KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Povilas Kanapickas

SAUGUS IR PATOGUS SLAPTAŽODŽIŲ GENERAVIMO IR SAUGOJIMO METODAS

Baigiamasis magistro darbas

Vadovas

Prof. dr. Algimantas Venčkauskas

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

KOMPIUTERIŲ KATEDRA

SAUGUS IR PATOGUS SLAPTAŽODŽIŲ GENERAVIMO IR SAUGOJIMO METODAS

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Vadovas

(parašas) Prof. dr. Algimantas Venčkauskas

(data)

Recenzentas

(parašas) Doc. dr. Nerijus Morkevičius

(data)

Projektą atliko

(parašas) Povilas Kanapickas

(data)

KAUNAS, 2017


ktu
1922

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Povilas Kanapickas

(Studento vardas, pavardė)

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Saugus ir patogus slaptažodžių generavimo ir saugojimo metodas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 ___ m. _____ ____ d.

Kaunas

Patvirtinu, kad mano **Povilo Kanapicko** baigiamasis projektas tema „Saugus ir patogus slaptažodžių generavimo ir saugojimo metodas” yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

_____ _____

(vardą ir pavardę įrašyti ranka) *(parašas)*

Kanapickas, P. Saugus ir patogus slaptažodžių generavimo ir saugojimo metodas. Magistro baigiamasis projektas / vadovas prof. Algimantas Venčkauskas; Kauno technologijos universitetas, Informatinkos fakultetas, Kompiuterių katedra. Kaunas, 2017. 70 psl.

SANTRAUKA

Didžioji dauguma interneto puslapių ir programų naudoja slaptažodį kaip pagrindinę priemonę vartotojo autentifikavimui. Norint naudoti slaptažodžius atsižvelgiant į saugumo rekomendacijas, jie yra itin nepatogūs vartotojui. Šį trūkūmą galima išspręsti naudojant slaptažodžių tvarkykles.

Darbe yra sukurtas hibridinis slaptažodžių tvarkymo metodas, kuris leidžia didžiąją dalį slaptažodžių generuoti ši pagrindinio slaptažodžio, o likusius išsaugoti slaptažodžių duomenų bazėje. Metodas paremtas kliento-serverio architektūra. Slaptažodžiai yra suskirstomi į rinkinius, prieiga prie kurių yra ribojama naudojant papildomą autentifikavimo priemonę: prieigos kodus. Piktavalis negali prieiti prie rinkinių, kurių prieigos kodų nežino. Slaptažodžių saugykla yra šifruojama taip, kad tam tikrai informacijai šifruoti yra reikalinga ta pati informacija, kaip ir autentifikavimui prie slaptažodžių tvarkyklės serverio. Atliktas sukurto prototipo tyrimas, kuriame vertintas slaptažodžių tvarkyklės atsparumas bandymams nutekinti joje esančią informaciją bei tvarkyklės sparta.

Kanapickas, P. Secure and convenient method of password generation and storage. Master's thesis / supervisor Prof. Algimantas Venčkauskas; Department of Computer Science, Faculty of Informatics, Kaunas University of Technology. – Kaunas, 2017. – 70 p.

SUMMARY

Passwords are used as the primary means of authentication in a large majority of internet services and various programs. Passwords are very inconvenient if the best usage practices are taken into account. This shortcoming can be solved by using password managers.

This work describes a hybrid method of password management, which generates passwords from a master passwords while allowing to simply store any password for which this scheme does not work. Method uses client-server architecture. The passwords are grouped into isolated clusters each of which can only be accessed by additional authentication information called access codes. An adversary can't access information in clusters that use access codes he does not have. The password store is encrypted in such a way that decryption requires the same information that would be required if the information is accesssed via authenticating with the password manager. An assessment of the implemented prototype was done, in which the speed of the password manager and resistance against attempts to access the stored information were assessed.

1. TURINYS

LENTELIŲ SĄRAŠAS....9

PAVEIKSLŲ SĄRAŠAS....10

TERMINŲ IR SUTRUMPINIMŲ SĄRAŠAS....11

ĮVADAS....12

1. PRIEMONIŲ SLAPTAŽODŽIAMS TVARKYTI ANALIZĖ....14

1.1. Saugių slaptažodžių naudojimo problema....14

1.2. Įprastinio autentifikavimo slaptažodžiais bruožai....14

1.3. Slaptažodžių tvarkyklės....16

1.3.1 Pritaikomumas autentifikavimo saugumo problemai spręsti....16

1.3.2 Slaptažodžių saugojimo metodai....17

1.3.3 Slaptažodžių generavimo metodai....20

1.3.4 Egzistuojančios slaptažodžių tvarkyklės....25

1.4. Išvados....29

2. SIŪLOMAS SLAPTAŽODŽIŲ TVARKYKLĖS MODELIS....31

2.1. Slaptažodžių tvarkyklei keliami tikslai ir jų prioritetai....31

2.2. Konceptuali slaptažodžių tvarkyklės schema....32

2.3. Reikalavimai slaptažodžių tvarkyklei....33

2.4. Slaptažodžių tvarkyklės architektūra....34

2.4.1 Slaptažodžių tvarkyklės klientas....36

2.4.2 Slaptažodžių tvarkyklės serveris....37

2.5. Slaptažodžių saugos modelis....38

2.6. Autentifikavimas tarp slaptažodžių tvarkyklės kliento ir serverio....43

2.7. Slaptažodžių generavimas iš pagrindinio slaptažodžio....46

2.8. Slaptažodžių perdavimas jų reikalaujančioms programoms....47

2.9. Išvados....49

3. SLAPTAŽODŽIŲ TVARKYKLĖS REALIZACIJA....50

3.1. Slaptažodžių tvarkyklės realizacijos struktūra..... 50
3.2. Duomenų bazės struktūra..... 52
3.3. Išvados..... 56
4. SLAPTAŽODŽIŲ TVARKYKLĖS TYRIMAS..... 57
4.1. Slaptažodžių tvarkyklės atsparumo saugumo grėsmėms įvertinimas..... 57
4.2. Slaptažodžių tvarkyklės spartos įvertinimas..... 62
4.3. Rezultatų apibendrinimas..... 67
5. IŠVADOS..... 68
LITERATŪRA..... 70

LENTELIŲ SĄRAŠAS

1 lentelė: Slaptažodžių tvarkyklių palyginimas....28
2 lentelė: Informacijos kiekis, kurį reikia prisiminti vartotojui....40
3 lentelė: Rakto-reikšmės duomenų bazės struktūra....54
4 lentelė: Keliamų grėsmių įgyvendinimo sudėtingumas piktavaliams....59
5 lentelė: Konkrečių atakų prieš slaptažodžių tvarkyklę nutekinama informacija....61
6 lentelė: Tyrime naudotos aparatinės ir programinės įrangos aprašas....62

PAVEIKSLŲ SĄRAŠAS

1 pav. Konceptuali slaptažodžių tvarkyklės schema 33
2 pav. Slaptažodžių tvarkyklės architektūra 35
3 pav. Slaptažodžių saugos sprendimo schema iš vartotojo pusės 39
4 pav. Slaptažodžių gavimo iš tvarkyklės veismo schema 41
5 pav. Slaptažodžių tvarkyklės duomenų saugyklos šifravimo schema 43
6 pav. Kliento-serverio autentifikavimo schema 45
7 pav. Sukurtos slaptažodžių tvarkyklės struktūra 50
8 pav. Prieigos prie slaptažodžių žurnalo šifravimo schema 55
9 pav. Slaptažodžio tarpinės reikšmės generavimo trukmė naudojant PBKDF2-HMAC-SHA256 generavimo funkciją 63
10 pav. Slaptažodžio tarpinės reikšmės generavimo trukmė naudojant scrypt generavimo funkciją 64
11 pav. Paskyros slaptažodžio generavimo trukmė naudojant PBKDF2-HMAC-SHA256 generavimo funkciją 64
12 pav. Paskyros slaptažodžio generavimo trukmė naudojant scrypt generavimo funkciją 65
13 pav. Išsaugoto slaptažodžio gavimo operacijos trukmė priklausomai nuo duomenų bazės dydžio 65
14 pav. Slaptažodžio gavimo trukmė priklausomai nuo tinklo delsos 66
15 pav. Slaptažodžio generavimo trukmė priklausomai nuo tinklo delsos 66

TERMINŲ IR SUTRUMPINIMŲ SĄRAŠAS

Autentifikavimas – objekto tapatybės patikrinimas

Grubios jėgos metodas (angl. *brute force*) – slaptažodžių spėjimo metodas, atliekamas perrenkant visas įmanomas simbolių kombinacijas

HMAC (angl. *hash-based message authentication code*) – kriptografine maišos funkcija paremtas autentifikavimo protokolas

HTTP (angl. *hypertext transfer protocol*) – informacijos perdavimo TCP tinklu protokolas

HTTPS – HTTP paremtas informacijos perdavimo protokolas, kuris naudoja SSL ryšio šifravimui

Kelių dedamųjų autentifikavimas (angl. *multi-factor authentication*) – autentifikavimo būdas, kurio metu naudojama daugiau nei viena autentifikavimo priemonė

Kriptografinė maišos funkcija (angl. *cryptographic hash function*) – algoritmas, apibrėžiantis vienkryptę bet kokio duomenų kiekio transformaciją į fiksuoto dydžio duomenų kiekį

PBKDF (angl. *password-based key derivation function*) – kriptografinių raktų generavimo iš slaptažodžių algoritmas

scrypt – kriptografinių raktų generavimo iš slaptažodžių algoritmas

SSL (angl. *secure sockets layer*) – protokolų rinkinys, skirtas ryšių interneto tinkle autentiškumo, konfidencialumo ir integralumo užtikrinimui

TOTP (angl. *time-based one-time password algorithm*) – vienkartinių slaptažodžių algoritmas

ĮVADAS

Šis darbas priklauso informacijos ir informacinių technologijų saugos studijų programai.

Darbo problematika ir aktualumas

Didžioji dauguma interneto puslapių ir programų naudoja slaptažodį kaip pagrindinę priemonę vartotojo autentifikavimui. Nors yra sukurta įvairių už slaptažodžius geresnių autentifikavimo priemonių, kurios neturi slaptažodžiams priskiriamų trūkumų, šių modernių priemonių skvarba yra palyginti nedidelė. Taip pat, šios priemonės nėra universalios ir paprastai skirtingos kiekvienam puslapiui, todėl tikėtina, kad nepaisant savo trūkumų, slaptažodžiai ir toliau išliks itin populiari autentifikavimo priemonė.

Augant kompiuterių skaičiavimo greičiui, ilgėja ir rekomenduojamas saugaus slaptažodžio ilgis. Šiuo metu jis yra toks, kad žmogus sunkiai gali atsiminti daugiau nei keletą rekomenduojamo sudėtingumo skirtingų slaptažodžių, taigi žmonės naudoja tuos pačius slaptažodžius skirtingoms paskyroms. Tai yra didelė problema, kadangi piktavaliams įsilaužus į bet kurį iš vartotojo naudojamų paslaugų tiekėjų, jie dažnai gali prieiti ir prie autentifikavimui naudojamos informacijos, iš kurios galima išskaičiuoti vartotojų naudojamus slaptažodžius. Šią informaciją piktavaliai gali panaudoti kitų paslaugų tiekėjų paskyrų nulaužimui, kadangi vartotojai paprastai naudoja tokius pačius, arba panašius vartotojo identifikatorius. Vienas iš būdų tų pačių slaptažodžių naudojimo skirtingose paskyrose problemos sprendimui yra slaptažodžių tvarkyklės, kuri galėtų sumažinti autentifikavimui reikalingos vartotojo atminties kiekį, sukūrimas.

Darbo metu bus tiriamas ir įgyvendinamas didelių atminties resursų nereikalaujantis slaptažodžių generavimo, naudojant pagrindinį slaptažodį, metodas. Didelis dėmesys kuriant metodą bus skirtas vartotojo patogumui – į sistemą bus galima saugoti jau esamus slaptažodžius jų nekeičiant.

Darbo tikslas ir uždaviniai

Tikslas: sukurti saugų ir patogų metodą skirtingiems slaptažodžiams generuoti ir slaptažodžių saugojimui.

Uždaviniai:

- Susipažinti su egzistuojančiais slaptažodžių saugojimo ar generavimo metodais
- Išanalizuoti slaptažodžių generavimo iš pagrindinio slaptažodžio metodą tiriančią literatūrą
- Pasiūlyti priemones algoritmo ar metodo įgyvendinimo lygmenyje, padedančias apsaugoti turimus slaptažodžius nuo nutekėjimo

- Pasiūlyti praktines priemones, padedančias apsaugoti turimus slaptažodžius nuo nutekėjimo
- Praktiškai realizuoti pasiūlytą slaptažodžių saugojimo ir generavimo iš pagrindinio slaptažodžio metodą ir konfidencialią informaciją nuo nutekėjimo apsaugančias priemones
- Atlikti sukurto metodo kiekybinį ir kokybinį įvertinimą.

Darbo struktūra

Darbe yra keturi pagrindiniai skyriai.

Pirmame skyriuje yra atliekama slaptažodžių naudojimo problemos analizė. Apžvelgiami egzistuojantys slaptažodžių generavimo iš pagrindinio slaptažodžio metodai. Analizuojami ir palyginami egzistuojantys slaptažodžių tvarkymo produktai, pateikiamos bendros išvados ir rekomendacijos.

Antrame skyriuje yra pateikiamas siūlomo slaptažodžių saugojimo ir generavimo metodo aprašas. Detalizuojama sistemos architektūra, naudojami protokolai, jų struktūrinės schemos.

Trečiame skyriuje yra aprašomos sudaryto metodo programinio modelio realizacijos detalės.

Ketvirtame skyriuje aprašomas sukurto metodo tyrimas. Sudaryta slaptažodžių tvarkyklė ištiriama tiek kokybiškai, tiek kiekybiškai. Metodo kokybiniame tyrime yra įvertinamas metodo atsparumas įvairioms saugumo grėsmėms, jis palyginamas su pirmame skyriuje apžvelgtomis egzistuojančiomis slaptažodžių tvarkyklėmis. Tvarkyklės kiekybiniame tyrime yra įvertinama tvarkyklės greitaveika.

Darbo pabaigoje yra pateikiamos viso projekto apibendrinimas ir naudotos literatūros sąrašas.

1. PRIEMONIŲ SLAPTAŽODŽIAMS TVARKYTI ANALIZĖ

1.1. Saugių slaptažodžių naudojimo problema

Slaptažodžius kaip autentifikavimo metodą naudoja didžioji dauguma interneto svetainių. Pagal saugumo specialistų rekomendacijas idealiu atveju kiekvienam puslapiui turi būti naudojamas unikalus, atitinkamo ilgio ir stiprus slaptažodis [1]. Deja, žmogus ypač sunkiai gali atsiminti daugiau nei keletą rekomenduojamo sudėtingumo skirtingų slaptažodžių [2]. Šią prieštarą dauguma vartotojų išsprendžia naudodami tą patį slaptažodį kelioms svetainėms. Toks sprendimas yra itin nesaugus, kadangi įsilaužimo į bet kurią iš naudojamų svetainių atveju, egzistuoja įsilaužimo į visas paskyras, naudojusias tą patį slaptažodį, galimybė. Ši rizika yra ganėtinai didelė, kadangi įsilaužimai net į dideles ir, manytina, gerą saugumo sprendimą turinčias svetaines yra palyginti dažnas reiškinys [3].

Autentifikavimo saugumo, kai naudojami slaptažodžiai, problema tikėtina nebus sprendžiama iš esmės keičiant autentifikavimo schemą. Nors saugumo atžvilgiu slaptažodžiai yra aplenkiami tokių sprendimų, kaip biometrinis ar kelių dedamųjų (angl. *multi-factor*) autentifikavimas, slaptažodžiai išlieka populiarūs dėl savo paprastumo, patogumo ir universalumo. Kadangi šie aspektai yra vieni pagrindinių, į kuriuos atsižvelgia vartotojai, tikėtina, jog slaptažodžiais paremtas autentifikavimas bus plačiai naudojamas ir ateityje [4].

Mažiausiai išteklių ir tik kliento pusėje pakeitimų kliento pusėje reikalaujantis būdas pagerinti slaptažodžių saugumą yra slaptažodžių tvarkyklės. Jų yra įgyvendinta palyginti daug, bet visoms joms galima priskirti vieną bendrą bruožą: dalis autentifikavimui naudojamos informacijos yra saugoma pačioje tvarkyklėje, o vartotojui reikia atsiminti gerokai mažiau informacijos. Tokiu būdu, vartotojas gali naudoti slaptažodžius, atitinkančius saugumo specialistų rekomendacijas be įprastai tam priskiriamų sunkumų.

1.2. Įprastinio autentifikavimo slaptažodžiais bruožai

Autentifikavimas slaptažodžiais priskiriamas „kažkas, ką žinai“ autentifikavimo metodų grupei ir nepasižymi jokiais likusių dviejų, „kažkas, ką turi“ ir „kažkas, kas esi“ autentifikavimo metodų grupių požymiais. Slaptažodžių naudojimo ciklas susideda iš 2 dėmenų:

– registracija, kai vartotojui sukuriama paskyra, priskiriamas prisijungimo vardas ir paprastai jo paties sugalvotas slaptažodis. Registracija atliekama pirmą kartą jungiantis prie svetainės. Svėtainė išsaugo registracijos metu suteiktą informaciją savo duomenų bazėje.

– prisijungimas, svetainei pateikiant vartotojo vardą ir slaptažodį. Svetainė sutikrina pateikta informaciją su išsaugota registracijos metu. Jei pateiktas ir išsaugotas slaptažodžiai sutampa,

prisijungiantis asmuo yra laikomas paskyros savininku.

Autentifikavimo slaptažodžiais metu slaptažodis yra vienintelė informacija, pagal kurią atliekamas autentifikavimas, todėl sistemos saugumas yra nusakomas slaptažodžio sudėtingumo ir potencialių bandymų jį atspėti greičio santykiu. Jei autentifikavimo sistemą atakuojanti šalis gali slaptažodį atspėti ar kokiu nors būdu patikrinti visus įmanomus slaptažodžio variantus, sistema tampa neveiksni ir atakuojanti šalis gali apsimesti slaptažodžio savininku ir gauti kelią prie ribotos prieigos resursų.

Slaptažodžiais paremtų autentifikavimo sistemų atakos paprastai yra atliekamos dviem būdais. Pirmasis, ir pats paprasčiausias, vadinamas grubios jėgos (angl. *brute force*) metodu. Šios atakos metu perrenkamos visos įmanomos slaptažodyje galimų naudoti simbolių kombinacijos, tikrinant, ar gautas slaptažodis yra teisingas. Paprastai pirmiausia yra tikrinami simboliai, kurie statistiškai yra naudojami dažniausiai, o retai naudojamiems simboliams, tokiems kaip skaičiai ar skyrybos ženklai, yra priskiriamas mažesnis statistinis svoris. Antrasis metodas vadinamas žodyno (angl. *dictionary*) ataka. Jos metu, skirtingai nuo grubios jėgos metodo, tikrinami tik tam tikri slaptažodžiai iš atakuojančios šalies sudaryto slaptažodžių žodyno. Į jį paprastai yra įtraukiami įvairiose kalbose naudojami žodžiai ir jau žinomi, paviešinti slaptažodžiai iš kada nors praeityje nulaužtų slaptažodžių duomenų bazių. Kartais abu atakos metodai yra kombinuojami: tikrinami ne tik žodyne esantys žodžiai, bet ir įvairios jų modifikacijos keičiant raides kitomis, prijungiant atsitiktinius simbolius ar žodžius [5].

Saugus slaptažodis turi būti atsparus minėtiems atakų tipams. Norint apsisaugoti nuo grubios jėgos metodo slaptažodis turi būti sudaromas iš kuo įvairesnių simbolių: didžiųjų ir mažųjų raidžių, skaičių ir skyrybos ženklų. Slaptažodis turi būti ilgas, idealiu atveju gerokai ilgesnis nei būtų galima nulaužti didelius resursus turinčios potencialios atakuojančios šalies. Taip pat, slaptažodyje turi būti naudojama kuo mažiau kalboje vartojamų žodžių, net jei žodžiai yra modifikuojami keičiant simbolius į kitus. Šie aspektai lemia, kad saugūs slaptažodžiai yra ne tik sunkiai prisimenami, bet ir sugalvojami.

Tai, kad saugūs slaptažodžiai yra sudėtingi naudoti, pagrindžia faktas, kad nepaisant plačiai prieinamų nurodymų vartotojams pasirinkti saugius slaptažodžius, dauguma vartotojų į juos neatsižvelgia. Įvairūs tyrimai identifikuoja silpnus vartojamus slaptažodžius tiek vykdant vidinius saugumo auditus įvairiose svetainėse, tiek analizuojant viešai prieinamas paviešintas slaptažodžių duomenų bazes iš nulaužtų svetainių [6][7]. Silpni slaptažodžiai lemia, kad autentifikavimo sistemą bent daliai vartotojų gali apeiti nors šiek tiek resursų į tai investavusios atakuojančios šalys.

Saugus autentifikavimas slaptažodžiais taip pat pasižymi tuo, kad reikalaujama labai daug vartotojo atminties resursų [8]. Atlikti tyrimai rodo, kad žmogus trumpalaikė atmintis yra ribota ir gali talpinti apie septynis elementus [9], o nesunkiai galima atsiminti tik vieną saugų slaptažodį [2]. Tačiau vidutinis vartotojas įvairiose svetainėse paprastai valdo nemažą kiekį paskyrų, todėl tiesiog fiziškai neįmanoma naudoti skirtingus slaptažodžius visoms paskyroms. Vienodų slaptažodžių naudojimas skirtingoms paskyroms yra plačiai vartojamas šios problemos sprendimo būdas [10][11]. Įvykus saugumo incidentui ir atakuojančiai šaliai gavus prisijungimo duomenų bazę iš kurios nors svetainės, tampa įmanomas visų paskyrų, naudojančių vienodą slaptažodį, saugumo sukompromitavimas.

Slaptažodžiais paremtos autentifikavimo sistemos saugumas taip pat priklauso nuo komunikacijos kanalo tarp vartotojo ir svetainės ar paslaugos konfidencialumo. Įprasto slaptažodžių naudojimo atveju į komunikacijos kanalą patenka ir fizinis pasaulis: vartotojas, suvesdamas slaptažodį klaviatūra, turi būti tikras, jog yra privačioje aplinkoje ir niekas negali matyti klaviatūra suvedamos informacijos. Jei vartotojas klaviatūra suveda slaptažodį neprivačioje aplinkoje, priklausomai nuo konkretaus atvejo, klaviatūra suvestus duomenis galima gana nesunkiai nuskaityti [12]. Net jeigu fizinis privatumas yra pakankamas, vartotojas turi būti užtikrintas, jog per klaidą neatskleidžia slaptažodžio jį suvesdamas į atakuojančios šalies kontroliuojamą norimos svetainės kloną [13]. Apibendrinant, galima daryti išvadą, kad egzistuoja pakankamai įrankių, kuriuos naudojanti atkakli tam tikro vartotojo paskyrų autentifikavimą atakuojanti šalis gali pasiekti savo tikslą.

1.3. Slaptažodžių tvarkyklės

1.3.1 Pritaikomumas autentifikavimo saugumo problemai spręsti

Slaptažodžių tvarkyklės yra programinės įrangos paketai, leidžiantys vartotojui naudoti slaptažodžius, atitinkančius saugumo ekspertų rekomendacijas ir kartu išvengti su tokiais slaptažodžiais susijusių patogumo problemų. Slaptažodžių tvarkyklės veikia kaip tarpininkas tarp vartotojo ir svetainės, saugantis vartotojo slaptažodžius arba juos generuojantis pagal tam tikras taisykles. Nepriklausomai nuo slaptažodžių tvarkyklės tipo, norėdamas prisijungti prie svetainės, vartotojas pirmiausia kreipiasi į slaptažodžių tvarkyklę su užklausa tam tikros svetainės slaptažodžiui gauti. Jį gavęs, vartotojas tęsia autentifikavimą su svetaine naudodamas tuos pačius veiksmus, kaip ir nenaudojant slaptažodžių tvarkyklės.

Slaptažodžių tvarkyklėms negalioja viena iš pagrindinių problemų, su kuria susiduriama norint autentifikavimo saugumą padidinti naudojant kitas autentifikavimo schemas, tokias kaip biometrinis

ar kelių dedamųjų autentifikavimas. Pastarąsias sukurti galima galybe skirtingų būdų, o pagrindiniai pakeitimai yra paslaugų tiekėjo pusėje. Tai lemia, kad skirtingi paslaugų tiekėjai naudoja skirtingus būdus autentifikuotis, kurie ne tik nesupaprastina, bet dar ir padaro autentifikavimą sudėtingesniu vartotojui. Tuo tarpu slaptažodžių tvarkyklės leidžia iš esmės pagerinti slaptažodžių autentifikavimo saugumą naudojant esamą infrastruktūrą. Paslaugų tiekėjo požiūriu, vartotojo elgsena nepasikeičia, tik naudojamas saugus slaptažodis. Dėl to, slaptažodžių tvarkyklės yra universalios: jas galima naudoti bet kur, kur priimamas slaptažodis.

Nors ir pripažįstama, kad dabartinė slaptažodžių naudojimo praktika yra nepatenkinama saugumo prasme, nėra iki galo aišku, ar šis trūkumas pats savaime yra pakankama priežastis visiškai pakeisti autentifikavimo schemą į kitą. Daugumoje atvejų, sistemos, į kurias įsilaužus potencialiai galima prieiti prie didelės vertės turto, yra suprojektuotos taip, kad pelnas, kurį atakuojanti šalis galėtų gauti apėjusi autentifikavimo sistemą, yra gerokai mažesnis už sąnaudas tam atlikti [14]: aukštos rizikos paskyros naudoja ne slaptažodžiais paremtą autentifikavimą, o likusiais atvejais paskyros yra pakankamai apribotos, kad yra sudėtinga pelningai panaudoti prisijungimo duomenis. Pavyzdžiui, yra žinoma, kad piniginių lėšų pasisavinimą iš sukompromituotų internetu valdomų banko sąskaitų apriboja ne tokių sąskaitų, o asmenų, kurie galėtų paimti pinigus iš bankomatų, trūkumas [15]. Net ir žemos vertės paskyrų, kai tikrojo savininko prieigos prie paskyros apribojimas yra didžiausias galimas nuostolis, atveju, slaptažodis nėra vienintelė autentifikavimo priemonė, todėl įvykus jo nuotekiui, galima atgauti paskyros kontrolę, o nuostoliai yra minimalūs [16]. Dėl to, nors kibernetinių nusikaltimų metu yra patiriami nemaži nuostoliai, sunku identifikuoti realius nuostolius patiriamus būtent dėl slaptažodžių autentfikavimo pažeidžiamumo [14]. Nesant aiškios alternatyvių autentifikavimo priemonių naudos, nenuostabu, kad pastarosios labai sunkiai skinasi kelią.

Slaptažodžių tvarkyklės tuo tarpu yra ne revoliucinis, o evoliucinis sprendimas su aiškiais privalumais ir trūkumais. Jos nereikalauja iš esmės jokių infrastruktūros pokyčių, yra paprastos naudoti ir lengvai suprantamos vartotojų. Menko vartotojų nepatogumo sąskaita slaptažodžių saugumas, nors ir yra fundamentaliai ribotas, yra labai pagerinamas.

1.3.2 Slaptažodžių saugojimo metodai

Slaptažodžius saugančios tvarkyklės sprendžia slaptažodžių saugumo problemą iš vartotojo perimdamos saugių slaptažodžių saugojimą ir pateikdamos slaptažodžius, kai vartotojui reikalinga prieiga. Prieiga prie slaptažodžių ribojama naudojant *pagrindinį* slaptažodį. Tai yra vienintelė informacija, kurią reikia prisiminti vartotojui.

Slaptažodžiai ir papildoma informacija, reikalinga jiems atkurti, tokia kaip svetainių adresai ir

paskyrų identifikatoriai, yra išsaugomi. Priklausomai nuo to, kur saugomi vartotojo slaptažodžiai, šio tipo tvarkykles galima suskirstyti į keturias grupes:

- Lokali saugykla. Informacija yra saugoma vartotojo kompiuteryje. Priklausomai nuo konkrečios tvarkyklės, informacijos saugumu tvarkyklė arba užsiima pati, arba perduoda atsakomybę operacinės sistemos slaptažodžių saugyklai.

  Pirmuoju atveju, informacija paprastai saugoma užšifruotame faile, talpinančiame duomenų bazę. Pagrindinis slaptažodis yra naudojamas šifravimo raktui atkurti, kuris naudojamas iššifruoti seniems slaptažodžiams ir užšifruoti naujiems.

  Antruoju atveju, informacija yra užšifruojama naudojantis raktu, kuris yra išvestas iš vartotojo paskyros slaptažodžio arba lokaliai saugomo privataus rakto. Kadangi abiem atvejais rakto saugumas nepriklauso nuo slaptažodžių tvarkyklės, saugoma informacija dažnai papildomai šifruojama pagrindiniu slaptažodžiu.

- Aparatinis modulis. Informacija yra saugoma išoriniame USB įrenginyje arba TPM modulyje. Išorinio įrenginio naudojimas užtikrina, kad nesant saugumo spragų pačiame įrenginyje, yra gerokai apribota visos slaptažodžių duomenų bazės nutekinimo galimybė, net jei naudojamas kompiuteris nėra saugus. Šio slaptažodžių saugojimo būdo pagrindinis trūkumas yra aparatinės įrangos palaikymo problemos įvairiose sistemose ir slaptažodžių praradimas modulio pametimo ar pavogimo atveju.

- Mobilusis telefonas. Informacija yra saugoma mobiliajame telefone ir priklausomai nuo slaptažodžio tvarkyklės, arba įvedama rankiniu būdu, arba perkeliama naudojant papildomą programinę įrangą. Šiuo atveju tvarkyklės savybės nesiskiria nuo aparatinio modulio, bet dėl gerokai didesnio paskirčių spektro, mobilųjį telefoną galima atakuoti įvairiais būdais, dėl oo nukenčia saugumas.

- Debesys. Informacija yra užšifruojama naudojant raktą, išvestą iš vartotojo pagrindinio slaptažodžio ir tada saugojama trečiosios šalies serveriuose. Pagrindinė šio informacijos saugojimo metodo problema yra ta, kad vartotojas niekaip negali daryti įtakos šifruotos informacijos saugumui. Jei vartotojo informacija nutekinama iš paslaugos tiekėjo, atakuojanti šalis gali bandyti gauti prieigą prie saugojamų slaptažodžių bandydama atspėti pagrindinį slaptažodį.

Jei nenaudojamas aparatinis modulis, o didžioji dauguma vartotojų būtent taip ir daro, tuomet informacijos nuotekio atveju atsiranda netiesioginės (angl. *offline*) atakos grėsmė slaptažodžiams.

Egzistuoja sprendimai, kaip paversti neakivaizdinę ataką į tiesioginę (angl. *online*) [17]. Jie remiasi didelio kiekio (N>10000) apgaulingų šifruotų slaptažodžių duomenų bazėje sukūrimu. Jei atakuojanti šalis gauna nutekintą slaptažodžių duomenų bazę, tuomet norint atspėti slaptažodį teisingam slaptažodžių rinkiniui, vidutiniškai teks iššifruoti N/2 karto daugiau rinkinių, nei šio sprendimo nenaudojant. Taip pat, net jeigu slaptažodžiai bus iššifruoti, atakuojanti šalis negali pasakyti, kuris iš slaptažodžių rinkinių yra teisingas. Tokiu atveju, tektų bandyti prisijungti naudojant iššifruotus slaptažodžius ir iš esmės vykdyti tiesioginę ataką galimų slaptažodžių kiekio nuo N iki 1 sumažinimui. Sudėtingiausia šio metodo dalis yra pakankamai tikroviškų slaptažodžių generavimas, kas yra netrivialus uždavinys [17].

Visos komercinės ir didžioji dauguma nekomercinių slaptažodžių tvarkyklių naudoja slaptažodžių saugojimo metodą. Kai kurios jų atlieka ir keletą papildomų funkcijų, padedančių vartotojams valdyti slaptažodžius:

- Automatinis formų užpildymas. Naršyklių įskiepius turinčios tvarkyklės automatiškai iš prisijungimo formų surenka slaptažodžių ir prisijungimo vardų poras ir, vartotojui paprašius, vėliau jas užpildo prisijungimo metu. Vartotojui nereikia pačiam įkelti slaptažodžio į prisijungimo formą naudojant iškarpinę, tokiu būdu sumažinama rizika, jog kuris nors neautorizuotas vartotojo procesas gaus privačią slaptažodžių informaciją. Taip pat, priklausomai nuo slaptažodžių tvarkyklės, labai sumažinama sukčiavimo (angl. *phishing*) atakos galimybė, kadangi automatiškai yra sekamas svetainės, į kurią yra suvedami slaptažodžiai, URL.
- Saugių slaptažodžių generatorius. Slaptažodžius saugančios tvarkyklės gali įkelti bet kokį vartotojo slaptažodį, pridedant naują paskyrą iškyla problema kaip jį patogiai sugeneruoti. Didžioji dalis tvarkyklių tai sprendžia leisdamos vartotojui sugeneruoti slaptažodį pagal norimas taisykles naudojant vidinį atsitiktinių skaičių generatorių.
- Atsarginės kopijos. Kadangi tvarkyklės duomenų bazė yra vienintelė vieta, kurioje saugomi slaptažodžiai, o slaptažodžiai paprastai būna sugeneruoti atskirai, vartotojui praradus duomenų bazę būtų prarandami visi slaptažodžiai. Dėl to, tvarkyklės leidžia vartotojui išsisaugoti paprastai šifruotą duomenų bazės kopiją kitoje vietoje, kas leistų atgaminti slaptažodžius duomenų bazės praradimo atveju.
- Sinchronizacija. Dauguma vartotojų naudoja daugiau kaip vieną įrenginį prisijungti prie interneto svetainių, todėl norint naudoti slaptažodžius saugančią tvarkyklę, jos duomenų bazė turi būti bent iš dalies perkeliama į visus naudojamus įrenginius. Rankiniu būdu atliekamas

perkėlimas yra itin nepatogus, gerokai padidėja duomenų bazės nutekinimo ir slaptažodžių praradimo galimybė, jei skirtingos duomenų bazės turi skirtingą slaptažodžių rinkinį ir viena iš jų yra nukopijuojama į visus įrenginius. Dėl to slaptažodžių tvarkyklės leidžia sinchronizuoti slaptažodžius tarp įrenginių. Kaip sinchronizacijos įrankis naudojami įvairūs mechanizmai – įgyvendinti tiek slaptažodžių tvarkyklės tiekėjo, tiek operacinės sistemos tiekėjo serveriuose. Kai kurios tvarkyklės leidžia keliems įrenginiams sinchronizuoti slaptažodžių duomenų bazes tarpusavyje be trečiosios šalies paslaugų.

- Internetinė prieiga. Tam tikrais atvejais vartotojui reikia gauti vieną jo slaptažodžių iš įrenginio, į kurį negalima instaliuoti jokios programinės įrangos. Šiai problemai spręsti kai kurios slaptažodžių tiekėjo serverius sinchronizacijai naudojančios tvarkyklės leidžia prie slaptažodžių prieiti tiesiog interneto naršykle, pateikus reikalingą informaciją serveryje esančiai slaptažodžių duomenų bazei dešifruoti.
- Vienkartiniai slaptažodžiai. Kai kuriais atvejais vartotojas negali užtikrinti savo fizinio privatumo ir suvesdamas savo pagrindinį slaptažodį, gali jį atskleisti, kas leistų atkakliai atakuojančiai šaliai gauti priėjimą prie slaptažodžių jei kada nors būtų nutekinta slaptažodžių duomenų bazė. Kai kurios tvarkyklės leidžia pasirinkti keletą vienkartinių slaptažodžių, kurie naudojami kaip pagrindinio slaptažodžio atitikmenys tuo atveju, kai suvedamos informacijos konfidencialumas negali būti užtikrintas.

1.3.3 Slaptažodžių generavimo metodai

Egzistuoja keletas slaptažodžių generavimo metodų. Bendru atveju visi jie remiasi tam tikrai paskyrai skirto slaptažodžio generavimo iš dviejų dedamųjų [18]:

- pagrindinio slaptažodžio, kuris yra itin sunkiai atspėjamas. Šioje schemoje nuo pagrindinio slaptažodžio konfidencialumo priklauso visų paskyrų saugumas, todėl turi būti kreipiamas ypatingas dėmesys jam apsaugoti
- paskyrai-specifinio identifikatoriaus, kuris yra lengvai prisimenamas. Identifikatoriaus sudarymo metodas priklauso nuo konkrečios slaptažodžio tvarkyklės, tačiau paprastai tam yra naudojamas svetainės domenas ir paskyros vardas.

Paskyrai slaptažodis tada yra generuojamas pagal šią formulę, kur f – konkrečioje slaptažodžio generavimo schemoje naudojama kriptografinė funkcija, turinti savybę, kad žinant slaptažodį ir vieną iš funkcijos argumentų, neįmanoma išskaičiuoti antrojo argumento.

$$slaptažodis = f(pagrindinis\ slaptažodis, identifikatorius)$$

Naudojant šią slaptažodžių valdymo schemą vartotojams slaptažodžių vartojimas tampa gerokai patogesnis ir saugesnis. Vartotojams nebereikia prisiminti skirtingų slaptažodžių kiekvienai svetainei. Taip pat, automatiškai yra generuojami sudėtingi slaptažodžiai, kurie nėra naudojami kelioms paskyroms. Skirtingai nuo slaptažodžius saugančių tvarkyklių, nebėra rizikos prarasti turimus slaptažodžius, kadangi visa slaptažodžiams generuoti reikalinga informacija yra prisimenama vartotojo.

Verta atkreipti dėmesį į tai, kad žinant vienai paskyrai sugeneruotą slaptažodį, nėra atskleidžiama informacija apie pagrindinį slaptažodį, kartu ir apie kitų paskyrų slaptažodžius. Taip pat, dviejų paskyrų slaptažodžiai $Q_1=f(P,D_1)$ ir $Q_2=f(P,D_2)$ niekada nėra tapatūs, todėl, priklausomai nuo konkrečios slaptažodžių tvarkyklės, galima išvengti sukčiavimo atakų. Tam įgyvendinti slaptažodžių tvarkyklė turėtų naudoti URL ar domeną kaip vieną iš dėmenų identifikatoriui formuoti. Tuomet, net jei vartotojas yra įviliojamas į tam tikro puslapio imitaciją norint išvilioti prisijungimo duomenis, atakuojanti šalis nieko nepeša, kadangi URL neatitinka vartotojo iš tikrųjų norimo puslapio URL [18].

Yra keletas galimų būdų slaptažodžiams generuoti iš pagrindinio slaptažodžio ir paskyros identifikatorių – anksčiau minėtos f kriptografinės funkcijos įgyvendinimui. Žemiau yra įvardinti keletas jų:

- Originaliame slaptažodžių generavimą iš paskyros identifikatorių siūliusiame straipsnyje [18] buvo naudojama įprastinė kriptografinė maišos funkcija, kuriai kaip argumentas buvo perduodami sujungti pagrindinis slaptažodis ir paskyrai skirtas identifikatorius. Tai yra, buvo naudojama $f(x,y)=h(x||y)$, kur h-kriptografinė maišos funkcija. Deja, priklausomai nuo maišos funkcijos, šis metodas yra nesaugus, kadangi žinant slaptažodį $a_1=h(x_1||y_1)$, piktavalis gali suskaičiuoti kitą slaptažodį $a_2=h(x_1||y_1||y_p)$, kur y_p - piktavalio parinkta reikšmė [19]. Minėtą maišos funkcijų trūkumą išspręstų kontroliniu parašu [20] (angl. *hash-based message authentication* code, *HMAC*) paremtos schemos. Kontrolinis parašas šiuo atveju būtų įgyvendinamas pasirenkant iš anksto žinomas dvi pagrindinio slaptažodžio ilgio simbolių sekas p_i ir p_o ir paskyros slaptažodį generuojant naudojant šią formulę: $f(x,y)=HMAC(x,y)=h(x\,XOR\,p_o||h(x\,XOR\,p_i||y))$, kur h-kriptografinė maišos funkcija, x - pagrindinis slaptažodis, o y - paskyros identifikatorius. Yra ir keletas modernesnių ir kriptografiškai saugesnių slaptažodžių generativimo schemų, kurios aprašytos žemiau.
- HKDF [21] yra kontroliniu parašu HMAC paremta raktų generavimo funkcija

$HKDF(s, k, info, L)$, kur s yra druskos (angl. *salt*) parametras, skirtas atsparumui grubios jėgos perrinkimui padidinti, k yra pirminis slaptažodis, *info* yra nebūtina papildoma informacija, o L yra išeities rakto ilgis. Funkcijos veikimo principas susideda iš dviejų žingsnių: tarpinio rakto generavimo ir rakto išskleidimo iki reikalaujamo ilgio. Tarpinis raktas t yra generuojamas pagal $t = HMAC(salt, k)$. Raktas iki reikalaujamo ilgio išskleidžiamas suskaičiuojant HMAC $N = \lceil L / L_{HMAC} \rceil$ kartų, kur L_{HMAC} yra HMAC funkcijos rezultato ilgis. Galutinis HKDF rezultatas yra pirmi L baitų iš $T = T_1 || T_2 || ... || T_N$, kur T_0 yra tuščia eilutė, o likę elementai $T_i = HMAC(t, T_{i-1} || info || i)$. Matome, kad trumpiems išeities raktams, kokie yra reikalaujami slaptažodžiai, $HKDF(s, k, info, L) = HMAC(HMAC(s, k), info || 1)$. Norint pritaikyti šią schemą slaptažodžiams generuoti iš dviejų dėmenų, pagrindinį slaptažodį galima naudoti k, o paskyros identifikatorių - s.

- PBKDF2 [22] (angl. *password-based key derivation function 2*) yra bet kokia dviejų parametrų pseudoatsitiktine funkcija paremta raktų generavimo funkcija $PBKDF2(H, k, s, c, L)$, kur H yra naudojama pseudoatsitiktinė funkcija (pvz. HMAC), k yra pradinis slaptažodis, s yra druskos (angl. *salt*) parametras, skirtas atsparumui grubios jėgos perrinkimui padidinti, c yra iteracijų skaičius, o L yra išeities rakto ilgis. Išeities raktas T skaičiuojamas generuojant $N = \lceil L / L_{HMAC} \rceil$ duomenų blokų, kurie yra sujungiami: $T = T_1 || T_2 || ... || T_N$. T_i yra generuojamas pagal tarpinę funkciją $T_i = F(k, s, c, i)$, kuri yra ekvivalenti XOR operacijai tarp c kartų iškviestos H funkcijos rezultato: $F(k, s, c, i) = U_1 \; XOR \; U_2 \; XOR \; ... \; XOR \; U_c$, kur $U_0 = H(k, s || i)$, o $U_i = H(k, U_{i-1})$. Norint pritaikyti šią schemą slaptažodžiams generuoti iš dviejų dėmenų, pagrindinį slaptažodį galima naudoti k, o paskyros identifikatorių - s. Deja, PBKDF schema leidžia jos reikšmes skaičiuoti paraleliai nereikalaujant daug atminties – šį algoritmą galima efektyviai įgyvendinti naudojant grafikos procesorius ar programuojamos logikos procesorius (angl. *field-programmable gate array, FPGA*) [23]. Tai reiškia, kad net ir labai didelė iteracijų skaičiaus c vertė yra potencialiai nepakankama atbaidyti atkakliam piktavaliui.

- scrypt [24] yra raktų generavimo funkcija, kurios reikšmėms skaičiuoti yra reikalingas palyginti didelis atminties kiekis: $scrypt(k, s, c, r, p, L)$, kur k yra pradinis slaptažodis, s yra druskos (angl. *salt*) parametras skirtas atsparumui grubios jėgos perrinkimui padidinti, c yra reikalaujamo procesoriaus laiko ar atminties parametras, r yra bloko dydžio parametras, p yra paralelizavimo parametras, o L yra išeities rakto ilgis. Scrypt algoritmas yra per sudėtingas, kad čia būtų galima pateikti pilną jo aprašą. Jis susideda iš trijų žingsnių:

- $p \times r \times 128$ baitų ilgio duomenų masyvo sugeneravimas naudojant PBKDF2 algoritmą su HMAC-SHA-256 funkcija kaip kontrolinio parašo funkcija.

- Šio masyvo reikšmių perskaičiavimas ir permaišymas c kartų. Kiekvieno permaišymo metu duomenys yra parenkami iš didelių atminties rėžių.

- Masyvo reikšmės maišos suskaičiavimas naudojant PBKDF2 algoritmą su HMAC-SHA-256 funkcija kaip kontrolinio parašo funkcija.

Norint pritaikyti šią schemą slaptažodžiams generuoti iš dviejų dėmenų, pagrindinį slaptažodį galima naudoti k, o paskyros identifikatorių - s. Šis algoritmas yra ypatingas tuo, kad yra itin sunkiai paralelizuojamas ir grafikos procesoriai ar programuojamos logikos procesoriai duoda tik labai ribotą pranašumą piktavaliui, norinčiam perrinkti slaptažodžius. Taip yra dėl to, kad pagrindinis algoritmo greičio apribojimas yra priėjimo prie didelio atminties kiekio greitis, kuris yra panašus skirtingose platformose [25].

Vienas iš potencialių slaptažodžių generavimo trūkumų yra tas, kad priklausomai nuo slaptažodžio tvarkyklės, vartotojas yra ganėtinai stipriai apribotas slaptažodžių keitimo požiūriu. Jei paskyros identifikatorius yra valdomas slaptažodžių tvarkyklės, tuomet vienintelis būdas pakeisti slaptažodį yra pagrindinio slaptažodžio keitimas, kas pareikalautų pakeisti visus slaptažodžius. Kita problema yra ta, kad jeigu slaptažodis neatitinka tam tikroje svetainėje taikomų reikalavimų (pavyzdžiui, yra per ilgas), vartotojas negali naudoti slaptažodžių tvarkyklės apskritai. Šias problemas galima apeiti kruopščiai parenkant generuojamų simbolių savybes ir vartotojui leidžiant nurodyti savo sugalvotą paskyros identifikatorių, todėl tampa įmanoma pakeisti slaptažodį nekeičiant pagrindinio slaptažodžio.

Susijusi, praktiškai neišsprendžiama problema yra ta, kad norint migruoti slaptažodžių valdymą iš kitos sistemos, būtina pakeisti visus jau naudojamus slaptažodžius. Tai reiškia, kad vien norint pradėti naudoti slaptažodžių tvarkyklę, reikalinga didelė laiko investicija. Palyginti su slaptažodžius išsaugančiomis tvarkyklėmis, tai yra ženklus trūkumas patogumo prasme, kadangi pastarąsias galima naudoti iškart, be jokių pakeitimų vartotojo paskyrose.

Slaptažodžių generavimo metodas yra rizikingesnis saugumo prasme lyginant su slaptažodžius išsaugančiomis tvarkyklėmis. Nepastebėto informacijos nuotekio atveju atakuojanti šalis gali prieiti ne tik prie visų praeityje naudotų slaptažodžių, kaip kad yra slaptažodžių saugojimo atveju, bet ir visų ateityje naudosimų slaptažodžių, net jei vartotojas nėra apie juos net ir pagalvojęs.

Kita problema yra tai, kad netinkamai parinkus slaptažodžių generavimo algoritmą, atakuojanti

šalis gali bandyti atspėti pagrindinį slaptažodį turėdama tik paskyros slaptažodį. Tai yra įmanoma dėl to, kad paskyros identifikatorius turi ypač mažai entropijos ir ataka iš esmės tampa ekvivalenti konkrečios maišos funkcijos inversijai.

Pagrindinio slaptažodžio nulaužimo riziką galima labai sumažinti naudojant didesnę sudėtingumo parametro c reikšmę anksčiau minėtuose slaptažodžių generavimo algoritmuose. Tokiu atveju iškyla problema: c reikšmė turi maksimalią reikšmę, kuriai esant slaptažodžiui generuoti užtrunkama pakankamai laiko, jog vartotojui slaptažodžių tvarkyklę tampa sudėtinga naudoti. Problemai spręsti yra pasiūlytas patobulintas slaptažodžių generavimo algoritmas, kai slaptažodžio generavimas yra sudarytas iš dviejų dalių [26]:

- Itin daug resursų reikalaujantis žingsnis, kuris atliekamas vieną kartą, o jo rezultatai išsaugomi lokaliai:

  $$V = f^{j}(vartotojo\ vardas \| pagrindinis\ slaptažodis)$$

- Mažai resursų reikalaujantis žingsnis, naudojantis praėjusio žingsnio rezultatą.

  $$slaptažodis = f^{k}(identifikatorius \| pagrindinis\ slaptažodis \| V)$$

Čia f – kriptografinė maišos funkcija, j ir k – funkcijos pakartojimų skaičius, $j \gg k$, V – lokaliai išsaugoma reikšmė.

Naudojant šį algoritmą, papildomas resursų kiekis, reikalingas atlikti grubios jėgos ataką prieš pagrindinį slaptažodį yra proporcingas j ir k santykiui. Kadangi V yra išsaugotas, vėlesnių naudojimų metu vartotojui slaptažodžio generavimo kaštai lieka tokie patys.

Lyginant saugojimo ir generavimo principu veikiančios slaptažodžių tvarkykles patogumo atžvilgiu, pastarosios yra šiek tiek sudėtingesnės naudoti ir suprasti vartotojui. Norint sugeneruoti slaptažodį, reikalingi du, su autentifikavimo forma svetainėse ar kitur visiškai nesusiję dėmenys: pagrindinis slaptažodis ir paskyros identifikatorius. Jeigu šiuos dėmenis bandoma pateikti per esančias autentifikavimo formas, vartotojai lengvai susimaišo, kas blogiausiais atvejais lemia pagrindinio slaptažodžio naudojimą vietoj paskyros slaptažodžio, taip labai padidinant jo nutekinimo galimybę [27].

Šiuo metu nėra gerą palaikymą turinčios slaptažodžių generavimo principu veikiančios slaptažodžių tvarkyklės. Visi šiuo metodu pagrįsti sprendimai yra prototipų stadijoje. Dėl to, skirtingai nuo subrendusių slaptažodžių saugojimo metodu veikiančių tvarkyklių, esamos tvarkyklės turi labai ribotą platformų palaikymą ir skurdų galimybių rinkinį.

1.3.4 Egzistuojančios slaptažodžių tvarkyklės

1.3.4.1 Naršyklių vidinės slaptažodžių tvarkyklės

Visos pagrindinės naršyklės – Internet Explorer, Safari, Opera, Chrome ir Firefox – turi vidines slaptažodžių tvarkykles, kurios naudojamos išsaugotų slaptažodžių tvarkymui. Deja, šių tvarkyklių galimybės yra labai ribotos, visais atvejais įgyvendinama tik slaptažodžių išsaugojimo galimybė, kai kuriais atvejais jų net nešifruojant. Visais atvejais, jei nenaudojamas pagrindinis slaptažodis, informacija yra išsaugoma tokiu būdu, kad gauti slaptažodžių informaciją atakuojančiai šaliai yra trivialus uždavinys [28]. Slaptažodžių saugyklos įgyvendinimas priklauso nuo platformos:

- Firefox ir Opera atveju išlaikomas tas pats funkcionalumas skirtingose platformose: slaptažodžiai yra šifruojami naudojant 3DES; jei nenaudojamas pagrindinis slaptažodis, naudojamas lokaliai išsaugotas raktas, kitu atveju raktas išvedamas iš pagrindinio slaptažodžio. Abiem atvejais yra įmanoma išvilioti pagrindinį slaptažodį iš vartotojo parodant *popup* langą, imituojantį pagrindinio slaptažodžio dialogą: naršyklės tokių bandymų neblokuoja [28].
- Windows platformoje nenaudojant pagrindinio slaptažodžio tiek Internet Explorer, tiek Chrome, tiek Safari naudoja standartines Windows Data Protection API funkcijas, taigi bet kuris agentas, turintis vartotojo teises, gali perskaityti jo išsaugotus slaptažodžius
- Linux platformoje Chrome naudoja Kwallet arba Gnome slaptažodžių saugyklas. Jei jų buvimas neidentifikuojamas, slaptažodžiai nešifruojami.

1.3.4.2 LastPass

LastPass [29] yra komercinė slaptažodžių tvarkyklė su plačiu galimybių rinkiniu ir platformų palaikymu. Tvarkyklė yra sukurta naudojant slaptažodžių saugojimo debesyse modelį. LastPass gali būti suinstaliuotas kaip naršyklės įskiepis pagrindinėse naršyklėse ir operacinėse sistemose – palaikoma Firefox, Chrome, Opera, Internet Explorer, Safari naudojant Windows, OSX, Linux, Android, Blackberry ar iOS operacines sistemas. Slaptažodžiai gali būti pasiekiami tiek per naršyklę, tiek parsisiuntus atskirą specializuotą programą. LastPass slaptažodžiai yra sinchronizuojami tarp visų tą pačią paskyrą naudojančių įrenginių, palaikomi vienkartiniai slaptažodžiai, šifruotos atsarginės kopijos, dviejų dėmenų autentifikavimas ir prieiga per naršyklę. Slaptažodžių informacija šifruojama naudojant AES256 ir PBKDF2-HMAC-SHA256 [30][31].

1Password

1Password[32] yra komercinė slaptažodžių tvarkyklė, gerokai brangesnė už LastPass, su

mažesniu platformų palaikymu ir galimybių rinkiniu. Skirtingai nuo LastPass, 1Password yra sukurta kaip atskira programa, o ne naršyklės įskiepis. Naudojamas slaptažodžių saugojimo modelis. Palaikomos Windows, OSX, iOS ir Android operacinės sistemos, slaptažodžiai gali būti sinchronizuojami įvairiais būdais; nėra privaloma juos kelti į centrinį serverį. Nepalaikomas dviejų dėmenų autentifikavimas. Slaptažodžių informacija yra šifruojama naudojant AES-256 ir PBKDF2-HMAC-SHA256[33].

1.3.4.3 KeePass

KeePass[34] yra nemokama, atviro kodo slaptažodžių tvarkyklė, naudojanti lokalų slaptažodžių saugojimo modelį. Tvarkyklė yra sukurta ir kaip atskira programa, ir kaip naršyklės įskiepis; gali pasigirti bene plačiausiu platformų palaikymu. Galimybių rinkinys yra ganėtinai ribotas – nėra būdo sinchronizuoti slaptažodžius, automatinių atsarginių kopijų, vienkartinių slaptažodžių ir t.t. Slaptažodžių informacija yra šifruojama naudojant AES-256 ir PBKDF2-HMAC-SHA-256 [33].

1.3.4.4 Roboform

Roboform [35] yra komercinė slaptažodžių tvarkyklė, sukurta kaip naršyklės įskiepis, naudojant slaptažodžių saugojimo modelį. Palaikomos įvairios platformos, įskaitant Windows, OSX, Linux, iOS ir Android operacinės sistemos. Pagrindinis slaptažodis neprivalomas, kai slaptažodžių duomenų bazė nėra užšifruojama. Palaikoma slaptažodžių sinchronizacija, dviejų dėmenų autentifikavimas. Slaptažodžiai gali būti saugomi palaikomuose USB moduliuose. Duomenims šifruoti yra naudojama AES, BlowFish arba RC6, kartu su PBKDF2-HMAC-SHA1 [31].

1.3.4.5 PwdHash

PwdHash [18] yra nemokamas, atviro kodo naršyklės įskiepis, naudojantis slaptažodžių generavimo metodą. PwdHash generuoja slaptažodžius pagal šią formulę, kur f yra kriptografinė maišos funkcija:

$$slaptažodis = f(pagrindinis\,slaptažodis || domenas)$$

Domenas šiuo atveju naudojamas kaip druskos (angl. *salt*) reikšmė maišos funkcijoje. Naudojant vieną pagrindinį slaptažodį galima sugeneruoti slaptažodžius neribotam skaičiui puslapių su skirtingais domenais. Konkrečiu atveju, vienam puslapiui galimas ne daugiau nei vienas slaptažodis, todėl šis sprendimas yra itin nelankstus. Taip pat, ši slaptažodžių tvarkyklė turi itin prastą vartotojo sąsają, kuri dėl menkos vartotojo klaidos gali nusiųsti pagrindinį vartotojo slaptažodį vietoj norimo slaptažodžio [27].

1.3.4.6 PasswordMultiplier

PasswordMultiplier [26] yra nemokamas atviro kodo Firefox naršyklės įskiepis, naudojantis slaptažodžių generavimo metodą. PasswordMultiplier generuoja slaptažodžius pagal šią formulę:

$$V = f^{j}(vartotojo\ vardas || pagrindinis\ slaptažodis)$$

$$slaptažodis = f^{k}(domenas || pagrindinis\ slaptažodis || V)$$

V yra naudojama kaip grubios jėgos atakas neutralizuojanti reikšmė, kurią suskaičiuoti užtrunka daug laiko ($j \gg k$). Ji sugeneruojama vieną kartą ir išsaugoma lokaliai. Esant tokiam slaptažodžių generavimo algoritmui, skirtingai nuo PwdHash, turint net ir keletą slaptažodžių neįmanoma atspėti pagrindinio slaptažodžio per protingą laiko tarpą. PasswordMultiplier, kaip ir PwdHash turi didelių vartotojo sąsajos problemų, kurių dėka užtenka menkos vartotojo klaidos pagrindiniam slaptažodžiui atskleisti [27].

1.3.4.7 Slaptažodžių tvarkyklių palyginimas

1 lentelėje yra pateikiamas nagrinėtų slaptažodžių tvarkyklių palyginimas. Slaptažodžių tvarkyklės lyginamos pagal šiuos parametrus:

- Ar slaptažodžių tvarkyklėje galima išsaugoti jau egzistuojančius slaptažodžius?
- Ar slaptažodžių tvarkyklė palaiko atsitiktinių slaptažodžių generavimą?
- Ar slaptažodžių tvarkyklė šifruoja ar kitaip apsaugo slaptažodžių duomenų bazę vartotojo duotu slaptažodžiu? Jei taip, ar metodas atitinka saugumo rekomendacijas?
- Ar slaptažodžių tvarkyklė gali generuoti slaptažodžius pagal pagrindinį slaptažodį?
- Ar galima sinchronizuoti slaptažodžius tarp kelių paskyrų?
- Ar slaptažodžių tvarkyklė priklauso nuo trečiosios šalies valdomos įrangos ar serverių?
- Ar slaptažodžių tvarkyklės kodas yra atviras?
- Ar slaptažodžių tvarkyklė veikia visose populiariose platformose: Windows, OSX, Linux, Android ir iOS?
- Ar yra įgyvendinama slaptažodžių izoliacija? Tai yra, ar vartotojo dažnai naudojamas autentifikavimo informacijos rinkinys jam nutekėjus potencialiai leidžia piktavaliui prieiti tik prie dalies tvarkyklėje išsaugotų slaptažodžių.

Matome, kad nei viena tvarkyklė neatitinka visų iškeltų tinkamumo parametrų. Visos

slaptažodžių sinchronizavimą palaikančios tvarkyklės yra uždaro kodo ir priklauso nuo trečiosios šalies serverių. Tai sukuria papildomą problemą: pradėjus naudoti konkrečią slaptažodžių tvarkyklę negalima būti garantuotam dėl to, ar trečiosios šalies paslauga išliks ir ateityje ar nesikeis jos kaina.

	Firefox	Internet explorer	Safari	Chrome	LastPass	1Password	KeePass	Roboform	PwdHash	PasswordMultiplier
Egzistuojančių slaptažodžių išsaugojimas	Taip	Taip	Taip	Taip	Taip	Taip	Taip	Taip	Ne	Ne
Naujų slaptažodžių generavimas	Ne	Ne	Ne	Ne	Taip	Taip	Taip	Taip	Taip	Taip
Duomenų bazės šifravimas	Taip	Taip	Taip	Ne visur	Taip	Taip	Taip	Taip	Ne	Ne
Slaptažodžių generavimas iš pagrindinio	Ne	Ne	Ne	Ne	Ne	Ne	Ne	Ne	Taip	Taip
Slaptažodžių sinchronizavimas	Ne	Ne	Ne	Ne	Taip	Taip	Ne	Taip	Iš dalies	Iš dalies
Naudoja trečiosios šalies įrangą	Ne	Ne	Ne	Ne	Taip	Taip	Ne	Taip	Ne	Ne
Tvarkyklės kodas yra atviras	Taip	Ne	Ne	Iš dalies	Ne	Ne	Taip	Ne	Taip	Taip
Platformų palaikymas	Taip	Ne	Ne	Taip	Taip	Taip	Taip	Taip	Taip	Taip
Informacijos izoliacija	Ne	Ne	Ne	Ne	Ne	Ne	Ne	Ne	Ne	Ne

1 lentelė: Slaptažodžių tvarkyklių palyginimas

1.4. Išvados

Atlikus priemonių slaptažodžiams tvarkyti analizę, išaiškinti keli svarbūs aspektai, į kuriuos atsižvelgus, galima sukurti veiksmingą slaptažodžių tvarkymo sistemą. Identifikuoti esamų slaptažodžių tvarkyklių trūkumai, į kuriuos atsižvelgta kuriant pažangesnį slaptažodžio generavimo ir saugojimo metodą antrame skyriuje.

- Slaptažodžiai yra universalus autentifikavimo metodas, naudojamas didžiojoje dalyje interneto svetainių. Slaptažodžiai yra patogus, paprastas ir lengvai idiegiamas autentifikavimo metodas. Deja, nenaudojant papildomos programinės įrangos slaptažodžiai tampa arba itin nepatogūs vartotojui, arba labai nesaugūs.
- Slaptažodžių trūkumai nėra pakankama priežastis visos autentifikavimo schemos keitimui. Alternatyvūs autentifikavimo metodai yra gerokai saugesni, tačiau sudėtingai diegiami ir neuniversalūs. Naudojant papildomą programinę įrangą slaptažodžiams tvarkyti įmanoma pašalinti pakankamai didelę dalį slaptažodžių saugumo problemų, kad papildomi kitų autentifikavimo schemų teikiami privalumai nebeatsveria jų trūkumų.
- Pagal veikimo metodą, visas slaptažodžių tvarkykles galima suskirstyti į dvi grupes: slaptažodžius išsaugančias ir juos generuojančias tvarkykles. Abiem atvejais gerokai

sumažėja vartotojui prisiminti reikalingos informacijos kiekis: slaptažodžius išsaugančių tvarkyklių atveju vartotojui tereikia prisiminti pagrindinį slaptažodį, o slaptažodžius generuojančių tvarkyklių atveju dar ir paprastą, lengvai prisimenamą identifikatorių kiekvienai paskyrai.

- Iš vartotojo perspektyvos, slaptažodžius išsaugančios tvarkyklės yra patogesnės. Jos naudoja paprastesnį slaptažodžių valdymo mechanizmą, vartotojui nereikia mokytis papildomų koncepcijų. Praktiškai nereikalingas esamų vartotojo sąsajų svetainėse modifikavimas. Taip pat, slaptažodžių generatoriaus atvejui vartotojui reikia prisiminti papildomą informaciją kiekvienai paskyrai. Slaptažodžių generavimo metodu veikiančios tvarkyklės reikalauja monolitinio slaptažodžių valdymo: norint naudoti sistemą, vartotojas privalo pakeisti visus savo slaptažodžius.
- Nėra sukurta atviro kodo slaptažodžių tvarkyklių, kurios nebūtų priklausomos nuo trečiosios šalies serverių, gebėtų sinchronizuoti slaptažodžius ir palaikytų tiek slaptažodžių saugojimą, tiek jų generavimą iš pagrindinio slaptažodžio.
- Ideali slaptažodžių tvarkyklė turėtų kombinuoti tiek slaptažodžių saugojimo, tiek jų generavimo metodų bruožus:
    - turi būti kiek įmanoma mažiau keičiama vartotojo sąsaja. Paprasčiausiu naudojimo atveju vartotojas turi matyti tik suvedamą slaptažodį
    - slaptažodžių tvarkymo sistema neturėtų būti monolitinė. Turi būti galimybė tvarkyti jau egzistuojančius slaptažodžius be pakeitimų
    - slaptažodžių sistema turi būti atspari praradimui. Idealiu atveju, vartotojo prisimenamos informacijos turi pakakti didžiajai daliai slaptažodžių atkurti
    - svarbu, kad vartotojas turėtų galimybę lengvai naudotis slaptažodžiais iš skirtingų įrenginių
    - idealiu atveju, vartotojas turi turėti galimybę nesaugoti slaptažodžių naudojant trečiųjų šalių paslaugas, kadangi taip padidėja slaptažodžių nuotekio galimybė.

2. SIŪLOMAS SLAPTAŽODŽIŲ TVARKYKLĖS MODELIS

Analizės dalyje apžvelgėme ir aptarėme esamus slaptažodžių tvarkymo metodus ir egzistuojančias slaptažodžių tvarkykles. Buvo tiriami du slaptžodžių tvarkyklių tipai: slaptažodžius saugančios tvarkyklės ir slaptažodžius generuojančios iš pagrindinio slaptažodžio tvarkyklės, jų privalumai ir trūkumai. Po to identifikuotos siektinos patogios ir saugios slaptažodžių tvarkyklės savybės. Į šiuos kriterijus atsižvelgiant buvo sudarytas naujas slaptažodžių tvarkyklės modelis.

Sudarytame modelyje slaptažodžių tvarkyklė naudoja hibridinį slaptažodžių tvarkymo modelį: geba tiek saugoti bet kokius slaptažodžius, tiek generuoti slaptažodžius iš pagrindinio slaptažodžio. Ypatingas dėmesys yra skiriamas praktiniam slaptažodžių tvarkyklės saugumui esant programinio įsilaužimo ar fizinės prieigos grėsmei. Tolesniuose skyriuose yra pateiktas detalus sudaryto modelio aprašas.

2.1. Slaptažodžių tvarkyklei keliami tikslai ir jų prioritetai

Atliekant slaptažodžių tvarkyklės reikalavimų analizę ir kuriant jos modelį, nebuvo apsieita be kompromisų. Be jų nėra apsieinama nors kiek sudėtingesnės programinės įrangos kūrimo procese, tačiau šiuo atveju, kuriant slaptažodžių tvarkykles, praktiškai kiekvienas dizaino sprendimas yra kompromisinis, kadangi, kaip išsiaiškinome analizės dalyje, slaptažodžių tvarkyklės turi būti tiek saugios, tiek patogios vartotojui, o abiejų šių dalykų praktiškai neįmanoma visiškai įgyvendinti vienu metu. Dėl to jau pačioje slaptažodžių tvarkyklės modelio kūrimo pradžioje reikia apsibrėžti, kokie konkretūs tikslai yra keliami jai, ir koks jų svarbos santykis tarpusavyje.

Pagrindiniai slaptažodžių tvarkyklei keliami tikslai yra patogumas vartotojui ir saugumas. Vartotojo patogumui yra skiriamas didesnis dėmesys, kadangi jei turima programinė įranga ar kuri jos dalis yra saugi, tačiau nepatogi ir dėl pastarojo faktoriaus vartotojas jos nenaudoja, tai įgyvendintas saugumo sprendimas netenka prasmės. Kurdami slaptažodžių tvarkyklę, šį kriterijų taikysime tik tam funkcionalumui, kuris bus naudojamas dažnai. Jei didesnis saugumas reikalauja vienkartinių nepatogumų vartotojui, pavyzdžiui, įdiegimo metu, tai saugumui bus skiriamas didesnis dėmesys.

Gerokai mažesnis prioritetas nei saugumui ir vartotojo patogumui yra skiriamas laikui, reikalingam vartotojui suvokti slaptažodžio tvarkyklės darbo principus, jog galėtų gebėti ja efektyviai naudotis (angl. *learning curve*). Laikoma, jog tai yra vienkartinė kaina, kurią sumoka vartotojas ir lygiai kaip vienkartinio nepatogumo vartotojui atveju, bet koks saugumo padidėjimas yra to vertas. Taip pat, pati slaptžodžių saugojimo ar generavimo koncepcija yra pakankamai

paprasta, todėl palyginti lengvai išmokstama ir suprantama.

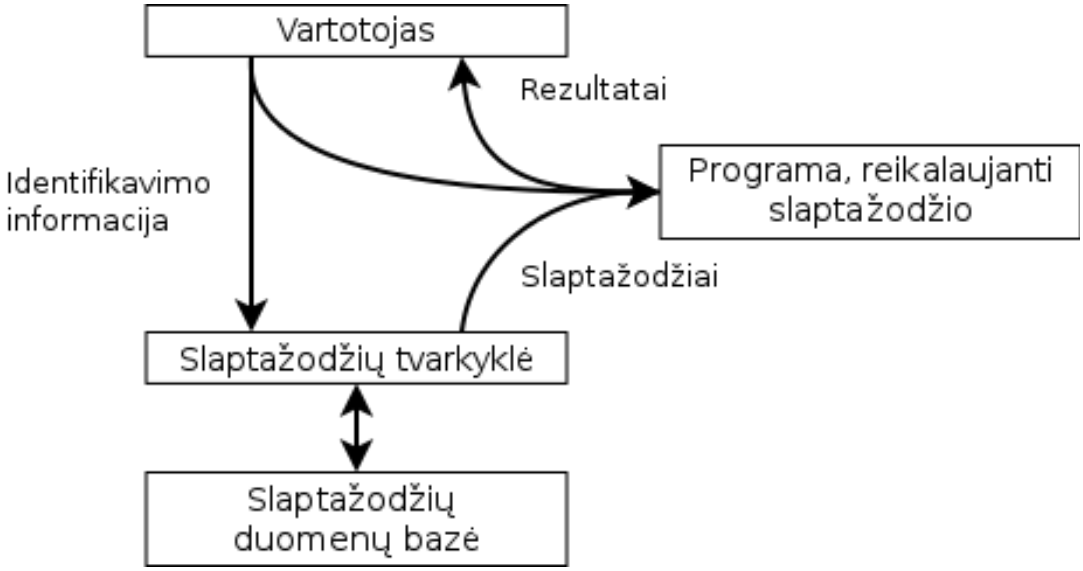
Visiems kitiems faktoriams, tokiems kaip sistemos paprastumas ar kaina, bus taikomas antraeilis dėmesys.

2.2. Konceptuali slaptažodžių tvarkyklės schema

Konceptuali slaptažodžių tvarkyklės schema yra pavaizduota 1 paveiksle. Joje nagrinėjamas pagrindinis konceptualus veiksmas, kurį galima atlikti naudojantis slaptažodžių tvarkykle: slaptažodžio gavimas iš slaptažodžių tvarkyklės ir jo perdavimas slaptažodžio reikalaujančiai kompiuterinei programai. Šio veiksmo metu, vartotojas perduoda paskyrą identifikuojančią informaciją tiek slaptažodžio reikalaujančiai programai, tiek slaptažodžių tvarkyklei. Slaptažodžių tvarkyklė iš slaptažodžių duomenų bazės gauna norimą slaptažodį ir perduoda slaptažodžio reikalaujančiai programai. Po autentifikavimo programa perduoda rezultatus vartotojui.

Verta pastebėti, jog pagal šį modelį vartotojas visai nebūtinai turi žinoti, kokį konkretų slaptažodį jis naudoja, tik tiek, jog jį naudojant autentikuotis pavyksta. Dėl šios savybės turime nemažą pačių slaptžodžių įgyvendinimo laisvę, o kartu ir papildomą saugumą, kadangi slaptažodis nepalieka informacinės sistemos fiziškai vartotojo darbo vietoje, pavyzdžiui, nėra rodomas ekrane.

Slaptažodžių duomenų bazė šioje schemoje yra abstraktus komponentas, tam tikrai identifikavimo informacijai visada grąžinantis tą pačią informaciją. Tai galima įgyvendinti nebūtinai fiziškai saugant konkrečius slaptažodžius. Naudojant šią architektūrą galima įgyvendinti tiek slaptažodžių saugojimą įgyvendinančią slaptažodžių tvarkyklę, tiek slaptažodžių generavimą įgyvendinančią slaptažodžių tvarkyklę praktiškai neatliekant modifikacijų pačiame slaptažodžių tvarkyklės lygmenyje.



1 pav. Konceptuali slaptažodžių tvarkyklės schema

Naginėjant schemą galima įvardinti bendras saugumo grėsmes ir pavojus, kurias būtina išspręsti kuriant slaptažodžių tvarkyklės architektūrą. Šiuo atveju egzistuoja dviejų tipų grėsmės: grėsmės slaptažodžių duomenų bazėje ilgą laiką saugomų duomenų konfidencialumui ir grėsmės iš vartotojui besinaudojant slaptažodžių tvarkykle gauto vieno ar kelių slaptažodžių konfidencialumui. Matome, kad slaptažodžių duomenų bazė turi būti gerokai stipriau apsaugota, kadangi bet kokia saugumo spraga joje gali lemti visų vartotojo slaptažodžių nutekėjimą. Tuo tarpu, slaptažodžių tvarkyklei ir ryšiui tarp jos ir vartotojo, ar slaptažodžio reikalaujančios programos gali būti skiriamas kiek mažesnis dėmesys, kadangi potencialiai gali nutekėti gerokai mažesnis duomenų kiekis. Į šiuos aspektus buvo atsižvelgiama kuriant slaptažodžių tvarkyklės architektūrą.

2.3. Reikalavimai slaptažodžių tvarkyklei

Slaptažodžių tvarkyklė yra išimtinai saugumo produktas, todėl jos architektūra tiesiogiai priklauso nuo funkcinių reikalavimų, kurie iškelti norimam saugumo lygiui užtikrinti. Šiame darbe kuriamam sprendimui buvo identifikuoti šie bendri funkciniai reikalavimai:

a) Vartotojo slaptažodžiai turi būti patogiai prieinami iš neribojamo skaičiaus kompiuterių ar mobiliųjų telefonų

b) Turi būti palaikomas tiek slaptažodžių saugojimo metodas, tiek slaptažodžių generavimo iš pagrindinio slaptžodžio metodas.

c) Įsilaužimas į vartotojo kompiuterį ar mobilųjį telefoną vartotojo teisėmis gali sudaryti galimybę nutekinti tik slaptažodžius, kurie tame įrenginyje buvo naudoti po įsilaužimo momento.

d) Vartotojui turi būti sudaryta galimybė suskirstyti slaptažodžius į kelias grupes, kurių saugumas nepriklausytų viena nuo kitos. Tokiu būdu siekiama apriboti įsilaužimo į vartotojo kompiuterį administratoriaus teisėmis žalą. Jeigu vartotojas nepastebi įsilaužimo administratoriaus teisėmis ir toliau naudojasi slaptažodžių tvarkykle, tai tam tikro slaptažodžių naudojimas gali sudaryti galimybę nutekinti tik toje pačioje grupėje esančius slaptažodžius.

e) Turi būti sudaryta galimybė naudoti kelių dėmenų autentifikavimą priėjimui prie tam tikrų slaptažodžių grupių ar slaptažodžių tvarkyklės konfigūracijos taip dar labiau sumažinant įsilaužimo į vartotojo kompiuterį administratoriaus teisėmis žalą.

f) Fizinis vartotojo stebėjimas ir fizinis priėjimas prie vartotojo kompiuterinės įrangos gali leisti prieiti prie vartotojo slaptažodžių tik tuo atveju, jei abu šie metodai yra naudojami kartu.

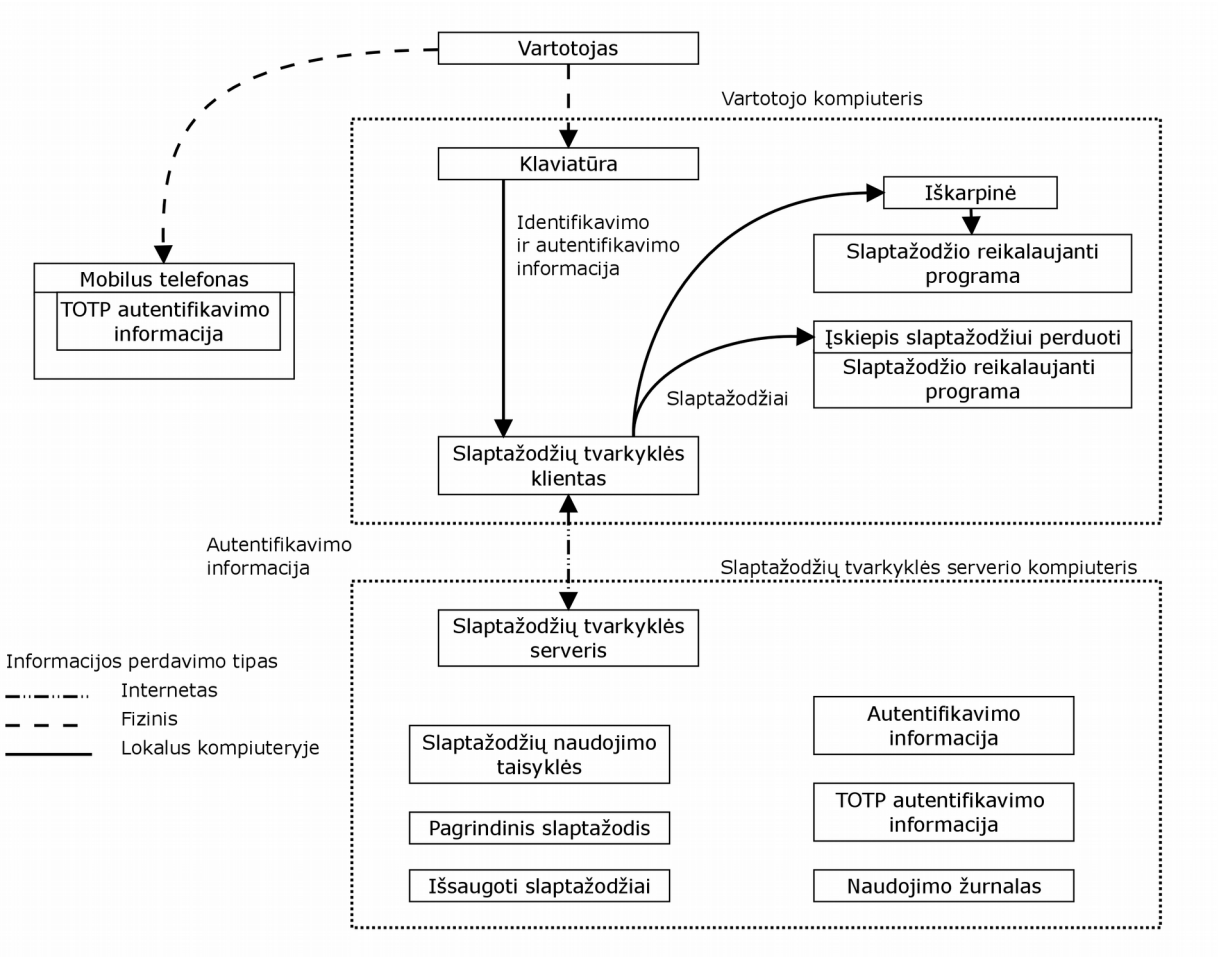
g) Jei naudojamas pagrindinis slaptažodis, turi būti įgyvendinta galimybė jį apsaugoti taip, kad nebūtų įmanoma jo nutekinti turint fizinį priėjimą prie visos vartotojo aparatinės įrangos.

h) Priėjimas prie slaptažodžių turi būti žurnalizuojamas

2.4. Slaptažodžių tvarkyklės architektūra

Slaptažodžių tvarkyklės architektūra, sukurta atsižvelgiant į identifikuotus funkcinius reikalavimus, yra pavaizduota 2 paveiksle. Slaptažodžių tvarkyklei įgyvendinti buvo pasirinkta kliento-serverio architektūra, atskiriant slaptažodžių duomenų bazę ir prieigą prie jos reguliuojančią slaptažodžių tvarkykės dalį nuo su vartotoju ir slaptažodžių reikalaujančiomis programomis bendraujančia dalimi.

Slaptažodžių tvarkyklės serveris atlieka visus veiksmus, susijusius su slaptažodžių saugojimu ir generavimu. Jame yra saugoma informacija apie tai, kokie autentifikavimo metodai turi būti naudojami prieš išsiunčiant slaptažodį tvarkyklės klientui. Taip pat, slaptažodžių tvarkyklės serveris atlieka slaptažodžių naudojimo žurnalizavimą. Slaptažodžių tvarkyklės klientas tuo tarpu neatlieka jokių tiesiogiai su slaptažodžių saugojimu ar generavimu susijusių funkcijų. Jis veikia kaip tarpininkas tarp vartotojo ir slaptažodžių reikalaujančių programų ir slaptažodžių tvarkyklės serverio. Vieninteliai su kriptografija susiję veiksmai, kuriuos jis atlieka yra autentifikavimo prie slaptažodžių tvarkyklės serverio informacijos saugojimas ir panaudojimas autentifikavimo metu. Plačiau slaptažodžių tvarkyklės klientas yra aprašytas 2.4.1 skyriuje, o slaptažodžių tvarkyklės klientas yra aprašytas 2.4.2 skyriuje.



2 pav. Slaptažodžių tvarkyklės architektūra

Tokia atskirtis yra būtina norint pasiekti siekiamą saugumo lygį saugomiems slaptažodžiams, kadangi tik tokiu atveju galima iškelti slaptažodžius saugančią slaptažodžių tvarkyklės dalį iš vartotojo kompiuterio. Šiuo būdu saugomų slaptažodžių saugumas gali būti potencialiai išsaugomas įsilaužimo į vartotojo kompiuterinę įrangą atveju. Kliento-serverio architektūra kartu leidžia gerokai paprasčiau įgyvendinti slaptažodžių prieinamumą iš kelių įrenginių. Verta paminėti, kad šiuo atveju slaptažodžių tvarkyklės serverio dalies iškėlimas iš vartotojo kompiuterio nėra privalomas, tai tik papildoma galimybė – vartotojas gali sukonfigūruoti tvarkyklę taip, kad tiek slaptažodžių tvarkyklės klientas, tiek serveris būtų viename kompiuteryje.

Papildomam sąsajos tarp slaptažodžių tvarkyklės kliento ir serverio autentifikavimui naudojamas laiku paremtas vienkartinių slaptažodžių algoritmas (angl. *time-based one-time password algorithm*) [36]. Plačiau šis algoritmas yra aprašytas 2.6 skyriuje.

2.4.1 Slaptažodžių tvarkyklės klientas

Sudarytoje slaptažodžių tvarkyklės architektūroje slaptažodžių tvarkyklės klientas atlieka

tarpininko vaidmenį tarp vartotojo, slaptažodžių reikalaujančių programų ir slaptažodžių tvarkyklės serverio. Slaptažodžių tvarkyklės klientas atlieka šiuos veiksmus:

- vartotojui pateikia jam patogią grafinę vartotojo sąsają
- priima iš vartotojo autentifikavimo ir paskyros identifikavimo informaciją
- saugo autentifikavimo su slaptažodžių tvarkyklės serveriu informaciją, jei tokia naudojama
- atlieka informacijos apsikeitimą su slaptažodžių tvarkyklės serveriu
- pateikia slaptažodį jo reikalaujančiai programai per iškarpinę, per specialų įskiepį konkrečiai programai ar kitu būdu.

Slaptažodžių tvarkyklės kliento grafinė sąsaja yra įgyvendinama dviem skirtingais lygmenimis: mažas minimalistinis langas slaptažodžiui gauti reikalingai informacijai suvesti ir slaptažodžių tvarkyklės kliento ir serverio nustatymų langas. Tuo siekiama dažniausiai vartotojo atliekamą veiksmą – slaptažodžio gavimą iš tvarkyklės – padaryti kuo patogesniu ir paprastesniu, o sudėtingesnė vartotojo sąsaja naudojama tik tada, kai reikalinga.

Į slaptažodžio gavimo langą įvedamas paskyros identifikatorius visais atvejais identifikuoja konkretų slaptažodį, todėl turi būti sudaryta kuo mažiau galimybių įvesti neteisingą paskyros identifikatorių ir įvedus neteisingai, tai pastebėti. To neužtikrinus, egzistuoja galimybė nepamatyti klaidų identifikatoriuje tam tikrais kritiniais momentais, pavyzdžiui registruojant naują paskyrą kokioje nors interneto svetainėje, todėl realus slaptažodis yra kitas, nei mano vartotojas ir vartotojas jo iš esmės nežino ir taip praranda prieigą prie ką tik sukurtos paskyros.

Klaidų paskyros identifikatoriuje prevencija atliekama šiais būdais:

- Visais atvejais, kai tam tikras paskyros identifikatorius naudojamas pirmą kartą, vartotojas įspėjamas ir reikalaujamas papildomas patvirtinimo veiksmas.
- Įvestas paskyros identifikatorius yra rodomas gerai matomoje vietoje papildomai ilgesnį laiką, negu yra nustatyta numatytųjų operacinės sistemos langų valdymo taisyklių, kurios dažniausiai gali paslėpti langą tik jam praradus fokusą. Šis funkcionalumas yra įgyvendinamas slaptažodžio gavimo langą sukonfigūruojant ignoruoti įprastą langų matomumo rikiavimo eilę ir jį laikant „virš kitų langų“. Po tam tikro laiko, kurį vartotojas gali keisti nustatymuose, šis langas yra paslepiamas. Tokiu būdu vartotojai gali ilgiau matyti suvestą paskyros identifikarorių ir taip pastebėti klaidas jame.

Slaptažodžių tvarkyklės klientas atlieka slaptažodžių pateikimą jų reikalaujančioms programoms. Nėra saugaus ir kartu veikiančio visose platformose būdo atlikti šiai operacijai, todėl kiekvienai platformai šis funkcionalumas yra įgyvendinamas naudojant konkrečios platformos įrankius. Detaliau tai yra aprašyta 2.8 skyriuje.

2.4.2 Slaptažodžių tvarkyklės serveris

Sudarytoje slaptažodžių tvarkyklės architektūroje slaptažodžių tvarkyklės serveris valdo slaptažodžių duomenų bazę ir atlieka klientų prieigos prie jos kontrolę. Slaptažodžių tvarkyklės serveris atlieka šiuos veiksmus:

- atlieka prieigos prie slaptažodžių kontrolę, klientų autentifikavimą
- įveda kiekvienos prieigos prie slaptažodžių duomenis į žurnalą
- atlieka neteisingo autentifikavimo įvykių monitoringą. Įvykių dažnumui peržengus nustatytas ribas, praneša vartotojui ir gali pradėti reikalauti papildomų autentifikavimo priemonių tam tikrai paskyrai
- generuoja naujus slaptažodžius naudojant atsitiktinių skaičių generatorių arba pagrindinį slaptažodį
- saugo esamus paskyrų identifikatorius ir juos atitinkančius slaptažodžius
- saugo pagrindinius slaptažodžius

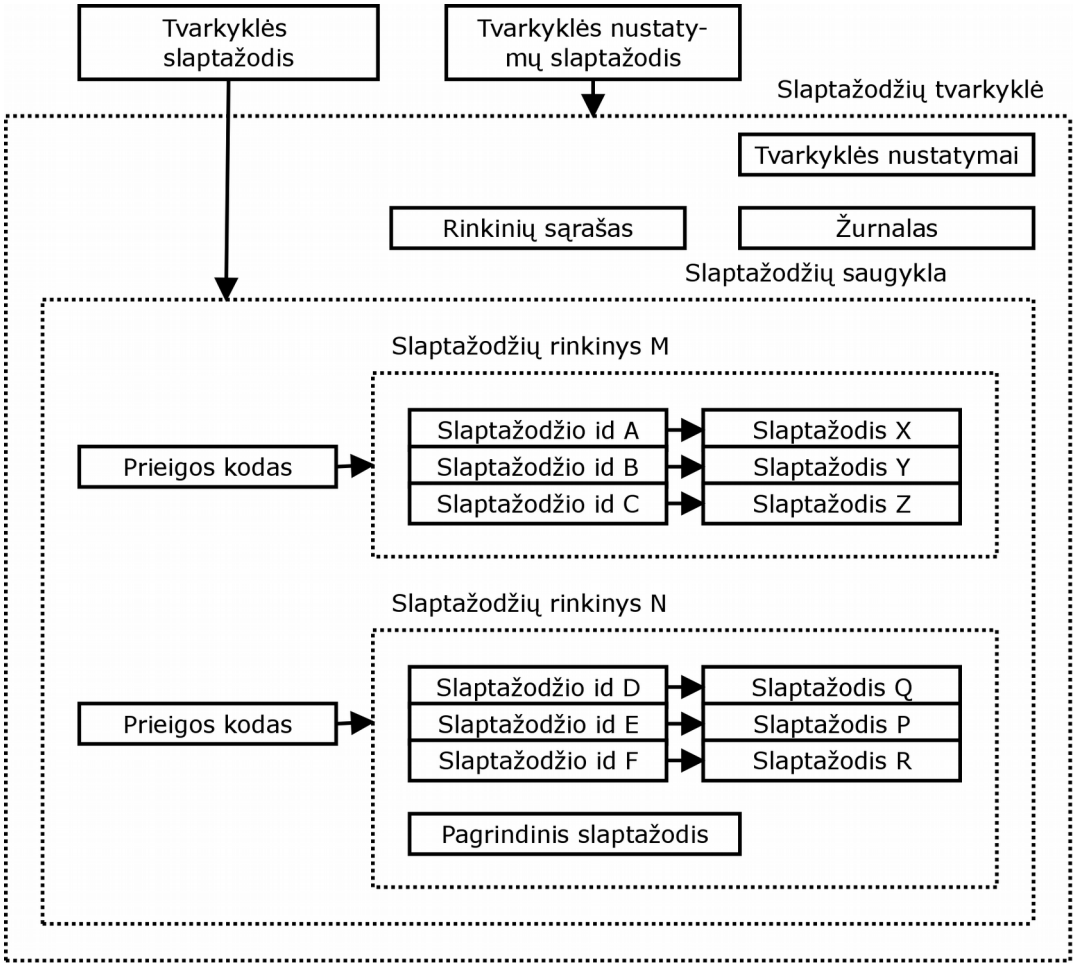
Sąsaja tarp slaptažodžių tvarkyklės kliento ir serverio yra šifruojama, jie vienas kitą autentikuoja. Slaptažodžių tvarkyklės serveris prieigą prie slaptažodžių ir klientų autentifikavimą atlieka keletu būdų, priklausomai nuo reikalaujamo informacijos saugumo. Tiek slaptažodžių tvarkyklė, tiek serveris gali saugoti papildomą autentifikavimui naudojamą informaciją. Detaliau tai yra aprašyta 2.6 skyriuje.

Slaptažodžių tvarkyklės serveris į žurnalą įrašo šiuos prieigą prie slaptažodžių identifikuojančius duomenis: data ir laikas, kliento IP adresas ir paskyros identifikatorius.

Paskyrų identifikatoriai, juos atitinkantys slaptažodžiai ir pagrindiniai slaptažodžiai saugomi duomenų bazėje. Slaptažodžių duomenų bazė yra šifruojama. Šifravimo raktui saugoti yra pasirenkama integruota duomenų saugykla, prie kurios neįmanoma prieiga iš išorės (pavyzdžiui į procesorių integruota Flash ar EEPROM tipo atmintinė), jei tokia saugykla yra. Serveryje yra įgyvendinta galimybė registruoti fizinį įsilaužimą iš išorės, kurio metu saugyklos šifravimo raktas yra ištrinamas.

2.5. Slaptažodžių saugos modelis

Reikalavimų analizėje 2.3 skyriuje buvo identifikuotas poreikis kuo didesnės apsaugos nuo galimybės nutekinti informaciją net ir tais atvejais, kai piktavalis turi fizinę prieigą prie vartotojo kompiuterinės įrangos ir potencialiai gali stebėti vartotoją prieš gaudamas minėtą prieigą. Tai reiškia, kad piktavalis gali potencialiai gauti visą informaciją, reikalingą prieiti prie slaptažodžių, kuriuos vartotojas naudojo piktavalio stebėjimo periodu. Norint sumažinti informacijos nuotekį atsižvelgiant į šį scenarijų, reikalingas kompleksinis sprendimas. Šio sprendimo supaprastinta schema iš vartotojo perspektyvos pateikta 3 paveiksle.



3 pav. Slaptažodžių saugos sprendimo schema iš vartotojo pusės

Sprendimo esmė yra ta, kad vartotojas slaptažodžius suskirsto į keletą skirtingų grupių, vadinamų slaptažodžių rinkiniais. Slaptažodžių rinkinys gali apimti slaptažodžių, generuojamų iš pagrindinio slaptažodžio, aibę, arba išsaugotų slaptažodžių grupę. Kiekvienam šių rinkinių priskiriama informacija serverio duomenų bazėje yra užšifruojamas skirtingu šifro raktu. Patys šifro raktai yra taip pat užšifruojami, tačiau tam yra naudojamas vartotojo duodamas slaptažodis, vadinamas prieigos kodu ir kuris nėra saugomas serveryje. Serveris užtikrina, kad neįmanoma

priskirti vienodų prieigos kodų prie skirtingų slaptažodžių rinkinių. Tokiu būdu kiekvienas prieigos kodas gali suteikti informacijos iššifruoti daugiausiai vieną slaptažodžių rinkinį – vieną pagrindinį slaptažodį arba vieną išsaugotų slaptažodžių grupę. Jeigu naudojami pakankamai sudėtingi prieigos kodai, piktavalis negali prieti prie slaptažodžių rinkinių, kurių prieigos kodų vartotojas nėra suvedęs stebint piktavaliui. Tokiu būdu vartotojas slaptažodžius gali susiskirstyti į kelis rinkinius pagal jų konfidencialumo lygmenį ar pagal kitą kriterijų. Tai, kad kiekvienas slaptažodžių rinkinys turi unikalius prieigos kodus leidžia juos naudoti slaptažodžių rinkinių identifikavimui.

Žemiau pateikiamos papildomos priemonės, naudojamos slaptažodžių saugumui užtikrinti:

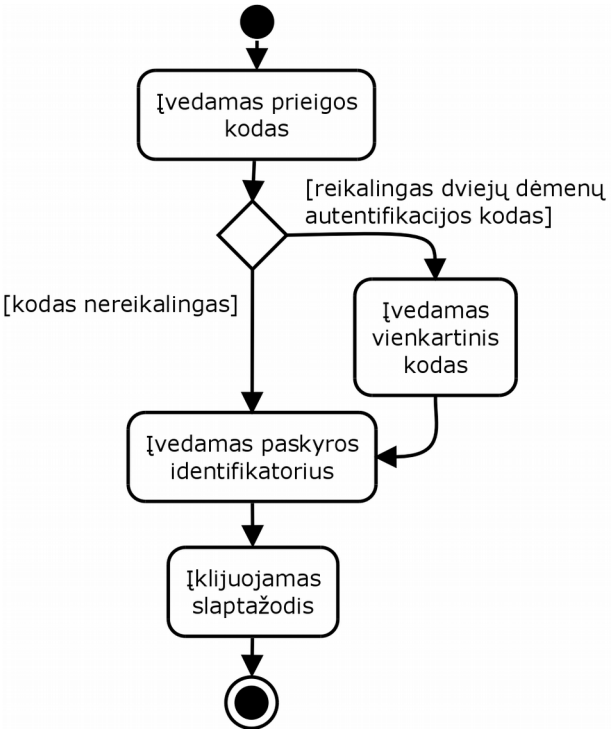
- vartotojo patogumui serverio slaptažodis yra naudojamas tik pirmą kartą autentikuojantis prie serverio naudojant tam tikrą klientą. Pavykus autentifikavimui serveris sugeneruoja atsitiktinį ilgą slaptažodį šiam klientui, jį išsaugo duomenų bazėje ir išsiunčia klientui, kuris atsiųstą slaptažodį taip pat išsaugo. Vėliau šį kliento slaptažodį galima naudoti autentifikavimui vietoje serverio slaptažodžio dažniausiai atliekamiems veiksmams: slaptažodžiams generuoti ir prieigai prie serveryje saugomų slaptažodžių. Šiems veiksmams atlikti visais atvejais yra reikalaujami papildomi vartotojo įvedami prieigos kodai, todėl rizikos, kad piktavaliui gavus prieigą prie kliento slaptažodžio, jis galės prieiti ir prie slaptažodžių, nėra.
- slaptažodžių tvarkyklės nuostatų keitimui yra naudojamas atskiras *nustatymų slaptažodis*.
- jeigu vartotojas suveda neegzistuojantį prieigos kodą kelis kartus, serveris panaikina leidimą klientui autentikuotis naudojantis jo kliento slaptožodžiu. Tokiu būdu pašalinama galimybė piktavaliui autentikuotis prie serverio ir atlikti prieigos kodų perrinkimą jam gavus prieigą prie vartotojo kompiuterinės įrangos.
- palaikomi laikini arba vienkartiniai prieigos kodai. Laikini slaptažodžiai leidžia tam tikrą skaičių kartų gauti prieigą prie serveryje esančio slaptažodžių rinkinio, po to sunaikinant šifravimui būtent tuo prieigos raktu naudotą informaciją. Tokiu būdu, jei vartotojas jaučiasi stebimas nesaugioje aplinkoje, jis gali naudoti vienkartinį slaptažodį nepalikdamas galimybės sukompromituoti autentifikavimo tarp slaptažodžių tvarkyklės kliento ir serverio mechanizmo, jeigu iškart po to piktavaliai gautų fizinį priėjimą prie vartotojo kompiuterio. Kartu sumažinama ir galimybė sukompromituoti serveryje saugomų slaptažodžių konfidencialumą jeigu piktavalis gautų fizinę prieigą prie serverio.
- vartotojui norint, kliento autentifikavimas arba priėjimas prie tam tikrų slaptažodžių rinkinių galimas tik papildomai autentifikuojant naudojant dviejų dėmenų autentifikavimą. Šis

metodas leidžia įregistruoti kliento slaptažodį serveryje nenaudojant mažiau saugių metodų.

Paminėti autentifikavimo metodai ir jų saugumą padidinančios priemonės leidžia įgyvendinti nuo sukompromitavimo maksimaliai apsaugotą sistemą. Nepaisant to, informacijos kiekis, kurį vartotojui reikia prisiminti, yra palyginti nedidelis (žr. 2 lentelę). Galima daryti išvadą, kad vienintelė informacija, kuri yra dažnai naudojama ir kartu sunki prisiminti yra prieigos kodas. Likę slaptažodžiai yra naudojami palyginti retai, vartotojams neturėtų iškilti problemų juos užsirašyti. Lyginant su dabartine situacija, kai vartotojui yra rekomenduojama kiekvienai paskyrai naudoti sudėtingą slaptažodį ir juos visus prisiminti, tai yra ženklus pagerėjimas.

Pavadinimas	Reikalingas sudėtingumas	Naudojimo dažnumas	Naudojimo atvejai
Tvarkyklės slaptažodis	Didelis Sunku prisiminti	Retas	Pirmąkart autentifikuojant su konkrečiu klientu
Nustatymų slaptažodis	Didelis Sunku prisiminti	Labai retas	Keičiant slaptažodžių tvarkyklės nustatymus, peržiūrint žurnalą
Prieigos kodas	Vidutinis Sunku prisiminti	Dažnas	Kaskart prieinant prie konkretaus slaptažodžių rinkinio
Paskyros identifikatorius	Mažas Lengva prisiminti	Dažnas	Kaskart prieinant prie konkretaus slaptažodžio

2 lentelė: Informacijos kiekis, kurį reikia prisiminti vartotojui



4 pav. Slaptažodžių gavimo iš tvarkyklės veismo schema

Dažniausiai slaptažodžių tvarkykle atliekamas veiksmas – slaptažodžių gavimo iš tvarkyklės – yra palyginti nesudėtingas ir nesunkus išmokti. Tai iliustruoja 4 paveikslas.

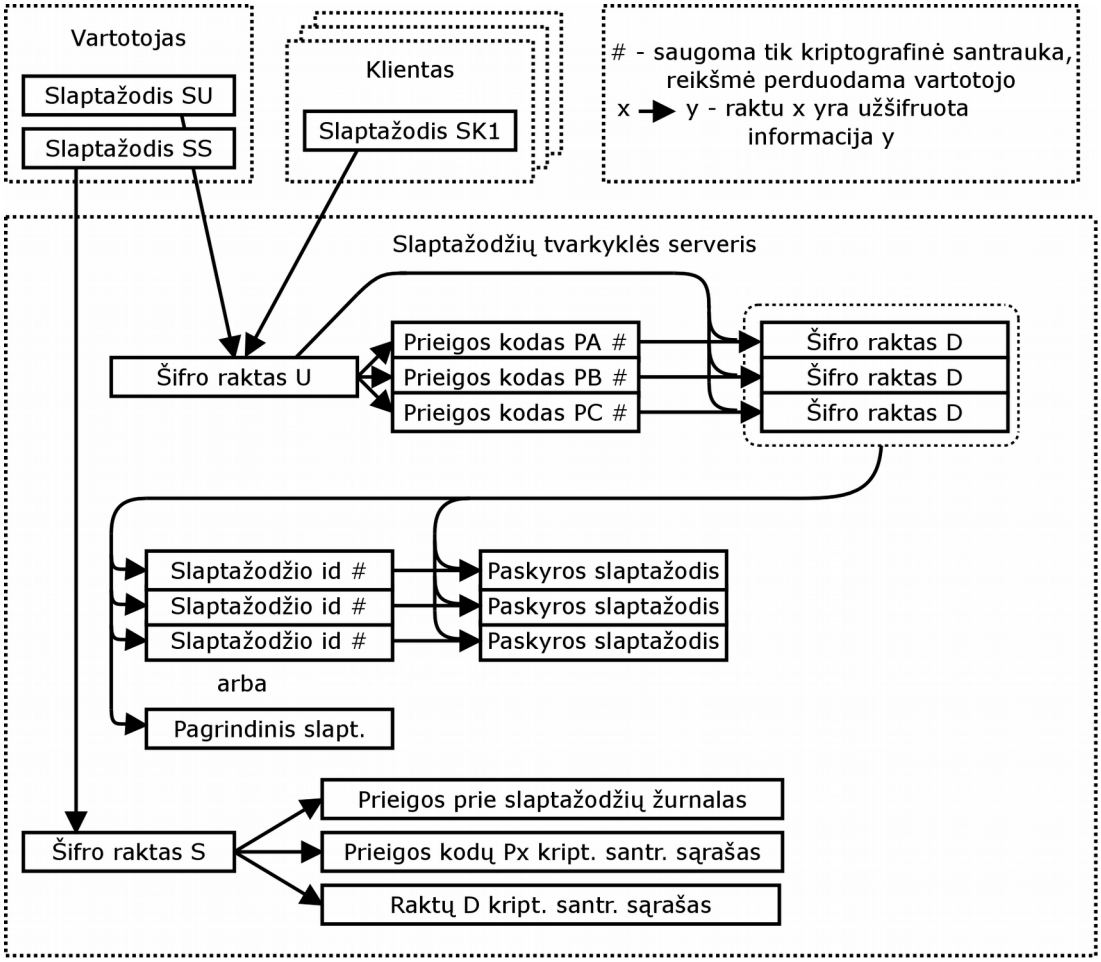
Slaptažodžių saugos sprendimui yra iškeltas uždavinys kiek įmanoma labiau sumažinti informacijos nuotekį net jei piktavalis turi priėjimą prie vartotojo aparatinės įrangos – tiek slaptažodžių tvarkyklės kliento, tiek serverio. Norint tai įgyvendinti nepakanka vien tik tinkamo vartotojo ar kliento autentifikavimo – duomenų bazėje esanti informacija turi būti šifruojama šifravimo raktais, kurie yra generuojami pagal vartotojo autentifikavimo metu suteikiamą informaciją.

Siūloma duomenų šifravimo schema yra pateikta 5 paveiksle. Šifravimas atliekamas naudojant AES algoritmą CCM režimu, kuris užtikrina ne tik duomenų konfidencialumą, bet ir integralumą. Duomenų bazėje nėra saugomi jokie vartotojo į serverį siunčiami autentifikavimo slaptažodžiai, tik jų kriptografinės maišos vertės. Vartotojo autentifikavimo veiksmo metu slaptažodžiai yra nusiunčiami į serverį ir leidžia jam iššifruoti jais užšifruotus duomenis. Žemiau pateikiamas detalus šifravimo ryšių sąrašas, kuris atvaizduotas 5 paveiksle:

- Visi vartotojui priskirtini su slaptažodžiais susiję duomenys saugykloje yra šifruojami šifro raktu U.
- Visi vartotojui priskirtini su tvarkyklės nustatymais susiję duomenys serveryje yra šifruojami šifro raktu S.
- Serveryje yra saugomos kelios užšifruotos rakto U kopijos: viena yra užšifruota tvarkyklės slaptažodžiu SU. Likusios – kliento slaptažodžiais $SK_1 ... SK_N$ po vieną kopiją kiekvienam klientui. Serveryje nėra saugomas slaptažodis SU, tik jo kriptografinės maišos reikšmė.
- Saugykloje saugoma viena rakto S kopija, kuri yra užšifruota nustatymų slaptažodžiu SS. Serveryje nėra saugomas slaptažodis SS, tik jo kriptografinės maišos reikšmė.
- Visa su konkrečiu slaptažodžių rinkiniu susijusi informacija yra šifruojama šifro raktu D. Kiekvienam slaptažodžių rinkiniui yra priskirtas atskiras toks šifro raktas.
- Saugykloje saugomos kelios užšifruotos rakto D kopijos – po vieną kopiją kiekvienam atitinkamo slaptažodžių rinkinio prieigos kodui P_X. Kiekviena rakto D kopija yra šifruojama raktu, kuris yra išvedamas iš atitinkamo prieigos kodo P_X ir šifro rakto S. Šifro raktas S yra naudojamas todėl, kad prieigos kodas turi nepakankamą entropijos kiekį.

Saugykloje nėra saugomos prieigos kodai P_X, tik jų kriptografinės maišos reikšmės, kurios yra užšifruotos raktu S.

- Jeigu slaptažodžių rinkinys yra paremtas pagrindiniu slaptažodžiu, pagrindinis slaptažodis saugykloje yra saugomas užšifruotas raktu D.
- Jeigu slaptažodžių rinkinys yra paremtas saugomais slaptažodžiais, saugykloje yra saugomos raktu D šifruotos paskyrų identifikatorių kriptografinės maišos reikšmės. Patys slaptažodžiai yra saugomi užšifruoti raktu, kuris yra išvedamas iš atitinkamo paskyros identifikatoriaus ir šifro rakto D. Šifro raktas D yra naudojamas todėl, kad paskyros identifikatorius turi nepakankamą entropijo kiekį.



5 pav. Slaptažodžių tvarkyklės duomenų saugyklos šifravimo schema

2.6. Autentifikavimas tarp slaptažodžių tvarkyklės kliento ir serverio

Ryšys tarp tvarkyklės kliento ir serverio dauguma atvejų vyks per nesaugų tinklą, todėl turi būti įgyvendintas kliento autentifikavimas serveriui, serverio autentifikavimas klientui ir ryšio tarp kliento ir serverio šifravimas. Autentifikavimui įgyvendinti egzistuoja galybė būdų, vieni iš jų labiau

saugūs, kiti – patogūs. Kaip minėta anksčiau, vartotojo patogumui skiriamas aukštas prioritetas, todėl tikslinga įgyvendinti kelis autentifikavimo mechanizmus, iš kurių būtų pasirenkamas patogiausias norimam saugumo lygiui įgyvendinti.

Norint išvengti impersonifikavimo atakų, visų pirma reikalingas serverio autentifikavimas klientui. Šios problemos sprendimas įgyvendinamas naudojant standartinius serverių autenfitikavimui naudojamus sprendimus – SSL sertifikatus. Jeigu serveris turi nekintamą IP adresą ar DNS įrašą, sertifikatų autentiškumas gali būti papildomai patvirtintas juos pasirašant tam tikram sertifikatų centrui.

Pirmo prisijungimo prie serverio metu slaptažodžių tvarkyklės kliento lange būtų rodomas SSL sertifikato identifikatorius (angl. *certificate fingerprint*), leidžiantis vartotojui įvertinti, ar jungiamasi prie norimo serverio nepriklausomai nuo to ar sertifiktas yra pasirašytas sertifikavimo centro (žr. 5 paveikslą). Tai reikalinga todėl, kad sertifikato pasirašymas negarantuoja, jog šis sertifikatas tikrai yra tinkamai išduotas: jau yra buvę atvejų, kai visuotinai patikimas šakninis SSL sertifikatas buvo sukompromituotas [37]. Po pirmo sėkmingo prisijungimo slaptažodžių tvarkyklės klientas išsaugo serverio SSL sertifikatą. Vėlesnių prisijungimų metu jis būtų sulyginamas su tuo metu serverio naudojamu SSL sertifikatu, taip išvengiant nepastebėtų sertifikato pasikeitimų.

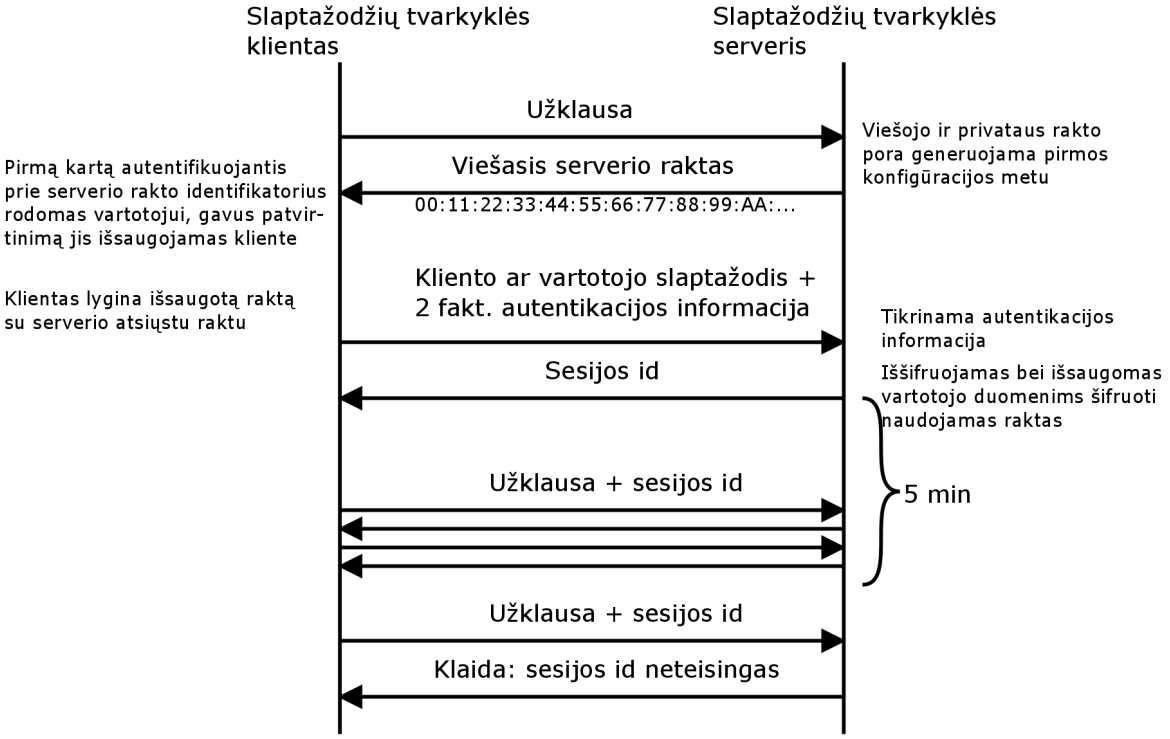
SSL sertifikatų naudojimas serverio autentiškumo užtikrinimui leidžia gerokai supaprastinti sprendimą ryšio tarp kliento ir serverio konfidencialumui ir integralumui užtikrinti – galima naudoti standartinį HTTPS protokolą.

Slaptažodžių tvarkyklės kliento autentifikavimas serveriui buvo detalizuotas 2.5 skyriuje analizuojant slaptažodžių saugą:

- vartotojo sugalvotas slaptažodis, naudojamas prieigai prie slaptažodžių.
- vartotojo sugalvotas slaptažodis, naudojamas prieigai prie tvarkyklės nustatymų.
- serverio kiekvienam klientui nusiųstas ir jame išsaugotas atsitiktinis ilgas slaptažodis. Kliento slaptažodis yra išsiunčiamas serverio pirmo sėkmingo autentifikavimo metu.
- dviejų dėmenų autentifikavimas naudojant pagalbinę programą mobiliajame telefone. Pirminiam autentifikavimui naudojamas kuris nors iš kitų dviejų autentifikavimo metodų.

Kadangi vartotojo slaptažodžiai yra naudojami ne tik autentifikavimui, bet ir duomenims šifruoti serverio pusėje, prieigą prie jų slaptažodžių tvarkyklės serveris turi turėti kiekvienos vartotojo operacijos metu. Tai įgyvendinama naudojant vartotojo prisijungimo sesijas: pavykus

vartotojo autentifikavimui, serveris sugeneruoja atsitiktinį sesijos identifikatorių S_{id}, iššifruoja ir išsaugo reikalingus duomenis pagal atsiųstą vartotojo slaptažodį, susieja juos su S_{id} ir atsiunčia S_{id} klientui. Sesijos metu klientas savo užklausas identifikuos S_{id}. Praėjus tam tikram laikui nuo sesijos sukūrimo, serveris ištrins S_{id} ir visą su juo susietą informaciją. Šis metodas leidžia sumažinti vartotojo slaptažodžių nutekėjimo galimybę lyginant su alternatyva saugoti juos slaptažodžių tvarkyklės kliente ir perduoti serveriui su kiekviena užklausa.



6 pav. Kliento-serverio autentifikavimo schema

Dviejų dėmenų autentifikavimui atlikti naudojamas laiku paremtas vienkartinių slaptažodžių algoritmas, TOTP (angl. *time-based one-time password algorithm*) [36]. Jis yra kiek pakoreguota kontroliniu parašu paremto vienkartinių slaptažodžių algoritmo, HOTP (angl. *HMAC-based one-time password algorithm*) [38], versija. Pastarasis algoritmas yra pagrįstas tos pačios paslapties K išsaugojimu slaptažodžių tvarkyklės serveryje ir kliento įrangoje, pavyzdžiui telefone. Vienkartiniai kodai generuojami pagal $HMAC-SHA1(K,C)$, kur C – serverio vartotojui atsiunčiamas kodas. Pagal šį algoritmą, tiek serveris, tiek vartotojo telefonas sugeneruoja 6 skaitmenų kodą, vartotojas šį kodą įveda į slaptažodžių tvarkyklės klientą, kuris kodą nusiunčia serveriui, o šis sulygina kodus ir patvirtina autentiškumą, jei kodai sutampa. Vartotojas tokį patį kodą gali sugeneruoti tik tada, jeigu jo telefone yra išsaugota ta pati paslaptis K ir vartotojas turi prieigą prie jo, tai yra šio sprendimo esmė. TOTP algoritmas nuo HOTP skiriasi tuo, kad C kodas yra išvedamas pagal tuo metu esantį

laiką serveryje ir vartotojo telefone taip, kad kiekvienas 30 sekundžių periodas atitiktų tą pačią C reikšmę. Šis algoritmas yra paprastas naudoti ir įgyvendinti, kadangi nereikia jokio papildomo ryšio kanalo tarp serverio ir vartotojo įrangos, visa informacija perduodama per vartotoją jam suvedant ar nuskenuojant kodus.

2.7. Slaptažodžių generavimas iš pagrindinio slaptažodžio

1.3.3 skyriuje buvo išnagrinėta slaptažodžių generavimo metodus aprašanti literatūra. Slaptažodžių iš pagrindinio slaptažodžio ir paskyros identifikatoriams generuoti pasirinktas metodas yra aprašomas šiame skyriuje.

Algoritmo paaiškinimo supaprastinimui apsibrėžkime funkciją $KDF(k,s,c,L)$, atliekančią slaptažodžio kriptografinį išvedimą iš dviejų dedamųjų k ir s su sudėtingumo parametru c ir išeities rakto ilgiu L. KDF įgyvendinimui gali būti naudojami bet kokie kriptografiškai saugūs slaptažodžių išvedimo algoritmai. Pavyzdžiui $PBKDF2(SHA256,k,s,c,L)$ arba $scrypt(k,s,c,r,p,L)$ su $r=1, p=1$.

Norint užtikrinti, kad piktavalis, žinantis vieną iš paskyrai sugeneruotų slaptažodžių, negalėtų atspėti pagrindinio slaptažodžio, naudojamas algoritmas, paremtas [26] aprašytų slaptažodžio generavimo iš dviejų dalių algoritmu. Šis yra patobulintas pakeičiant kriptografinės maišos funkciją į modernius slaptažodžių išvedimo įrankius. Slaptažodžio generavimas yra sudarytas iš dviejų dalių:

- Itin daug resursų reikalaujantis žingsnis, kuris atliekamas vieną kartą, o jo rezultatai išsaugomi lokaliai:

  $$V = KDF(\textit{vartotojo vardas}, \textit{pagrindinis slaptažodis}, j)$$

- Mažai resursų reikalaujantis žingsnis, naudojantis praėjusio žingsnio rezultatą.

  $$\textit{paskyros slaptažodis} = KDF(\textit{paskyros identifikatorius}, V, k)$$

Čia j ir k – KDF sudėtingumo parametrai, $j \gg k$, V – lokaliai išsaugoma reikšmė.

Naudojant šį algoritmą, slaptažodžio generavimas yra atliekamas dviem dalimis: lėta, vieną kartą atliekama dalis, kurios metu sugeneruojama ir išsaugoma tarpinė reikšmė V, ir greita dalis, kuriuos metu yra generuojamas paskyros slaptažodis. Kadangi pirmoji dalis yra atliekama vieną kartą, j galima padaryti kiek norima didelį taip nesudarant nepatogumų vartotojui. Šio algoritmo rezultatas yra tas, kad atspėti pagrindinį slaptažodį žinant vieną ar daugiau paskyros slaptožodžių yra nepalyginamai sunkiau net daug resursų turinčiam piktavaliui.

Kaip minėta, KDF įgyvendinimui gali būti naudojami bet kokie kriptografiškai saugūs

slaptažodžių išvedimo algoritmai. Slaptažodžių tvarkyklės prototipas, aprašytas 3 dalyje, įgyvendina tiek $PBKDF2(SHA256, k, s, c, L)$, tiek $scrypt(k, s, c, r, p, L)$ su $r=1, p=1$, kurios yra palyginamos 4 dalyje.

2.8. Slaptažodžių perdavimas jų reikalaujančioms programoms

Konfidencialus informacijos perdavimas tarp skirtingų programų jų nemodifikuojant yra sudėtingas praktinis uždavinys dėl skirtumų tarp įvairių platformų. Net ir nekonfidencialiai informacijai perduoti egzistuoja tik vienas universalus būdas – iškarpinė. Konfidencialiai informacijai perduoti tinkančio universalaus būdo nėra, todėl kiekvienai platformai šis funkcionalumas turėtų būti įgyvendinamas naudojant konkrečios platformos įrankius. Kiekvienas metodas, jo privalumai ir trūkūmai aprašomi žemiau.

- Iškarpinė. Metodas palaikomas visose platformose kaip langų tvarkyklės dalis. Metodo pagrindinis trūkumas yra tas, kad jo negalima naudoti konfidencialiems duomenims praktiškai visose platformose, nes bet kuri vartotojo teisėmis veikianti programa gali perskaityti iškarpinėje saugomus duomenis [39][40][41]. Saugumo požiūriu ši problema yra gerokai mažesnė, nei atrodo, kadangi kenkėjiškai programinei įrangai gavus priejimą prie vartotojo kompiuterio jo teisėmis, prie slaptažodžių prieiti galima ir kitais būdais. Iškarpinėje saugomus duomenis reikia saugoti tik nuo atsitiktinio paviešinimo naudojant programas, kurios skirtos iškarpinėje esančių duomenų tvarkymui. Puikus šios problemos pavyzdys yra KDE programų pakete esantis įrankis, automatiškai be vartotojo žinios diske išsaugantis 10 paskiausiai į iškarpinę įdėtų teksto gabaliukų [42].
- Iškarpinė naudojant turinio peradresavimą. Kai kuriose platformose egzistuoja metodai peradresuoti iškarpinės įklijavimo veiksmą, kai tam tikrai programai leidžiama nurodyti įklijavimui naudojamą tekstą vietoj tuo metu iškarpinėje esančių duomenų. Šis metodas yra ekvivalentus iškarpinei, išskyrus tai, kad jis leidžia apsaugoti slaptažodžius nuo atsitiktinio slaptažodžių nutekėjimo..
- Teksto surinkimas naudojant virtualų įvesties įrenginį. Kai kuriose platformose galima kurti virtualius įvesties įvykius, kurie iš praktinės pusės nesiskiria nuo įvesties įvykių, generuojamų klaviatūros paspaudimų metu. Šį metodą galima panaudoti įklijavimo veiksmui, kuris iš vartotojo pusės praktiškai nesiskirtų nuo iškarpine atliekamo įklijavimo. Metodo iškvietimui reiktų naudoti kitokią klavišų kombinaciją, negu priskirta iškarpinei. Metodui galioja tie patys saugumo trūkūmai, kaip ir iškarpinei, tačiau skirtingai nuo iškarpinės, jis nesudaro rizikos atsitiktiniam slaptažodžių nutekėjimui.

- Įskiepis konkrečiai programai. Šis metodas yra labiausiai invazyvus iš pasirinktų metodų, tačiau leiždia atlikti nuo atsitiktinio nutekėjimo apsaugotą slaptažodžio perdavimą bet kurioje platformoje. Jį įgyvendinti galima bet kuriai programai, palaikančiai įskiepius. Šiek tiek sudėtingesnis metodas, nereikalaujantis įskiepių palaikymo, yra dinaminės bibliotekos simbolių peradresavimas, kurio metu galima perimti iškarpinės įklijavimo veiksmą iš programos ir pateikti savo duomenis.

Visos stacionariuose kompiuteriuose naudojamos operacinės sistemos, kurias slaptažodžių tvarkyklė turi palaikyti, turi tiek pilną iškarpinės palaikymą, tiek programinio klaviatūros klavišų paspaudimų generavimo palaikymą naudojant virtualų įvesties įrenginį. Pirmasis šių metodų yra patogesnis vartotojui, tuo tarpu antrasis yra gerokai saugesnis, kadangi išnyksta galimybė nepiktybinėms iškarpinės turinį registruojančioms programoms jį išsaugoti ir potencialiai nutekinti. Antrojo metodo pagrindinė problema yra ta, kad vartotojas turi nurodyti slaptažodžių tvarkyklei kuriuo metu jis yra pasirinkęs slaptažodžio įvesties elementą, į kurį reikia suvesti slaptažodį. Paprasčiausias būdas tai atlikti yra unikali, vartotojo pasirenkama klavišų kombinacija, šis būdas ir naudojamas slaptažodžių tvarkyklėje. Slaptažodžių tvarkyklė turi leisti vartotojui naudoti abu nagrinėtus slaptažodžių perdavimo metodus pasirenkant norimą.

Toliau aprašomi konkretūs slaptažodžių perdavimo metodai, kuriuos galima naudoti tam tikroje platformoje:

- **Windows** platformoje iškarpinėje esantiems duomenims pakeisti naudojama `SetClipboardData` funkcija, o ištrinti iškarpinėje esančiam slaptažodžiui: `EmptyClipboard`. Klavišų simuliavimas atliekant `SendInput` WINAPI funkciją. Ši funkcija atlieka klavišų simuliavimą operacinės sistemos branduolio lygmenyje, todėl metodo saugumas yra toks pat, kaip įvedant slaptažodį iš klaviatūros. Norint naudoti klavišų simuliaciją, slaptažodžių tvarkyklei turi būti priskirtos papildomos teisės naudoti neįgaliesiems skirtų technologijų palaikymą (angl. *accessibility*). Prieigai prie iškarpinės papildomų teisių nereikia.
- **macOS** ir **OSX** platformose iškarpinei valdyti naudojamas `NSPasteboard` programinė sąsaja. Tuo tarpu klavišų simuliavimui naudojama `CGEventCreateKeyboardEvent` funkcija. Skirtingai nuo Windows, klavišų simuliavimas atliekamas ne operacinės sistemos branduolio lygmenyje, o kompozitoriuje. Nepaisant to, kadangi kompozitorius yra izoliuotas nuo vartotojo programų, metodo saugumas yra toks pat, kaip įvedant slaptažodį iš klaviatūros. Norint naudoti klavišų simuliaciją, slaptažodžių tvarkyklei turi būti priskirtos

papildomos teisės naudoti neįgaliesiems skirtų technologijų palaikymą.

- **Linux** platformoje iškarpinei valdyti naudojama X11 serverio sąsaja. Klavišų simuliacijai galima naudoti tiek X11 galimybes, tiek siųsti generuojamus paspaudimus per operacinės sistemos branduolį naudojant `uinput2` modulį. Linux atmainose, naudojančiose X11 serverį, slaptažodžių saugumo užtikrinti nėra įmanoma, kadangi bet kuri programa gali klausytis visų klaviatūros paspaudimų, todėl nėra prasmės naudoti `uinput2` modulį klavišų simuliacijai.

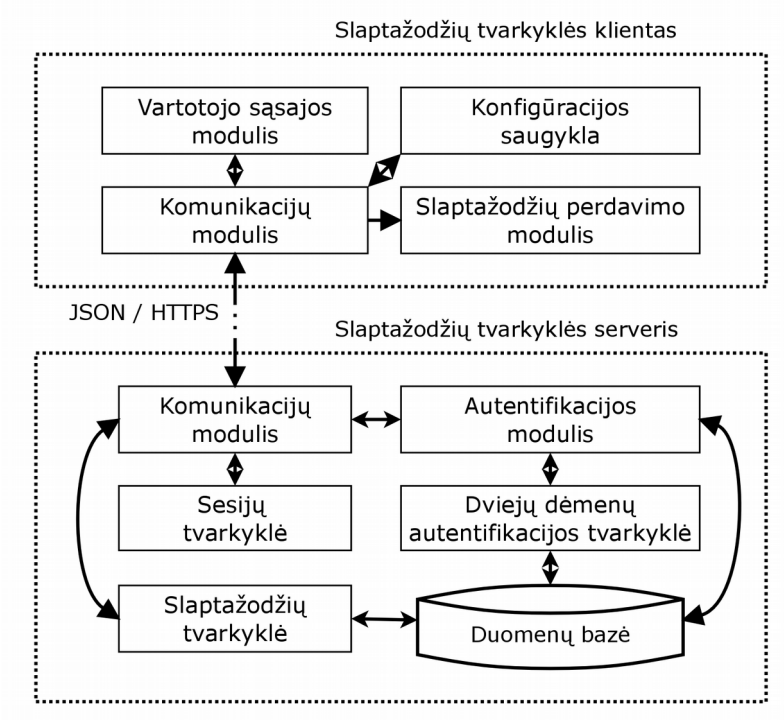
2.9. Išvados

- Atsižvelgiant į analizės dalyje identifikuotas siektinas patogios ir saugios slaptažodžių tvarkyklės savybes buvo sudarytas slaptažodžių tvarkyklės modelis.
- Slaptažodžių tvarkyklė geba tiek saugoti bet kokius slaptažodžius, tiek generuoti juos iš pagrindinio slaptažodžių.
- Slaptažodžius vartotojas gali suskirstyti į izoliuotus slaptažodžių rinkinius, kas leidžia sumažinti potencialų informacijos nuotekį.
- Slaptažodžių generavimo algoritmas yra atsparus bandymams atspėti pagrindinį slaptažodį net jeigu piktavalis turi didelius skaičiavimo resursus.
- Slaptažodžių tvarkyklė įgyvendinama naudojant kliento ir serverio modelį.
- Slaptažodžių saugykla yra šifruojama taip, kad tam tikrai informacijai iššifruoti yra reikalinga ta pati informacija, kaip ir autentifikavimui prie slaptažodžių tvarkyklės serverio.
- Informacijos kiekis, kurį vartotojui reikia prisiminti yra palyginti nedidelis. Prieigos kodai yra vienintelė infromacija, kuri yra ir sunkiai prisimenama, ir dažnai naudojama.

3. SLAPTAŽODŽIŲ TVARKYKLĖS REALIZACIJA

Antrame skyriuje apžvelgėme įvairius metodus ir algoritmus, kurie bus naudojami slaptažodžių tvarkyklei įgyvendinti. Šiame skyriuje apžvelgsime konkrečias realizacijos detales.

3.1. Slaptažodžių tvarkyklės realizacijos struktūra



7 pav. Sukurtos slaptažodžių tvarkyklės struktūra

Sukurtos slaptažodžių tvarkyklės realizacijos struktūra pavaizduota 7 paveiksle. Slaptažodžių tvarkyklės klientą sudaro šie moduliai:

- **Vartotojo sąsajos modulis** – atlieka informacijos pateikimą vartotojui ir leidžia vartotojui įvesti komandas. Šio modulio funkcionalumą galima įgyvendinti naudojant tiek grafinę, tiek komandinės eilutės sąsają.
- **Konfigūracijos saugykla** – saugo URL iki vartotojo naudojamo slaptažodžių tvarkyklės serverio. Taip pat, priklausomai nuo vartotojo pasirinktų parinkčių, saugo serverio tik šiam klientui sugeneruotą autentifikavimo raktą, kuris naudojamas vietoj vartotojo slaptažodžio kliento autentifikavimui.
- **Slaptažodžių perdavimo modulis** – atlieka iš slaptažodžių tvarkyklės serverio gautų slaptažodžių perdavimą to reikalaujančioms programoms, pagal programos tipą parenka saugiausią iš įgyvendintų perdavimo būdų.

- **Komunikacijų modulis** – jungiasi prie slaptažodžių tvarkyklės serverio, sudaro šifruotą ryšio kanalą, perduoda ir priima duomenis. Modulis atsakingas už ryšio su serveriu konfidencialumą ir integralumą. Serverio autentifikavimas yra atliekamas naudojant sertifikatus, kliento autentifikavimas yra atliekamas naudojant slaptažodžius.

Slaptažodžių tvarkyklės serverį sudaro šie moduliai (žr. 7 paveikslą):

- **Komunikacijų modulis** – atsakingas už ryšio su slaptažodžių tvarkyklės klientais palaikymą. Kadangi visos slaptažodžių tvarkyklės serverio paslaugos yra tiesiogiai įgyvendintos kituose moduliuose, nėra paslaugų tvarkyklės modulio, kuris organizuotų serverio darbą, vietoje to, šis funkcionalumas yra įgyvendintas komunikacijų modulyje.
- **Sesijų tvarkyklė** – atlieka aktyvių vartotojo sesijų monitoringą. Kiekvienai sesijai galima tam tikram laikui išsaugoti iš duomenų bazės iššifruotus vartotojo raktus, kad slaptažodžių tvarkyklės klientui nereikėtų vėl siųsti šiems raktams gauti reikalingos informacijos taip sumažinant konfidencialios informacijos judėjimo tinklu dažnumą.
- **Autentifikavimo modulis** – autentifikuoja vartotoją ir tvarko vartotojo informacijai užšifruoti duomenų bazėje naudojamus raktus.
- **Dviejų dėmenų autentifikavimo tvarkyklė** – saugo su vartotoju susietus antro dėmens raktus. Modulis atsakingas už teisingą vartotojo autentifikavimą pagal sugeneruotą vienkartinį kodą.
- **Slaptažodžių tvarkyklė** – modulis atsakingas už slaptažodžių saugojimą, generavimą ir prieigos prie slaptažodžių žurnalizavimą.

Slaptažodžių tvarkyklės prototipe buvo pasinaudota šiomis atviro kodo bibliotekomis:

- boost – didelis bendro pobūdžio bibliotekų rinkinys. Šios bibliotekos projekte naudojamos nuo platformos nepriklausančio failų sistemos ir tinklo ryšių funkcionalumui įgyvendinti
- Crypto++ – daug su kriptografija susijusių modulių įgyvendinanti biblioteka. Ši biblioteka projekte naudojama visiems kriptografinėms funkcijoms įgyvendinti.
- sqlite – duomenų bazių valdymo biblioteka. Projekte biblioteka naudojama įgyvendinti duomenų saugojimą duomenų bazėje.
- libscrypt – biblioteka, įgyvendinanti scrypt raktų išvedimo algoritmą
- Simple-Web-Server – biblioteka, įgyvendinanti supaprastintą HTTP/HTTPS serverį

C++ kalba naudojantis boost bibliotekos sąsajomis

- uri – biblioteka, skirta veiksmų su URI elementais palengvinimui

3.2. Duomenų bazės struktūra

2.5 skyriuje apžvelgtas slaptažodžių tvarkyklėje naudojamas abstraktus slaptažodžių saugos modelis. Jis numato, kad visa informacija duomenų bazėje yra šifruojama vienu ar kitu raktu. Tai komplikuoja duomenų bazės realizaciją, kadangi prie šifruotų duomenų neįmanoma efektyviai prieiti. Pavyzdžiui, neįmanoma naudoti įprastinio reliacinės duomenų bazės modelio, kai duomenys yra saugomi lentelėse ir yra susiejami tam tikrų langelių reikšmėmis.

Šiai problemai spręsti buvo pasirinktas rakto-reikšmės duomenų saugojimo modelis. Šifruotų duomenų saugojimui realizuoti pasinaudota kriptografinių maišos funkcijų vienkryptiškumo savybe. Visi užšifruoti duomenys yra saugomi naudojant šį šabloną:

- Šifruojamus duomenis pavadinkime D, šifro raktą pavadinkime K. Laikykime, kad K turi pakankamai entropijos. Taip pat, šifruojamiems duomenims priskirkime trumpą identifikatorių I, kuris nurodytų duomenų rūšį. Pavadinkime naudojamą kriptografinės maišos algoritmą $h(x)$, o šifravimo algoritmą $e(k,d)$.
- Duomenų bazės raktui yra naudojama $h(I\|K)$ arba $h(I\|h(K))$ rezultatas.
- Duomenų bazės reikšmei yra naudojama $e(K,D)$ rezultatas

Naudojant šį šabloną, žinant šifro raktą D ir duomenų rūšį I galima suskaičiuoti duomenų bazės rakto reikšmę, pagal ją efektyviai pasiekti duomenų bazėje išsaugotus duomenis ir juos iššifruoti. Toliau apraše norėdami aprašyti vieną pagal šį šabloną duomenų bazėje išsaugotą rakto-reikšmės porą naudosime notaciją $h(I\|K)\rightarrow e(K,D)$.

Kai kuriais atvejais duomenų bazėje norime išsaugoti keletą tuo pačiu šifro raktu D šifruojamų elementų $D_1..D_n$ kaip sąrašą, kur n – elementų skaičius. Tai įgyvendinti turime saugojant kiekvieną elementą kaip atskirą reikšmę, kadangi maksimalus duomenų bazėje saugomos reikšmės ilgis gali būti apribotas. Tokiam sąrašui saugoti yra naudojama šis šablonas:

- Išsaugojama rakto-reikšmės pora $h(I\|K\|\texttt{.num})\rightarrow e(K,n)$
- Kiekvienam elementui D_i išsaugojama rakto-reikšmės pora $h(I\|K\|i)\rightarrow e(K,D_i)$

Toliau norėdami aprašyti pagal šį šabloną duomenų bazėje išsaugotą elementų sąrašą naudosime notaciją $h_{sąrašas}(I\|K)\rightarrow e(K,D_{1..n})$.

3 lentelėje aprašoma konkreti naudojama duomenų bazės schema. Kaip minėta 2.5 skyriuje, duomenys yra šifruojami AES algoritmu CCM rėžime, o naudojama kriptografinė maišos funkcija yra SHA512. Naudojami tokie sutrumpinimai:

- IU – vartotojo identifikatorius slaptažodžių tvarkyklėje
- SU – vartotojo slaptažodis
- SS – vartotojo nustatymų slaptažodis
- U – šifro raktas, naudojamas vartotojo duomenims šifruoti
- SK – tam tikro kliento slaptažodis
- PK – prieigos prie tam tikro slaptažodžių rinkinio kodas
- PI – paskyros identifikatorius
- S – šifro raktas vartotojo nustatymams ir žurnalui šifruoti
- D – šifro raktas, naudojamas tam tikram slaptažodžių rinkiniui šifruoti
- $\check{Z}$ – raktas prieigai prie žurnalo eilutės

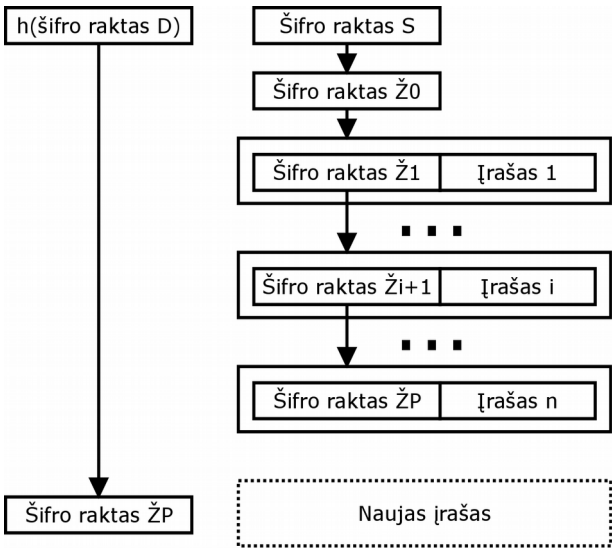
Darbe realizuotoje slaptažodžių tvarkyklėje kiekvienas duomenų bazės apraše nurodytas raktas yra saugomas su $h(UI)$ kaip priešdėliu (angl. *prefix*), kad būtų galima identifikuoti tam tikram vartotojui priklausančius duomenis ir įgyvendinti vartotojo ištrynimo operaciją. Paprastumo dėlei duomenų bazės apraše nagrinėjame vieno vartotojo atvejį ir šią informaciją praleidžiame.

Saugoma rakto-reikšmės pora	Paaiškinimai
$h(\texttt{userkey.}\|IU) \rightarrow e(SU, U)$	Prieiga prie U pagal vartotojo slaptažodį
$h(\texttt{userkey\_client.}\|h(SK_i)) \rightarrow e(SK_i, U)$	Prieiga prie U pagal klientų slaptažodžius
$h_{sąrašas}(\texttt{clientlist.}\|U) \rightarrow e(U, \{h(SK_i), pavadinimas\})$	Vartotojo klientų sąrašas. Žinant $h(SK_i)$, galima išrinti atitinkamą rakto-reikšmės porą iš `userkey_client` duomenų rinkinio.
$h(\texttt{settkey.}\|U) \rightarrow e(U\|SS, S)$	Prieiga prie S pagal vartotojo nustatymų slaptažodį
$h(\texttt{2fa.}\|U) \rightarrow e(U, \{raktas, nustatymai\})$	Prieiga prie dviejų dėmenų autentifikavimo nustatymų ir raktų
$h(\texttt{userkeymisc.}\|U) \rightarrow e(U, \{nustatymai\})$	Nekritinės vartotojo parinktys
$h(\texttt{settkeymisc.}\|S) \rightarrow e(S, \{nustatymai\})$	Nekritinės vartotojo parinktys
$h_{sąrašas}(\texttt{dbdesc.}\|S) \rightarrow e(S, \{h(D_i), slapt.rinkinio\,info.\})$	Informacija apie slaptažodžių rinkinius: tipas, pavadinimas.
$h(\texttt{db.}\|D) \rightarrow e(D, \{tipas, pagr.slapt.info.\})$	Slaptažodžio rinkinio tipas, pagrindinis slaptažodis, jeigu toks yra, su juo susijusi informacija
$h(\texttt{ac.}\|U\|h(PK)) \rightarrow e(U\|PK, \{tipas, D\})$	Prieiga prie D pagal prieigos kodą
$h_{sąrašas}(\texttt{acdesc.}\|h(D)\|S) \rightarrow e(S, \{h(PK_i), tipas, pavadinimas\})$	Prieigos kodų sąrašas, atitinkantis tikrą slaptažodžių rinkinį. Žinant $h(PK_i)$, galima šrinti atitinkamą rakto-reikšmės porą iš `ac` duomenų rinkinio.
$h(\texttt{pw.}\|D\|PI) \rightarrow e(D\|PI, \{slaptažodis\})$	Prieiga prie išsaugotų slaptažodžių
$h_{sąrašas}(\texttt{pwlist.}\|h(D)\|S) \rightarrow e(S, \{PK_i\})$	Sąrašas prieigos kodų, kada nors naudotų su tam tikru slaptažodžiu rinkiniu. Sąraše informacija nepilna: `pwlog` duomenų rinkinys tarp raktų $\check{Z}_{pirmas\,ne\,pwlist}$ ir $\check{Z}_{po\,paskutinio}$ turės papildomų prieigos kodų
$h(\texttt{pwlog.keys.}\|D\|PI) \rightarrow e(D\|PI, \{\check{Z}_{pirmas}, \check{Z}_{pirmas\,ne\,pwlist}\})$	Saugo raktus, pagal kuriuos galima skaityti prieigos prie slaptažodžių žurnalą
$h(\texttt{pwlog.}\|h(D)\|\check{Z}) \rightarrow e(\check{Z}, \{PK, \check{Z}_{sekantis}, info\})$	Saugo prieigos prie tam tikro slaptažodžių rinkinio žurnalą
$h(\texttt{pwlog.next.}\|h(D)) \rightarrow e(h(D), \{\check{Z}_{po\,paskutinio}\})$	Saugo raktą naujam įrašui į `pwlog` duomenų rinkinį

3 lentelė: Rakto-reikšmės duomenų bazės struktūra

Duomenų bazės struktūra yra palyginti nesudėtinga, išskyrus duomenų rinkinius,

detalizuojančius prieigos prie slaptažodžių žurnalą (duomenų rinkiniai `pwlist`, `pwlog.keys`, `pwlog`, `pwlog.next`). Pagrindinė sudėtingesnę struktūrą lėmusi priežastis yra ta, kad slaptažodžių tvarkyklės serveris turi gebėti perskaityti žurnalo įrašus tada ir tik tada, kai turi S raktą, tačiau rašyti į žurnalą jis turi gebėti ir tada, kai to rakto neturi. Šiai problemai spręsti naudojama šifravimo schema yra pavaizduota 8 paveiksle. Schemos esmė yra vienkrypčio tiesinio sąrašo (angl. *linked list*) formavimas. Pirmasis raktas $\check{Z}_0$ yra užšifruojamas raktu S, raktu $\check{Z}_i$ yra šifruojama i-tasis įrašas žurnale, kuriame yra išsaugotas sekantis raktas $\check{Z}_{i+1}$, kol įraše išsaugotas raktas $\check{Z}P$ rodo į neegzistuojantį įrašą, kuris dar tik bus sukurtas. Dar viena $\check{Z}P$ kopija yra užšifruota raktu $h(D)$. Tai leidžia turint raktą $h(D)$ prijungti naujus įrašus prie sąrašo išlaikant sąrašo struktūrą.



8 pav. Prieigos prie slaptažodžių žurnalo šifravimo schema

Naudojama informacijos saugojimo ir šifravimo duomenų bazėje schema leidžia užtikrinti ne tik tai, kad neturint reikalingų slaptažodžių neįmanoma prieiti prie vartotojo konfidencialios informacijos, bet ir tai, kad potencialus piktavalis negali sužinoti apie tos informacijos pasiskirstymą. Pavyzdžiui, laikykime, kad piktavalis turi prieigą prie slaptažodžių saugyklos ir žino vartotojo identifikatorių IU. Tai leidžia jam identifikuoti visą informaciją susijusią su vartotoju, tačiau ji tėra rinkinys kriptografinės maišos reikšmių ir jas atitinkančios šifruotos informacijos. Piktavalis negali įvertinti koks yra šios informacijos pasiskirstymas – kiek slaptažodžių rinkinių saugoma, kiek skirtingų slaptažodžių yra šiuose rinkiniuose ir pan.

Slaptažodžių tvarkyklės serveris naujo vartotojo sukūrimo metu į duomenų bazę įkelia atsitiktinį skaičių rakto-reikšmės porų $h(R) \rightarrow e(R,R)$, kur R - kiekvienai porai sugeneruotas skirtingas atsitiktinis skaičius. Dėl to potencialus piktavalis negali padaryti jokių užtikrintų prielaidų

ne tik apie konkrečiam vartotojui priskirtinos saugomos informacijos pobūdį, bet ir kiekį. Jeigu piktavalis nežino vartotojo nustatymų slaptažodžio SS, jis negali užtikrintai įvertinti net ar vartotojos naudojasi slaptažodžių tvarkykle, ar duomenų bazėje saugomi tik atsitiktinės vartotojo sukūrimo metu įkeltos rakto-reikšmės poros.

3.3. Išvados

- Realizuota antrame skyriuje aprašytą metodą įgyvendinanti slaptažodžių tvarkyklė.
- Slaptažodžių tvarkyklės klientas įgyvendina vartotojo sąsają, saugo autentifikavimo prie serverio slaptažodį ir perduoda slaptažodžius jų reikalaujančioms programoms
- Slaptažodžių tvarkyklės serveris saugo konfidencialią su vartotojo slaptažodžiais susijusią informaciją, jai šifruoti naudojamus raktus, vartotojo autentifikavimo informaciją ir jai šifruoti naudojamus raktus.
- Neturint informacijai iššifruoti reikalingų raktų, piktavaliui sudėtinga įvertinti saugomos informacijos kiekį ar pobūdį: slaptažodžių rinkinių, prieigos kodų prie jų ar jiems priklausančių slaptažodžių skaičių.

4. SLAPTAŽODŽIŲ TVARKYKLĖS TYRIMAS

Trečiame skyriuje apžvelgėme konkrečias antrame skyriuje pasiūlytos slaptažodžių tvarkyklės modelio realizacijos detales. Šiame skyriuje sukurta slaptažodžių tvarkyklė bus įvertinta dviem parametrais:

- Slaptažodžių tvarkyklės atsparumas saugumo grėsmėms. Įvardytos saugumo grėsmės, aktualios šiai slaptažodžių tvarkyklės realizacijai, analitiniu metodu bus įvertintas tvarkyklės atsparumas.
- Slaptažodžių tvarkyklės spartos įvertinimas. Bus atsakyta į klausimą, kokios minimalios spartos aparatinės įrangos reikalauja slaptažodžių tvarkyklės serveris ir ar įmanoma jį naudoti įdiegtus į nebrangius kompiuteriukus, tokius kaip Raspberry Pi.

4.1. Slaptažodžių tvarkyklės atsparumo saugumo grėsmėms įvertinimas

Modeliuojant slaptažodžių tvarkyklės atsparumą saugumo grėsmėms laikome, kad piktavalis, atakuojantis vartotoją, yra motyvuotas būtent šio vartotojo slaptažodžių tvarkyklėje saugomos informacijos išgavimu. Piktavaliai, ieškantys atsitiktinių spragų, gebės pasinaudoti žemiau aprašytų grėsmių poaibiu. Pasirinktos šias grėsmes ir jų kombinacijas:

- **Vartotojo fizinis stebėjimas**. Piktavalis gali fiziškai sekti vartotoją, arba stebėti jį vaizdo kameromis ar kitais metodais. Laikome, kad piktavalis potencialiai gali pamatyti viską, kas matoma vartotojo kompiuterio ekrane ir viską, ką vartotojas įveda klaviatūra.
- **Slaptažodžių tvarkyklės kliento programinis stebėjimas**. Piktavalis gali nepastebimai įsilaužti į vartotojo kompiuterinę įrangą, kurioje jis gali prieiti prie slaptažodžių tvarkyklės kliento ir nepastebimai stebėti vartotojo atliekamus veiksmus. Laikoma, kad piktavalis gauna administratoriaus teises ir gali stebėti šiuos duomenų šaltinius:
  - informacija, kuri yra įvedama klaviatūra
  - informacija, kuri yra rodoma ekrane
  - kietojo disko turinys ir pokyčiai jame
  - interneto srauto turinys
- **Fizinė prieiga prie slaptažodžių tvarkyklės kliento**. Šia grėsme apibrėžiame slaptažodžių tvarkyklės kliento programinio stebėjimo grėsmės atveju piktavalio galimų atlikti veiksmų

poaibį. Laikoma, kad piktavalis gali gauti pilną prieigą prie kompiuterio kietojo disko turinio tam tikru laiko momentu.

- **Fizinė prieiga prie vartotojo įrenginio naudojamo dviejų dėmenų autentifikavimui**. Laikome, kad piktavalis gali gauti pilną prieigą prie vartotojo mobiliojo telefono duomenų laikmenos turinio, įskaitant dviejų dėmenų autentifikavimui naudojamo rakto.
- **Slaptažodžių tvarkyklės serverio programinis stebėjimas**. Piktavalis gali nepastebimai įsilaužti į slaptažodžio tvarkyklės serveriui naudojamą kompiuterinę įrangą, kurioje jis gali prieiti prie slaptažodžių tvarkyklės serverio ir stebėti vartotojo atliekamus veiksmus. Laikoma, kad piktavalis gauna administratoriaus teises ir gali stebėti šiuos duomenų šaltinius:
  - kietojo disko turinys ir pokyčiai jame
  - interneto srauto turinys
- **Fizinė prieiga prie slaptažodžių tvarkyklės serverio**. Šia grėsme apibrėžiame slaptažodžių tvarkyklės serverio programinio stebėjimo grėsmės atveju piktavalio galimų atlikti veiksmų poaibį. Laikoma, kad piktavalis gali gauti pilną prieigą prie kompiuterio kietojo disko turinio tam tikru laiko momentu.

4 lentelėje yra pateiktas pasirinktų grėsmių sudėtingumo piktavaliui įgyvendinti ir tikimybės, kad grėsme bus bandoma pasinaudoti, įvertinimas. Nagrinėjame du atvejus:

- Piktavalis atsitiktinai gauna galimybę pasinaudoti grėsme prieš atsitiktinį asmenį. Vėliau tyrime laikysime, kad šiuo atveju praktiškai neegzistuoja tikimybė, kad piktavalis pasinaudos keliomis grėsmėmis vienu metu.
- Atkaklus piktavalis siekia pasinaudoti grėsme prieš konkretų asmenį. Vėliau tyrime laikysime, kad šiuo atveju egzistuoja nemaža tikimybė, kad piktavaliui pavyks pasinaudoti keliomis grėsmėmis vienu metu.

Grėsmė	Sudėtingumas atsitiktiniam piktavaliui, įvykio tikmybė	Sudėtingumas atkakliam piktavaliui, įvykio tikmybė
Vartotojo fizinis stebėjimas	Žemas, Tikėtina	Didelis, Mažai tikėtina
Slaptažodžių tvarkyklės kliento programinis stebėjimas	Vidutinis, Tikėtina	Vidutinis, Tikėtina
Fizinė prieiga prie slaptažodžių tvarkyklės kliento	Didelis, Mažai tikėtina	Labai didelis, Labai mažai tikėtina
Fizinė prieiga prie vartotojo įrenginio naudojamo dviejų dėmenų autentifikavimui	Vidutinis, Tikėtina	Didelis, Mažai tikėtina
Slaptažodžių tvarkyklės serverio programinis stebėjimas	Labai didelis, Mažai tikėtina	Labai didelis, Mažai tikėtina
Fizinė prieiga prie slaptažodžių tvarkyklės serverio	Didelis, Labai mažai tikėtina	Labai didelis, Labai mažai tikėtina

4 lentelė: Keliamų grėsmių įgyvendinimo sudėtingumas piktavaliams

Vartotojo fizinio stebėjimo grėsmę įgyvendinti ir atsitiktinai pamatyti vartotojo ekrane informaciją ar jo suvedamą slaptažodį yra palyginti nesudėtinga atsitiktiniam piktavaliui. Pakanka, kad jis turėtų prieigą prie kur nors viešoje vietoje, kurioje žmonės potencialiai naudojasi kompiuteriais, esančios vaizdo kameros siunčiamų duomenų. Kur kas didesnis sudėtingumas šia grėsme pasinaudoti atkakliam piktavaliui, kadangi autentifikavimo veiksmas yra palyginti nedažnas vartojant kompiuterį. Vadinasi, piktavalis turėtų sąlyginai ilgą laiką sekti vartotoją, kad pamatytų bent vieną į klaviatūrą suvedamą slaptažodį.

Slaptažodžių tvarkyklės kliento programinio stebėjimo grėsmę įgyvendinti piktavaliui reikėtų kokiu nors būdu programiškai įsilaužti į vartotojo kompiuterinę įrangą. Užsikrėtimas virusais ir kita kenkėjiška programine įranga yra ganėtinai dažnas reiškinys, o vartotojui užtenka suklysti vieną kartą, todėl laikome, kad šia grėsme yra atsitiktiniam piktavaliui yra vidutiniškai sunku pasinaudoti. Panaši situacija ir atkaklaus piktavalio atveju: yra nemažai žinomų atakų, kurios gali būti įgyvendinamos atkakliam piktavaliui būnant fiziškai netoli vartotojo. Pavyzdžiui, piktavalis galėtų bandyti įsilaužti į vartotojo WiFi tinklą arba bandyti apgauti vartotojo įrangą, kad ji prisijungtų prie piktavalio kontroliuojamo WiFi taško. Turint tiesioginę prieigą prie vartotojo interneto srauto galima išnaudoti kurį nors iš galybės naršyklių, jų įskiepių pažeidžiamumų ar kitokią spragą.

Fizinės prieigos prie slaptažodžių tvarkyklės kliento, serverio ar dviejų dėmenų autentifikavimui naudojamo įrenginio, gavimo grėsmės įgyvendinimui tiek atsitiktinis, tiek atkaklus piktavalis turėtų apvogti vartotoją. Tai yra palyginti rizikinga, kadangi tokiu atveju praktiškai

neegzistuoja tikimybė, jog vartotojas vagystės nepastebės, o nusikaltimo išaiškinimo tikimybė taip pat gerokai didesnė. Vienintelis skirtumas tarp šių trijų grėsmių yra tas, kad dviejų dėmenų autentifikavimas yra paprastai atliekamas telefonu, o jį vartotojas gali pamesti, kas padidina atsitiktinio pasinaudojimo tikimybę.

Slaptažodžių tvarkyklės serverio programinio stebėjimo grėsmę įgyvendinti, piktavaliui reikėtų kokiu nors būdu programiškai įsilaužti į vartotojo valdomą slaptažodžių tvarkyklės serverį. Tai turėtų būti itin sudėtinga, kadangi tikėtina, kad slaptažodžių tvarkyklės serveris bus vienintelis paslaugų paketas paleistas konkrečiame serveryje, dėl ko atakos paviršius bus itin mažas ir menka klaidų ir pasinaudojimo jomis tikimybė.

Matome, kad fizinė prieiga prie slaptažodžių tvarkyklės kliento visais atvejais yra sudėtingesnė ir duoda mažiau naudos, negu programinis kliento stebėjimas, todėl tolesnėje analizėje nagrinėsime tik programinį slaptažodžių tvarkyklės kliento stebėjimą.

Nagrinėdami potencialaus informacijos nuotekio iš slaptažodžių tvarkyklės dydį turime visų pirma atsižvelgti į tai, kad jis yra apribotas slaptažodžių tvarkyklės architektūros:

- Joks piktavalis negali automatiškai jungtis prie slaptažodžių tvarkyklės serverio, jeigu jis nežino vartotojo ar kliento slaptažodžio.
- Joks piktavalis jokiais atvejais negali iššifruoti jokios informacijos, esančios slaptažodžių rinkinyje, kurio prieigos kodai jam nėra žinomi.

Nagrinėjame šiuos informacijos nutekėjimo atvejus:

- Slaptažodžių, gautų iš slaptažodžių tvarkyklės kliento vartotojo ar jo įrangos stebėjimo periodu, nutekėjimas
- Slaptažodžių rinkinių, identifikuojamų daugkartiniais prieigos kodais, panaudotais vartotojo ar jo įrangos stebėjimo periodu, nutekėjimas, jeigu prieiga prie tvarkyklės neapsaugota dviejų dėmenų autentifikavimu.
- Slaptažodžių rinkinių, identifikuojamų vienkartiniais prieigos kodais, panaudotais vartotojo ar jo įrangos stebėjimo periodu, nutekėjimas, jeigu prieiga prie tvarkyklės neapsaugota dviejų dėmenų autentifikavimu.
- Slaptažodžių rinkinių, identifikuojamų daugkartiniais prieigos kodais, panaudotais vartotojo ar jo įrangos stebėjimo periodu, nutekėjimas, jeigu prieiga prie tvarkyklės apsaugota dviejų dėmenų autentifikavimu.

- Slaptažodžių rinkinių, identifikuojamų vienkartiniais prieigos kodais, panaudotais vartotojo ar jo įrangos stebėjimo periodu, nutekėjimas, jeigu prieiga prie tvarkyklės apsaugota dviejų dėmenų autentifikavimu.

Teorinio informacijos nuotekio iš slaptažodžių tvarkyklės nagrinėjimo rezultatai pateikiami 5 lentelėje. Joje yra pateiktas sąrašas grėsmių kombinacijų, pasinaudojimą kuriomis modeliuojame ir kokios informacijos nutekėjimą jos sąlygoja. Rezultatus galima interpretuoti taip:

- Norint nutekinti bet kokią informaciją iš slaptažodžių tvarkyklės, būtina programinė prieiga prie slaptažodžių tvarkyklės kliento arba serverio.
- Vartotojo fizinis sekimas jokiais atvejais neatneša naudos.
- Vienkartinių prieigos kodų naudojimas leidžia apsisaugoti nuo viso slaptažodžių rinkinio nutekinimo, jeigu piktavalis neturi programinės prieigos prie serverio.
- Dviejų dėmenų autentifikavimas padeda apriboti informacijos nuotekį, jeigu piktavalis neturi prieigos prie slaptažodžių tvarkyklės serverio.
- Slaptažodžių suskirstymas į slaptažodžių rinkinius leidžia visiškai apsaugoti slaptažodžių rinkinius, kurių prieigos kodai nėra žinomi piktavaliui.
- Palyginę su egzistuojančiomis slaptažodžių tvarkyklėmis aprašytomis 1.3 skyrelyje, galime teigti, kad darbe sukurtas sprendimas yra saugesnis tuo, kad leidžia naudotis dviejų dėmenų autentifikavimu ir suskirstyti slaptažodžius į slaptažodžių rinkinius.

	Atakos					Nutekanti informacija				
Eil. nr.	Vartotojo fizinis stebėjimas	Slaptažodžių tvarkyklės kliento programinis stebėjimas	Fizinė prieiga prie vartotojo įrenginio dviejų dėmenų autentifikavimui	Slaptažodžių tvarkyklės serverio programinis stebėjimas	Fizinė prieiga prie slaptažodžių tvarkyklės serverio	Slaptažodžiai, gauti iš kliento	Slaptažodžių rinkiniai, vienk. prieigos kodai, naudojamas 2 dėm. aut.	Slaptažodžių rinkiniai, daugk. prieigos kodai, naudojamas 2 dėm. aut.	Slaptažodžių rinkiniai, vienk. prieigos kodai, nenaudojamas 2 dėm. aut.	Slaptažodžių rinkiniai, daugk. prieigos kodai, nenaudojamas 2 dėm. aut.
1	nesvarbu	–	nesvarbu	–	nesvarbu	–	–	–	–	–
2	nesvarbu	taip	–	–	–	nuteka	–	–	–	iš dalies
10	nesvarbu	taip	taip	–	–	nuteka	–	iš dalies	–	iš dalies
24	nesvarbu	taip	nesvarbu	–	taip	nuteka	–	nuteka	–	nuteka
27	nesvarbu	nesvarbu	nesvarbu	taip	nesvarbu	nuteka	nuteka	nuteka	nuteka	nuteka

5 lentelė: Konkrečių atakų prieš slaptažodžių tvarkyklę nutekinama informacija

4.2. Slaptažodžių tvarkyklės spartos įvertinimas

Slaptažodžių tvarkyklės sparta tyrime buvo nagrinėjama pagal šiuos parametrus:

- Atskirų veiksmų trukmė priklausomai nuo naudojamų kriptografinių funkcijų slaptažodžių išvedimui ir pakartojimų parametrų joms.
- Atskirų veiksmų trukmė simuliuojant didelės delsos (angl. *latency*) kanalą tarp slaptažodžių tvarkyklės kliento ir serverio.

Tyrimui naudojamos kompiuterinės ir programinės įrangos parametrai pateikti 6 lentelėje.

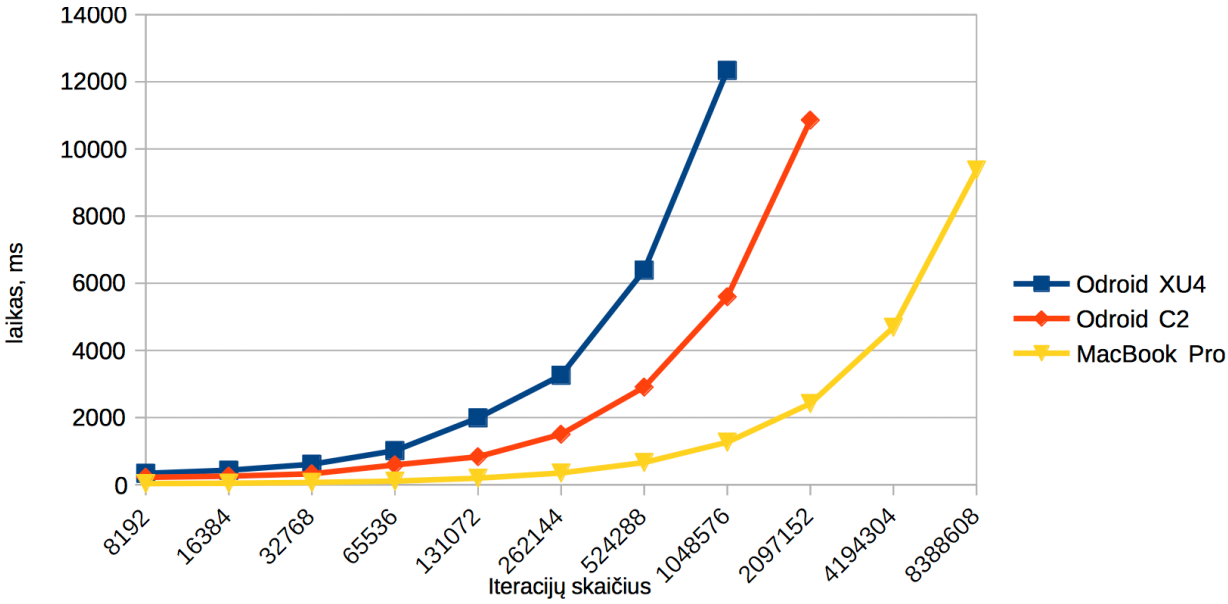
Nešiojamasis kompiuteris MacBook Pro	
Procesorius	Intel Core i7, 2.5GHz, 4 branduoliai, 8 gijos
Atmintis	16 GB DDR3 1600 MHz
Kietasis diskas	500GB Apple SSD
Grafikos procesorius	AMD Radeon R9 M370X 2GB ir Intel Iris Pro 1.5GB
Tinklo plokštė	AirPort Extreme ir Apple Thunderbold-Ethernet adapteris
Operacinė sistema	macOS Sierra
Testavimo platforma Odroid XU4	
Procesorius	Exynos5422 aštuonių branduolių procesorius – 4x Cortex A7 1.4GHz, 4x Cortex A15 2GHz
Atmintis	2GB LPDDR3 933 MHz
Kietasis diskas	Sandisk Ultra 16GB microSDHC kortelė 80MB/s
Grafikos procesorius	ARM Mali-T628 MP6 600MHz
Tinklo plokštė	Realtek RTL8153 USB3-Ethernet kontroleris
Operacinė sistema	Ubuntu 16.04 armhf
Testavimo platforma Odroid C2	
Procesorius	Amlogic S905 keturių branduolių procesorius – 4x Cortex A53 2.0GHz
Atmintis	2GB DDR3 912 MHz
Kietasis diskas	Sandisk Ultra 16GB microSDHC kortelė 80MB/s
Grafikos procesorius	ARM Mali-450 MP3 750MHz
Tinklo plokštė	Integruotas Gbit Ethernet kontroleris
Operacinė sistema	Ubuntu 16.04 arm64

6 lentelė: Tyrime naudotos aparatinės ir programinės įrangos aprašas

Didelės tinklo delsos simuliavimui buvo naudojamas “Network Link Conditioner” įrankis, platinamas kartu su macOS Sierra operacine sistema. Visų tyrimų metu visų branduolių Odroid XU4 ir Odroid C2 testavimo platformose taktiniai dažniai buvo nustatyti į 1.3GHz, o Cortex A15

branduoliai Odroid XU4 testavimo platformoje buvo išjungti.

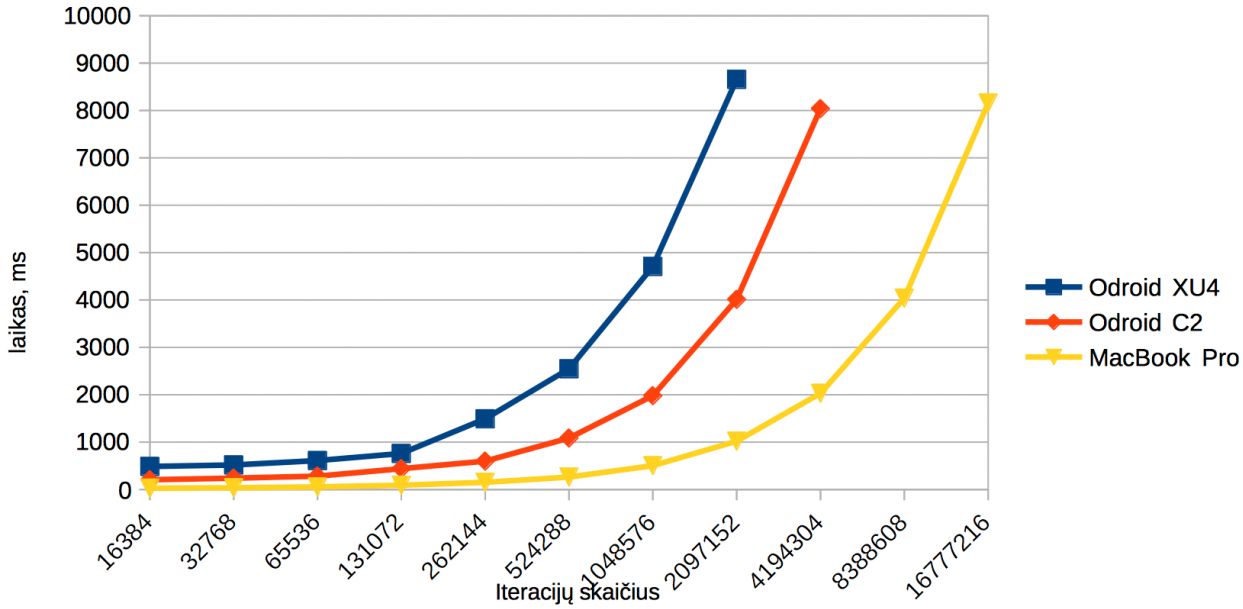
9 paveiksle pavaizduota pagrindinio slaptažodžio tarpinės reikšmės V (žr. 2.7 skyrių) skaičiavimo trukmė naudojant PBKDF2-HMAC-SHA256 slaptažodžių generavimo funkciją priklausomai nuo jos iteracijų parametro c. Jeigu laikome, kad vartotojui priimtina 10 sekundžių palaukti pirmą kartą konfigūruojant pagrindinį slaptažodį, tinkama c reikšmė yra apie 1 milijonas iteracijų.



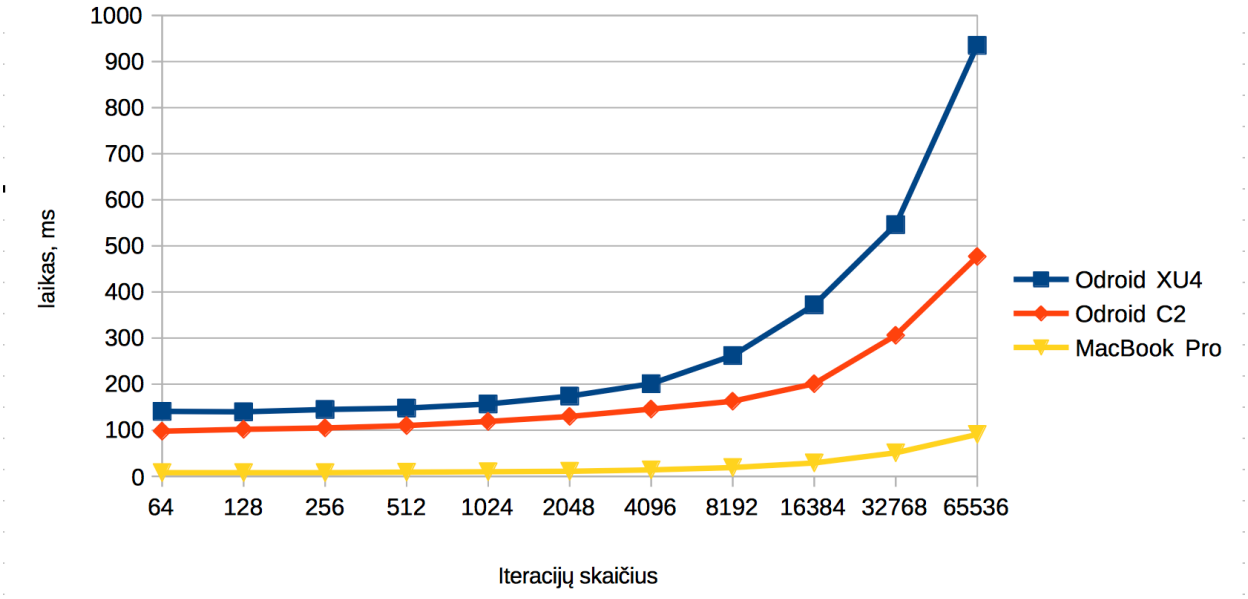
9 pav. Slaptažodžio tarpinės reikšmės generavimo trukmė naudojant PBKDF2-HMAC-SHA256 generavimo funkciją

10 paveiksle pavaizduota pagrindinio slaptažodžio tarpinės reikšmės V skaičiavimo trukmė naudojant scrypt slaptažodžių generavimo funkciją priklausomai nuo jos iteracijų parametro c. Jeigu laikome, kad vartotojui priimtina 10 sekundžių palaukti pirmą kartą konfigūruojant pagrindinį slaptažodį, tinkama c reikšmė yra apie 2 milijonai iteracijų.

11 paveiksle yra pavaizduota paskyros slaptažodžio generavimo iš pagrindinio slaptažodžio trukmė naudojant PBKDF2-HMAC-SHA256 slaptažodžių generavimo funkciją priklausomai nuo jos iteracijų parametro c. Matome, kad net ir esant mažam iteracijų parametrui c, Odroid XU4 kompiuteriukas užtrunka apie 140ms, Odroid C2 – 100ms, o MacBook Pro – 10ms. Tikėtina, kad tiek daug laiko užtrunka todėl, kad slaptažodžių tvarkyklė kiekvienam slaptažodžio generavimo veiksmui užpildo žurnalo įrašą ir laukia, kol duomenų saugojimo įrenginys, šiuo atveju, MicroSD kortelė, patvirtins įrašą.

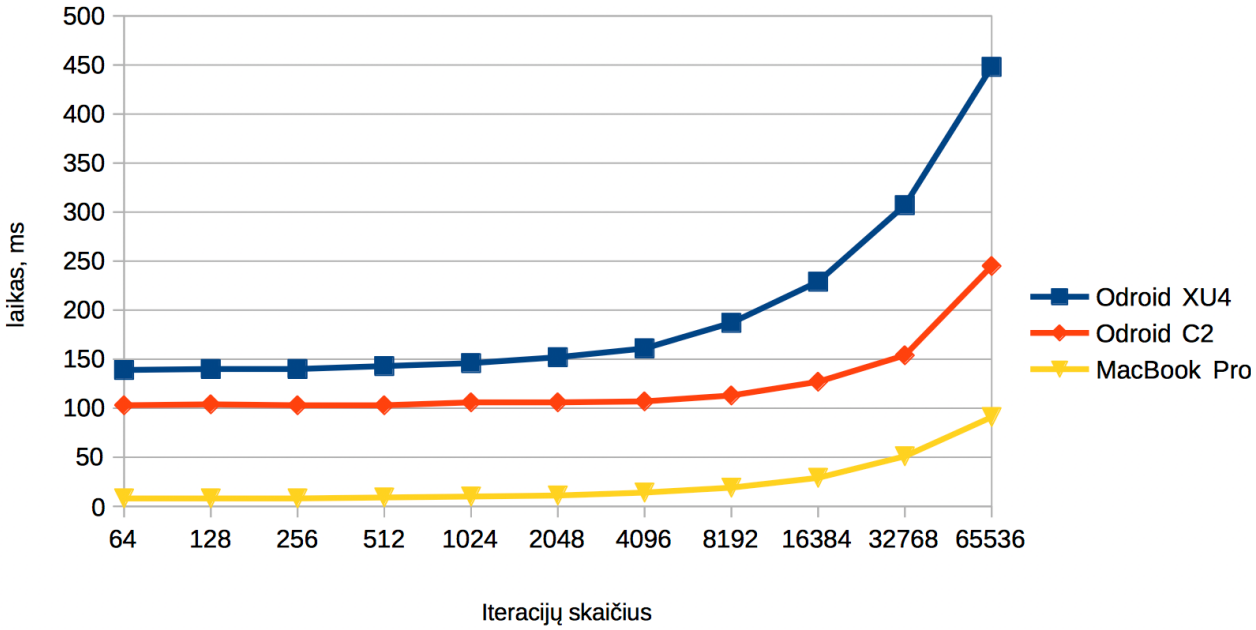


10 pav. Slaptažodžio tarpinės reikšmės generavimo trukmė naudojant scrypt generavimo funkciją



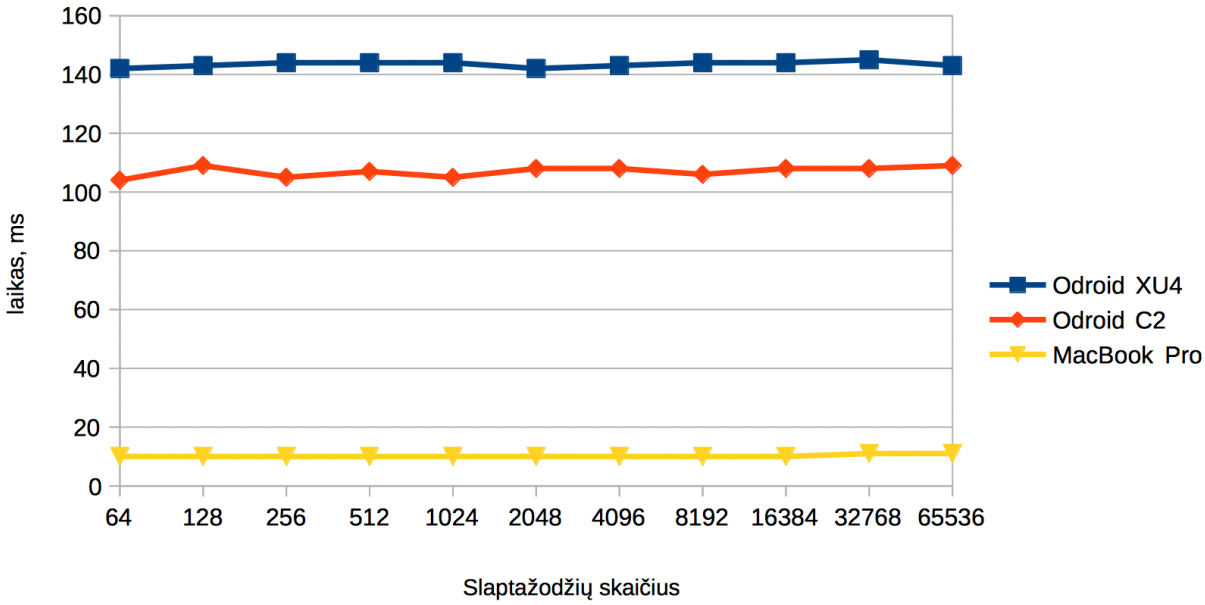
11 pav. Paskyros slaptažodžio generavimo trukmė naudojant PBKDF2-HMAC-SHA256 generavimo funkciją

12 paveiksle pavaizduota paskyros slaptažodžio generavimo iš pagrindinio slaptažodžio trukmė naudojant scrypt slaptažodžių generavimo funkciją priklausomai nuo jos iteracijų parametro c. Matome, kad esant mažam iteracijų parametrui c, operacijos trukmė atitinka PBKDF2-HMAC-SHA256 slaptažodžių generavimo funkcijos atvejį.



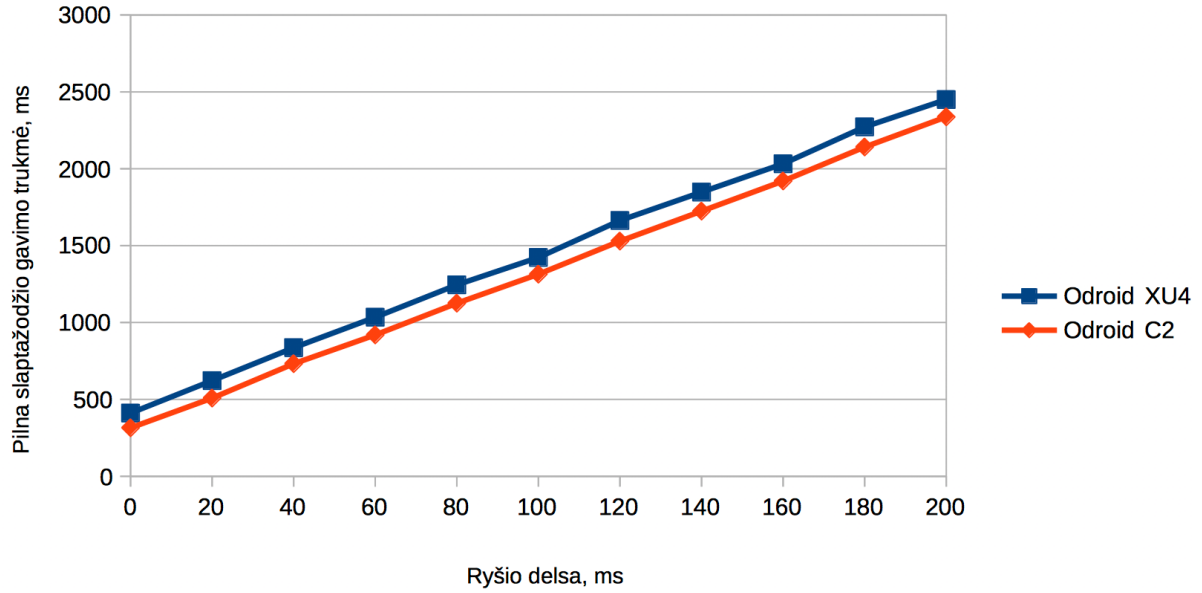
12 pav. Paskyros slaptažodžio generavimo trukmė naudojant scrypt generavimo funkciją

13 paveiksle pavaizduota išsaugoto paskyros slaptažodžio gavimo iš slaptažodžių tvarkyklės operacijos trukmė priklausomai nuo slaptažodžių tvarkyklėje jau saugomų slaptažodžių skaičiaus. Matome, kad operacijos trukmė nepriklauso nuo slaptažodžių skaičiaus nagrinėtame intervale – iki 65 tūkstančių slaptažodžių.



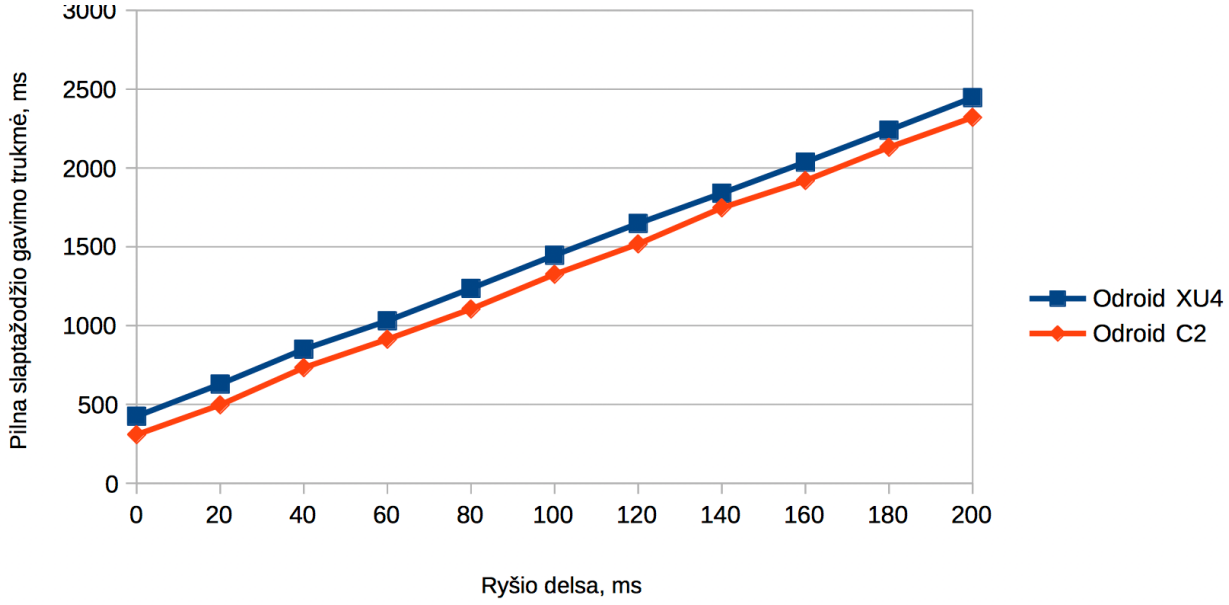
13 pav. Išsaugoto slaptažodžio gavimo operacijos trukmė priklausomai nuo duomenų bazės dydžio

14 paveiksle pavaizduota išsaugoto paskyros slaptažodžio gavimo iš slaptažodžių tvarkyklės operacijos trukmė priklausomai nuo tinklo delsos. Ryšys tarp slaptažodžių tvarkyklės kliento ir serverio įgyvendinamas HTTPS protokolu.



14 pav. Slaptažodžio gavimo trukmė priklausomai nuo tinklo delsos

15 paveiksle pavaizduota slaptažodžio generavimo iš pagrindinio slaptažodžio operacijos trukmė priklausomai nuo tinklo delsos. Ryšys tarp slaptažodžių tvarkyklės kliento ir serverio įgyvendinamas HTTPS protokolu.



15 pav. Slaptažodžio generavimo trukmė priklausomai nuo tinklo delsos

Analizuodami gautus duomenis, matome, kad turimos testavimo platformos slaptažodžio grąžinimo veiksmą atlieka 100-200ms nepriklausomai nuo slaptažodžių rinkinio tipo. Nors ši trukmė yra keliasdešimt kartų didesnė, nei tiriamo nešiojamojo kompiuterio, didžioji dalis vartotojo pusėje matomos veiksmo trukmės susidaro dėl tinklo delsos, jei ji yra bent 20ms. Matome, kad scrypt algoritmas yra labiau tinkamas pagrindinio slaptažodžio tarpinės reikšmės V skaičiavimui, kadangi

šios operacijos greičio skirtumas tarp testinių platformų ir nešiojamojo kompiuterio yra palyginti mažas – tik 2-4 kartai, o patį algoritmą sunku įgyvendinti naudojant grafikos ar programuojamos logikos procesoriu. Tinkamas iteracijų parametro c, atsižvelgus į tikslinę kompiuterinę įrangą, dydis turėtų būti 2 mln. Tuo tarpu galutinio paskyros slaptažodžio išvedimui yra labiau tinkamas PBKDF2-HMAC-SHA256, kadangi ši operacija yra paprastesnė, o pilnam slaptažodžio išvedimui reikalingi pajėgumai vis tiek daugiausia sunaudojami V skaičiuoti. Matome, jog net pasirinkus didelį tinklo delsos dydį, slaptažodžio gavimo veiksmą slaptažodžių tvarkyklė vis tiek atlieka greičiau, nei per 2 sekundes.

4.3. Rezultatų apibendrinimas

Apibendrinant tyrimo dalies rezultatus, galima padaryti šias išvadas:

- Sukurta slaptažodžių tvarkyklė dėl prieigos kodų naudojimo kai kuriais aspektais yra saugesnė, nei esami sprendimai. Be prieigos kodų neįmanoma prieiti prie jais apsaugotų slaptažodžių.
- Slaptažodžių tvarkyklę įdiegus į Raspberry-Pi klasės kompiuterinę techniką, operacijoms atlikti reikalingas laikas nėra per didelis, kad atbaidytų vartotoją: slaptažodžio gavimo veiksmas yra įvykdomas greičiau, nei per 200ms, o didesnės, nei 20ms delsos tinkle dominuos pastarojo sukeltas uždelsimas.
- Slaptažodžio tarpinės reikšmės V skaičiavimui geriausias scrypt algoritmas su iteracijų parametro c reikšme lygia 2^{20} (apie 2 mln.).
- Slaptažodžių tvarkyklė gali įgyvendinti slaptažodžių sinchronizaciją nenaudojant trečiųjų šalių paslaugų. Jos kodas yra platinamas pagal atviro kodo licenziją, todėl, skirtingai nuo egzistuojančių slaptažodžių tvarkyklių, bet kuris asmuo gali audituoti visus tvarkyklės aspektus.

5. IŠVADOS

1. Slaptažodžiai yra universalus autentifikavimo metodaus, naudojamas didžiojoje dalyje interneto svetainių dėl savo patogumo, paprastumo ir mažų įdiegimo sąnaudų.
2. Norint naudoti slaptažodžius atsižvelgiant į saugumo rekomendacijas, jie yra itin nepatogūs vartotojui. Šį trūkūmą galima išspręsti naudojant slaptažodžių tvarkykles.
3. Sukurtas hibridinis slaptažodžių tvarkymo metodas, kuris leidžia didžiąją dalį slaptažodžių generuoti ši pagrindinio slaptažodžio, o likusius išsaugoti slaptažodžių duomenų bazėje. Metodas paremtas kliento-serverio architektūra.
4. Slaptažodžiai yra suskirstomi į rinkinius, prieiga prie kurių yra ribojama naudojant papildomą autentifikavimo priemonę: prieigos kodus. Piktavalis negali prieiti prie rinkinių, kurių prieigos kodų nežino.
5. Slaptažodžių saugykla yra šifruojama taip, kad tam tikrai informacijai šifruoti yra reikalinga ta pati informacija, kaip ir autentifikavimui prie slaptažodžių tvarkyklės serverio.
6. Neturint informacijai iššifruoti reikalingų slaptažodžių, piktavaliui sudėtinga įvertinti saugomos informacijos kiekį ar pobūdį: slaptažodžių rinkinių, prieigos kodų prie jų ar jiems priklausančių slaptažodžių skaičių.
7. Slaptažodžių generuoti yra naudojamas algoritmas, atsparus bandymams atspėti pagrindinį slaptažodį net jeigu piktavalis turi didelius skaičiavimo resursus.
8. Informacijos kiekis, kurį vartotojui reikia prisiminti yra palyginti nedidelis. Prieigos kodai yra vienintelė infromacija, kuri yra ir sunki prisiminti ir dažnai naudojama.
9. Slaptažodžių tvarkyklę įdiegus į Raspberry-Pi klasės kompiuterinę techniką, operacijoms atlikti reikalingas laikas nėra per didelis, kad atbaidytų vartotoją: slaptažodžio gavimo veiksmas yra įvykdomas greičiau, nei per 200ms, o didesnės, nei 20ms delsos tinkle dominuoja pastarojo sukeltas uždelsimas. Ilgiausia potenciali operacijos trukmė yra apie 2 sekundes.
10. Slaptažodžio tarpinės reikšmės V skaičiavimui geriausias scrypt algoritmas su iteracijų parametro c reikšme lygia 2^{20} (apie 2 mln.).
11. Sukurta slaptažodžių tvarkyklė dėl prieigos kodų naudojimo kai kuriais aspektais yra saugesnė, nei esami sprendimai. Be prieigos kodų neįmanoma prieiti prie jais apsaugotų

slaptažodžių, kartu palaikomas dviejų dėmenų autentifikavimas. Slaptažodžių sinchronizacija gali būti įgyvendinama nenaudojant trečiųjų šalių paslaugų. Bet kuris asmuo gali audituoti visus tvarkyklės aspektus.

LITERATŪRA

[1] MENEZES A.J., P.C. VAN OORSCHOT, S.A. VANSTONE. Handbook of Applied Cryptography. CRC Press, 1996. ISBN 978-0849385230.

[2] YAN J., A. BLACKWELL, R. ANDERSON, A. GRANT. The memorability and security of passwords - some empirical results. 1993.

[3] Proactive Credential Monitoring. https://pwnedlist.com/. Prieiga 2017-05-10.

[4] HERLEY C., P.C. VAN OORSCHOT, A. S. PATRICK. Passwords: If We're So Smart, Why Are We Still Using Them? *Financial Cryptography*. 2009

[5] John the Ripper, Password cracker, http://www.openwall.com/john/. Prieiga 2016-01-11.

[6] DELL'AMICO M., P. MICHIARDI, Y. ROUDIER. Password Strength: An Empirical Analysis. *INFOCOM*. 2010

[7] BONNEAU J. The science of guessing: analyzing an anonymized corpus of 70 million passwords. *IEEE Symposium on Security and Privacy*, 538-552, 2012

[8] ADAMS A., M.A. SASSE. Users are not the enemy. Communications of the ACM. 42(12), 40-46. 1999

[9] HUITT W. The Information Processing Approach to Cognition. *Educational Psychology Interactive*, 2003

[10] FLORENCIO D., C. HERLEY. A large-scale study of web password habits. 2007

[11] GEBBER E., P.B. GIBBONS, Y. MATIAS, A. MAYER. How to make personalized web browsing simple, secure, and anonymous. 1997

[12] OLZAK T. Keystroke logging (keylogging). *SIN*. 2008

[13] JAGATIC T.N., N.A. JOHNSON, M. JAKOBSSON, F. MENCZER. Social phishing. *ACM*, 50, 94-100, 2005

[14] HERLEY C., P. VAN OORSCHOT. A research agenda acknowledging the persistence of passwords. 2012

[15] FLORÊNCIO D., C. HERLEY, Phishing and Money Mules, 2010

[16] HERLEY C. So Long, And No Thanks for the Externalities: The Rational Rejection of

Security Advice by Users. *NSPW*. 2009

[17] BOJINOV H., E. BURSZTEIN, X. BOYEN, D. BONEH. Kamouflage: Loss-resistant password management. 2010

[18] ROSS B., C. JACKSON, N. MIYAKE, D. BONEH, J.C. MITCHELL. Stronger Password Authentication Using Browser Extensions. 2005

[19] DUONG T., J. RIZZO, Flickr's API Signature Forgery Vulnerability, 2009

[20] KRAWCZYK H., M. BELLARE, R. CANNETTI, IETF RFC 2104: HMAC: Keyed-Hashing for Message Authentication. 1997

[21] KRAWCZYK H. Cryptographic Extraction and Key Derivation: The HKDF Scheme. *IACR Cryptology ePrint Archive*, 264, 2010.

[22] KALISKI B. IETF RFC 2898: PKGS #5: Password-Based Cryptography Specification Version 2.0, 2000

[23] DURMUTH M., T. GUNEYSU, M. KASPER, C. PAAR, T. YALCIN, R. ZIMMERMANN. Evaluation of Standardized Password-based Key Derivation against Parallel Processing Platforms. 2012

[24] PERCIVAL C., S. JOSEFSSON. IETF RFC 7914: The scrypt Password-Based Key Derivation Function. 2016

[25] ALWEN J., B. CHEN, K. PIETRZAK, L. REYZIN, S. TESSARO. Scrypt is Maximally Memory-Hard. 2016

[26] HALDERMAN J.A., B. WATERS, E.W. FELTEN. A Convenient Method for Securely Managing Passwords. 2005

[27] CHIASSON S., P.C. OORSCHOT, R. BIDDLE. A Usability Study and Critique of Two Password Managers. 2006

[28] ZHAO R., C. YUE. All Your Browser-saved Passwords Could Belong to Us: A Security Analysis and a Cloud-based New Design. *Proceedings of the third ACM conference on Data and application security and privacy*, 333-340. 2013. ISBN: 978-1-4503-1890-7.

[29] LastPass password manager. https://lastpass.com/. Prieiga 2017-05-10.

[30] ZHAO R., C. YUE, K. SUN. Vulnerability and Risk Analysis of Two Commercial Browser and Cloud Based Password Managers. *ASE*, 2013

[31] LI Z., W. HE, D. AKHAWE, D. SONG, The Emperor's New Password Manager: Security Analysis of Web-based Password Managers, 2014

[32] 1Password password manager, https://agilebits.com/onepassword. Prieiga 2017-05-10.

[33] GASTI P., K.B. RASMUSSEN. On The Security of Password Manager Database Formats. 2012

[34] KeePass password manager. http://keepass.info/. Prieiga 2017-05-10.

[35] Roboform password manager. https://www.roboform.com. Prieiga 2017-05-10.

[36] M'RAIHI D., S. MACHANI, M. PEI, J. RYDELL, IETF RFC 6238. TOTP: Time-Based One-Time Password Algorithm, 2011

[37] Microsoft Security Advisory 2607712, Fraudulent Digital Certificates Could Allow Spoofing, 2011.

[38] M'RAIHI D., M. BELLARE, F. HOORNAERT, D. NACCACHE, O. RANEN, IETF RFC 4226: HOTP: An HMAC-Based One-Time Password Algorithm, 2005.

[39] ZHANG X., W. DU, Attacks on Android Clipboard,

[40] N S P a s t e b o a r d k l a s ė s d o k u m e n t a c i j a , https://developer.apple.com/reference/appkit/nspasteboard. Prieiga 2017-05-10.

[41] Storing Data to and Reading from the Clipboard, https://msdn.microsoft.com/en-us/library/e2ft7ez4.aspx. Prieiga 2017-05-10

[42] RODRIGUES P., C. PFEIFFER, The Klipper Handbook, 2015