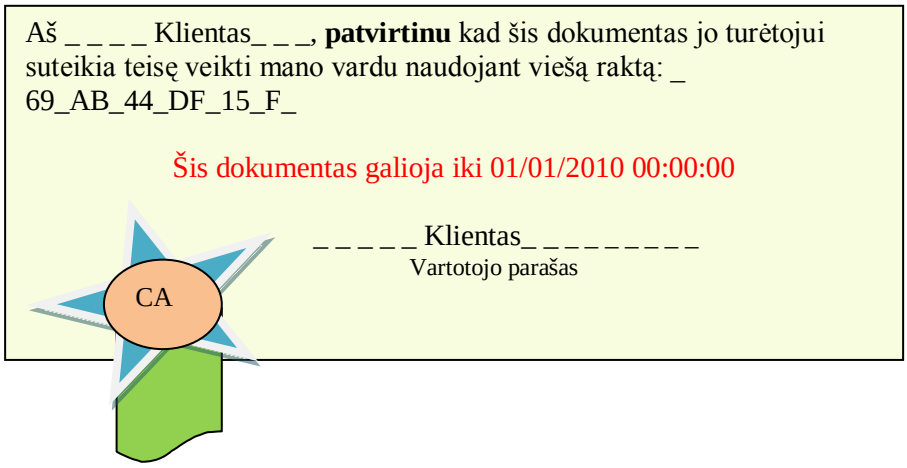


3 pav. Užduoties vykdymo prašymas kitos šalies vardu

Tačiau, tai nėra saugus sprendimas, nes kiekvienas gali tvirtinti, jog veikia Kliento vardu. Vienas iš galimų sprendimo būdų būtų Serveriui 2 susisiekti su Klientu kiekvieną kartą, kai yra gaunamas prašymas Kliento vardu. Tačiau, tai yra nepatogu, nes, patvirtindamas užduoties tikrumą, Klientas kas kartą turės įvedinėti savo privataus rakto slaptažodį. O įsivaizduojant, kad užduotis Z susideda iš 20 mažesnių užduočių ir, kad kiekviena mažesnė užduotis yra pasiunčiama skirtingoms organizacijoms, reiškia, kad Klientas bus „užpiltas“ žinutėmis, kuriose bus sakoma: „Serveris 1 paprašė manęs įvykdyti užduotį jūsų vardu ir ar jūs patvirtinate jog tai tiesa”.

Arba galimas kitas sprendimo būdas būtų kažkaip priversti Serverį 2 patikėti, kad Serveris 1 yra Klientas. Kitaip sakant, reikia rasti legalų būdą, kad Serveris 1 galėtų parodyti, jog jis veikia Kliento vardu. Tai įmanoma Klientui paskolinant savo viešą ir privatų raktą Serveriui 1, kad jis galėtų atlikti abipusę autentifikaciją, tačiau tai yra nepriimtina, nes privatus raktas turi išlikti paslaptimi ir jo siuntimas kitai organizacijai (nepriklausomai nuo to kaip ja pasitikima) yra didelis saugumo pažeidimas, t. y. tai pažeidžia privataus rakto apsaugos koncepciją.

Kitas daug geresnis ir priimtinesnis sprendimo būdas būtų išduoti Serveriui 1 įgaliojimus arba kitaip sakant atitinkamą sertifikatą. Tačiau vien tik įgaliojimų išdavimas tai dar ne viskas. Leidimas veikti kieno nors vardu yra besąlygiškai rizikingas. Pasitikėti kitu galima tol, kol jis atliks atitinkamą užduotį. Nes kitaip kas nors iš Serverio 1 organizacijos šiuo sertifikatu gali pasinaudoti ateityje Kliento vardu siekiant blogų tikslų. Todėl tokio sertifikato gyvavimo laikas turi būti ribotas (paprastai 12 valandų) (žr. 4 pav.). Ir tai dar reiškia, kad jei šis sertifikatas bus sukompromituotas, įsilaužėlis negalės iš jo „išpešti“ daug naudos.



4 pav. Teisių perdavimo sertifikato pavyzdys

Ką tik aptarti sertifikatai, GRID tinkluose yra vadinami proxy sertifikatais. Pagal Webster's žodyną “proxy” yra vadinamas instrumentas, kuriuo asmuo yra įgalinamas veikti kito vardu. Proxy sertifikatas leidžia jo turėtojui veikti Kliento vardu, kas dar vadinama teisių perdavimu (credential delegation). Be viso to, proxy sertifikatas, kitaip nei paprastas X.509 sertifikatas, turi dar papildomų saugumą užtikrinančių parametrų, kurie leidžia apriboti funkcionalumą dar labiau (pavyzdžiui, nurodant, kad proxy sertifikatas gali būti naudojamas tik atitinkamai užduočiai įvykdyti ar apribotas naudotis tik atitinkamais resursais, o ne visais, ką leistų tikras sertifikatas) [34].

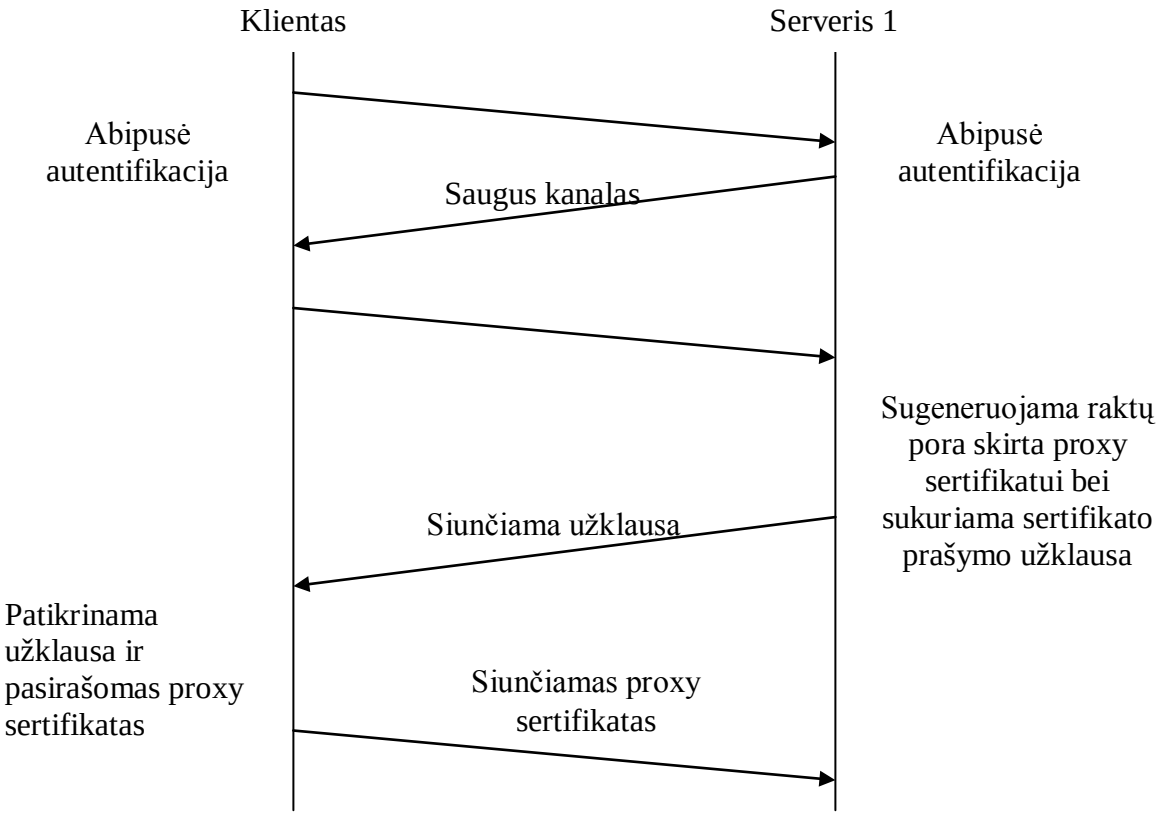
Faktiškai, šis sertifikatas labai panašus į X.509 skaitmeninį sertifikatą, išskyrus tai, kad jis yra pasirašytas ne sertifikatų centro, bet vartotojo. Įsitikinti jo autentiškumu galima patikrinus Kliento parašą (Klientas sertifikatą pasirašo savo privačiu raktu). O sukurtas privatus ir viešas raktas, proxy sertifikatui nėra nei vartotojo nei sertifikatų centro, todėl galima sakyti, kad jis yra niekieno. Proxy sertifikatas turi privataus ir viešo rakto porą, kuri yra sugeneruota specialiai proxy sertifikatui. Ši raktų pora yra abipusiškai patvirtinta abiejų šalių (šiuo atveju Kliento ir

Serverio 1), todėl Kliento organizacija, šių raktų turėtojui leidžia veikti jo vardu (šiuo atveju Serveriui 1).

Naudojantis proxy sertifikatais yra išsprendžiama ne tik teisių perdavimo problema, bet ir panaikinama būtinybė įvedinėti tikro privataus rakto slaptažodį kas kartą, kai reikalinga autentifikacija. Šiuo atveju slaptažodį panaudoti reikia tik vieną kartą – pasirašant proxy sertifikatą. O tada visas reikalingas autentifikacijas jau atlieka šis sertifikatas. Proxy sertifikatas turi dar vieną privalumą, kad jis gali būti sukuriamas ir naudojamas vietinėje sistemoje, taip apsaugant visus bendravimus ir tam nebenaudojant tikrojo privataus ir viešo rakto. Taip sumažinama grėsmė, kad bus pažeista tikrojo privataus rakto apsauga, dažnai įvedinėjant slaptažodį (slaptažodis tokiu atveju būtų visada reikalingas tik pasirašant proxy sertifikatą) [34].

Paskutinis svarbus dalykas, ką reikia aptarti, tai proxy sertifikato generavimo mechanizmas ir kaip užtikrinamas to mechanizmo saugumas (žr. 5 pav.). Remiantis tuo pačiu pavyzdžiu su Klientu, Serveriu 1 ir Serveriu 2 procesas yra toks:

1. Klientas ir Serveris 1 naudodami SSL įvykdo abipusę autentifikaciją naudodami savo X.509 sertifikatus ir sukuria autentifikuotą, vientisą bendravimo kanalą.
2. Serveris 1 sugeneruoja viešo ir privataus raktų porą, kurie bus skirti proxy sertifikatui.
3. Serveris 1 pasinaudodamas šiais raktais sukuria sertifikato prašymo užklausą, kurią siunčia Klientui apsaugotu kanalu. Ši sertifikato prašymo užklausa savyje turi sugeneruotą viešą, bet ne privatų raktą.
4. Kadangi Klientas sutinka duoti savo teises Serveriui 1, tai pats Klientas patikrina ar sertifikato prašymo užklausa atitinka proxy sertifikato profilį ir pasirašo pasinaudodamas savo privačiu raktu.
5. Klientas pasinaudodamas saugiu kanalu siunčia pasirašytą sertifikatą atgal Serveriui 1.
6. Serveris 1 gavęs proxy sertifikatą gali veikti Kliento vardu [31].



5 pav. Proxy sertifikato generavimo mechanizmas [31]

Pabrėžiama, kad nei proxy sertifikato nei Kliento privatus raktas niekada nėra siunčiamas tarp Kliento ir Serverio 1.

Proxy sertifikato tikrinimas galima sakyti, jog vyksta identiškai kaip ir tikro sertifikato. Skirtumas yra tik tas, kad proxy sertifikatas yra pasirašytas ne sertifikatų centro, o vartotojo. Tęsiant pavyzdį, proxy sertifikatas yra pasirašytas Kliento, todėl reikia patikrinti jo viešo rakto autentiškumą. Kadangi Serveris 2 tikėtina, jog neturės Kliento sertifikato, todėl jis yra atsiunčiamas kartu su proxy sertifikatu, o tai leidžia patvirtinti proxy sertifikatą. Tada belieka patikrinti sertifikatų centro parašą, nes jis yra pasirašęs ant Kliento sertifikato. Taigi visą šią grandinę parašų galima surasti proxy sertifikate [34].

1.2. Autentifikacijos ir autorizacijos vykdymo aplinkos

Prieš pateikiant bet kokius duomenis, informacijos tiekėjas turi įsitikinti, kad perduoda duomenis būtent tam asmeniui, kuris padarė užklausą. Atitinkamai užklausą padaręs asmuo turi autentifikuoti save, t. y. įrodyti savo tapatybę. Vienas iš saugiausių būdų tai padaryti – autentifikuotis naudojant skaitmeninį sertifikatą [23].

Taigi teigiama, jog autentifikacija yra reikalinga vartotojo tapatybei identifikuoti. Tam padaryti dažniausia naudojamas susietas vartotojo vardas ir slaptažodis. Šios sistemos pagrindas yra vartotojų ir jų atributų duomenų bazė (LDAP, NIS, RDBVS, MS Active Directory). Taip pat išskiriami tokie vartotojus identifikuojantys duomenys, kaip vartotojo vardas ir slaptažodis, skaitmeniniai sertifikatai, identifikavimo kortelės, biometriniai duomenys ir kita.

Autorizacija nustato ar jau identifikuotas vartotojas turi teisę naudotis paslauga. Jos metu tikrinama ar vartotojas priklauso tokį leidimą turinčiai grupei, ar yra tinkamas jo saugumo lygis [12]. Taigi galime teigti, kad šie du procesai yra susiję.

Išsiaiškinus autentifikacijos ir autorizacijos esmę, toliau bus analizuojamos dvi realizavimo galimybės. Bus mėginama išskirti esminius jų panašumus ir skirtumus.

1.2.1. Ne WS (Interneto paslaugomis) paremta autentifikacija ir autorizacija

GRID ne internetinių paslaugų autentifikacijos ir autorizacijos komponentai naudoja APIs (Aplikacijų kūrimo interfeisus) ir įrankius autentifikacijos, autorizacijos ir sertifikatų valdymui. Autentifikacijos API naudoja viešo rakto infrastruktūros (PKI) technologijas, X.509 sertifikatus ir TLS. Papildomos autentifikacijos galimybės, kai naudojamas teisių perdavimo mechanizmas, yra paremtos X.509 proxy sertifikatais. Autorizacijos palaikymas reikalauja dviejų API formų naudojimo. Pirmasis iš jų vykdo bendrą autorizaciją, kuri leidžia vykdyti kliento kvalifikacija paremtą priėjimo kontrolę (X.509 sertifikatų grandinė). Antrasis teikia paprastą priėjimo kontrolės sąrašą, kuris sužymi autorizuotas nutolusias esybes vietinėje sistemoje vartotojų vardais. Antrasis mechanizmas taip pat leidžia iškvietimus, kurie leidžia trečiai šaliai nepaisyti standartinių taisyklių ir šis mechanizmas paprastai naudojamas „Gatekeeperyje“ ir GripFTP serveryje. Be šitų dviejų mechanizmų, yra įvairių žemesnio lygio APIs ir įrankių sertifikatų valdymui [17].

1.2.2. WS (Interneto paslaugomis) paremta autentifikacija ir autorizacija

GRID interneto paslaugos bendravimui, žinutėms perdavinėti naudoja SOAP per HTTP. WS autentifikacija ir autorizacija naudojant JAVA realizuoja WS saugumo standartą ir WS saugaus pokalbio specifikaciją, kurie užtikrina SOAP žinučių apsaugą.

Suteikiamos galimybės:

- Siuntėjo autentifikacija.
- Žinučių šifravimas.
- Integruota žinučių apsauga.
- Apsauga nuo „replay“ atakų.

Java WS autentifikacija ir autorizacija vykdoma sukuriant saugų kanalą naudojant HTTP per SSL/TLS (HTTPS), per kurį transportuojamos žinutės. Šis saugumo mechanizmas palaiko visas saugumo ypatybes teikiamas SSL/TLS, be to, turi galimybę palaikyti X.509 proxy sertifikatus. Autorizacijos komponentas leidžia infrastruktūrai naudoti atributus ir apsaugoti priėjimą prie išteklių pagal priėjimų politiką. Tai leidžia sukonfigūruoti ir vykdyti autorizacijos politiką įvairiais lygiais (konteineriams, paslaugoms ar resursams). Tai taip pat leidžia kliento pusės autorizaciją, t. y. leidžia autorizuoti paslaugas, kurias klientai gali naudoti. Šis mechanizmas yra lankstus ir jis gali būti sukonfigūruotas taip, kad būtų galima naudoti įvairius mechanizmus atributų kolekcijoms ir politikos nustatymui. Be viso to, jis taip pat leidžia daugialypį autorizacijos modulio realizavimą, pavyzdžiui, palaikyti gridmap pagrindu paremtą autorizaciją, iškviesti modulius, kurie naudoja SAML protokolą ir naudojantis juo gali iškviesti papildomas paslaugas, kurios nuspręstų dėl autorizacijos ir pan. [17].

Apžvelgus šias dvi autentifikacijos ir autorizacijos realizavimo galimybes, negalima pastebėti labai didelių skirtumų. WS autentifikacijos ir autorizacijos mechanizmas yra universalesnis, jis gali būti realizuojamas daugiau nei vienoje platformoje. Ne WS A&A realizacija galima tik linux sistemoje, o WS A&A realizacija gali būti vykdoma ir Windows 2000 platformoje ar Solaris. Abu šie realizavimo modeliai taip pat gali būti atliekami naudojant tiek Java technologijas tiek ir C.

1.3. Saugumą užtikrinančios paslaugos

Atlikus analizę, kokiais būdais galima realizuoti GSI ir kokių komponentų realizavimas reikalingas, toliau bus nagrinėjamos atskirų saugumo paslaugų realizavimo galimybės. Viena iš paslaugų, kuri gali būti realizuota GRID tinkle, tai įgaliojimų saugojimo paslauga, vadinama MyProxy. Toliau bus truputį plačiau apžvelgiama kam ji reikalinga.

1.3.1. MyProxy

MyProxy yra realaus laiko rėžimu veikianti įgaliojimų saugykla. Joje galima saugoti X.509 proxy įgaliojimus apsaugotus slaptažodžiais, kuriuos vėliau galima panaudoti tinkle. Tai panaikina poreikį kopijuoti privačius raktus ir sertifikatų failus tarp kompiuterių. MyProxy taip pat gali būti naudojamas autentifikuojantis prie GRID portalų ir atnaujinant įgaliojimus su darbų valdytojais (job managers).

Kadangi standartiniai interneto saugumo protokolai nepalaiko GRID portalų poreikio, todėl reikalinga ši programinė įranka, kuri tai realizuoja. Ji leidžia per interneto naršyklę pasiekti GRID portalus, be to, leidžia vartotojui juos pasiekti iš tokių vietų, kur jo priėjimas prie GRID nebūtų įmanomas [25].

1.3.2. Teisių (įgaliojimų) perdavimo paslauga

Įgaliojimų perdavimo paslauga sukuria aplinką, kuri leidžia perduoti įgaliojimus hostingo aplinkai. Ji taip pat suteikia įgaliojimų atnaujinimo būdus. Tai yra GRID interneto paslauga, kuri paremta WS-Trust ir WSRF specifikacijomis. Ši paslauga palengvina vartotojo įgaliojimų perdavimą kitoms tame pačiame konteineryje veikiančioms paslaugoms. Svarbu paminėti ir tai, kad įgaliojimų perdavimo paslauga leidžia įgaliojimus atnaujinti nuotoliniu būdu. Šis komponentas iš tikro susideda iš dviejų susijusių paslaugų – faktorių perdavimo paslaugos ir perdavimo paslaugos [7].

1.3.3. Bendruomenės autentifikacijos paslauga

Kita svarbi paslauga yra bendruomenės autentifikacijos paslauga (CAS), kuri leidžia virtualiai organizacijai išreikšti politiką ir yra susijusi su resursų paskirstymu tarp įvairaus kieko svetainių. CAS serveris tvirtina virtualios organizacijos vartotojus, suteikdamas jiems tam tikras priėjimų teises prie išteklių. Serveriai atpažįsta vartotojus ir suteikia tik jiems skirtas teises. CAS gali būti naudojama kelioms paslaugoms ir ji šiuo metu yra palaikoma per GridFTP serverį ir interneto paslaugas [28].

1.4. Sertifikatų centras

Ankstensėje dalyje išnagrinėjus GSI kocepcijas galima teigti, kad sertifikatų centras (CA) yra neatsiejama GRID dalis. CA vartotojams išduoti sertifikatai yra naudojami autentifikacijai, autorizacijai, norint naudotis paslaugomis ir pan., t.y. sertifikatai naudojami kiekviename žingsnyje.

Norint turėti savo sertifikatų centrą ir jį realizuoti, pirmiausia reikėtų apžvelgti kaip GRID saugumo federacija (Grid Trust Federation) apibūdina tradicinio X.509 PKI sertifikatų centro minimalius reaikalavimus. Tradiciniai X.509 viešo rakto sertifikatų centrai išduoda ilgalaikius sertifikatus galutiniams vartotojams, kurie patys valdo ir kontroliuoja savo raktų porą ir savo aktyvacijos duomenis. Šie CA veikia kaip nepriklausoma patikima trečioji šalis tiek sertifikatų turėtojams tiek tų sertifikatų tikrintojams. Šie sertifikatų centrai naudoja ilgalaikius pasirašymo raktus, kurie yra saugomi saugioje aplinkoje, kurią apibūdina klasikinis X.509 viešo rakto sertifikatų centro su saugia infrastruktūra autentifikacijos profilis. Šis autentifikacijos profilis yra valdomas EUGridPMA [19].

Toliau būtina peržiūrėti paminėto profilio reikalavimus sertifikatų centrui, nes visi šiame profilyje paminėti reikalavimai turės būti realizuoti sertifikatų centre.

1.4.1. Bendra architektūra

Šalyje, dideliam regione ar tartautinėje organizacijoje turi būti vienas sertifikatų centras. Pagrindinis tikslas yra aptarnauti kiek įmanoma didesnę bendruomenę su mažu kiekiu patikimų sertifikatų centrų. Norint pasiekti patikimumo, reikia tikėtis, kad kiekvienas CA bus prižiūrimas ilgą laiką kokios nors institucijos ar organizacijos, o ne kad bus sukurtas vienam projektui įgyvendinti.

CA struktūra kiekvienam regione neturi laikytis tradicinio hierarhinio modelio, bet ten turi būti viena galinė esybė išduodanti CA. Kiekvienam CA yra pageidaujama turėti platų registracijos centrų (RA) pripažinimą. RA valdo galinių esybių identifikaciją ir autentifikuoja jų prašymus, kurie perduodami į sertifikatų centrus. CA valdo tikslias užduotis: pasirašymą, sertifikatų išdavimą ir sertifikatų atšaukimo sąrašų tvarkymą [14].

1.4.2. Identiškumas

Bet kuris vienas subjektas atskiru vardu turi būti susietas tik su viena esybe. Per visą gyvavimo ciklą CA neturi būti susieta su bet kuria kita esybe. Tai neprieštarauja pirmiau nustatytam reikalavimui, kad viena esybė gali turėti daugiau nei vieną susietą subjekto vardą, pavyzdžiui, kai raktas naudojamas skirtingiems tikslams. Privatus raktas susietas su bet kuriuo sertifikatu neturi būti atskleistas arba pasidalintas su galinėmis esybėmis, išskyrus su ta, kuri išdavė sertifikatą.

Identiškumo tikrinimo taisyklės

PKI CA turi nustatyti registravimo institucijos (RA) vaidmenį, ir šios registracijos institucijos yra atsakingos už visų galutinių esybių tapatybių tikrinimą, pavyzdžiui, asmenų ir tinklo esybių.

Kad RA galėtų patvirtinti asmens tapatybę, subjektas turi kreiptis į RA gyvai ir turi pateikti nuotrauką ir galiojantį oficialų dokumentą, įrodantį, kad subjektas yra tas kuo skelbiasi, kaip apibrėžta sertifikatų centro CP / CPS dokumente. Tuo atveju, kai prašoma ne asmeninio serifikato, RA turėtų patvirtinti tapatybę ir asmens tinkamumą naudotis saugumo metodais. Prašant sertifikato hostui ir paslaugoms, RA turėtų įsitikinti, kad prašytojas yra tikrai įgaliotas asocijuoto FQDN savininko arba atsakingo administratorius. RA turi patvirtinti sertifikato pasirašymą. CA arba RA turi turėti dokumentus, kurie pagrįstų tapatybę visam laikui. CA yra atsakingas už šių bylų archyvavimą. Visas bendravimas tarp CA ir RA dėl sertifikato išdavimo arba sertifikato statuso pakeitimo turi būti saugus ir kontroliuojamas. CP / CPS turi aprašyti, kaip RA ar CA yra informuojami apie pakeitimus, kurie gali turėti įtakos sertifikato statusui. Visais atvejais sertifikato prašymas pateiktas sertifikatui gauti, turi būti susietas su asmens tapatybės tikrinimo aktais.

Sertifikato galiojimo pabaiga, atnaujinimas ir naujo sertifikato išdavimas

Sertifikatas, kurio privatus raktas yra valdomas programinės įrangos pagrindu turėtų būti iš naujo išduotas, o neatnaujintas. Sertifikatas susisietas su privačiu raktu, kuris patalpintas techniniame tokene (token) gali būti pratęstas iki 5 metų (kai RSA rakto ilgis 2.048 bitai), arba iki 3 metų (kai RSA rakto ilgis yra 1024 bitų). Sertifikatas negali būti atnaujintas ar pakartotinai

išduotas daugiau nei 5 metams be tapatybės ir tinkamumo patikrinimo, ir ši tvarka turi būti aprašyta CP / CPS [14].

Įgaliojimų atėmimas iš institucijos

Akredituota institucija gali būti pašalinama iš akredituotų institucijų sąrašo, jeigu ji nesilaiko autentifikacijos profilio dokumento arba nesilaiko IGTF federacijos dokumento.

1.4.3. Eksploataciniai reikalavimai

CA sistemos turi būti tokioje saugioje aplinkoje, kur priėjimas yra kontroliuojamas ir jį turi tik atitinkamai apmokyti darbuotojai. CA kompiuteris, kuriame vyksta sertifikatų pasirašymas, turi būti specialus. Jis turi vykdyti tik tas paslaugas, kurios būtinos CA pasirašymo operacijoms. CA pasirašymo kompiuteris gali būti:

- On-line: sertifikatų išdavimo kompiuteris yra tiesiogiai ar netiesiogiai sujungtas (laidais, bevieliu ryšiu arba kitomis priemonėmis) su kitu kompiuteriniu prietaisu (tai apima ir periferinę įrangą, kuri pati yra sujungta su įtaisais, kurie nėra sertifikatus išduodančio aparato sudėtinė dalis);
- visiškai off-line: nuolat atjungtas nuo bet kokio tinklo.

CA rakto minimalus ilgis turi būti 2048 bitai ir sertifikatų centro sertifikatas turi galioti ne mažiau kaip du kartus ilgiau nei šis sertifikatų centras išduoda ilgiausios trukmės sertifikatus galiniams vartotojams, ir šis galiojimo laikas negali būti ilgesnis nei 20 metų.

Programine įranga pagrįstas CA privatus raktas turi būti apsaugotas bent 15 simbolių ilgio slaptažodžiu ir jis gali būti žinomas tik atsakingo sertifikavimo institucijos asmens. On-line CA naudojanti HSM turi imtis panašios ar net aukštesnio lygio apsaugos. Šifruoto rakto kopija turi būti laikoma off-line laikmenoje, saugioje vietoje, kur priėjimas yra kontroliuojamas.

On-line CA

Tuo atveju, kai CA kompiuteris yra aprūpintas bent FIPS 140-2 3 lygį palaikančiu techninės įrangos saugumo moduliu arba lygiavertėmis priemonėmis ir CA sistema naudojama FIPS 140-2 3 lygiu, taip saugant CA privatų raktą, tada CA kompiuteris gali būti prijungtas prie itin saugomo / stebimo tinklo, galbūt prieinamo iš interneto. Saugi aplinka turi būti dokumentuota

ir patvirtinta PMA, o šis dokumentas prieinamas PMA. Atitinkama architektūra (išsamiai aprašyta „on-line CA gairių dokumente“) apima:

- autentifikavimo / prašymų priėmimo serveris, tinkamai saugomas ir prijungtas prie viešojo tinklo, ir atskira pasirašymo sistema, prijungta per privatų ryšį, kuri tik vykdo patvirtintus pasirašymo prašymus bei registruoja visus sertifikatų išdavimus (modelis A);
- autentifikavimo / prašymų priemimo serveris, kuris turi HSM aparatūrą yra prijungtas prie tinklo, kuris rūpinasi ateinančiu į CA srautu ir jį aktyviai stebi nuo invazijų ir saugo naudodamas paketus tikrinančią užkardą (modelis B), arba lygiavertis apsaugos lygis turi būti apibrėžtas PMA.

On-line CA architektūra turi vesti išduotų ir atšauktų sertifikatų žurnalą. Žurnalas turi būti apsaugotas nuo suklastojimo.

Sertifikatų politika ir praktinių pareiškimų identifikacija

Kiekvienas CA turi turėti sertifikavimo politiką ir sertifikatų praktinį pareiškimą (CP / CPS dokumentas) ir turi jam priskirti globaliai unikalų identifikatorių (OID). CP / CPS dokumentai turi būti struktūrizuoti pagal tai kaip apibrėžia RFC 3647. Kai CP / CPS yra keičiami, dokumento OID turi būti pakeistas ir esminiai pakeitimai turi būti pranešti akredituotai PMA ir naujus sertifikatus pasirašyti pagal naują CP / CPS galima tik, kai CP / CPS paketimai yra patvirtinti. Visi CP / CPS, pagal kuriuos yra išduoti galiojantys sertifikatai, turi būti prieinami internete.

Sertifikatas ir CRL profilis

Akredituota institucija privalo teikti ir leisti platinti X.509 sertifikatus sertifikatų centrui, kad būtų patvirtinti galutinių vartotojų sertifikatai. Visi sertifikatai, įskaitant visus galutinio vartotojo sertifikatus, turi atitikti šį autentifikavimo profilį bei taip pat turi atitikti GRID sertifikatų profilį, kaip apibrėžta Open Grid Forum GFD.125.

Sertifikatų centras išduoda X.509 sertifikatus galiniams vartotojams, kurie paremti sugeneruota pareiškėjo šifruota informacija, arba paremti šifruota informacija, kuri gali būti laikoma tik saugioje įrangoje.

Galinių vartotojų raktai turi būti bent 1024 bitų ilgio, o sertifikatai maksimaliai gali turėti 1 metų ir 1 mėnesio galiojimo laiką.

Galinio vartotojo sertifikato plėtiniai:

- policyIdentifier turi būti įtrauktas ir jame turi būti nurodytas OID, kuris identifikuotų CP dokumentą, pagal kurį buvo išduotas sertifikatas, ir čia turi būti nurodyti tik OIDs;
- policyIdentifier turi apimti OID šio profilio: 1.2.840.113612.5.2.2.1;
- CRLDistributionPoints turi būti įtraukti ir turi būti įrašytas bent viens HTTP URL;
- OCSP URI gali būti įtrauktas į AuthorityInfoAccess plėtinį tik tada, jei OCSP atsakiklis veikia kaip paskirta paslauga, arba tai nustatyta išduodačio sertifikatus CA;

Jei CommonName komponentas naudojamas kaip DN dalis, jame turėtų būti nurodytas tinkslus sertifikatą naudosinčios institucijos ar asmens vardas.

Sertifiaktų centras privalo skelbti CRL, ir šie CRL turėtų būti suderinti su RFC5280.

Atšaukimas

CA turi skelbti CRL. CA turi reaguoti kaip įmanoma greičiau, tačiau bent per vieną darbo dieną, kai gaunamas sertifikato atšaukimo prašymas. Nustačius jo nebegaliojimą CRL turi būti atnaujinamas nedelsiant. CA vartotojams išduotų sertifikatų, didžiausias CRL gyvavimo laikas turi būti ne daugiau kaip 30 dienų. CA turi išduoti naują CRL skirtą off-line CA‘s ne mažiau kaip prieš 7 dienas iki to laiko, kuris nurodytas nextUpdate lauke. Bent prieš 3 dienas iki to laiko, kuris nurodytas nextUpdate lauke, kai on-line CA automatiškai išduoda CRL, ir iš karto, kai būna įvykdytas atšaukimas. CRL turi būti paskelbtas saugykloje ir turi būti prieinamas internetu kuo greičiau. Atšaukimo prašymas gali būti pateiktas galinių vartotojų, RA ir CA. Tokie prašymai turi būti tinkamai autentifikuoti. Kiti gali prašyti panaikinti sertifikatą, jei jie gali įrodyti, kad privatus raktas buvo kompromituotas.

CA rakto keitimas

Kai CA kriptografiniai duomenys turi būti pakeisti, toks perėjimas turi būti valdomas. Nuo to laiko, kai nauji kriptografiniai duomenys gaunami, tik naujas raktas turi būti naudojamas sertifikatų pasirašymui. Seno ir naujo rakto galiojimas turi sutapti mažiausiai tiek laiko, koks gali būti ilgiausias vartotojo sertifikato galiojimo laikas. Senas, tačiau vis dar galiojantis sertifikatas,

turi būti prieinamas, kad būtų galima patikrinti senus sertifikatus – ir slaptus raktus, pasirašant CRL – kol baigsis visų sertifikatų, pasirašytų naudojant senąjį raktą, galiojimo laikas [14].

1.4.4. Svetainės apsauga, publikavimas ir saugyklos atsakomybė

Šifruoto privataus rakto slaptažodis turi būti saugomas offline terpėje, atskirtas nuo šifruoto rakto ir saugomas saugioje vietoje, kur tik įgaliotas sertifikavimo institucijos personalas turi prieigą.

Kiekviena institucija savo vartotojams turi skelbti atitinkamą informaciją:

- CA sertifikatą arba visus CA sertifikatus;
- HTTP ar HTTPS URL CA sertifikatą PEM formatu;
- HTTP URL suformatuotą CRL PEM ar DER formatu;
- HTTP ar HTTPS URL tinklalapio, kur pateikta bendra CA informacija;
- CP ir / arba CPS dokumentus;
- Oficialų elektroninio pašto adresą, kur būtų galima pateikti klausimus ar pranešti apie gedimus;
- Fizinį arba pašto adresą.

CA turi užtikrinti pasitikėjimo vientisumą. Be to, CA turi pateikti savo patikimumo tvirtinimą, kad būtų tikima saugykla, nurodant akreditavimo PMA, taikant metodą, nurodytą patikimos saugyklos politikoje. Saugykla turi būti vykdoma bent geriausių pastangų principu, t. y. turi būti nuolat prieinama. Pagrindinė institucija turi suteikti PMA ir federacijai – pagal jos akreditavimą – neribotas teises platinti šią informaciją.

1.4.5. Auditas

CA turi įrašinėti ir archyvuoti visus sertifikatų prašymus, kartu ir visus išduotus sertifikatus, visus panaikinimų prašymus, visus išduotus CRL ir sertifikatus išduodančio kompiuterio prisijungimus, atsijungimus, perkrovimus.

CA turi saugoti šiuos duomenis mažiausiai trejus metus, o tapatybės patvirtinimo įrašai turi būti saugomi bent tol, kol yra galiojančių sertifikatų susijusių su šiais duomenimis. Šie įrašai turi būti pateikiami išorės auditoriams. Kiekvienas CA turi sutikti būti patikrintas kitos akredituotos

CA, kai siekiama patikrinti ar CA laikosi atitinkamų taisyklių ir procedūrų, nurodytų jos CP / CPS dokumentuose.

CA turi atlikti CA ir RA personalo veiklos auditą bent kartą per metus. CA ir RA personalo sąrašas turėtų būti išlaikomas ir patikrinamas bent vieną kartą per metus [14].

1.4.6. Atstatymas po kompromitacijos ir nelaimių

CA turi turėti tinkamas atkūrimo procedūras po kompromitacijų bei nelaimių ir turi būti pasirengęs aptarti šią procedūrą PMA. Procedūros neturi būti atskleidžiamos politikoje ir praktinėse ataskaitose.

Kruopštus abonentas

CA turėtų dėti reikiamas pastangas, siekdama užtikrinti, kad galutiniai vartotojai suprastų savo privačių duomenų saugojimo svarbą. Kai privačiam raktui saugoti naudojama programinė įranga, vartotojas privalo apsaugoti savo privatų raktą stipriu slaptažodžiu, t.y. bent 12 simbolių ilgio ir remtis dabartine pažangiausia praktika, pasirenkant stiprius slaptažodžius. Privačius raktus, susijusius su hostu ir paslaugos sertifikatu, galima laikyti be slaptažodžio, bet apsauga turi būti užtikrinta kitais adekvačiais sisteminiais apsaugos metodais.

Vartotojai turi prašyti panaikinti sertifikatą kaip galima greičiau, t.y. per vieną darbo dieną po to, kai pastebimas privataus rakto praradimas arba sukompromitavimas, arba, jei sertifikato duomenys nebegalioja.

Apibendrinant galima teigti, kad bendra sertifikatų centro apsauga priklauso nuo visų paminėtų punktų. Kadangi norima realizuoti on-line CA, tai reikia atkreipti dėmesį į labai aukštus saugumo reikalavimus. Anksčiau buvo minėti du on-line CA realizavimo architektūros modeliai, kur pagal modelį A prašymų priėmimo serveris ir pasirašymo sistema turi būti atskirtos fiziškai. O pagal modelį B teigiama, kad šios dvi funkcijos gali būti atliekamos viename serveryje, tačiau šis serveris turi naudoti HSM aparatūra. Taigi siekiant padidinti CA saugumą šiuos du modelius galėtų apjungti į vieną ir be to, kad išskirti atliekamas funkcijas į du serverius, privačiam CA raktui saugoti dar reikėtų naudoti HSM. Tokiu atveju nulaužus CA serverį ir pavogus HSM‘o PIN‘ą, nebūtų galima pavogti CA sertifikato rakto (HSM'e būtų saugomas CA privatus raktas).

Pirmo modelio realizavimui galima naudoti OpenCA atviro kodo paketą. Be to, OpenCA paketas atitinka visus ankčiau aprašytus reikalavimus sertifikatų centrui, kaip CRL, visų veiksmų registravimą, priimtus ir atšauktus sertifikatų prašymus ir t. t. [21]. Taigi naudojant šį paketą pirmo modelio realizacijai, off-line dalyje dar reikėtų patalpinti CA privatų raktą į HSM bei sinchronizuoti visą veikimą.

1.5. Neišspręstų problemų formulavimas

Aliekant GSI analizę buvo paminėta, jog svarbiausia GSI autentifikacijos koncepcija yra sertifikatai. Šiais elektroniniais dokumentais kiekvienas vartotojas ir paslauga yra identifikuojami GRID tinkle. Vadinasi, sertifikatų išduoti reikia labai daug ir kiekvienas GRID tinklas norėdamas efektyviai veikti turi turėti savo sertifikatų centrą. Sertifikatų centro sertifikatas sertifikatų pasitikėjimo grandinėje yra pats svarbiausias ir tokio sertifikato sukompromitavimas reikštų visų pasitikėjimų nutrūkimą, todėl sertifikatų centro apsaugai turi būti skiriamas pats didžiausias dėmesys. Dėl to iškyla problema kaip maksimaliai apsaugoti sertifikatų centro privatų raktą nuo bet kokios galimybės jį sukompromituoti.

Aptariant GRID sistemas pastebima, kad proxy sertifikatai atlieka labai didelę reikšmę GRID saugumui užtikrinti, jie naudojami vieno prisijungimo galimybei realizuoti, teisių perdavimui. Tačiau iki šiol egzistuoja problema susijusi su teisių perdavimų sekimu. Kai kuriais atvejais būtų naudinga žinoti kas dalyvavo teisių perdavimo grandinėje (pavyzdžiui, sertifikatą priimanti šalis gali norėti atmesti proxy sertifikatą, kuris buvo kurtas atitinkamame domene). Arba įmanoma, kad proxy sertifikatas buvo pavogtas ir iš jo buvo sugeneruota dar daug kitų proxy sertifikatų, kuriais yra naudojamasi nelegaliai. Tai būtų galima spręsti praplečiant proxy sertifikatą ir į jį įrašant šio sertifikato gavėjo identifikatorių (IP adresą, domeno informaciją ar kita) [34].

Kita problema susijusi su proxy sertifikatų naudojimu yra ta, kad sukompromitavus proxy sertifikatą taip pat nėra galimybės tokių sertifikatų atšaukti, todėl pavogus šią informaciją nelabai ką galima padaryti. Tai yra guodžiamasi tuo, kad proxy sertifikatai yra riboto galiojimo ir, kad tai gali padaryti gana ribotą žalą [2].

Suformulavus esamas problemas GRID sistemose, tolimesnis žingsnis ką reikia atlikti, tai ištirti kaip jos sprendžiamos įvairiose sistemose ir kokius sprendimo būdus siūlo kiti autoriai.

1.6. Išvados

1. Atlikus GSI analizę galima išskirti pagrindinius saugą užtikrinančius elementus: sertifikatų centrą ir proxy sertifikatus, nes jų pagalba yra užtikrinamas saugus darbas GRID tinkle.
2. Analizuojant sertifikatų centą išskiriama, kad sertifikatų centro sertifikatas sertifikatų pasitikėjimo grandinėje yra pats svarbiausias ir tokio sertifikato sukompromitavimas reikštų visų pasitikėjimų nutrūkimą, todėl sertifikatų centro apsaugai turi būti skiriamas pats didžiausias dėmesys.
3. Aptariant GRID sistemas pastebima, kad proxy sertifikatai atlieka labai didelę reikšmę GRID saugumui užtikrinti, jie naudojami vieno prisijungimo galimybei realizuoti, teisių perdavimui. Tačiau iki šiol egzistuoja problema susijusi su teisių perdavimų sekimu, jų atšaukimu bei kontroliavimu.

2. SUFORMULUOTŲ PROBLEMŲ SPRENDIMŲ ANALIZĖ

Atlikus GRID saugumo infrastruktūros analizę buvo suformuluotos dvi problemos, viena susijusi su sertifikatų centro apsauga, tiksliau privataus rakto apsauga, bei kita su proxy sertifikatų valdymu. Radus ir apibrėžus problemas tolimesnis žingsnis, kuris turi būti atliktas, tai išanalizuoti programinius paketus ir kitų autorių siūlomus šių problemų sprendimo būdus.

2.1. Sertifikatų centro problemos sprendimas

Ankstesnėje darbo dalyje jau buvo aptarti galimi sertifikatų realizavimo modeliai bei sertifikatų centro svarba ir apibrėžta tai, kad sertifikatų centro privataus rakto apsaugai turi būti skiriama didžiausia įmanoma apsauga. Kalbant toliau apie galimus sertifikatų centrų modelius norėtusi aptarti vieną mokslinį straipsnį „A CA-Based Security e-Government System“, kuriame pateikiamas dar vienas sertifikatų centro modelis [6]. Šiame straipsnyje yra kalbama apie elektroninės valdžios saugumo sistemą bei sertifikatų dalinimą jos institucijoms, kas reikalauja ypač didelės apsaugos. Kaip ir elekroninėje valdžioje, taip ir GRID tinkle sertifikatai yra išduodami daugeliui esybių (tinklo įrenginiams, paslaugoms, vartotojams). Straipsnyje aptariami trys sertifikatų centro lygmenys:

1) CA centras, kuriame yra vykdomas sertifikatų pasirašymas.
2) RA (registracijų centras), kuriame yra pateikiamos vartotojų užklausos.
3) Asmenys, departamentų varotojai, kurie prašo sertifikatų.

Be to, papildomai dar yra aptariamas pirmo lygio saugumas ir minima, jog CA privatus raktas tuti būti apsaugotas užšifruojant jį techninę įranga, kitaip sakant naudojant kriptografinę intelektualiąją kortelę [6].

Toliau nagrinėjant šią problemą reikia apžvelgti programinius paketus leidžiančius realizuoti sertifikatų centrą ir išanalizuoti, kokias saugumo priemones jie naudoja.

GRID sistemoms diegti yra naudojamos įvairios tarpinės programinės įrangos, vienos iš populiariausių yra Globus toolkit bei gLite. Šios abi tarpinės programinės įrangos savyje turi įrankius leidžiančius realizuoti sertifikatų centrus. Be šių GRID tarpinėse programinėse įrangose naudojamų paketų taip pat yra keletas specializuotų sertifikatų centro realizavimui naudojamų paketų. Kiekvienas iš jų truputį toliau bus aptartas plačiau.

Globus toolkit SimpleCA paketas

Globus toolkit programinėje įrangoje yra SimpleCA paketas leidžiantis įdiegti paprastą sertifikatų centrą, kuris leidžia išduoti X.509 sertifikatus Globus Toolkit vartotojams ir paslaugoms. SimpleCA yra lengvai naudojama programa, kurioje yra tipinis sistemos administratorius, kuris gali kurti sertifikatus naudodamasis nesunkiu interfeisu. Ši programa turi keletą funkcijų ir pasirinkimų, nei daugelis kitų CA paketų. Ji leidžia keisti visus pasirinkimus komandine eilute arba konfiguraciniame faile, be to, programa yra su pradiniu konfiguraciniu failu, kuris turi praktiškus pradinius nustatymus, taigi sistemos administratorius gali lengvai išduoti sertifikatus be didelių žinių. Išsamiau analizuojant šį paketą, jo diegimą ir suteikiamas saugumo galimybes, pasigendama to, kad nėra sakoma apie galimybę privatų raktą saugoti intelektualioje kortelėje, minima tik privataus rakto apsauga slaptažodžiu, todėl galima teigti, kad šis paketas nėra vienas iš geriausių pasirinkimų sertifikatų centro realizacijai [16].

gLite tarpinės programinės įrangos paketo MyProxy įrankis

Sekantis populiarus GRID tarpinės programinės įrangos paketas yra gLite. Šiame pakete yra galimybė naudoti MyProxy paslaugą [13], o ši paslauga be įgaliojimų saugojimo taip pat gali veikti kaip serifikatų centras. Šis paketas atitinka certifikatų centrams keliamus reikalavimus bei yra akredituotas NCSA MyProxy CA ir NERSC Online CA ir gali išduoti trumpalaikius sertifikatus. Be viso to, šis paketas leidžia privatų raktą apsaugoti naudojant HSM [33], todėl žiūrint iš saugumo pusės tai yra geresnis pasirinkimas nei jau aptartas SimpleCA paketas.

EJBCA ir OpenCA programiniai paketai

EJBCA ir OpenCA programiniai paketai yra atviro kodo programinė įranga skirta sertifikatų centro realizacijai [3]. Iš esmės jie vienas nuo kito labai nesiskiria, abu jie leidžia privatų raktą saugoti HSM įrenginyje bei suteikia galimybę atskirti sertifikatų prašymų priėmimo serverį ir pasirašymų serverį fiziškai. Taip pat šie paketai suteikia daug kitų galimybių jau aptartų analizuojant minimalius reikalavimus sertifikatų centrui bei leidžia išduoti ilgalaikius sertifikatus, skirtingai nei prieš tai aptarti programiniai paketai. Šie paketai skiriasi tik tuo, kad jie realizuoti naudojant skirtingas technologijas [11, 22].

2.2. Proxy sertifikatų problemos sprendimo būdai

Proxy sertifikatai GRID tinkle yra naudojami teisių perdavimui, vieno prisijungimo galimybei realizuoti, tačiau nors šie sertifikatai ir yra trumpalaikiai, tačiau jų turėtojai įgauna visas šį sertifikatą išdavusios esybės suteiktas teises. Kaip jau buvo apibrėžta formuluojant problemą, šio sertifikato naudojimui reikia naujų mechanizmų ar priemonių, jeigu norima sumažinti grėsmę pasinaudoti jais nelegaliai. Toliau bus nagrinėjami keli siūlomi sprendimo būdai.

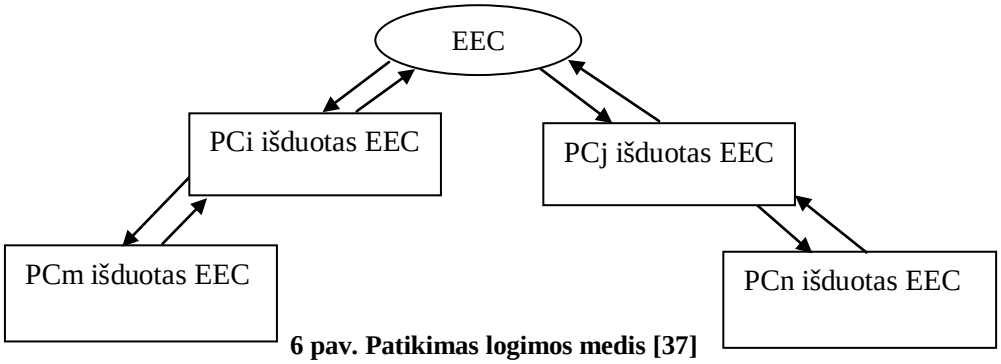
Pirmasis iš būdų būtų naudoti patikimų proxy sertifikatų sąrašą (Proxy Certificate Trust List). Šis sprendimas yra paremtas viešo rakto infrastruktūra, kai sukuriamas papildomas ir nepriklausomas modulis trečioje patikimoje šalyje vadinamoje sertifikatų registrų centru (CRA) [37].

Pagrindinės CRA funkcijos yra:

1) Valdyti proxy sertifikatais perduodamų įgaliojimų ryšius.
2) Priiminėti ir dalintis informacija apie proxy sertifikatus.
3) Generuoti proxy sertifikatų patikimus sąrašus.
4) Atšaukti sukompromituotus ir pasibaigusius proxy sertifikatus.

Veikimo algoritmas

Kai yra išduodamas kiekvienas proxy sertfikatas, jo išdavėjas užregistruoja naują proxy sertifikatą į CRA. CRA gavęs tokį pranešimą patvirtina, kad proxy sertifikatas yra išduotas. Kai įvykdomas patvirtinimas CRA sukuria naujo proxy sertifikato šaką vadinamame patikimame logikos medyje (TrustLogicTree). Kai naudojantis proxy sertifikatu išduodamas sekantis proxy sertifikatas, CRA įrašą apie naujai sukurtą sertifikatą talpina toje šakoje žemiau. 6 paveiksle pateikiamas šis algoritmas, kur EEC – galinė esybė, o PC – proxy sertifikatas.



6 pav. Patikimas logimos medis [37]

Remiantis 6 paveiksle pateiktu logikos medžiu yra generuojamas patikimų proxy sertifikatų sąrašas apie kuriuos yra pateikiama visa informacija, t. y. ar sertifikatas nėra atšauktas ir panašiai. Šis sąrašas yra naudojamas tikrinant proxy sertifikatų galiojimą ir jei šiame sąraše yra pažymėta, jog toks sertifikatas negalioja, tai jis yra atmetamas. O kai yra sukompromituojamas kuris nors proxy sertifikato privatus raktas, į CRA yra pranešama apie tai ir visų proxy sertifikatų esančių atitinkamoje logikos medžio šakoje statusas pakeičiamas į negaliojančius. O ši informacija atsispindi į PCTL, taigi taip įvykdomas proxy sertifikatų atšaukimas, užkertant tokių sertfikatų naudojimusi po sukompromitavimo [37].

Antrasis siūlomas sprendimo būdas taip pat siūlo mechanizmą, kaip būtų galima atšaukti proxy sertifikatus, tik ši idėja remiasi MyProxy (įgaliojimų saugojimo saugykla). Yra siūloma į MyProxy paslaugą įdėti sluoksnį, kuris palaikytų proxy sertifikatų atšaukimą. Ši sistema leistų GRID administratoriams atšaukti proxy sertifikatus. Kai sertifikatų centras atšauks EEC (galinio vartotojo sertifikatą išduotą sertifikatų centro), proxy sertifikatai taip pat būtų automatiškai atšaukti. Pagrindinė idėja paremta tuo, kad MyProxy serveris kurdamas proxy sertifikatą įtrauktų 160 bitų hash reikšmę nurodančią sertifikato atšaukimo informaciją. Ši hash reikšmė gaunama taip: tarkime sertifikato galiojimo laikas yra N laiko vienetų, tada MPS atsitiktinai sugeneruoja dvi 20 bitų reikšmes X0 ir Y0 ir užšifruoja jas vienakrypte hash funkcija H. Nuosekli X0 hash reišmė yra: X1 = H(X0), X2 = H(X1), … XN = H(XN-1); ir Y1 = H(Y0). Tada MPS sukuria proxy sertifikatą, į kurį patalpina informaciją apie sertifikato savininką, viešą proxy sertifikato raktą, sertifikato serijos numerį, galiojimo laikus, Y1 ir XN reišmes ir visa tai pasirašo privačiu raktu. Be viso to, šita informacija yra patalpinama į DS (direktyvinius serverius). Pats tokio proxy sertifikato tikrinimas remiasi tuo, kokia reikšmė yra gaunama iš DS, jei DS atsiunčia X(N-i), tai vadinasi sertifikatas yra aktyvus, o jei atsiunčiama Y0, vadinasi sertifikatas atšauktas. Minėtas „i“ reiškia bet kokį laiko vienetą [38].

Analizuojant egzistuojančiose sistemose ir moksliniuose straipsniuose pateikiamus problemų sprendimo būdus, galima surasti keletą idėjų, kurios padeda sukurti savus šių problemų sprendimo modelius.

Kalbant apie sertifikatų centro modelį, kuris siūlomas e-valdžios sistemoms, buvo apibrėžta ypač didelė šios sistemos apsauga, o tai siūloma daryti atskiriant visas pagrindines sertifikatų centro dalis fiziškai. Toks fizinis atskyrimas galimas ir modeliuojant sertifikatų centro modelį GRID sistemai.

Atliekant antrosios problemos siūlomų sprendimų analizę pastebima tai, kad pagrindinis dėmesys skiriamas mechanizmams, kurie leistų atšaukti proxy sertifikatus. Tačiau pasigesta

sprendimo būdo leidžiančio grieščiau tokius sertifikatus dalinti, nes jei šie sertifikatai būtų išduodami atsakingiau, tai sumažėtų poreikis ir juos atšaukti. Viename iš straipsnių siūloma į proxy sertifikatus talpinti papildomas hash reikšmes, kurių pagalba būtų galima atšaukti sertifikatus. Tačiau hash reikšmių talpinimas galėtų būti naudojamas ir kitokiai paskirčiai, tai yra įgaliojimus gaunančios esybės identifikacijai, pavyzdžiui, paskaičiuojant tos esybės sertifikato hash reikšmę ir ją patalpinus į proxy sertifikatą.

Taigi išanalizavus GRID sistemų saugumą užtikrinačius komponentus, suformulavus esamas problemas bei atlikus šių problemų sprendimo budus, tolimesnė užduotis yra suprojektuoti esamų problemų sprendimo modelius, remiantis iki šiol padarytomis išvadomis bei iškeltomis idėjomis.

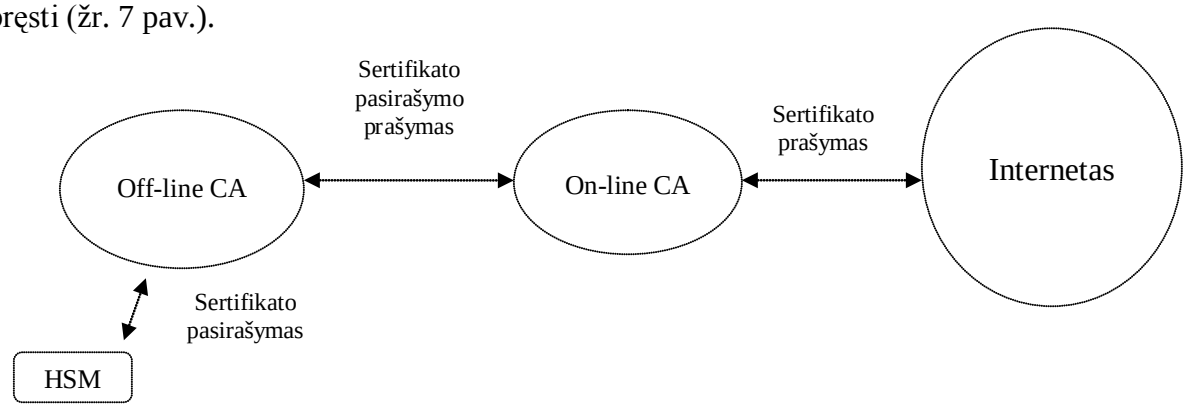
2.3. Išvados

1. Atlikus sertifikatų centro problemos sprendimo būdų analizę, išskirtas sekantis aspektas, kad norint padidinti sertifikatų centro apsaugą, sertifikatų pasirašymo funkciją reikia atskirti fiziškai nuo CA off-line dalies, o tai leis apsaugoti sertifikatų centro privatų raktą nuo sukompromitavimo.
2. Atlikus proxy sertifikatų problemos sprendimo būdų analizę, pasigesta sprendimo būdo leidžiančio grieščiau tokius sertifikatus dalinti. Todėl reikalingas naujas sprendimas leidžiantis proxy sertifikatus išduoti konkrečioms esybėms, taip panaikinant galimybę jais pasinaudoti nelegaliai. Tai galima atlikti į proxy sertifikatą patalpinant papildomą jo gavėjo sertifikato hash reikšmę.

3. PROBLEMŲ SPRENDIMO MODELIŲ PROJEKTAVIMAS

3.1. Sertifikatų centro modelio projektavimas

Analizės dalyje buvo minėta, kad GRID saugumo federacijos (Grid Trust Federation) tradicinio X.509 PKI sertifikatų centro minimaliuose reikalavimuose pateikiami du sertifikatų centrų realizacijos metodai, tai yra siūloma sertifikatų pasirašymo funkciją fiziškai atskirti tinkle arba tai daryti naudojant HSM. Kaip jau kalbėjome, sertifikatų centro saugai turi būti skiriamas didžiausias dėmesys, todėl siekiant maksimalios privataus rakto apsaugos, kuris naudojamas pasirašant sertifikatus, būtų protinga naudoti ir fizinį atskyrimą tinkle ir dar naudoti HSM. Šitoks sprendimas taip pat buvo aptartas nagrinėjant e-valdžios sertifikatų sistemos modelį. Taigi remiantis šiais teiginiais galima suprojektuoti siūlomą sertifikatų centro modelį šiai problemai spręsti (žr. 7 pav.).



7 pav. Suprojektuoto sertifikatų centro modelis

Septintame paveiksle Off-line CA dalis ir On-line CA dalis turėtų būti įdiegtos fiziškai atskirtose mašinose. Off-line CA dalis tiesioginio ryšio su viešaisiais tinklais neturi turėti. Ši dalis turėtų vykdyti sertifikatų pasirašymo prašymų priėmimą, kai to prašoma iš On-line CA dalies, bei kaupti informaciją apie pasirašytus sertifikatus duomenų bazėje, generuoti CRL (atšauktų sertifikatų sąrašus). Kadangi privatus raktas yra svarbiausias sertifikatų centro elementas, tai jis turi būti patalpintas į HSM, kas užtikrintų papildomą apsaugą norint juo pasinaudoti. Iš HSM privatus raktas paimtas negali būti, jis gali būti nebent ištrintas. Vadinasi, privataus rakto pasisavinimas yra neįmanomas. Gavus prašymą pasirašyti sertifikato prašymo užklausą ir pasirašyti jį privačiu raktu Off-line dalis turės kreiptis papildomai į HSM, kur bus įvykdomas pasirašymas. Visi sertifikatų išdavimo prašymai savaime suprantama ateis iš viešojo tinklo, asmenų norinčių gauti sertifikatus, kurie užtikrintų jų autentiškumą elektroninėje erdvėje.

3.2. Proxy sertifikatų valdymo mechanizmų projektavimas

Kaip jau buvo aptarta pirmame skyriuje, didžiausią vaidmenį GRID tinkluose atlieka proxy sertifikatai. Jie yra naudojami perduodant vartotojų įgaliojimus kitoms paslaugoms ar vartotojams. Kalbant apie teisių perdavimą, galima dar aptarti vieną gyvenimišką atvejį. Realiame gyvenime įgaliojimai niekada nėra suteikiami neįvardintiems asmenims, nes kitu atveju tokiu įgaliojimu galėtų pasinaudoti bet kas pasisavinęs šį dokumentą. Taigi, kadangi įgaliojimai yra suteikiami konkrečiam asmeniui tai šiais įgaliojimais galima pasinaudoti tik tada, kai įgaliojimą pateikiantis asmuo įrodo savo tapatybę, o ne vien parodo, kad turi kito asmens gautą įgaliojimą. Grįžtant prie proxy sertifikatų, pastebėta, kad kai jie išduodami kitai organizacijai, pačiame įgaliojime (proxy sertifikate) nėra nurodoma kam jis yra išduotas ir kas juo gali naudotis. Šiuo metu patvirtinimas yra grindžiamas tik tuo, kad šalis turi proxy sertifikato privatų raktą, bet visada yra galimybė, kad jis gali būti pasisavintas. O tokiu atveju visomis vartotojo teisėmis gali pasinaudoti proxy sertifikato privatų raktą pavogęs asmuo, nes proxy sertifikato privatus raktas nėra papildomai užšifruotas slaptažodžiu ir jį jis gali lengvai naudoti autentifikuojantis su kitomis sistemomis, taip apsimesdamas kitu vartotoju. Apie tai, taip pat buvo kalbėta pirmame skyriuje, kur buvo aiškiai apibrėžta, jog proxy sertifikato privatus raktas yra saugus tik tiek, kiek jį saugo failų sistemos teisės. Taigi iš to seka išvada, kad norint pašalinti grėsmę proxy sertifikatu pasinaudoti nelegaliai, būtina jį išduodant nurodyti ir kam jis yra išduodamas. Kaip žinoma, GRID paslaugos turi savo sertifikatus (juos yra išdavusios sertifikatų sistemos), kuriuos naudoja autentifikuojantis su kitomis paslaugomis ar vartotojais, taip jos gali identifikuoti save ir jais patvirtinti savo tapatybę. Iš to seka, kad proxy sertifikate galėtume patalpinti patikimo sertifikato žymę tos paslaugos, kuriai perduodami įgaliojimai. O ta žymė galėtų būti tos paslaugos turimo sertifikato vienkrypte funkcija paskaičiuota reikšmė, pavyzdžiui, naudojant MD5. Apie tokios informacijos talpinimą į sertifikatus buvo kalbama analizuojant problemų sprendimo būdus (žr. 2.2. poskyryje).

Toliau norint nuspręsti kur tokią žymę būtų galima patalpinti, buvo padarytas paprastas eksperimentas. Šio eksperimento metu buvo siunčiama tokia užduotis į GRID tinklą (BalticGrid), kuri rezultatu gražintų visus sertifikatus, kurie buvo kuriami užduočiai keliaujant GRID elementais iki darbinio mazgo. Užduočiai pasiųsti buvo naudojamas asmeninis Kęstučio Pauliko BalticGrid sertifikatas, todėl pateikiant proxy sertifikatų subjekto ir išdavėjo laukus bus matomas šio asmens vardas ir pavardė.

Pirmiausia pateiksime vieną proxy sertifikatą, kuris buvo generuotas tarp GRID vartotojo sąsajos ir resursų brokerio. Jo pagalba nuspręsime kur būtų galima talpinti papildomą proxy sertifikato gavėjo identifikatorių. GRID resursų brokeriui išduotas proxy sertifikatas pateiktas 8 paveiksle.

```
     Certificate:
         Data:
             Version: 4 (0x3)
             Serial Number: 2796 (0xaec)
             Signature Algorithm: md5WithRSAEncryption
             Issuer: DC=org, DC=balticgrid, OU=ktu.lt, CN=Kestutis Paulikas,
CN=proxy
             Validity
                 Not Before: Apr 29 08:20:57 2010 GMT
                 Not After : May  3 08:08:57 2010 GMT
             Subject: DC=org, DC=balticgrid, OU=ktu.lt, CN=Kestutis Paulikas,
CN=proxy, CN=proxy
             Subject Public Key Info:
                 Public Key Algorithm: rsaEncryption
                 RSA Public Key: (512 bit)
                     Modulus (512 bit):
                         00:da:c8:77:36:9a:b9:22:9e:82:2e:a5:dc:8a:f9:
                         85:9a:cd:f1:a1:3a:71:79:a9:dd:03:d5:37:9b:2c:
                         1e:bb:08:ce:8d:ef:61:77:8b:87:b2:67:40:e3:ed:
                         21:c6:49:f0:bf:15:fe:05:46:0c:f4:80:e4:83:c9:
                         94:27:ee:3c:21
                     Exponent: 65537 (0x10001)
         Signature Algorithm: md5WithRSAEncryption
             ad:97:fe:50:53:83:e4:a4:01:e6:a5:45:81:cd:53:d2:d6:2f:
             27:06:8d:70:bb:71:08:54:49:11:c5:38:ed:d6:a8:35:d0:99:
             c4:e9:a4:01:f9:d7:df:2c:ae:2e:05:4e:ba:34:c4:a8:a5:99:
             a6:f8:73:7b:21:d7:d4:75:a4:9c
```

8 pav. Resursų brokeriui išduotas proxy sertifikatas

Pagal šį paveikslą matome, kad „*Subject*“ laukas yra geriausiai tinkamas, kuriame galima patalpinti papildomą informaciją apie šio sertifikato naudotoją (paslaugą ar vartotoją, kuriam yra suteikiami įgaliojimai). Šiuo metu proxy sertifikatas GRID sistemose atpažįstamas iš šiame lauke prirašomos „CN=proxy“ reikšmės. Be to, šiame paveiksle dar matoma kas šį sertifikatą išdavė, tai yra, kad šis sertifikatas buvo pasirašytas kito proxy sertifikato (DC=org, DC=balticgrid, OU=ktu.lt, CN=Kestutis Paulikas, CN=proxy). Tarkime, resursų brokerio sertifikato MD5 reikšmė lygi 2e09c6e379158da805f6cb1522810509, tai mūsų aptartu atveju į proxy sertifikato „*Subject*“ lauką reikėtų vietoj „CN=proxy“ talpinti reikšmę „CN=proxy 2e09c6e379158da805f6cb1522810509“. Taigi autentifikuojantis su tokiu proxy sertifikatu būtų galima resursų brokerio papildomai reikalauti pateikti savo tikrą sertifikatą, o gavus jį, paskaičiuoti MD5 reikšmę ir ją palyginti su reikšme įrašyta proxy sertifikate. Ir jei ji sutaptų, dar

reikėtų prašyti, kad ši paslauga taip pat patvirtintų, jog ji yra šio sertifikato savininkė, t. y. reikėtų atlikti dar vieną autentifikacijos procesą. Proxy sertifikato valdymo mechanizmą detaliau pateiksime vėliau, o dabar reikėtų grįžti prie proxy sertifikato projektuojamo naujo formato. Kaip jau minėjome, į GRID paleistos užduoties gauti rezultatai grąžino visą eilę sertifikatų, kurie buvo generuoti užduočiai keliaujant GRID elementais iki darbinio mazgo. Žemiau pateikiama šių sertifikatų visa grandinė, rodanti tik sertifikato išdavėjo ir subjekto laukus:

1. Skaičiuojamojo elemento proxy sertifikatas, kurį pasirašė resursų brokeris.
   issuer= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy/CN=proxy
   subject=/DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy/CN=proxy/CN=limited proxy
2. Resursų brokerio proxy sertifikatas, kurį pasirašė vartotojo proxy sertifikatas.
   issuer= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy
   subject= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy/CN=proxy
3. Vartotojo proxy sertifikatas, kurį pasirašė pats vartotojas.
   issuer= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas
   subject= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy
4. Vartotojo sertifikatas, kurį išdavė sertifikatų centras.
   issuer= /DC=org/DC=balticgrid/CN=Baltic Grid Certification Authority
   subject= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas
5. Sertifikatų centro sertifikatas, kuris yra paties sertifikatų centro pasirašytas.
   issuer= /DC=org/DC=balticgrid/CN=Baltic Grid Certification Authority
   subject= /DC=org/DC=balticgrid/CN=Baltic Grid Certification Authority

Kaip matome iš šios grandinės pirmasis proxy sertifikatas yra generuojamas su savimi pačiu, kurį pasirašo pats vartotojas. Šis proxy sertifikatas dažniausiai skiriasi nuo kitų, nes jame patalpinamos VO (virtalios organizacijos) vartotojui suteiktos teisės GRID tinkle, o vartotojas ar kitos paslaugos generuodamos proxy sertifikatą dažniausiai papildomų praplėtimų netalpina, t. y. nededa jokių apribojimų. Kaip jau minėjome pirmame skyriuje šie apribojimai – tai galimas ribojimas proxy sertifikatu naudotis visomis vartotojui VO suteiktomis teisėmis, suteikiant teisę naudotis tik dalimi iš jų. Tarkime vartotojo sertifikato MD5 reikšmė yra 5a77f2eeede2dc2a8a5be1bcbabad9c3, resursų brokerio sertifikato MD5 reikšmė yra 2e09c6e379158da805f6cb1522810509 ir skaičiuojamojo elemento sertifikato MD5 reikšmė yra 6c7915014d81757726d8c5b6bb35c90f, tada pagal aptartą naują sertifikato formatą sertifikatų grandinė turėtų būti tokia:

1. Skaičiuojamojo elemento proxy sertifikatas.
   issuer= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy 5a77f2eeede2dc2a8a5be1bcbabad9c3/CN=proxy 2e09c6e379158da805f6cb1522810509
   subject=/DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy 5a77f2eeede2dc2a8a5be1bcbabad9c3/CN=proxy 2e09c6e379158da805f6cb1522810509/CN=limited proxy 6c7915014d81757726d8c5b6bb35c90f
2. Resursų brokerio proxy sertifikatas.
   issuer= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy 5a77f2eeede2dc2a8a5be1bcbabad9c3
   subject= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy 5a77f2eeede2dc2a8a5be1bcbabad9c3/CN=proxy 2e09c6e379158da805f6cb1522810509
3. Vartotojo proxy sertifikatas.
   issuer= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas
   subject= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas/CN=proxy 5a77f2eeede2dc2a8a5be1bcbabad9c3
4. Vartotojo sertifikatas.
   issuer= /DC=org/DC=balticgrid/CN=Baltic Grid Certification Authority
   subject= /DC=org/DC=balticgrid/OU=ktu.lt/CN=Kestutis Paulikas
5. Sertifikatų centro sertifikatas.
   issuer= /DC=org/DC=balticgrid/CN=Baltic Grid Certification Authority
   subject= /DC=org/DC=balticgrid/CN=Baltic Grid Certification Authority

Pagal pateiktą sertifikatų grandinę pritaikius naują proxy sertifikatų formatą, matome, kad MD5 reikšmės pateks ne tik į „*Subject*“ lauką, bet ir į „*Issuer*“, nes „*Issuer*“ laukas sekančiame proxy sertifikate yra toks, koks buvo prieš tai buvusio proxy sertifikato „*Subject*“ laukas. Apibendrinant galime teigti, kad šis naujas formatas iš realizacinės pusės yra patogus, nes iš esmės generavimo procesas nesikeičia, tik atsiranda papildomas veiksmas skaičiuojant tikro sertifikato MD5 reikšmę. Šios reikšmės patalpinimas neturėtų sukelti problemų, nes į „*Subject*“ lauką norimą informaciją galima rašyti laisvai.

Apibrėžus naują proxy sertifikatų formatą toliau reikia suprojektuoti, kaip šie sertifikatai turi būti naudojami, tikrinami arba kitaip sakant, kaip turėtų būti vykdoma autentifikacija naudojantis jais.

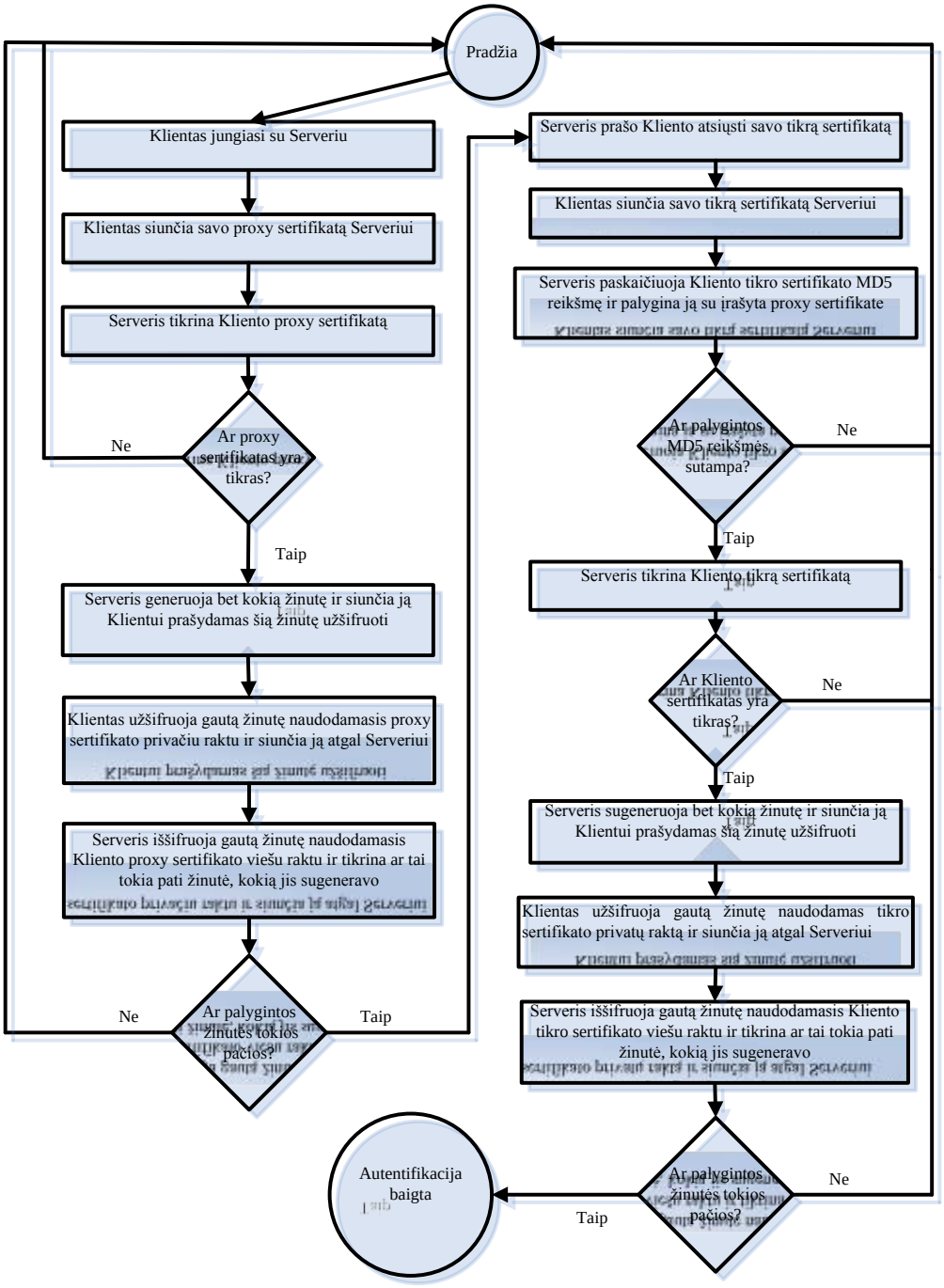
Atliekant proxy sertifikato patikrinimą jau anksčiau aptartu atveju reikėtų ne tik įsitikinti ar jo naudotojas turi jo privatų raktą, bet ir patikrinti ar šio sertifikato naudotojas yra tas kam suteikti šie įgaliojimai. Kitaip sakant proxy sertifikato naudotojas turėtų įrodyti, kad jis turi tikro sertifikato privatų raktą. Šį mechanizmą dar galima pavadinti dviguba autentifikacija, kai esybė naudojanti proxy sertifikatą turės patvirtinti savo kaip esybės autentiškumą ir pačio proxy sertifikato autentiškumą. Visą naujo proxy sertifikato sukūrimo mechanizmą galima padalinti į dvi dalis, tai yra:

1. Dviejų šalių dviguba abipusė autentifikacija.
2. Naujos raktų poros skirtos proxy sertifikatui generavimas, sertifikato prašymo generavimas ir tos užklausos pasirašymas.

Pirmiausia bus suprojektuotas autentifikacijos algoritmas, tačiau dar prisimenant vykdyto eksperimento rezultatus matoma, kad pirmas proxy sertifikatas yra išduodamas vartotojo sau pačiam ir įgaliojimus vartotojas perduoda pats sau. Tai reiškia, kad autentifikacijos mechanizme reikia nagrinėti šiuos abu atvejus, nes kiekvienu iš jų autentifikacijos procesas turi būti vykdomas skirtingai.

Tuo atveju, kai vartotojas išsiduoda proxy sertifikatą sau pačiam, autentifikacija vykdoma tik su VO, kuri pateikia vartotojo teises GRID tinkle. O tada pats vartotojas pasirašo proxy sertifikatą ir jam nereikia vykdyti antros autentifikacijos, kad sau pačiam dar patvirtintų savo sertifikatą. Šiuo atveju proxy sertifikato sukūrimo mechanizmas beveik nesiskiria nuo esančio GRID sistemose dabar, skirtumas yra tik toks, kad sukuriamas jau aptarto naujo formato sertifikatas, kur į proxy sertifikatą vartotojas patalpina savo patikimo (išduoto CA) sertifikato MD5 reikšmę.

Kitu atveju, kai proxy sertifikatas jau išduodamas vartotojo paslaugai ar vienos paslaugos kitai paslaugai, autentifikacijos mechanizmas bus kitoks, kaip jau vadinome – dvigubas. Taigi esybei (klientui) naudojančiai proxy sertifikatą reikės atlikti dvigubą tapatybės įrodymą: 1) naudojantis proxy sertifikatu; 2) savo patikimu sertifikatu. O esybė (serveris) turės naudoti savo tikrą sertifikatą kaip ir dabar bei taip turės įrodyti savo tapatybę vieną kartą. Todėl toliau bus pateiktas algoritmas, pagal kurį esybė (klientas) turės autentifikuotis su kita esybe (serveriu), sudarytą iš dviejų dalių. Kliento ir serverio visas autentifikacijos algoritmas pavaizduotas 9 paveiksle.



9 pav. Autentifikacija naudojantis proxy sertifikatu

Kai klientas ir serveris sėkmingai įvykdys suprojektuotą autentifikacijos atlgoritmą, toliau serveris generuos naują raktų porą skirtą proxy sertifikatui ir naudodamasis jais sugeneruos proxy sertifikato prašymo užklausą. Ši užklausa turės būti jau aptarto naujo formato, kur subjekto lauke serveris turės talpinti savo sertifikato MD5 reikšmę. Po to, serveris šią užklausą kaip ir įprasta siųs klientui, kuris patikrinęs subjekto lauke įrašytą MD5 reikšmę su paskaičiuota serverio sertifikato MD5 reikšme pasirašys proxy sertifikatą ir jį siųs atgal serveriui. Toliau serveris jau

galės naudotis šiuo proxy sertifikatu ir veikti vartotojo vardu bei išduoti kitus proxy sertifikatus kitiems serveriams.

Apibendrinant galime teigti, kad sertifikatų centro privataus rakto apsaugai naudojamas HSM padidins jo apsaugą ir sumažins galimybę jį sukompromituoti, nes privatus raktas iš HSM įrenginio negalės būti paimtas. Blogiausiu atveju įsilaužėlis, privačiu raktu galės pasinaudoti neteisėtai ir išsiduoti sertifikatą sau, tačiau tokie sertifikatai vėliau galės būti atšaukti, o CA privatus raktas liks nepaliestas. Kalbant apie proxy sertifikatus, naujo formato ir naujo autenifikacijos mechanizmo naudojimas, leis panaikinti grėsmę pasinaudoti šiais sertifikatais nelegaliai, nes jų naudotojai papildomai turės įrodyti savo tapatybę.

3.3. Išvados

1. Atlikus sertifikatų centro modelio projektavimą, remiantis GRID saugumo federacijos rekomendacijomis ir sprendimo budų analize sertifikatų centro privatus raktas bus papildomai patalpintas į HSM įrenginį. Kadangi naujų sertifikatų pasirašymui bus naudojamas privatus raktas patalpintas HSM, todėl reikės sinchronizuoti CA of-line dalį su HSM.
2. Proxy sertifikatų projektavimo metu naudojantis atlikto eksperimento rezultatais buvo nuspręsta papildomą hash reikšmę talpinti į „Subject“ lauką, ten talpinant proxy sertifikato gavėjo CA išduoto sertifikato MD5 reikšmę. Šių sertifikatų naujo formato naudojimas reikalauja dvigubos autentifikacijos.

4. SUPROJEKTUOTŲ MODELIŲ REALIZACIJOS

4.1. Sertifikatų centro realizacija

Aptarus CA sertifikato saugumo reikalavimus ir susipažinus su dviem on-line CA realizavimo architektūros modeliais, buvo nutarta šiuos abu modelius sujungti į vieną, išskiriant CA atliekamas funkcijas į du serverius on-line ir off-line bei off-line dalyje naudoti HSM, CA privataus rakto saugojimui. Sertifikato centro modelis pateiktas septintame paveiksle. Taigi visos sistemos realizavimui reikalingos priemonės ir reikalingi programiniai paketai bus apžvelgti šioje dalyje. Be to, sertifikatų centro realizacijos rezultatai bus iliustruojami paveikslais.

Realizacijai atlikti buvo pasirinktas jau antrame darbo skyriuje aptartas nemokamas OpenCA paketas, nes šiame pakete numatoma galimybė sertifikatų pasirašymo ir užklausų priėmimo funkcijas išskirti į dvi dalis, kitaip sakant off-line ir on-line dalis.

Naudoti programiniai paketai:

- OpenCA;
- Apache;
- mod_ssl;
- OpenSSL;
- OpenLDAP;
- Perl.

Pagal sertifikatų centro modelį (žr. 7 pav.) galime matyti, kad ryšį su internetu turės tik on-line dalis, t. y. šioje dalyje bus viešas interfeisas, kuriuo galės naudotis vartotojai. Jie galės pateikti prašymus sertifikatams gauti, gauti certifikatų centro sertifikatą PEM ir kitais formatais, peržiūrėti galiojačius sertifikatus ir kita. Sertifikatų centro off-line dalis bus fiziškai atskirta, neturės tiesioginio ryšio su viešaisiais tinklais ir ši sertifikatų centro dalis atliks tik sertifikatų pasirašymo procedūrą, kurie paskui per on-line dalį bus perduodami vartotojams. Siekiant padidinti sertifikatų centro apsaugą, sertifikatų centro privatus raktas bus saugomas fizinėje laikmenoje – HSM, todėl net ir nulaužus CA serverį ir pavogus HSM PIN (jei taip atsitiktų), nebus galima pavogti CA sertifikato privataus rakto.

Sertifikatų centro (OpenCA) viešas interfeisas

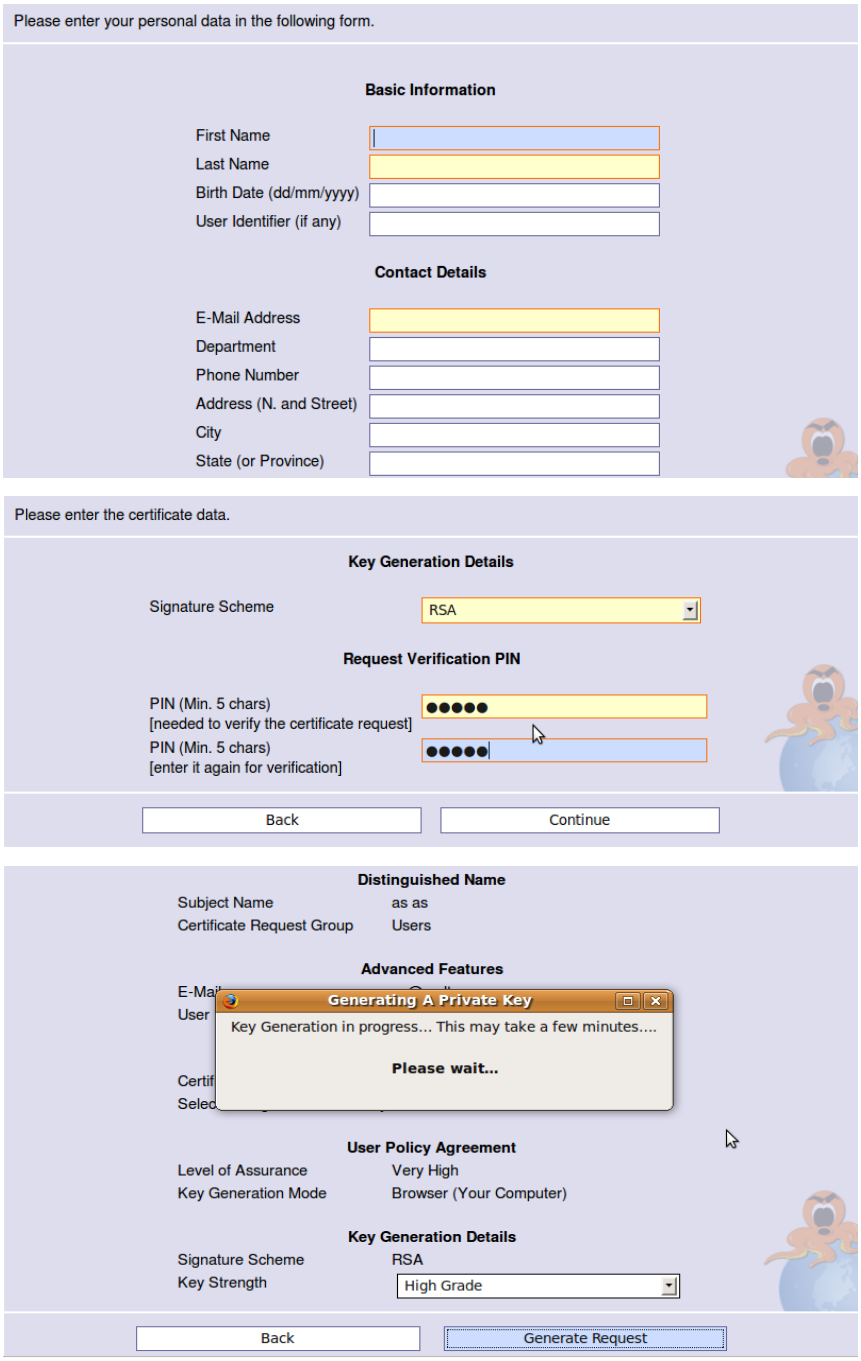
Žemiau pateiktame dešimtame paveiksle matomas viešas interfeisas kurį mato paprasti vartotojai. Šio interfeiso pagalba, vartotojai gali:

- gauti CA sertifikatą CRT, PEM, DER, CER, TXT formatais;
- gauti CRL DER, PEM ir TXT formatais;
- pateikti sertifikato prašymą;
- įdiegtį savo sertifikatą (reikia žinoti serijos numerį bei PIN, kuris buvo naudotas prašant sertifikato);
- atšaukti savo sertifikatą (jei jis būtų sukompromituotas);
- peržiūrėti šio sertifikatų centro vartotojų aktyvius, atšauktus, pasibaigusius sertifikatus ir kita.

10 pav. OpenCA viešas interfeisas

Sertifikato prašymas

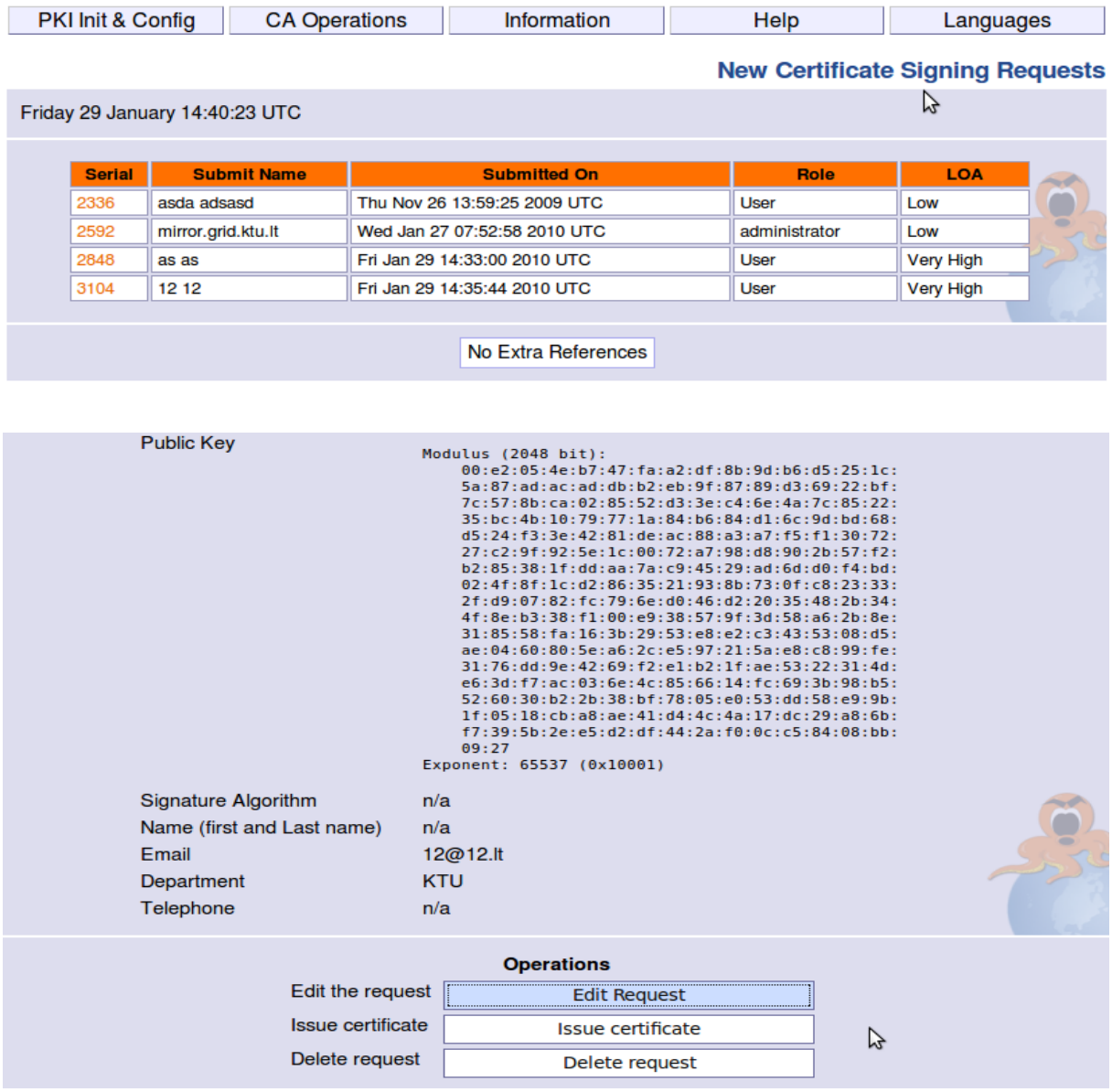
Vartotojas norėdamas gauti sertifikatą, pirmiausia turi užpildyti keletą formų, kuriose jis turi pateikti informaciją apie save ir sertifikatą, įvesti slaptažodį, sutikti su vartotojo sertifikato išdavimo sąlygomis, nurodyti rakto saugumo laipsnį bei sugeneruoti prašymą. Vienuoliktame paveiksle matoma keletas formų, kurias pildo vartotojas, o šio paveikslo apačioje matoma, kaip vykdomas privataus rakto generavimas. Kur generuojamas privatus raktas pasirenka pats vartotojas. Šis raktas gali būti generuojamas serverio pusėje arba vartotojo kompiuteryje.



11 pav. Sertifikato prašymas

Sertifikatų patvirtinimas

Visi vartotojų sertifikatų prašymai matomi viešame interfeise bei CA interfeise. Sertifikatų prašymus peržiūrėti, panaikinti, priimti ar pakeisti gali tik asmuo, kuris turi priėjimą prie CA interfeiso (žr. 12 pav.). Pagal sertifikatų centro modelį, ši dalis yra off-line dalyje. Priimant sertifikato prašymą būtina įvesti root privataus rakto slaptažodį.



12 pav. Sertifikato patvirtinimas

Sertfikatų atšaukimas

Klientams besinaudojant savo sertifikatais, jie gali būti sukompromituoti, t. y. gali būti pažeista privataus rakto apsauga. Tokiu atveju vartotojai gali atšaukti savo sertifikatus naudodamiesi realizuoto sertifikatų centro viešu interfeisu. Kai sertifikatai yra atšaukiami, šie sertifikatai yra įtraukiami į atšauktų sertifikatų sąrašą, kuris yra pasirašytas sertifikatų centro. Žemiau pateiktas sugeneruotas realizuoto sertifikatų centro CRL pavyzdys TXT formatu, tačiau viešojo interfeiso pagalba galima jį taip pat parsisiųsti PEM, DER formatais. Pavyzdyje aiškiai matomos CRL paskutinio atnaujimo bei sekančio atnaujinimo datos, tačiau kaip buvo minėta CRL gali būti atnaujintas anksčiau, tai yra iš karto, kai yra atšaukiamas koks sertifikatas. CRL

galioja 30 dienų bei jame matomi atšauktų sertifikatų serijos numeriai, kur mūsų pateiktame pavyzdyje matomi trys atšaukti sertifikatai.

Sugeneruoto CRL pavyzdys:

```
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: /DC=KPI/C=LT/L=Kaunas/ST=Kaunas/O=KPI/OU=KTU/CN=Donatas
Vilijosius/emailAddress=donatas.vilijosius@stud.ktu.lt
        Last Update: Jan 29 14:27:10 2010 GMT
        Next Update: Feb 28 14:27:10 2010 GMT
        CRL extensions:
            X509v3 CRL Number:
                12
Revoked Certificates:
    Serial Number: 07
        Revocation Date: Nov 26 12:29:11 2009 GMT
    Serial Number: 0C
        Revocation Date: Jan 29 14:09:20 2010 GMT
    Serial Number: 0D
        Revocation Date: Jan 29 14:18:43 2010 GMT
    Signature Algorithm: sha256WithRSAEncryption
        a3:18:8f:67:76:c3:4a:61:f4:49:a4:4b:0f:a3:18:14:43:3f:
        95:47:9c:54:2c:ad:df:fc:fb:c7:d1:4d:a5:b5:41:a7:06:e1:
        98:92:51:06:63:d7:3c:d8:ab:37:3a:12:6c:87:12:1d:ba:de:
        50:cd:28:45:1e:59:cb:aa:d4:58:46:63:6e:19:8b:2e:41:ba:
        f7:23:61:df:b6:67:0e:95:da:e8:2b:b7:50:d3:bd:81:39:f6:
        8d:43:49:0a:d0:97:f5:38:f0:3b:9f:e7:97:89:01:eb:c5:e5:
        19:44:4d:a5:08:f8:93:73:db:d3:7b:8b:fe:db:7a:7d:88:0a:
        1d:64:d0:af:98:0e:3e:76:6b:a3:b1:0d:a0:24:8e:34:35:88:
        28:ea:5b:32:a4:f6:2f:e0:e7:14:05:de:e4:03:76:33:e6:5e:
        22:34:81:c3:90:26:75:8d:68:35:e1:ab:1e:e8:94:07:90:f2:
        61:1c:85:36:3c:05:1d:88:a5:45:58:a9:71:cb:a8:65:d6:a5:
        dc:b8:5f:13:af:6e:d7:25:fe:80:bb:e5:37:5a:6d:95:6c:43:
        9e:76:d1:2a:50:ee:70:56:95:2d:1a:dd:e0:13:e3:6d:b4:90:
        35:b7:10:f7:96:ac:9e:99:72:b4:63:c2:81:01:20:86:72:cc:
        c3:10:ac:e5:8e:c8:13:ec:5c:6a:4d:dc:c6:09:d2:26:95:86:
        6f:7f:f3:d4:79:5e:c4:59:a3:54:0a:eb:1f:05:4b:92:99:71:
        1d:d3:10:ab:26:f9:1f:34:c2:98:e7:37:c3:c6:0a:5c:cc:f4:
        d3:77:6d:da:cb:56:8c:c2:03:34:0d:42:b8:aa:e6:b3:65:c3:
        30:72:ba:c1:e1:cb:e1:0d:d7:55:cb:28:75:11:cb:05:65:85:
        56:dc:38:ba:da:04:4c:30:e6:8c:e9:14:de:0e:98:d9:1d:02:
        f4:56:7b:ad:6a:b4:09:55:ec:56:fb:05:b0:ab:5d:3b:ce:42:
        a1:8f:fe:a1:32:7a:ec:c3:b7:93:6d:d7:c3:b2:fa:d6:42:bd:
        80:ae:19:cf:3e:e1:8a:33:e2:85:37:2e:5a:63:13:af:b7:7b:
        2b:4d:ca:af:2a:4d:57:c9:a2:34:2f:46:d1:a1:14:6b:5c:64:
        03:48:8f:19:9a:c3:ef:53:e0:c1:d5:b7:34:a0:39:88:aa:c4:
        cd:cd:91:6f:e6:59:fe:08:de:43:86:58:fd:90:37:fc:a8:0b:
        98:9e:c4:46:de:27:87:02:a8:89:79:71:1f:1e:6d:1b:8f:ec:
        cd:01:86:c0:48:36:aa:1f:68:b4:69:f0:7e:8e:53:7a:12:cd:
        52:d7:a4:92:7b:59:c2:47:4c:51:e5:28:f4:05:f0:f4:0e:11:
        ef:6c:5e:3a:e0:6a:1b:30:9a:79:52:a8:b8:a0:7c:de:1c:20:
        e1:a8:52:2f:5d:d3:04:06:f5:b0:9d:d1:20:d9:9f:3e:02:b2:
        0f:76:2e:15:ab:25:c0:3b:95:85:43:14:ee:4c:66:03:d0:23:
        b8:87:87:8c:60:66:f1:69:11:a2:8a:82:c4:8a:6c:9d:99:d3:
        da:27:33:f3:e1:b6:12:ab:ab:a6:32:c7:9f:81:e8:93:4b:ac:
        07:46:4d:28:bf:b7:e3:ec:25:9e:ea:57:ef:20:fb:32:fa:7c:
        62:47:00:23:34:49:3b:bf:f7:8a:f5:52:d6:68:92:2f:03:c9:
        1e:8f:07:b6:f7:1e:d4:66:85:22:f3:26:85:ad:b2:e4:12:3c:
        30:2c:9a:0b:4d:ed:3d:33:58:d3:7a:32:94:47:0b:63:04:3d:
        db:4c:10:1a:cf:c9:14:7a:97:a1:70:9d:22:44:4f:bf:91:64:
        27:86:a1:e4:3f:a3:7d:ec:d5:48:de:41:ff:93:94:20:c4:77:
        40:81:89:a2:b2:f1:69:42:fd:0f:81:fd:38:d7:a4:0f:e3:da:
        10:6b:cf:32:2d:ae:c8:23:e8:49:88:f4:a7:8b:13:f7:6b:28:
        c5:81:0b:8e:04:25:2d:26:fb:5d:1e:e9:f2:a8:77:43:be:50:
        9d:b9:fd:cf:34:8c:cf:61:df:7e:03:91:39:f2:48:6e:1f:ad:
        f2:88:e2:ef:1e:73:12:7c:6c:50:1f:93:cc:67:99:b3:47:a5:
        c7:3b:ae:d7:55:d2:04:f2:66:3a:4f:86:9b:51:7e:9e:4a:7a:
        c4:7e:fb:03:1d:0d:63:5e:93:59:19:45:cd:ad:8e:6a:5d:87:
        b7:f1:89:f1:c3:f8:fc:d7:47:2f:03:70:2c:52:9a:28:3d:ab:
        67:8a:83:63:16:13:db:94:6f:3c:46:b7:ce:f3:d7:46:95:b0:
        fb:96:e0:b9:6a:cb:75:59:7f:ff:3d:85:3e:3f:04:13:d3:a0:
```

```
        8e:bc:34:3d:91:30:34:48:28:3f:e1:a2:d4:cf:86:40:2d:90:
        8c:82:52:02:d2:c3:72:d9:6e:f0:15:eb:6f:0e:1e:14:8b:a6:
        56:b2:06:30:9f:1c:21:3a:16:f0:fc:7c:cc:28:f3:95:de:b4:
        de:0f:a2:77:14:1d:cf:f8:48:60:06:d8:18:3c:be:a4:aa:8a:
        f1:ec:c1:92:38:76:52:11:7b:c9:c9:66:5d:99:e1:70:91:4a:
        94:ad:e9:9b:52:b2:55:ba:50:53:cd:55:16:2e:f2:b7:1c:97:
        0b:c9:9e:c6:0b:02:94:83:51:8b:73:8e:15:f7:87:ca
-----BEGIN X509 CRL-----
MIIFRTCCAS0CAQEwDQYJKoZIhvcNAQELBQAwgawxEzARBgoJkiaJk/IsZAEZFgNL
UEkxCzAJBgNVBAYTAkxUMQ8wDQYDVQQHDAZLYXVuYXMxDzANBgNVBAgMBkthdW5h
czEMMAoGA1UECgwDS1BJMQwwCgYDVQQLDANLVFUxGzAZBgNVBAMMEkRvbmF0YXMg
VmlsaWpvc2l1czEtMCsGCSqGSIb3DQEJARYeZG9uYXRhcy52aWxpam9zaXVzQHN0
dWQua3R1Lmx0Fw0xMDAxMjkxNDI3MTBaFw0xMDAyMjgxNDI3MTBaMDwwEgIBBxcN
MDkxMTI2MTIyOTExWjASAgEMFw0xMDAxMjkxNDA5MjBaMBICAQ0XDTEwMDEyOTE0
MTg0M1qgDjAMMAoGA1UdFAQDAgEMMA0GCSqGSIb3DQEBCwUAA4IEAQCjGI9ndsNK
YfRJpEsPoxgUQz+VR5xULK3f/PvH0U2ltUGnBuGYklEGY9c82Ks3OhJshxIdut5Q
zShFHlnLqtRYRmNuGYsuQbr3I2HftmcOldroK7dQ072BOfaNQ0kK0Jf1OPA7n+eX
iQHrxeUZRE2lCPiTc9vTe4v+23p9iAodZNCvmA4+dmujsQ2gJI40NYgo6lsypPYv
4OcUBd7kA3Yz5l4iNIHDkCZ1jWg14ase6JQHkPJhHIU2PAUdiKVFWKlxy6hl1qXc
uF8Tr27XJf6Au+U3Wm2VbEOedtEqUO5wVpUtGt3gE+NttJA1txD3lqyemXK0Y8KB
ASCGcszDEKzljsgT7FxqTdzGCdImlYZvf/PUeV7EWaNUCusfBUuSmXEd0xCrJvkf
NMKY5zfDxgpczPTTd23ay1aMwgM0DUK4quazZcMwcrrB4cvhDddVyyh1EcsFZYVW
3Di62gRMMOaM6RTeDpjZHQL0VnutarQJVexW+wWWq107zkKhj/6hMnrsw7eTbdfD
svrWQr2ArhnPPuGKM+KFNy5aYxOvt3srTcqvKk1XyaI0L0bRoRRrXGQDSI8ZmsPv
U+DB1bc0oDmIqsTNzZFv5ln+CN5Dhlj9kDf8qAuYnsRG3ieHAqiJeXEfHm0bj+zN
AYbASDaqH2i0afB+jlN6Es1S16SSe1nCR0xR5Sj0BfD0DhHvbF464GobMJp5Uqi4
oHzeHCDhqFIvXdMEBvWwndEg2Z8+ArIPdi4VqyXAO5WFQxTuTGYD0CO4h4eMYGbx
aRGiioLEimydmdPaJzPz4bYSq6umMsefgeiTS6wHRk0ov7fj7CWe6lfvIPsy+nxi
RwAjNEk7v/eK9VLWaJIvA8kejwe29x7UZoUi8yaFrbLkEjwwLJoLTe09M1jTejKU
RwtjBD3bTBAaz8kUepehcJ0iRE+/kWQnhqHkP6N97NVI3kH/k5QgxHdAgYmisvFp
Qv0Pgf0416QP49oQa88yLa7II+hJiPSnixP3ayjFgQuOBCUtJvtdHunyqHdDvlCd
uf3PNIzPYd9+A5E58khuH63yiOLvHnMSfGxQH5PMZ5mzR6XHO67XVdIE8mY6T4ab
UX6eSnrEfvsDHQ1jXpNZGUXNrY5qXYe38Ynxw/j810cvA3AsUpooPatnioNjFhPb
lG88RrfO89dGlbD7luC5ast1WX//PYU+PwQT06COvDQ9kTA0SCg/4aLUz4ZALZCM
glIC0sNy2W7wFetvDh4Ui6ZWsgYwnxwhOhbw/HzMKPOV3rTeD6J3FB3P+EhgBtgY
PL6kqorx7MGSOHZSEXvJyWZdmeFwkUqUrembUrJVulBTzVUWLvK3HJcLyZ7GCwKU
g1GLc44V94fK
-----END X509 CRL-----
```

HSM realizacija

Įdiegus OpenCA paketą atskiromis dalimis (off-line CA ir on-line CA), toliau reikia apsaugoti svarbiausią sertifikatų centro elementą, privatų raktą. Kaip anksčiau buvo apibrėžta, reikia į HSM patalpinti privatų raktą. Privataus rakto patalpinimas buvo atliktas naudojantis OpenSC programine įranga. HSM ir off-line dalies bendravimas buvo realizuotas sukonfigūruojant OpenCA taip, kad sertifikatų pasirašymams būtų naudojamas privatus raktas patalpintas HSM įrenginyje. Pasirašymams vykdyti, naudojant HSM įrenginyje patalpintą privatų raktą, taip pat naudojama OpenSC programinė įranga, o OpensCA ir OpenSC paketų darbo sinhronizacija realizuota naudojant tokia konfigūraciją:

```
<token>
            <name>CA</name>
            <type>OpenSC</type>
            <mode>standby</mode>
            <option>
                <name>SHELL</name>
                <value>/usr/local/ssl/bin/openssl</value>
            </option>
            <option>
                <name>WRAPPER</name>
                <value></value>
            </option>
```

```
<option>
    <name>KEY</name>
    <value>slot_0-id_45</value>
</option>
<option>
    <name>PASSWD_PARTS</name>
    <value>1</value>
</option>
<option>
    <name>PEM_CERT</name>
    <value>/opt/openca/var/openca/crypto/cacerts/cacert.pem</value>
</option>
<option>
    <name>DER_CERT</name>
    <value>/opt/openca/var/openca/crypto/cacerts/cacert.der</value>
</option>
<option>
    <name>TXT_CERT</name>
    <value>/opt/openca/var/openca/crypto/cacerts/cacert.txt</value>
</option>
<option>
    <name>CHAIN</name>
    <value>/opt/openca/var/openca/var/crypto/chain</value>
</option>
<option>
    <name>OPENCA_SV</name>
    <value>/usr/local/bin/openca-sv</value>
</option>
<option>
    <name>TMPDIR</name>
    <value>/opt/openca/var/openca/tmp</value>
</option>
<option>
    <name>CONFIG</name>
    <value>/opt/openca/etc/openssl/openssl.cnf</value>
</option>
<option>
    <name>RANDFILE</name>
    <value>/opt/openca/var/openca/crypto/.rand</value>
</option>
<option>
    <name>ENGINE</name>
    <value>pkcs11</value>
</option>
<option>
    <name>PRE_ENGINE</name>
    <value>SO_PATH:/usr/local/lib/opensc/engine_pkcs11.so</value>
</option>
<option>
    <name>PRE_ENGINE</name>
    <value>ID:pkcs11</value>
</option>
<option>
    <name>PRE_ENGINE</name>
    <value>LIST_ADD:1</value>
</option>
<option>
    <name>PRE_ENGINE</name>
    <value>LOAD</value>
</option>
<option>
    <name>PRE_ENGINE</name>
    <value>MODULE_PATH:/usr/local/lib/pkcs11/opensc-pkcs11.so</value>
</option>
<option>
    <name>CARDDRIVER</name>
    <value>flex</value>
</option>
<option>
    <name>CARDREADER</name>
    <value>0</value>
</option>
<option>
    <name>PKCS15_INIT</name>
```

```
            <value>/usr/local/bin/pkcs15-init</value>
        </option>
        <option>
            <name>PKCS15_TOOL</name>
            <value>/usr/local/bin/pkcs15-tool</value>
        </option>
        <option>
            <name>OPENSC_TOOL</name>
            <value>/usr/local/bin/opensc-tool</value>
        </option>
        <option>
            <name>DEBUG</name>
            <value>1</value>
        </option>
    </token>
```

Apibendrinant galime teigti, kad realizuojant sertifikatų centrą buvo siekiama kuo labiau apsaugoti sertifikatų centro privatų raktą, todėl sertifikatų centro atliekamos funkcijos buvo atskirtos, t.y. sertifikatų pasirašymo veiksmas buvo fiziškai atskirtas nuo išorės. Jau šis faktas pagal saugios infrastruktūros autentifikacijos profilį turėtų užtikrinti reikiamą apsaugą, tačiau siekdami maksimalios apsaugos sertifikatų centro privatų raktą dar papildomai apsaugojome fiziškai jį patalpindami į HSM. Tai suteiks papildomą apsaugą tuo atveju, jei būtų nulaužtas sertifikatų centras. Tokia apsauga reikalinga dėl to, kad sukompromitavus sertifikatų centrą, tuo pačiu būtų nutraukta pasitikėjimo grandinė, t.y. vartotojų sertifikatai taptų nepatikimi. HSM leidžia apsaugoti privatų raktą nuo perėmimo, net jei ir būtų fiziškai įsilaušta į sertifikatų centrą. Blogiausiu atvejų juo būtų galima pasinauoti, pasirašant sertifikatą nelegaliai, tačiau toks nelegaliai pasirašytas sertifikatas gali būti atšauktas, o pati sertifikatų sistema nebūtų sukompromituota.

4.2. Proxy sertifikatų modelio realizacija

Trečiame skyriuje suprojektavus naują proxy sertifikatų išdavimo ir valdymo mechanizmą, toliau buvo atliekama jo realizacija. Šio modelio autentifikacijai vykdyti buvo naudojamasi „openssl s_server“ komanda (serverio paleidimui, kuris visą laiką „klausosi“) ir „openssl s_client“ komanda (klientui jungiantis su serveriu ir bandant autentifikuotis su serveriu). Norint apjungti visą algoritmą į vieną buvo parašyta keletas bash skriptų, kurie naudojami jungiantis su serveriu ir bandant klientui autentifikuotis su juo, sertifikato užklausos generavimui bei jos pasirašymui. Autentifikuojantis klientas su serveriu pirmą kartą turi naudoti proxy sertifikatą, o antrą kartą naudoti savo patikimą sertifikatą išduotą CA. Be bash skriptų, visas algoritmas buvo aprašytas php programavimo kalba, kurios pagalba kviečami bash skriptai, atliekamas MD5 reikšmių tikrinimas ir taip pat naudojantis ssh funkcijomis jungiamasi prie serverio ir ten

nusiunčiant reikiamus sertifikatus. Reikia pastebėti, kad visą proxy sertifikato sukūrimo mechanizmą atliekančios sistemos prototipo veikimas paremtas dviguba autentifikacija. Tai yra nagrinėjamas atvejis, kai vartotojas proxy sertifikatą išduoda serveriui jau naudodamasis savo proxy sertifikatu. Kaip jau buvo aprašyta ankstesnėse dalyse pirmą proxy sertifikatą vartotojas GRID tinkle išsiduoda sau pačiam, o toks atvejis nereikalauja dvigubos autentifikacijos. Vadinasi, šis atvejis atitinka realizuoto mechanizmo tik kelis etapus, todėl jis bus tik bendrai aprašytas pateikiant visą proxy sertifikato išdavimo mechanizmą, kai proxy sertifikatas išduodamas proxy sertifikato.

Pagal algoritmą klientas atlieka tokius etapus:

1. Autentifikuojasi su serveriu naudodamasis proxy sertifikatu.
2. Serveriui siunčiamas tikras sertifikatas, kuris lygina paskaičiuotą tikro sertifikato MD5 reikšmę su reikšme įrašyta proxy sertifikate.
3. Klientas autentifikuojasi CA išduotu patikimu sertifikatu.
4. Serveris generuoja naujo proxy sertifikatui skirtą raktų porą, sugeneruoja proxy sertifikato prašymo užklausą ir ją siunčia klientui.
5. Klientas gavęs proxy sertifikato prašymą jį patikrina ir jei viskas gerai pasirašo.

Pirmo punkto realizacija buvo atlikta naudojantis jau minėta openssl s_client komanda kuri jungiasi prie openssl serverio (serveris paleistas naudojantis openssl s_server komanda, kuris reikalauja kliento sertifikato) ir autentifikuojasi. Visa tai realizuota bash scriptu pateiktu tryliktame paveiksle.

```
connect_server:
#!/bin/bash
proxycert="$1".pem
proxycertkey="$1"key.pem
if [ $1 ]; then
a=$(echo "QUIT" | openssl s_client -connect 192.168.72.132:9000 -CAfile
CA_visi.pem -cert $proxycert -key $proxycertkey -state | grep "Verify return
code: 0 (ok)")
res=$?
if [ $res -eq 0 ] ; then
echo Autentifikacija sekminga
else
echo Autentifikacija nesekminga
fi
echo "------"
echo $a
echo "----"
else
echo "1 argumentas proxy sertifikatas"
fi
```

13 pav. Autentifikacija su serveriu

Šiame skripte nurodomas serverio adresas, portas, proxy sertifikatas naudojamas autentifikacijai bei CA failas, kuriame yra kliento patikimi CA sertifikatai. Jei autentifikacija su serveriu sėkminga iš serverio gaunama „Verify return code: 0 (ok)" žinutė ir tada vykdomas sekantis žingsnis. Jei autentifikacija nesėkminga tolimesni algortimo veiksmai yra nutraukiami.

Pabaigus pirmą autentifikaciją yra vykdomas antras punktas, t. y. serveriui siunčiamas vartotojo sertifikatas. Tai atliekama naudojantis php ir pagalbinėmis ssh funkcijomis, kurios leidžia serveriui nusiųsti vartotojo sertifikatą. Tada serveris gavęs vartotojo sertifikatą, skaičiuoja jo MD5 reikšmę ir ją palygina su reikšme esančia proxy sertifikate (žr. 14 pav.).

```
md5.php:
<?php
exec("openssl x509 -noout -subject -in /home/donatas/proxy/test/proxycert.pem
-subject", &$subject);
echo "<br>";
foreach ($subject as $subj){
  echo "Proxy ".$subj;
  echo "<br>";
  echo "<br>";
}

$ilgis = strlen($subj);
$pradzia = $ilgis-32;
$md51 = substr("$subj",$pradzia,$ilgis);
echo "Proxy MD5_1=";
echo $md51;
echo "<br>";

exec("md5sum /home/donatas/proxy/test/usercert.pem | sed "s/usercert.pem//g" |
sed -n "s/ .*//p"", &$md5);
echo $md5[1];

foreach ($md5 as $md52){
  echo "Users MD5_2=";
  echo $md52;
  echo "<br>";
  echo "<br>";
}

if($md51==$md52){
echo "Tikro sertifikato MD5 ir proxy sertifikate esanti MD5 reiksmes
sutampa<br>";
$MD5reiksmes = "OK";
echo "<br>";
echo $MD5reiksmes;
}else
echo "Klaida: MD5 reiksmes nesutampa, galimas bandymas apsimesti proxy
sertifikato savininku <br>";
?>
```

14 pav. MD5 reikšmės tikrinimas

Atlikus MD5 reikšmių patikrinimą, klientas gauna pranešimą ar reikšmės yra lygios, tai yra pranešimą „OK" arba jei nelygios „Klaida: MD5 reiksmes nesutampa, galimas bandymas

apsimesti proxy sertifikato savininku“. Jei gaunamas pranešimas „OK“, tada pereinama į sekantį etapą ir atliekama vartotojo autentifikacija su serveriu, naudojant tikrą kliento sertifikatą. Tai realizuota naudojantis beveik tuo pačiu skriptu, kuris pateiktas prie pirmos autentifikacijos, o skirtumas yra tik toks, kad šį kartą vartotojas turi įvesti tikro privataus rakto slaptažodį, nes, kaip žinoma, privatus raktas turi būti laikomas saugiai. Todėl anksčiau pateiktame kode openssl komanda pasipildo tik „–pass“ argumentu.

Atlikus dvigubą abipusę autentifikaciją vykdomas ketvirtas punktas, kur serveris generuoja raktų porą bei sukuria sertifikato prašymo užklausą, kurioje, formuojant „*Subject*“ lauką, kaip jau aprašėme projektinėje dalyje, pridedama standartinė „/proxy“ ir papildoma savo sertifikato MD5 reikšmė. Sugeneruota proxy sertifikato užklausa nuo dabar GRID sistemose generuojamų skiriasi tik tuo, kad „*Subject*“ lauke papildomai dar įrašoma serverio sertifikato MD5 reikšmė, o tai realizuota šiuo bash skriptu (žr. 15 pav.).

```
genproxy:
#!/bin/bash
usercert=/home/donatas/proxy/test/"$1".pem

kam_isduodam_cert=/home/donatas/proxy/test/"$2".pem
proxy=/home/donatas/proxy/test/"$3"
proxycsrfile=/var/www/proxy/"$3".csr
proxykeyfile=/var/www/proxy/"$3"key.pem
conf=/home/donatas/proxy/test/csr.conf
if  [ $1 ]  && [ $2 ]  && [ $3 ]

then

subject=$(openssl x509 -noout -subject -in $usercert | sed "s/subject= //g")

echo Kliento proxy subject= $subject
md5reiksm=$(md5sum $kam_isduodam_cert | sed -n "s/ .*//p")
prideti="/CN=proxy "
gsubject="$subject$prideti$md5reiksm"

echo Naujo proxy subject= $gsubject
openssl req -new -config $conf -out $proxycsrfile -keyout $proxykeyfile -subj
"$gsubject"

echo Proxy sertifikato requestas sugeneruotas... siunciama klientui

else
echo "1 kas isduoda proxy sertifikata"
echo "2 kam isduodamas proxy sertifikatas"
echo "3 naujo proxy vardas"
fi
```

15 pav. Proxy sertifikato prašymo užklausos generavimas

Proxy sertifikato prašymo užklausai generuoti naudojamame skripte nurodoma: kas išduoda proxy sertifikatą (klientas), kam išduodamas šis sertifikatas (serveris) bei naujo proxy sertifikato

vardas. Toliau šis prašymas yra siunčiamas klientui, o tam realizuoti buvo naudojama PHP programavimo kalba, kurios pagalba pirmiausia įvykdomas pateiktas skriptas ir tada pasinaudojant papildomomis ssh funkcijomis užklausa nusiunčiama klientui (žr. 1 Priedas).

Toliau yra vykdomas paskutinis penktas algoritmo etapas, t. y. proxy sertifikato prašymo užklausos pasirašymas ir pasirašyto sertifikato siuntimas atgal serveriui. Klientas gavęs pranešimą iš serverio apie sėkmingai sugeneruotą raktų porą ir gavęs sertifikato prašymo užklausą ją pirmiausia patikrina ir tik tada pasirašo. Klientas tikrindamas gautą užklausą, tikrina ar serveris nebando proxy sertifikato gauti ne sau, tai yra vykdo analogišką veiksmą jau aptartą antrame šio algoritmo etape, tik šį kartą jau klientas skaičiuoja serverio tikro sertifikato MD5 reikšmę ir palygina ją su reikšme įrašyta proxy sertifikato prašyme. Jei palyginus, šios reikšmės sutampa, tada klientas pasirašo užklausą savo proxy sertifikato privačiu raktu, o tai realizuota atitinkamu bash skriptu (žr. 16 pav.).

```
signproxy:
#!/bin/bash
proxy_csr=/home/klientas/proxy/test/$1.csr
cert=/var/www/cert/$2.pem
certkey=/var/www/cert/$2key.pem
proxycert=/var/www/cert/$1.pem
if [ $1 ] && [ $2 ] ; then
if [ $3 ]; then
keypass="pass:$3"
#slaptazodzio reikia tik kai pasiraso pats useris
openssl x509 -req -md5 -CAcreateserial -in $proxy_csr -days 1 -CA $cert -CAkey
$certkey -extfile csr.conf -extensions v3_proxy -out $proxycert -passin
$keypass
rez=$?
if [ $rez -eq 0 ]; then
echo Proxy sertifikatas sekmingai pasirasytas
else echo Pasirasant ivyko klaida
fi
else
#is proxy isduodamas kitas proxy
openssl x509 -req -md5 -CAcreateserial -in $proxy_csr -days 1 -CA $cert -CAkey
$certkey -extfile csr.conf -extensions v3_proxy -out $proxycert
rez=$?
if [ $rez -eq 0 ]; then
echo Proxy sertifikatas sekmingai pasirasytas
else
echo Pasirasant ivyko klaida
echo $a
fi
fi
else
echo 1 requestas kuris yra pasirasomas
echo 2 privatus raktas kuriuo pasirasoma
echo 3 privataus rakto slaptazodis, jei reikia
fi
```

16 pav. Proxy sertifikato prašymo užklausos pasirašymas

Pateiktas bash skriptas gali būti naudojamas ne vien tik serverio proxy sertifikato prašymo užklausai pasirašyti, bet taip pat ir pasirašant, kai proxy sertifikatas išduodamas sau pačiam. O tokiu atveju klientas turėtų vykdyti ketvirto etapo veiksmą (sugeneruoti proxy sertifikato prašymą) bei pats jį pasirašyti jau nebevykdydamas užklausos subjekte įrašytos MD5 reikšmės tikrinimo.

Kai proxy sertifikato užklausa yra sėkmingai pasirašoma, tada klientas jau pasirašytą sertifikatą siunčia serveriui, o tai realizuota vėl naudojantis PHP programavimo kalba ir papildomomis ssh funkcijomis. Visus aprašytus etapus realizuojantis PHP kodas pateiktas 2 priede.

Realizuotas proxy sertifikatų naudojimo mechanizmas naudojant MD5 reikšmes proxy sertifikatuose turi dar vieną ankščiau neminėtą pliusą. Tokių naujo tipo proxy sertifikatų naudojimas ir aptartas jų valdymo mechanizmas leidžia taip pat tokius sertifikatus atšaukti. Visų pirma pavogus proxy sertifikato privatų raktą įsilaužėliui įgaliojimai nėra perduodami, nes jis negali įvykdyti antrosios autentifikacijos, kurios metu tikrinamas proxy sertifikato naudotojas. Vadinasi, proxy sertifikatas gali būti sukompromituotas tik tada, kai yra pavogiamas serverio tikras privatus raktas. Tačiau jei taip atsitiktų, serveris privalėtų apie tai pranešti CA, kuri šio serverio sertifikatą atšauktų, o tuo pačiu ir visus šio sertifikato išduotus proxy sertifikatus. Proxy sertifikatai būtų menamai atšaukti (jie nebutų įtraukti į specialius atšauktų sertifikatų sąrašus), nes naudojant juos ir reikiant autentifikuotis antrą kartą, butų gaunama klaida, kad serverio sertifikatas yra atšauktas, o tai reikštų, kad ir proxy sertifikatas negalioja.

4.3. Išvados

1. Realizuotas sertifikatų centras atitinka visus GRID saugumo federacijos sertifikatų centrams keliamus reikalavimus bei papildomai užtikriną CA privataus rakto apsaugą nuo galimo sukompromitavimo.
2. Realizuotas proxy sertifikatų valdymo mechanizmas, reikalauja atlikti dvigubą autentifikaciją, kurios metu atsiranda galimybė šiuos sertifikatus „atšaukti“, jie automatiškai tampa negaliojančiais jei yra atšaukiamas šį sertifikatą naudojančios esybės CA išduotas sertifikatas.

5. EKSPERIMENTAI IR TESTAVIMAS

Atlikus sertifikatų centro modelio ir prototipinio proxy sertifikatų veikimo mechanizmo realizacijas, toliau, siekiant nustatyti kaip papildomų apsaugos elementų įdiegimas ar esamų mechanizmų patobulinimai paveikė sistemos ar esamų algoritmų našumą, reikia atlikti eksperimentus.

5.1. Sertifikatų sertifikato darbo našumo eksperimentas

Buvo atliktas eksperimentas, kurio metu buvo tikrinama kaip pasikeitė sertifikatų centro darbo našumas, kai sertifikatų centro privatus raktas buvo patalpintas į HSM (Aladdin eToken PRO 64k). Kadangi HSM naudojamas tik tada kai reikia vykdyti sertifikatų prašymo užklausų pasirašymą, tai buvo tirta tik pačio pasirašymo užtrunkamas laikas.

Kai pasirašymui naudojamas privatus raktas buvo patalpintas lokaliame diske, buvo naudotas batch skriptas ir jame keičiamas pasirašymo vykdymo ciklų skaičius (žr. 17 pav.).

```
#!/bin/bash
START=$(date +%s%N)
for ((  i = 1 ;  i <= 500;  i++  ))
do
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
done
END=$(date +%s%N)
DIFF=$(( $END - $START ))
echo "It took $DIFF nanoseconds"
```

17 pav. Pasirašymas su lokaliame diske esančiu privačiu raktu

Kai pasirašymui naudojamas privatus raktas buvo patalpintas HSM‘e, buvo naudotas batch skriptas ir jame keičiamas pasirašymų vykdymo skaičius (žr. 18 pav).

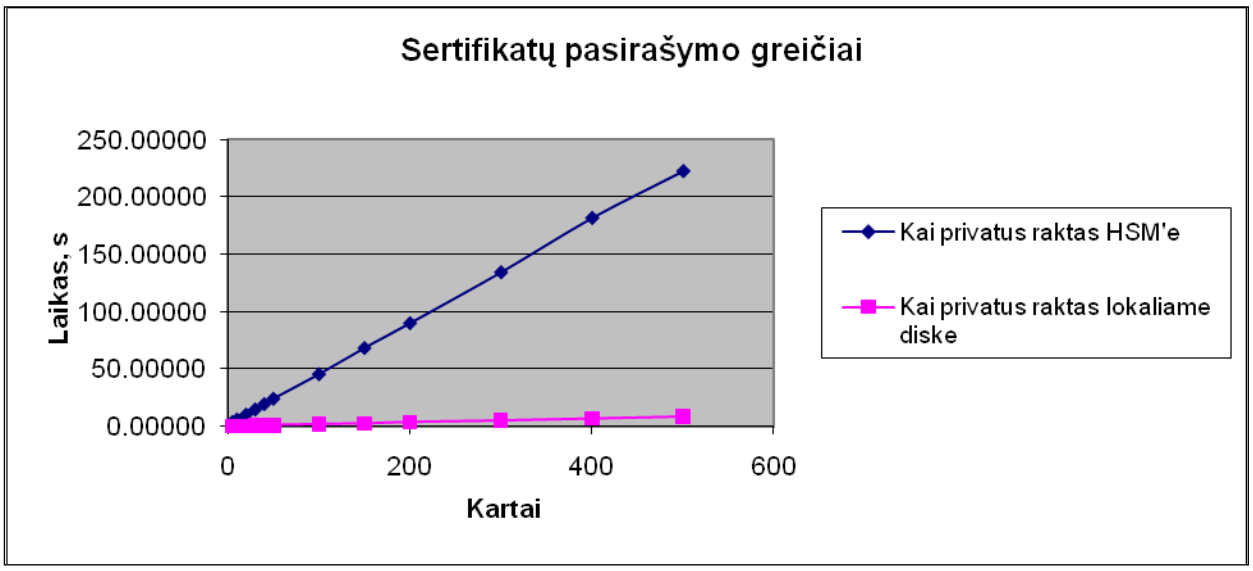
```
 #!/bin/bash
START=$(date +%s%N)
cat << EOF | openssl

engine dynamic -pre SO_PATH:/usr/lib/engines/engine_pkcs11.so -pre
ID:pkcs11 -pre LIST_ADD:1 -pre LOAD -pre MODULE_PATH:opensc-pkcs11.so

x509 -engine pkcs11 -signkey slot_0-id_45 -keyform engine -in req.pem
-out cert.pem -passin "pass:1986"
x509 -engine pkcs11 -signkey slot_0-id_45 -keyform engine -in req.pem
-out cert.pem -passin "pass:1986"
x509 -engine pkcs11 -signkey slot_0-id_45 -keyform engine -in req.pem
-out cert.pem -passin "pass:1986"
x509 -engine pkcs11 -signkey slot_0-id_45 -keyform engine -in req.pem
-out cert.pem -passin "pass:1986"
x509 -engine pkcs11 -signkey slot_0-id_45 -keyform engine -in req.pem
-out cert.pem -passin "pass:1986"

EOF

END=$(date +%s%N)
DIFF=$(( $END - $START ))
echo "It took $DIFF nanoseconds"
```

18 pav. Pasirašymas su HSM įrenginyje esančiu privačiu raktu

Naudojantis pateiktais skriptais buvo gautos laiko reikšmės nano sekundėmis. Gauti rezultatai pateikiami 1 lentelėje, o lentelės rezultatai pateikiami grafiku, kur vaizduojama laiko priklausomybė nuo parašų kiekio, kai parašui naudotas sertifikatų centro privatus raktas patalpintas lokaliame diske ir HSM‘e (žr. 19 pav.).

1 lentelė. Sertifikatų pasirašymų greičiai

	Kai privatus raktas HSM'e			**Kai privatus raktas lokaliame diske**		
Kartai	Laikas, ns	Laikas, s	Vidutinis vieno parašo laikas, s	Laikas, ns	Laikas, s	Vidutinis vieno parašo laikas, s
5	4108328142	4.10833	0.82167	72450647	0.07245	0.01449
10	6274711328	6.27471	0.62747	166299230	0.16630	0.01663
20	10576742557	10.57674	0.52884	334896803	0.33490	0.01674
30	15008398293	15.00840	0.50028	428128126	0.42813	0.01427
40	19777144067	19.77714	0.49443	603871401	0.60387	0.01510
50	24384399001	24.38440	0.48769	830718021	0.83072	0.01661
100	45723790908	45.72379	0.45724	1548491941	1.54849	0.01548
150	68507314984	68.50731	0.45672	2270589142	2.27059	0.01514
200	90158533794	90.15853	0.45079	3503135508	3.50314	0.01752
300	134401662033	134.40166	0.44801	4890915052	4.89092	0.01630
400	181699401734	181.69940	0.45425	6694576173	6.69458	0.01674
500	222413599434	222.41360	0.44483	8476912632	8.47691	0.01695



19 pav. Sertifikatų pasirašymo greičių grafikas

Pagal gautus rezultatus pateiktus lenteleje matome, kad vidutinis vieno parašo laikas sekundėmis yra beveik vienodas visą laiką, nepriklausomai nuo to kiek, kartų buvo bandoma pasirašyti vienu metu. Tai matome ir iš grafiko, kad užtruktas laikas pasirašant sertifikatus tiesiškai priklauso nuo parašų kiekio. Laiko priklausomybė nuo parašų kiekio matoma abiem atvejais, kai pasirašymui naudojamas privatus raktas buvo lokaliame diske ir HSM‘e.

Atlikto eksperimento rezultatuose taip pat matomas didelis sertifikatų centro darbo našumo sumažėjimas kai sertifikatų centro privatus raktas buvo patalpintas į HSM. Norint apskaičiuoti kiek kartų sumažėjo sertifikatų centro darbo našumas, kai privataus rakto apsaugai buvo naudotas HSM, imame minimalius vidutinius vieno parašo laikus, kai privatus raktas buvo lokaliame diske ir kai HSM‘e, nes laikome, kad tuo metu kompiuteris, su kuriuo buvo atliekamas eksperimentas, buvo mažiausiai užimtas kitais procesais, kas galėjo įtakoti skirtingas vidutines parašo reikšmes darant bandymus. Vadinasi, darbo našumas sumažėjo $0.44483 / 0.01427 \approx 31.17$ karto.

5.2. Proxy sertifikatų išdavimo procedūrų testavimas

Atlikus proxy sertifikatų išdavimo mechanizmo realizaciją toliau buvo atliekami eksperimentai siekiant išnagrinėti kaip pasikeitė atskirų proxy sertifikatų išdavimo etapų našumas. Buvo vykdyti eksperimentai nagrinėjantys autentifikacijos, proxy sertifikatų prašymo generavimo bei šio prašymo pasirašymo veiksmus, kurių gautus rezultatus aptarsime atskirai.

Autentifikacijos algoritmo testavimas

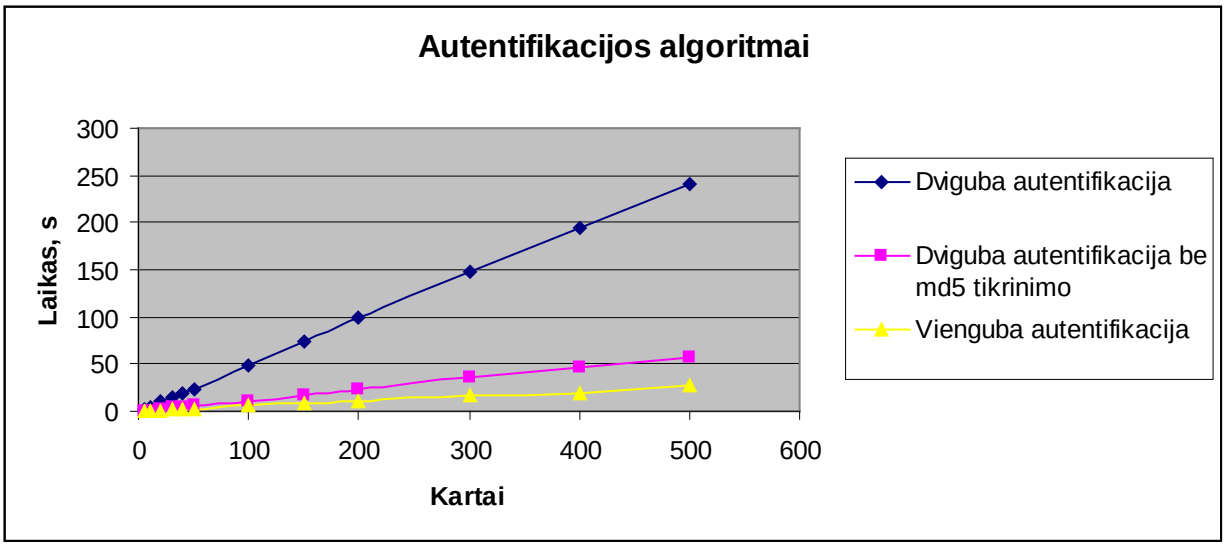
Pirmo eksperimento metu, buvo skaičiuojamos autentifikacijai sugaištas laikas imant kai autentifikacija vykdoma nuo 5 iki 500 kartų. Nagrinėti atvejai, kai autentifikacija vykdoma naudojant realizuotą dvigubą autentifikaciją su MD5 reikšmių tikrinimu, dvigubą autentifikaciją be MD5 reikšmių tikrinimo ir viengubą autentifikaciją. Dviguba autentifikacija be MD5 reikšmių tikrinimo buvo skaičiuota todėl, kad buvo siekiama pamatyti MD5 reikšmės tikrinimo įtaka esamame realizuotame algoritme. Kaip jau žinome šiuo metu realiose GRID sistemose naudoajantis proxy sertifikatais yra atliekama vienguba autentifikacija, kurios metu tikrinamas tik pats proxy sertifikatas.

Atliekant dvigubos autentifikacijos atvejį buvo atliekami 4.2. poskyryje aprašyti 1-3 etapai. Dvigubos autentifikacijos be MD5 reikšmės tikrinimo ekperimentui buvo vykdyti 4.2. poskiryje aprašytų 1 ir 3 etapų veiksmai, o viengubai autentifikacijai atlikti buvo vykdytas tik pirmas etapas. Kiekvieno atvejo metu buvo apjungti paminėtų etapų realizacijų kodai į vieną algoritmą, kurie buvo papildyti įterpiant ciklą, o atliktų ekperimentų visos laiko reikšmės pateiktos 2 lentelėje.

2 lentelė. Autentifikacijos eksperimento rezultatai

	Dviguba autentifikacija		Dviguba autentifikacija be MD5 tikrinimo		Vienguba autentifikacija	
Kartai	Laikas, s	Vidutinis laikas, s	Laikas, s	Vidutinis laikas, s	Laikas, s	Vidutinis laikas, s
5	3	0.60	1	0.20	0	0.00
10	5	0.50	1	0.10	1	0.10
20	10	0.50	2	0.10	1	0.05
30	15	0.50	4	0.13	2	0.07
40	19	0.48	5	0.13	3	0.08
50	24	0.48	6	0.12	3	0.06
100	49	0.49	11	0.11	6	0.06
150	74	0.49	17	0.11	8	0.05
200	99	0.50	23	0.12	11	0.06
300	148	0.49	35	0.12	17	0.06
400	194	0.49	47	0.12	20	0.05
500	240	0.48	57	0.11	27	0.05

Toliau lentelės rezultatai pateikiami grafiku, kur vaizduojama laiko priklausomybė sekundėmis nuo atliktų autentifikacijų skaičiaus, visais paminėtais algoritmais (žr. 20 pav.).



20 pav. Autentifikacijos algoritmų našumo grafikas

Pagal gautus rezultatus, kurie pateikti lentelėje ir grafike matome, kad realizuoto autentifikacijos algoritmo našumas yra stipriai prastesnis nei dabar naudojamas autentifikacijos mechanizmas GRID sistemose. Tačiau panaikinus MD5 tikrinimą matome, kad algoritmo našumas stipriai pagerėjo ir lyginant su vienguba autentifikacija jam alikti užtrunkama apie du kartus ilgiau. Todėl galima teigti, kad sertifikato siuntimas serveriui naudojant ssh ir tada MD5 reikšmės tikrinamas labiausiai pablogina realizuoto autentifikacijos mechanizmo našumą.

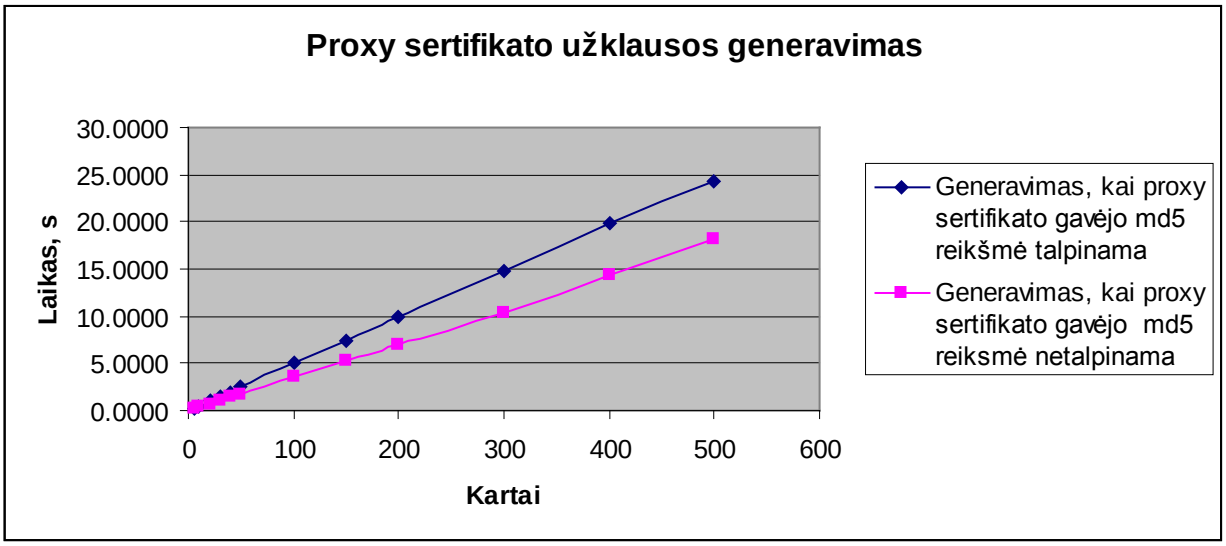
Proxy sertifikato prašymo užklausos generavimo testavimas

Antro eksperimento metu, buvo vykdomas proxy sertifikato užklausos generavimo testavimas. Vienu atveju, generuojant proxy sertifikato užklausą yra skaičiuojama serverio patikimo (sertifikatų centro išduoto) sertifikato MD5 reikšmė, kuri pridedama sertifikato užklausos „Subject“ lauke. Kitu atveju, kai generuojama standartinė proxy sertifikato prašymo užklausa. Taigi šiam eksperimentui atlikti buvo naudotas 4.2. poskyryje ketvirtame etape pateiktas šios užklausos generavimo kodas. Standartinės užklausos generavimui buvo naudotas tas pats kodas, tik buvo nevykdomas MD5 reikšmės skaičiavimas ir ji nebuvo pridedama „Subject“ lauke, o buvo pridedama tik standartinė „/proxy“ reikšmė. Visi testavimo rezultatai gauti įterpiant ciklą ir atitinkamą algoritmą vykdant nuo 5 iki 500 kartų, o gauti testavimo rezultatai pateikti 3 lentelėje.

3 lentelė. Proxy sertifikato užklausos generavimas

	Generavimas, kai proxy sertifikato gavėjo MD5 reikšmė talpinama			Generavimas, kai proxy sertifikato gavėjo MD5 reiksmė netalpinama		
Kartai	Laikas, ns	Laikas, s	Vidutinis laikas, s	Laikas, ns	Laikas, s	Vidutinis laikas, s
5	257772866	0.2578	0.0516	210721424	0.2107	0.0421
10	473055971	0.4731	0.0473	392615556	0.3926	0.0393
20	978994313	0.9790	0.0489	709614286	0.7096	0.0355
30	1485432273	1.4854	0.0495	1123048126	1.1230	0.0374
40	2004245361	2.0042	0.0501	1510494247	1.5105	0.0378
50	2549084147	2.5491	0.0510	1742352694	1.7424	0.0348
100	5049987099	5.0500	0.0505	3549740846	3.5497	0.0355
150	7344585957	7.3446	0.0490	5375693667	5.3757	0.0358
200	9852348502	9.8523	0.0493	7076256776	7.0763	0.0354
300	14872899159	14.8729	0.0496	10436669577	10.4367	0.0348
400	19872250913	19.8723	0.0497	14448691633	14.4487	0.0361
500	24370497821	24.3705	0.0487	18090005075	18.0900	0.0362

Toliau lentelės rezultatai pateikiami grafiku, kur vaizduojama laiko priklausomybė nuo sugeneruojų proxy sertifikato užklausų skaičiaus, kai papildoma MD5 reikšmė skaičiuojama ir patalpinama į užklausą ir kai ne. (žr. 21 pav.).



21 pav. Proxy sertifikato prašymo užklausų generavimo našumo grafikas

Pagal gautus rezultatus, kurie pateikti lentelėje ir grafike matome, kad proxy sertifikato prašymo generavimas laiko prasme abiem nagrinėtais atvejai skiriasi nežymiai. Nors MD5 reikšmės skaičiavimas ir jos talpinimas algoritmo našumą pablogina apie 38%, bet kadangi vieno proxy sertifikato generavimo užklausa užtrunka vidutiniškai 0.05 s, tai galime teigti, kad laiko prasme tai priimtina.

Proxy sertifikato užklausos pasirašymo testavimas

Trečio ekperimento metu buvo testuojama, kokį poveikį pasirašymo veiksmui padarys papildomas MD5 reikšmės tikrinimas. Kaip žinome, į proxy sertifikato prašymo užklausą turi būti patalpinama papildoma serverio patikimo sertifikato paskaičiuota MD5 reikšmė, todėl klientas prieš pasirašydamas šią užklausą privalo dar patikrinti ar serveris nebando proxy sertifikato gauti ne sau, t.y kokiai nors trečiai šaliai. Šiam eksperimentui atlikti buvo paimtas 4.2. poskyryje pentame etape pateiktas programos kodas, kuris papildomai buvo modifikuotas tam atvejui kai reikalingas MD5 reikšmės tikrinimas (žr. 22 pav.).

```
#!/bin/bash
START=$(date +%s%N)
for ((  i = 1 ;  i <= 500;  i++  ))
do
proxy_csr=exp1.csr
cert=donatocert.pem
certkey=donatocertkey.pem
proxycert=exp1.pem
md5reiksm=$(md5sum $cert | sed "s/"${cert}"//g" | sed "s/ //g")
ar_yra=$(openssl req -noout -subject -in $proxy_csr | grep $md5reiksm)
rez=$?
if [ $rez -eq 0 ]; then
keypass="pass:123456"
#slaptazodzio reikia tik kai pasiraso pats useris
openssl x509 -req -md5 -CAcreateserial -in $proxy_csr -days 1 -CA $cert -CAkey
$certkey -extfile
csr.conf -extensions v3_proxy -out $proxycert -passin $keypass
echo Done
else
echo Nera
fi
done
END=$(date +%s%N)
DIFF=$(( $END - $START ))
echo "It took $DIFF nanoseconds"
```

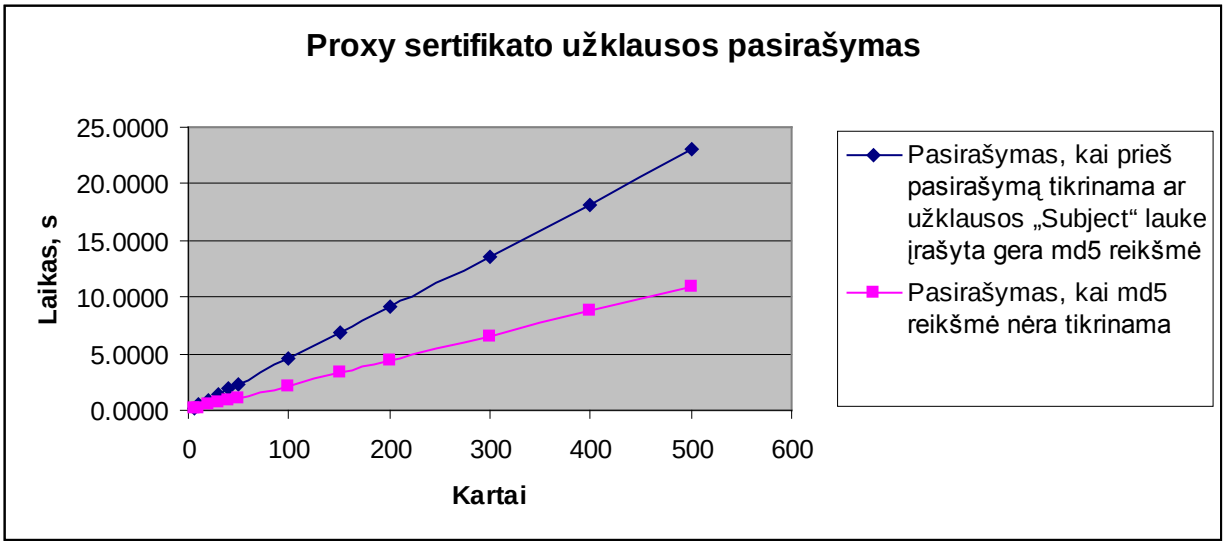
22 pav. Proxy sertifikato prašymo užklausos pasirašymas su MD5 rekšmės tikrinimu

Atliekant testavimą buvo bandoma atitinkamus algoritmus vykdyti nuo 5 iki 500 kartų ir buvo paimtos kiekvienu atveju gautos laiko reikšmės, kurios pateiktos 4 lenlelėje.

4 lentelė. Proxy sertifikato užklausos pasirašymas

	Pasirašymas, kai prieš pasirašymą tikrinama ar užklausos „Subject“ lauke įrašyta gera MD5 reikšmė			Pasirašymas, kai MD5 reikšmė nėra tikrinama		
Kartai	Laikas, ns	Laikas, s	Vidutinis vieno parašo laikas, s	Laikas, ns	Laikas, s	Vidutinis vieno parašo laikas, s
5	229077853	0.2291	0.0458	114140453	0.1141	0.0228
10	455433378	0.4554	0.0455	259349930	0.2593	0.0259
20	965754685	0.9658	0.0483	443215377	0.4432	0.0222
30	1393800795	1.3938	0.0465	649035878	0.6490	0.0216
40	1876882167	1.8769	0.0469	874375668	0.8744	0.0219
50	2329443072	2.3294	0.0466	1079049994	1.0790	0.0216
100	4571034792	4.5710	0.0457	2151321751	2.1513	0.0215
150	6785256230	6.7853	0.0452	3307730523	3.3077	0.0221
200	9102817786	9.1028	0.0455	4329977988	4.3300	0.0216
300	13593061448	13.5931	0.0453	6477207912	6.4772	0.0216
400	18109102742	18.1091	0.0453	8735206787	8.7352	0.0218
500	23134894888	23.1349	0.0463	10939856617	10.9399	0.0219

Toliau lentelės rezultatai pateikiami grafiku, kur vaizduojama laiko priklausomybė nuo proxy sertifikato užklausų parašų skaičiaus, kai vienu atveju tikrinama užklausos „*Subject*“ lauke įrašyta MD5 reikšmė , o kitu atveju ne (žr. 23 pav.).



23 pav. Proxy sertifikato prašymo užklausos pasirašymų grafikas

Pagal gautus rezultatus, kurie pateikti lentelėje ir grafike matome, kad proxy sertifikato prašymo pasirašymas, kai tikrinama MD5 reikšmė „*Subject*“ lauke užtrunka apytiksliai du kartus ilgiau nei kai ji yra netikrinama. Nors MD5 reikšmės tikrinimas algoritmo našumą pablogina, bet

kadangi vieno proxy sertifikato užklausos pasirašymas užtrunka vidutiniškai 0.05 s, tai galime teigti, kad laiko prasme tai priimtina.

Taigi atlikus eksperimentus matome, kad realizuoto sertifikatų centro modelio ir proxy sertifikatų valdymo mechanizmo našumai pablogėjo. Tačiau kalbant apie sertifikatų išdavimą reikia prisiminti, kad sertifikatai yra išduodami asmenims tik tada kai jie įrodo savo tapatybę fiziškai pateikdami dokumentus ir kadangi prieš sertifikato išdavimą reikalingas žmogaus įsikišimas, tai pasirašymo greitis nėra esminis šios funkcijos reikalavimas. Kadangi sertifikatų centrui svarbiausia yra apsauga, o tai mūsų atveju yra padidinta, galime teigti, kad buvo pasiekti norimi rezultatai. Toliau kalbant apie proxy sertifikatus reikia pastebėti, kad atliekant eksperimentus MD5 reikšmių tikrinimas buvo įvertintas dviejų eksperimentų metu, kai tai vykdoma dvigubos autentifikacijos metu ir proxy sertifikato pasirašymo metu. Pirmiausia dvigubos autentifikacijos metu, kliento sertifikato siuntimas ir MD5 reikšmės tikrinimas užtruko vidutiniškai 0.37 s (gauta iš dvigubos autentifikacijos su MD5 reikšmės tikrinimu ir be MD5 reikšmės tikrinimo vidutinių laiko reikšmių skirtumo). Toliau turint omenyje, kad viengubos autentifikacijos metu serveris su klientu apsikeičia sertifikatais, o tai vidutiniškai užtruko 0.055 s ir kad MD5 reikšmės tikrinimas užtruko vidutiniškai 0.024 s (gauta iš vidutinių laiko reikšmių skirtumo, kai prieš pasirašymą tikrinama MD5 reikšmė ir kai ji netikrinama), galime teigti, kad mūsų realizuotas sertifikato siuntimas serveriui naudojant ssh yra neefektyvus, nes 0.37 s >> 0.055 s + 0.024 s. Toliau apibendrinant proxy sertifikato testavimo rezultatus reikėtų paminėti jog žinoma, kad pateikta vartotojo užduotis į GRID tinklą resursų brokeryje gali užgaišti nuo 30 s iki 90 s, kol bus surasti laisvi resursai, po to pati užduotis darbiniame mazge gali būti vykdoma iki 48 h (priklauso nuo jos sudėtingumo). Iš to seka, kad nors naujo proxy sertifikato išdavimo našumas suprastėjo, tačiau tai neturėtų įtakoti bendro GRID tinklo darbo našumo. O naujas proxy sertifikatų valdymo mechanizmas leidžia padidinti saugumą, nes jis kontroliuoja šių sertifikatų išdavimą, naudojimą bei suteikia galimybę juos „atšaukti“ (jie automatiškai atmetami, jei yra atšauktas proxy sertifikato naudotojo CA išduotas sertifikatas).

5.3. Tolimesni darbai

Atlikus realizuoto sertifikatų centro ir proxy sertifikatų valdymo mechanizmo eksperimentus, galima suformuluoti tolimesnius darbus, kuriuos reikėtų atlikti norint toliau tęsti jau atliktus darbus. Įdiegtas sertifikatų centro modelis visiškai atitinka keliamus reikalavimus ir gali būti naudojamas tikrų sertifikatų išdavimui, tik sertifikatų centrui reikia gauti patvirtintą

sertifikatą leidžiantį sertifikatus pasirašinėti galiniams vartotojams. Kalbant apie naują proxy sertifikatų valdymo mechanizmą, darbe buvo atlikta prototipinė realizacija, kurią eksperimentuojant buvo surasta neefektyvių vietų, todėl jas reikėtų tobulinti. Toliau šį mechanizmą reikėtų diegti į realų GRID tinklą ir tęsti tolimesnius eksperimentus bei tyrinėjimus.

5.4. Išvados

1. Atlikto sertifikatų pasirašymo eksperimento rezultatuose matomas didelis sertifikatų centro darbo našumo sumažėjimas kai sertifikatų centro privatus raktas buvo patalpintas į HSM. Sertifikatų pasirašymo našumas sumažėjo $\approx$ 31.17 karto. Tačiau, kadangi sertifikatų centrui svarbiausia yra ne greitis, o apsauga, kuri buvo padidinta, galima teigti, kad pasiekti norimi rezultatai.
2. Atliekant proxy sertifikatų experimentus buvo pastebėta, kad sertifikato siuntimas serveriui antro etapo metu naudojant ssh yra neefektyvus. Taip pat autentifikacijos algoritmo našumo pablogėjimas neturėtų įtakoti bendro GRID darbo, nes pačios užduoties resursų paieškai ir jos vykdymui sugaištama daug daugiau laiko.

IŠVADOS

1. Atlikus GSI analizę išskiriami pagrindiniai saugą užtikrinantys elementai: sertifikatų centras ir proxy sertifikatai, nes jų pagalba yra užtikrinamas saugus darbas GRID tinkle. Kadangi šių elementų apsaugai turi būti skiriamas didžiausias dėmesys, todėl reikia naujų spendimo būdų esamoms problemoms spręsti.
2. Išanalizavus esamų problemų sprendimo būdus sertifikatų centro papildomai apsaugai užtikrinti galima pasinaudoti išanalizuotu elektroninės valdžios saugumo modeliu ir sertifikatų pasirašymo funkciją papildomai atskirti fiziškai nuo CA off-line dalies. O proxy sertifikatų problemų sprendimo būduose pasigesta metodų kaip kontroliuoti šių sertifikatų išdavimą. Todėl reikalingas naujas sprendimas leidžiantis proxy sertifikatus išduoti konkrečioms esybėms, t.y. patalpinant papildomą jo gavėjo sertifikato hash reikšmę.
3. Naujas sertifikatų centro modelis suprojektuotas remiantis GRID saugumo federacijos rekomendacijomis tik dar papildomai sertifikatų pasirašymo funkciją atskiriant fiziškai ir tai atliekant HSM įrenginyje. Proxy sertifikatų esamas formatas projektavimo metu papildytas į „Subject“ lauką talpinant proxy sertifikato gavėjo CA išduoto sertifikato MD5 reikšmę. O autentifikacija naudojantis šiais naujo formato sertifikatais reikalauja vykdyti dvigubą autentifikaciją.
4. Realizuotas sertifikatų centras atitinka visus GRID saugumo federacijos sertifikatų centrams keliamus reikalavimus bei papildomai užtikriną CA privataus rakto apsaugą nuo galimo sukompromitavimo. Realizuotos proxy sertifikatų dvigubos autentifikacijos metu atsiranda galimybė šiuos sertifikatus „atšaukti“. Jie automatiškai tampa negaliojančiais jei yra atšaukiami šiuos sertifikatus naudojančių esybių CA išduoti sertifikatai.
5. Atliktų eksperimentų rezultatai parodė, kad sertifikatų centro pasirašymo funkcijos ir proxy sertifikatų naudojimo mechanizmo našumai pablogėjo. Tačiau pagrįsta, kad šis kriterijus neturėtų žymiai įtakoti GRID darbo našumo, todėl galima teigti, jog pasiekti norimi rezultatai, nes šių elementų apsauga sustiprinta.

LITERATŪRA

1. ADAMS, C.; FARRELL, S. *Internet X.509 Public Key Infrastructure Certificate Management Protocols* [interaktyvus]. 1999, [žiūrėta 2010-03-21]. Prieiga per internetą: <http://www.ietf.org/rfc/rfc2510.txt>.
2. *Basic Certificate and Proxy Concepts* [interaktyvus]. 2007, [žiūrėta 2010-01-19]. Prieiga per internetą: <http://www.gridpp.ac.uk/deployment/users/certinfo.html>.
3. *Certificate authority* [interaktyvus]. [žiūrėta 2010-04-10]. Prieiga per internetą: <http://medlibrary.org/medwiki/Certificate_Authority>.
4. *Certificates* [interaktyvus]. [žiūrėta 2010-01-20]. Prieiga per internetą: <http://www-unix.globus.org/toolkit/docs/4.0/security/key-index.html#s-security-key-certificates>.
5. *Confidential Communication* [interaktyvus]. [žiūrėta 2010-01-20]. Prieiga per internetą: <http://www-unix.globus.org/toolkit/docs/4.0/security/key-index.html#s-security-key-confcommunication>.
6. CONG, S.; FANG, Z.; ZHANG, C.; LIU, X.; ZHANG, Y. *A CA-Based Security e-Government System* [interaktyvus]. 2006, [žiūrėta 2010-04-05]. Prieiga per internetą: <http://sf.library.lt:2098/stamp/stamp.jsp?tp=&arnumber=4019097>.
7. *Delegation Service* [interaktyvus]. [žiūrėta 2010-03-20]. Prieiga per internetą: <http://www.teragridforum.org/mediawiki/index.php?title=Delegation_Service>.
8. *Delegation, Single Sign-On and Proxy Certificates* [interaktyvus]. [žiūrėta 2010-01-20]. Prieiga per internetą: <http://www-unix.globus.org/toolkit/docs/4.0/security/key-index.html#s-security-key-delegation>.
9. *Digital Signature Algorithm* [interaktyvus]. [žiūrėta 2010-02-15]. Prieiga per internetą: <http://www.fact-index.com/d/di/digital_signature_algorithm.html>.
10. *Digital Signatures* [interaktyvus]. [žiūrėta 2010-04-05]. Prieiga per internetą: <http://www-unix.globus.org/toolkit/docs/4.2/4.2.1/security/key/#security-key-digitalsig>.
11. *EJBCA Enterprise PKI* [interaktyvus]. [žiūrėta 2010-04-10]. Prieiga per internetą: <http://ejbca.sourceforge.net/>.
12. FRANCKEVIČIUS, Arūnas. *Vartotojų autentifikavimas ir autorizavimas eLABa talpyklose* [interaktyvus]. 2007, [žiūrėta 2010-01-14]. Prieiga per internetą: <www.labt.lt/renginiai/20070529/20070529_Franckevicius.pps>.

13. *gLite...* [interaktyvus]. [žiūrėta 2010-04-07]. Prieiga per internetą: <http://glite.web.cern.ch/glite/>.
14. GROEP, D. *Authentication Profile for Classic X.509 Public Key Certification Authorities with secured infrastructure* [interaktyvus]. 2008, [žiūrėta 2010-03-25]. Prieiga per internetą: <http://www.eugridpma.org/guidelines/IGTF-AP-classic-4-2.pdf>.
15. *GT 4.0 Security: Key Concepts* [interaktyvus]. [žiūrėta 2010-01-20]. Prieiga per internetą: <http://www-unix.globus.org/toolkit/docs/4.0/security/key-index.html>.
16. *GT 4.0: Credential Management: SimpleCA* [interaktyvus]. [žiūrėta 2010-04-05]. Prieiga per internetą: <http://www-unix.globus.org/toolkit/docs/4.0/security/simpleca/>.
17. *GT 4.2.1: Security* [interaktyvus]. [žiūrėta 2010-02-27]. Prieiga per internetą: <http://www-unix.globus.org/toolkit/docs/4.2/4.2.1/security/>.
18. *GT Security (GSI)* [interaktyvus]. [žiūrėta 2010-03-23]. Prieiga per internetą: <http://www.globus.org/toolkit/security/>.
19. *Guidelines and Authentication Profiles: Classic X.509 CAs with secured infrastructure* [interaktyvus]. [žiūrėta 2010-03-25]. Prieiga per internetą: <http://www.eugridpma.org/guidelines/classic>.
20. *How PGP works* [interaktyvus]. [žiūrėta 2010-01-20]. Prieiga per internetą: <http://www.pgpi.org/doc/pgpintro/>.
21. *Installation* [interaktyvus]. [žiūrėta 2010-04-05]. Prieiga per internetą: <http://www.openca.org/~madwolf/ch03s03.html>.
22. *Introduction to OpenCA LABS* [interaktyvus]. [žiūrėta 2010-04-14]. Prieiga per internetą: <http://www.openca.org/>.
23. *Kam reikalinga autentifikacija?* [interaktyvus]. [žiūrėta 2010-02-20]. Prieiga per internetą: <http://www.ssc.lt/?name=menu&act=show&do=91,99,167&L=lt>.
24. *Koncepcija* [interaktyvus]. [žiūrėta 2010-03-23]. Prieiga per internetą: <http://lkz.mch.mii.lt/Zodynas/Visas.asp>.
25. *My Proxy Credential Management Service* [interaktyvus]. 2010, [žiūrėta 2010-02-27]. Prieiga per internetą: <http://grid.ncsa.illinois.edu/myproxy/>.
26. *Mutual Authentication* [interaktyvus]. [žiūrėta 2010-01-20]. Prieiga per internetą: <http://www-unix.globus.org/toolkit/docs/4.0/security/key-index.html#s-security-key-mutualauthentication>.
27. *Nolo's Plain-English Law Dictionary* [interaktyvus]. [žiūrėta 2010-02-10]. Prieiga per internetą: <http://www.nolo.com/dictionary/confidential-communication-term.html>.

28. PEARLMAN, L.; WELCH, V.; FOSTER, I.; KESSELMAN, C.; TUECKE, S. *A Community Authorization Service for Group Collaboration* [interaktyvus]. [žiūrėta 2010-03-28]. Prieiga per internetą: <http://www.globus.org/alliance/publications/papers/CAS_2002_Revised.pdf>.
29. *Public key certificate* [interaktyvus]. [žiūrėta 2010-02-10]. Prieiga per internetą: <http://technet.microsoft.com/en-us/library/cc737812(WS.10).aspx>.
30. *Public-key cryptography* [interaktyvus]. [žiūrėta 2010-01-20]. Prieiga per internetą: <http://www.tataelxsi.com/whitepapers/pub_key2.pdf?pdf_id=public_key_TEL.pdf>.
31. RAGHUNATHAN, S.; MIKLER, A. R.; COZZOLINO, C. Secure agent computation: X.509 Proxy Certificates in a multi-lingual agent framework. *Journal of Systems and Software* [interaktyvus]. 2005, [žiūrėta 2010-02-03]. Prieiga per internetą: <http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V0N-4BYRXDW-1&_user=5674488&_coverDate=02%2F15%2F2005&_rdoc=1&_fmt=high&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1327682387&_rerunOrigin=google&_acct=C000049863&_version=1&_urlVersion=0&_userid=5674488&md5=eb58dcb41ee0c01f2be963893bf46970>.
32. *Securing Private Keys* [interaktyvus]. [žiūrėta 2010-01-20]. Prieiga per internetą: <http://www-unix.globus.org/toolkit/docs/4.0/security/key-index.html#s-security-key-securingprivatekeys>.
33. *The MyProxy Certificate Authority* [interaktyvus]. [žiūrėta 2010-04-08]. Prieiga per internetą: <http://grid.ncsa.illinois.edu/myproxy/ca/>.
34. TUECKE, S.; WELCH, V.; ENGERT, D.; PEARLMAN, L.; THOMPSON, M. *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile* [interaktyvus]. 2004, [žiūrėta 2010-02-20]. Prieiga per internetą: <http://www.ietf.org/rfc/rfc3820.txt>.
35. WELCH, Von. *X.509 Proxy Certificates for Dynamic Delegation* [interaktyvus]. [žiūrėta 2010-02-20]. Prieiga per internetą: <http://middleware.internet2.edu/pki04/proceedings/proxy_certs.pdf>.
36. WRIGHT, D. J. *Public-key cryptography* [interaktyvus]. 1999, [žiūrėta 2010-01-20]. Prieiga per internetą: <http://www.math.okstate.edu/~wrightd/crypt/crypt-intro/node16.html>.
37. XIN, L.; OGAWA, M. Proxy Certificate Trust List for Grid Computing. *Information and Media Technologies* [interaktyvus]. 2006, [žiūrėta 2010-04-15]. Prieiga per internetą: <http://www.jaist.ac.jp/jinzai/Paper/JSSST05.pdf>.

38. ZHAO, S.; AGGARWAL, A.; KENT, R. D. *A Framework for Revocation of Proxy Certificates in a Grid* [interaktyvus]. 2007, [žiūrėta 2010-04-20]. Prieiga per internetą: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4287911>.

SANTRUMPŲ SĄRAŠAS

API *(Application Programming Interface)* – aplikacijų kūrimo aplinka.

CA *(Certificate Authority)* – sertifikatus išduodanti organizacija.

CAS *(Central Authentication Service)* – bendruomenės autentifikacijos paslauga.

CN *(Common Name)* – bendras sertifikato vardas.

CP *(Certificate Policy)* – sertifikatų centro politika.

CPS *(Certification Practice Statement)* – sertifikatų centro veiklos ataskaita.

CRA *(Certificate Registration Authority)* – sertifikatų registrų centras.

CRL *(Certificate Revocation List)* – atšauktų sertifikatų sąrašas.

DER *(Distinguished Encoding Rules)* – sertifikatų kodavimo formatas.

DN *(Distinguished Name)* – skiriamasis vardas.

EEC *(End Entity Certificate)* – galinio vartotojo sertifikatas išduotas CA.

EUGridPMA *(European Policy Management Authority for Grid Authentication)* – Europos GRID autentifikacijos politikų valdymo tarnyba.

FIPS *(Federal Information Processing Standard)* – federalinės informacijos apdorojimo standartas.

FQDN *(Fully Qualified Domain Name)* – kvalifikuotas domeno vardas.

GSI *(Globus Security Infrastructure)* **–** GRID saugumo infrastruktūra**.**

HSM *(Hardware Security Module)* – technine įranga paremtas saugumo modulis.

HTTP *(HyperText Transfer Protocol)* – standartinis protokolas skirtas HTML puslapių skelbimui ir skaitymui.

HTTPS (*HyperText Transfer Protocol Secure*) – tai šifruota versija HTTP protokolo, tekstas šifruojamas SSL protokolu.

IETF *(Internet Engineering Task Force)* – pagrindinius interneto standartus bei protokolus prižiūrinti organizacija.

IGTF (*The International Grid Trust Federation*) – tarptautinė gridų pasitikėjimo federacija.

IP *(Internet Protocol)* – interneto protokolas.

LDAP *(Lightweight Directory Access Protocol)* – protokolas naudojamas priėjimui prie kataloginių duomenų.

MD5 *(Message-Digest Algorithm 5)* – žinutės santraukos algoritmas.

MPS *(MyProxy Server)* – MyProxy serveris.

NCSA *(National Center For Supercomputing Applications)* – Nacionalinis superkompiuterių centras.

NERSC *(National Energy Research Scientific Computing Center)* – Nacionalinis energijos tyrinėjimų skaičiavimų centras.

NIS *(Network Information Service)* – kliento-serverio katalogų tarnybos protokolas.

OCSP *(Online Certificate Status Protocol)* – interneto protokolas naudojamas gauti X.509 skaitmeninio sertifikato atšaukimo statusą.

OID *(Object Identifier)* – objekto identifikatorius.

OU *(Organizational Unit)* – organizacijos padalinys.

PC *(Proxy Certificate)* – proxy sertifikatas.

PCTL *(Proxy Certificate Trust List)* – patikimų proxy sertifikatų sąrašas.

PEM *(Privacy Enhanced Mail)* – užkoduotas sertifikatas DER formatu.

PKI *(Public Key Infrastructure)* – viešo rakto infrastruktūra.

RA *(Registration Authority)* – sertifikatų registrų centras.

RDBVS *(Relational Database Management Systems)* – reliacinės duomenų bazių valdymo sistemos.

RFC *(Request for Comments)* – IETF paskelbti aiškinamieji dokumentai, standartai.

SAML *(Security Assertion Markup Language)* – XML paremtas standartas skirtas autentifikacijos ir autorizacijos duomenų apsikeitimui tarp domenų.

SOAP (*Simple Object Access Protocol*) – protokolas naudojamas siuntimui visokių tipų duomenų tarp programų.

SSL *(Secure Sockets Layer)* – kriptografinis protokolas, skirtas informacijos, sklindančios internete apsaugojimui šifruojant.

TLS *(Transport Layer Security)* **–** TLS protokolas užtikrina slaptumą ir integralumą duomenų siuntime.

URI *(Uniform Resource Identifier)* – trumpas tekstas identifikuojantis šaltinį internete.

URL *(Uniform Resource Locator)* – globalus dokumentų ar kitų resursų adresas internete.

VO (*Virtual Organization*) – virtuali organizacija.

WS *(Web Services)* – internetinės paslaugos.

WS A&A *(WS Authentication & Authorization)* – interneto paslaugomis paremta autentifikacija ir autorizacija.

WSRF *(Web Services Resource Framework)* – OASIS paskelbtos specifikacijos interneto paslaugoms.

WS-Trust *(Web Services Trust Language)* – saugus pranešimų apsikeitimo mechanizmas.

X.509 – sertifikatas aprašantis asimetrinių kriptografinių algoritmų naudojimą pasirašant elektroniniu parašu.

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

1 PRIEDAS

PHP skriptas vykdomas serverio pusėje, kurio pagalba vykdoma realizacijoje aprašyta 4.2. poskyryje ketvirtame etape, t. y. genproxy bash skripto iškvietimas bei proxy sertifikato užklausos siuntimas vartotojui (genproxy.php).

```
genproxy.php
<?php
//////////////////////////
$kas = "proxycert";
$kam = "serveriscert";
$naujas = "serverisproxy";
$IP = "192.168.72.135";
$user = "klientas";
$pass = "klientas";
//////////////////////////
$command = "sh /home/donatas/proxy/test/genproxy ".$kas." ".$kam." ".$naujas;

exec($command, &$output);
echo "<br>";
foreach ($output as $out){
  echo $out;
  echo "<br>";
}
//siunciama klientui
if(!($con = ssh2_connect($IP, 22)))
{
echo "Klaida: Negalima sukurti susijungimo";
}
else
{
// Meginama prisijungti naudojant prisijungimo varda ir slaptazodi vartotojo
if(!ssh2_auth_password($con, $user, $pass))
{
echo "Klaida: Autentifikacija neatlikta";
} else
{
// Viskas gerai, prisijungta

//siunciam proxy sertifikato requesta
$adresas = "/var/www/proxy/".$naujas.".csr";
$adresas2 = "/home/klientas/proxy/test/serverisproxy.csr";

if( ssh2_scp_send($con, $adresas, $adresas2, 0777))
{
echo "OK";
}
}
}
?>
```

2 PRIEDAS

Forma, kurios pagalba įvedami pradiniai duomenys, t. y. sertifikatų failų vardai ir vartotojo tikro sertifikato slaptažodis (index.html).

```
index.html:
<html>
<body>
<br>
<div align= "center">
<form action="start.php" method="post">
<table style="color:blue" border = "1" bordercolor = "red" >
<tr>
<td width ="250">
Proxy sertifikato vardas:
</td>
<td>
<input type="text" name="proxycert" />
<br>
</td>
</tr>
<tr>
<td width= "250">
Vartotojo sertifikato vardas:
</td>
<td>
<input type="text" name="usercert"/>
<br>
</td>
</tr>
<tr>
<td width="250">
Vartotojo privataus rakto slaptazodis:
</td>
<td>
<input type="text" name="usercertpass" />
<br>
</td>
</tr>
<tr>
<td colspan = "2" align="center">
<input type="submit" value="Kurti proxy" />
</td>
</tr>
</table>
</form>
</div>

</body></html>
```

PHP skriptas, kuriuo vykdomi realizacijoje aprašyti visi penki etapai ir kurio pagalba stabdomas darbas jei kuriame nors etape įvyksta klaida (start.php).

```
start.php
<html>
<head>
<title>Proxy sertifikato generavimas</title>
</head>
<body>
<?php
//////////////////////////////
$usercert = $_POST['usercert'];
$usercertpass = $_POST['usercertpass'];
$proxycert = $_POST['proxycert'];
$IP = "192.168.72.132";
$user = "donatas";
$pass = "donatas";
//////////////////////////////
echo "-----------------------------------------------------------<br>";
echo "| Autentifikacija proxy sertifikatu |<br>";
echo "-----------------------------------------------------------<br>";

$output = "";
$command = "sh connect_server ".$proxycert;
exec($command, &$output);

$i =0;
foreach ($output as $line){
  if($i==2) $result = $line;
  echo $line;
  echo "<br>";
  $i=$i+1;
}

if($result=="Verify return code: 0 (ok)")
{
echo "Autentifikacija proxy sertifikatu baigta<br>";
echo "-----------------------------------------------------------<br>";
echo "-------------------------------------------<br>";
echo "| MD5 reiksmiu tikrinimas |<br>";
echo "-------------------------------------------<br>";

//////////////////////md5

if(!($con = ssh2_connect($IP, 22)))
{
echo "Klaida: Negalima sukurti susijungimo";
}
else
{
// Meginama prisijungti naudojant prisijungimo varda ir slaptazodi vartotojo
if(!ssh2_auth_password($con, $user, $pass))
{
//echo "Klaida: Autentifikacija neatlikta";
} else
{
// Viskas gerai, prisijungta

//kopijuojam testa i nutolusia masina
$adresas = $proxycert.".pem";
$adresas2 = "/home/donatas/proxy/test/proxycert.pem";
$adresas1 = $usercert.".pem";
```

```
$adresas12 = "/home/donatas/proxy/test/usercert.pem";

if( ssh2_scp_send($con, $adresas, $adresas2, 0644)
    &&
    ssh2_scp_send($con, $adresas1, $adresas12, 0644))
{
echo "Vartotojo sertifikatas nusiustas<br><br>";

$html = implode('', file('http://192.168.72.132/md5.php'));
echo $html;

$length = strlen($html);
$MD5reismes = $html[$length-2].$html[$length-1];
echo "<br>";
}else { echo "Vartotojo sertifikato nusiusti nepavyko<br>";}

}
}
}
else
echo "Autentifikacija proxy sertifikatu nesekminga<br>";

if($result=="Verify return code: 0 (ok)" && $MD5reismes == "OK")
{

echo "MD5 reiksmiu tikrinimas baigtas<br>";
echo "-----------------------------------------------------<br>";

echo "-----------------------------------------------------<br>";
echo "| Autentifikacija tikru sertifikatu |<br>";
echo "-----------------------------------------------------<br>";
$output2 = "";
$command2 = "sh connect_server2 ".$usercert." ".$usercertpass;
exec($command2, &$output2);

$k=0;
foreach ($output2 as $line){
  if($k==2) $result2 = $line;
  echo $line;
  echo "<br>";
  $k=$k+1;
}

if($result2=="Verify return code: 0 (ok)")
{
echo "Autentifikacija tikru sertifikatu baigta<br>";
echo "-----------------------------------------------------<br>";

}
else
echo "-----------------------------------------------------<br>";
}
//////////////////////////////////////

if($result2=="Verify return code: 0 (ok)")
{
echo "-------------------------------------------------------------<br>";
echo "| Naujo proxy sertifikato sukurimas |<br>";
echo "-------------------------------------------------------------<br>";
```

```
$html2 = implode('', file('http://192.168.72.132/genproxy.php'));
echo $html2;
echo "<br>";
$length = strlen($html2);
$Nusiusta = $html2[$length-2].$html2[$length-1];

if ($Nusiusta == "OK"){
$request = "serverisproxy";
$command3 = "sh signproxy ".$request." ".$proxycert;
echo "<br>";
$output2 = "";

exec($command3, &$output2);

$k=0;
foreach ($output2 as $line){
  echo $line;
  echo "<br>";
  $k=$k+1;
}

////////
if(!($con = ssh2_connect($IP, 22)))
{
echo "Klaida: Negalima sukurti susijungimo";
}
else
{
// Meginama prisijungti naudojant prisijungimo varda ir slaptazodi vartotojo
if(!ssh2_auth_password($con, $user, $pass))
{
echo "Klaida: Autentifikacija neatlikta";
} else
{
// Viskas gerai, prisijungta

//kopijuojam proxy sertifikata
$adresas = "/var/www/cert/".$request.".pem";
$adresas2 = "/var/www/proxy/".$request.".pem";

if( ssh2_scp_send($con, $adresas, $adresas2, 0644))
{
echo "<br>";
echo "Pasirasytas sertifikatas nusiustas serveriui<br>";
echo "-------------------------------------------";
}
}
}
//////////////////////////////////////

}
}
//////////////////////////////////////
echo "<br>";
echo "Pabaiga";
?>
</body>
</html>
```