Kauno technologijos universitetas

Informatikos fakultetas

Patikimas atsarginis algoritmas dviejų faktorių autentifikacijos procesui

Baigiamasis magistro studijų projektas

Lukas Navašinskas

Projekto autorius

doc. Jonas Čeponis

Vadovas

Kaunas, 2026

Kauno technologijos universitetas

Informatikos fakultetas

Patikimas atsarginis algoritmas dviejų faktorių autentifikacijos procesui

Baigiamasis magistro studijų projektas

Informacijos ir informacinių technologijų sauga (6211BX008)

Lukas Navašinskas

Projekto autorius

doc. Jonas Čeponis

Vadovas

doc. Tomas Adomkus

Recenzentas

Kaunas, 2026

Kauno technologijos universitetas

Informatikos fakultetas

Lukas Navašinskas

Patikimas atsarginis algoritmas dviejų faktorių autentifikacijos procesui

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Lukas Navašinskas

Patvirtinta elektroniniu būdu

Navašinskas Lukas. Patikimas atsarginis algoritmas dviejų faktorių autentifikacijos procesui. Magistro studijų baigiamasis projektas vadovas dr. Jonas Čeponis; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Informacijos ir informacinių technologijų sauga.

Reikšminiai žodžiai: Sistema, 2FA, autentifikacija, antras faktorius, USB autentifikatorius.

Kaunas, 2026. 73 p.

Santrauka

Šiame magistro baigiamajame darbe sprendžiama atsarginio dviejų faktorių autentifikavimo (2FA) problema: kai naudotojas praranda telefoną su antro faktoriaus programėle, jis netenka prieigos prie savo paskyrų. Darbo tikslas - sukurti USB laikmenos pagrindu veikiantį atsarginį 2FA sprendimą, kuris naudoja asimetrinę kriptografiją ir veikia be papildomos programinės įrangos diegimo bei be administratoriaus teisių.

Darbe atlikta esamų 2FA sprendimų analizė ir nustatyti jų trūkumai. Suprojektuota ir įgyvendinta sistema iš keturių komponentų: autentifikavimo serverio, žiniatinklio programos, USB diegimo įrankio ir nešiojamos USB autentifikatoriaus programos. Sprendimas naudoja ECDSA P-256 raktų porą, kurios privatusis raktas saugomas tik USB laikmenoje, AES-256-GCM šifravimą su PBKDF2-HMAC-SHA256 (600 000 iteracijų) raktų saugyklai, Argon2id naudotojų slaptažodžių maišai ir JWT (RS256) prieigos žetonams. Iššūkio ir atsako protokolas pritaikytas pagal WebAuthn Level 2 specifikaciją: iššūkio galiojimas - 120 sekundžių.

Sprendimo saugumo lygis patikrintas penkiais eksperimentiniais scenarijais: USB kopijavimo, pakartojimo atakos, pasibaigusio iššūkio, neteisingo USB slaptažodžio ir pakeistų autentifikavimo duomenų. Visi atakų scenarijai atmesti, o teisėtas autentifikavimas užtruko apie 45 sekundes. Sukurta sistema atitinka Zero-Trust architektūros principus ir gali būti pritaikyta tiek komerciniams, tiek nekomerciniams poreikiams.

Navašinskas Lukas. Reliable Backup Algorithm for Two-factor Authentication Process. Master's Final Degree Project supervisor Dr. Jonas Čeponis; Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer Sciences, Information and Information Technology Security.

Keywords: system, 2FA, authentication, second factor, USB authenticator.

Kaunas, 2026. 73 pages.

Summary

This master's thesis addresses the problem of backup Two-Factor authentication (2FA): when a user loses the phone holding their second-factor application, access to their accounts is lost as well. The aim of the thesis is to create a USB-based backup 2FA solution that relies on asymmetric cryptography and works without installing additional software and without administrator privileges.

The thesis analyses existing 2FA solutions and identifies their shortcomings. A four-component system is designed and implemented: an authentication server, a web application, a USB deployment tool, and a portable USB authenticator application. The solution uses an ECDSA P-256 key pair, whose private key is stored only on the USB drive; AES-256-GCM encryption with PBKDF2-HMAC-SHA256 (600 000 iterations) for the key store; Argon2id for hashing user passwords; and JWT (RS256) for access tokens. The challenge-response protocol is adapted from the WebAuthn Level 2 specification: the challenge lifetime is 120 seconds.

The security level of the solution was verified through five experimental scenarios: USB cloning, replay attack, expired challenge, incorrect USB password, and tampered authentication data. All attack scenarios were rejected, while a legitimate authentication took approximately 45 seconds. The resulting system complies with Zero-Trust architecture principles and can be adapted for both commercial and non-commercial use.

Turinys

Lentelių sąrašas **7**
Paveikslų sąrašas **8**
Santrumpų ir terminų sąrašas **9**
Įvadas **10**
1. Dviejų faktorių autentifikavimo (2FA) ir atsarginių algoritmų problema **12**
1.1. Dviejų faktorių autentifikavimo sistemų saugos apžvalga 12
1.2. Dviejų faktorių autentifikavimo standartai ir tipai 14
1.3. Atsarginiai autentifikavimo metodai 19
1.4. Esamų dviejų faktorių autentifikavimo sistemų pažeidžiamumas 21
1.5. Dviejų faktorių autentifikavimo atsarginių metodų realizacijos 24
1.6. Išvados ir tolimesni 2FA atsarginių algoritmų tyrimo uždaviniai 25
2. Atsarginio dviejų faktorių autentifikavimo projektas **27**
2.1. Atsarginis 2FA algoritmas naudojant USB laikmeną 27
2.2. Patikimo atsarginio 2FA algoritmo topologija 30
2.3. Pagrindinis autentifikavimo procesas 36
2.4. Atsarginis autentifikavimo procesas 38
2.5. USB laikmenos 2FA paslaugos konfigūravimo procesas 40
2.6. Atsarginio dviejų faktorių projekto išvados 40
3. Atsarginio dviejų faktorių autentifikavimo prototipas **42**
3.1. Atsarginio USB 2FA prototipo realizacija 42
3.2. USB autentifikatoriaus diegimo prototipas 44
3.3. Autentifikavimo proceso realizacija naudotojo kompiuteryje 48
3.4. Nuotolinio autentifikacijos serverio prototipas 54
3.5. Prototipo realizacijos išvados 57
4. Eksperimentinis atsarginio 2FA algoritmo tyrimas **58**
4.1. Testavimo aplinka ir vertinimo kriterijai 58
4.2. Funkcinis prototipo patikrinimas 59
4.3. Patikimumo scenarijai 66
4.4. Pritaikomumo patikrinimas 69
4.5. Eksperimento išvados 69
5. Išvados ir rekomendacijos **71**
4. Literatūros sąrašas **72**

Lentelių sąrašas

1 lentelė. Populiariausių 2FA standartų naudojimo patirties (UX) palyginimas 18
2 lentelė. Populiariausių 2FA standartų pažeidžiamumo palyginimas 23
3 lentelė. Šifravimo algoritmų palyginimas 29
4 lentelė. Prototipe naudojami įrankiai ir technologijos 44
5 lentelė. Autentifikacijos API galutinių taškų specifikacija 54
6 lentelė. Autentifikacijos serverio duomenų bazės lentelės 56
7 lentelė. Testavimo aplinkos parametrai 58
8 lentelė. Patikimumo scenarijų rezultatų santrauka 69

Paveikslų sąrašas

1 pav. Kibernetinių atakų tendencijos tarp industrijų, matuojamos žalos (JAV doleriai) metrika.... 13
2 pav. „Google Prompt“ paspaudimo autentifikacijos metodas 15
3 pav. 2FA metodų autentifikacijos trukmė [6] 17
4 pav. „Yubico“ „Bio“ kartos U2F saugumo raktas 22
5 pav. Sistemos loginiai sluoksniai 28
6 pav. Patikimo atsarginio 2FA algoritmo topologija 31
7 pav. Patikimo atsarginio 2FA algoritmo topologija. Naudotojo sistema 32
8 pav. Patikimo atsarginio 2FA algoritmo topologija. Atvirkštinis tarpinis serveris 33
9 pav. Patikimo atsarginio 2FA algoritmo topologija. IAM serveriai 34
10 pav. Patikimo atsarginio 2FA algoritmo topologija. Replikų serveris 35
11 pav. Pagrindinis autentifikavimo procesas, srauto diagrama 37
12 pav. Atsarginis autentifikavimo procesas, srauto diagrama 39
13 pav. Sistemos realizacijos loginiai sluoksniai 42
14 pav. Prototipo architektūra 43
15 pav. USB autentifikatoriaus diegimo sekų diagrama 45
16 pav. USB autentifikatoriaus raktų šifravimo-saugojimo sekų diagrama 47
17 pav. Prototipo autentifikavimo proceso topologija 49
18 pav. USB 2FA programos Authenticator.exe paleidimo srauto diagrama 50
19 pav. USB slaptažodžio iššifravimo sekų diagrama 51
20 pav. USB autentifikavimo esybių diagrama 52
21 pav. USB 2FA prototipo autentifikavimo sekų diagrama 53
22 pav. USB diegimo įrankio pradinis langas. 59
23 pav. USB 2FA konfiguracijos langas 60
24 pav. USB konfiguracijos patvirtinimo langas 60
25 pav. Prisijungimo puslapis 61
26 pav. Pagrindinio (TOTP) neveikiančio 2FA langas 62
27 pav. Atsarginio USB 2FA langas 63
28 pav. USB autentifikatoriaus programos langas 64
29 pav. USB autentifikacijos puslapis 65
30 pav. Apsaugotas organizacijos puslapis 66
31 pav. USB autentifikacijos klaida 67

Santrumpų ir terminų sąrašas

Santrumpos ir terminai:

SFA - vieno veiksnio autentifikavimas (angl.: Single factor authentication);

MFA - kelių veiksnių autentifikavimas (angl.: Multifactor authentication);

TOTP - laiku pagrįstas vienkartinis slaptažodis (angl.: Time-based One-Time Password);

NFC - artimojo lauko ryšys (angl.: Near Field Communication). Trumpų atstumų belaidžio ryšio technologija, leidžianti įrenginiams keistis duomenimis labai mažu atstumu;

HMAC - kriptografinė funkcija, naudojama pranešimų integralumui ir autentiškumui užtikrinti naudojant slaptą raktą (angl.: Hash-based Message Authentication Code)[1];

HOTP - HMAC paremtas vienkartinis slaptažodis (angl.: HMAC-based One-Time Password);

U2F - universalus antras veiksnys (angl.: Universal 2nd Factor);

UX - naudotojo patirtis yra tai, kaip naudotojas sąveikauja su produktu, sistema ar paslauga. Tai apima asmens supratimą apie sistemos naudingumą, naudojimo paprastumą ir efektyvumą (angl.: User experience);

IAM - identiteto ir prieigos valdymas (angl.: Identity and access management): procesų, technologijų ir taisyklių rinkinys, skirtas centralizuotai valdyti naudotojų tapatybes, autentifikavimą ir autorizavimą, užtikrinant, kad kiekvienam naudotojui būtų suteikiama tik minimali būtina prieiga prie organizacijos išteklių;

SLA - paslaugų lygio susitarimas (angl.: Service level agreement) tarp paslaugos teikėjo ir naudotojo, kuriame nustatomi paslaugos prieinamumo, našumo, kokybės rodikliai bei atsakomybės ribos;

CORS - ištekliaus bendrinimas tarp skirtingų šaltinių (angl.: Cross-Origin Resource Sharing). Naršyklių saugumo mechanizmas, leidžiantis serveriams kontroliuoti, kurie domenai gali siųsti užklausas į jų išteklius. Tai apsaugo nuo neteisėto prieigos prie duomenų iš kitų svetainių, užtikrinant, kad tik patikimi šaltiniai galėtų pasiekti API ar kitus serverio išteklius;

Cross-site scripting (XSS) ataka – kenkėjiško kodo (dažniausiai „JavaScript“) injekcijos technika, kai užpuolikas įterpia savo scenarijų į patikimą tinklalapį. Tokiu būdu jis gali pasinaudoti naudotojo naršykle, pavogti slapukus, seanso duomenis ar atlikti veiksmus naudotojo vardu. XSS atakos kelia grėsmę tinklapių saugumui ir naudotojų privatumui;

AES-NI - aparatūrinio lygmens AES instrukcijų rinkinys (angl.: Advanced Encryption Standard New Instructions): procesoriaus instrukcijų plėtinys, skirtas pagreitinti AES šifravimo ir dešifravimo operacijas aparatinio lygmens vykdymu. AES-NI sumažina programinio šifravimo sąnaudas, padidina našumą ir sumažina pažeidžiamumą šoninių kanalų (angl.: side-channel) atakoms;

Nulinis pasitikėjimas (angl.: Zero-Trust) – kibernetinio saugumo modelis, kuriame kiekvienas prieigos bandymas laikomas potencialiai nesaugiu, todėl prieš suteikiant tik būtiniausią prieigą nuolat tikrinamos naudotojų tapatybės, įrenginiai ir kontekstas;

[1] HMAC funkcija: https://en.wikipedia.org/wiki/HMAC

Įvadas

Sparčiai besivystant informacinėms technologijoms ir vis dažnėjant kibernetinėm atakom, saugus ir patikimas autentifikacijos procesas tampa esminiu prioritetu tiek organizacijoms, tiek įvairių internetinių sistemų naudotojams. Viena iš populiariausių šiuolaikinių kibernetinio saugumo praktikų – dviejų faktorių autentifikacija (2FA)[1]. 2FA užtikrina aukštesnį saugumo lygį nei tradiciniai vieno faktoriaus metodai. 2FA sprendimai, pasitelkiantys papildomą saugumo sluoksnį, siekia apsaugoti naudotojus nuo neteisėtos prieigos prie jautrių duomenų ir sistemų.

Tačiau, kaip rodo moksliniai tyrimai [1,2,3,4,5,6], 2FA turi pažeidžiamumų. Pagrindinis antro faktoriaus autentifikacijos metodas gali tapti nepasiekiamas arba nebepatikimas dėl techninių trikdžių, naudotojo klaidų ar kibernetinių atakų. Esamų 2FA sprendimų trūkumai atskleidžia poreikį kurti ir diegti patikimus atsarginius algoritmus. Patikimas atsarginis algoritmas turi užtikrinti nenutrūkstamą autentifikacijos procesą net ir sąlygomis, kai sistemos patiria kibernetines atakas, ar kitus trikdžius, kurie padaro pirminį 2FA metodą, nepasiekiamu ar nepatikimu.

Šio darbo tikslas - išanalizuoti dviejų faktorių autentifikacijos spragas ir sukurti patikimą atsarginio autentifikavimo sprendimą. Sprendimas turi sumažinti rizikas, užtikrinti nenutrūkstamą sistemos veikimą ir išlaikyti aukštą naudotojo patirties (UX) kokybę. Tai ypač aktualu kritinėse sistemose – sveikatos priežiūros, finansų ar aviacijos srityse, kur autentifikavimo sutrikimai gali sukelti finansinių nuostolių arba kelti pavojų žmonių gyvybėms.

Tikslas ir uždaviniai

Pagrindinis projekto tikslas - suprojektuoti ir sukurti patikimą atsarginį autentifikavimo algoritmą dviejų faktorių autentifikacijos (2FA) procesui, kuris užtikrintų autentifikavimo tęstinumą sutrikus pirminiam 2FA metodui. Siekiant šio tikslo, sprendžiami šie uždaviniai:

1. Išanalizuoti esamas autentifikavimo metodų problemas ir pažeidžiamumus, daugiausia dėmesio skiriant gedimo ar nepasiekiamumo atvejams;
2. Sukurti veiksmingą alternatyvų autentifikavimo metodą, kuris galėtų pakeisti pirminį 2FA sprendimą kritinėse situacijose;
3. Išanalizuoti ir pasirinkti naudojamas technologijas užtikrinančias laisvą komercinį ir nekomercinį naudojimą;
4. Įgyvendinti eksperimentinę dalį, integruojant sukurtą atsarginį autentifikavimo algoritmą į testavimo aplinką;
5. Įvertinti sukurtą metodą bei palyginti rezultatus su esamais 2FA sprendimais.

Dokumento struktūra

Šis dokumentas sudarytas iš penkių pagrindinių skyrių:

1. Įvadas – pristatoma tema, aktualumas, tikslai ir uždaviniai, aprašoma dokumento struktūra;
2. Dviejų faktorių autentifikavimo (2FA) ir atsarginių algoritmų problema – aptariamos 2FA saugumo spragos, dažniausiai pasitaikančios problemos ir iššūkiai;
3. Dviejų faktorių autentifikavimo atsarginio autentifikavimo algoritmo pasiūlymas – pristatomas inovatyvus 2FA atsarginio metodo sprendimas;
4. Eksperimentinis dviejų faktorių autentifikavimo atsarginio algoritmo vertinimas – pateikiamas testavimo planas, atliktų testavimų rezultatai ir jų analizė;
5. Išvados ir rekomendacijos – apibendrinami darbo rezultatai, pateikiamos įžvalgos ir rekomendacijos tolimesniems tyrimams.

1. Dviejų faktorių autentifikavimo (2FA) ir atsarginių algoritmų problema

Dviejų faktorių autentifikavimas (2FA) yra vienas iš svarbiausių kibernetinio saugumo elementų, užtikrinančių jautrių duomenų apsaugą nuo neteisėtos prieigos. Dviejų faktorių autentifikavimas yra kelių faktorių autentifikavimo (angl.: Multi-Factor Authentication, MFA) atvejis, kai naudojami du nepriklausomi autentifikavimo veiksniai. MFA gali apimti tris ir daugiau veiksnių, tačiau praktikoje plačiausiai naudojamas būtent 2FA dėl balanso tarp saugumo ir naudojimo patogumo. Meyer ir kitų mokslininkų atliktas tyrimas, su dideliu Microsoft Azure Active Directory naudotojų duomenų rinkiniu, nustatė, kad kelių faktorių autentifikavimas sumažina paskyrų kompromitavimo riziką 99,22%, o daugiau nei 99,99% MFA apsaugotų paskyrų liko saugios tyrimo laikotarpiu[7]. Tačiau šiuolaikiniai 2FA sprendimai susiduria su reikšmingais iššūkiais, ypač tais atvejais, kai pirminis autentifikavimo metodas tampa nepatikimas ar nepasiekiamas. Šiame skyriuje aptariami 2FA privalumai ir trūkumai bei analizuojamas atsarginių algoritmų vaidmuo, siekiant užtikrinti autentifikavimo proceso patikimumą ir saugumą.

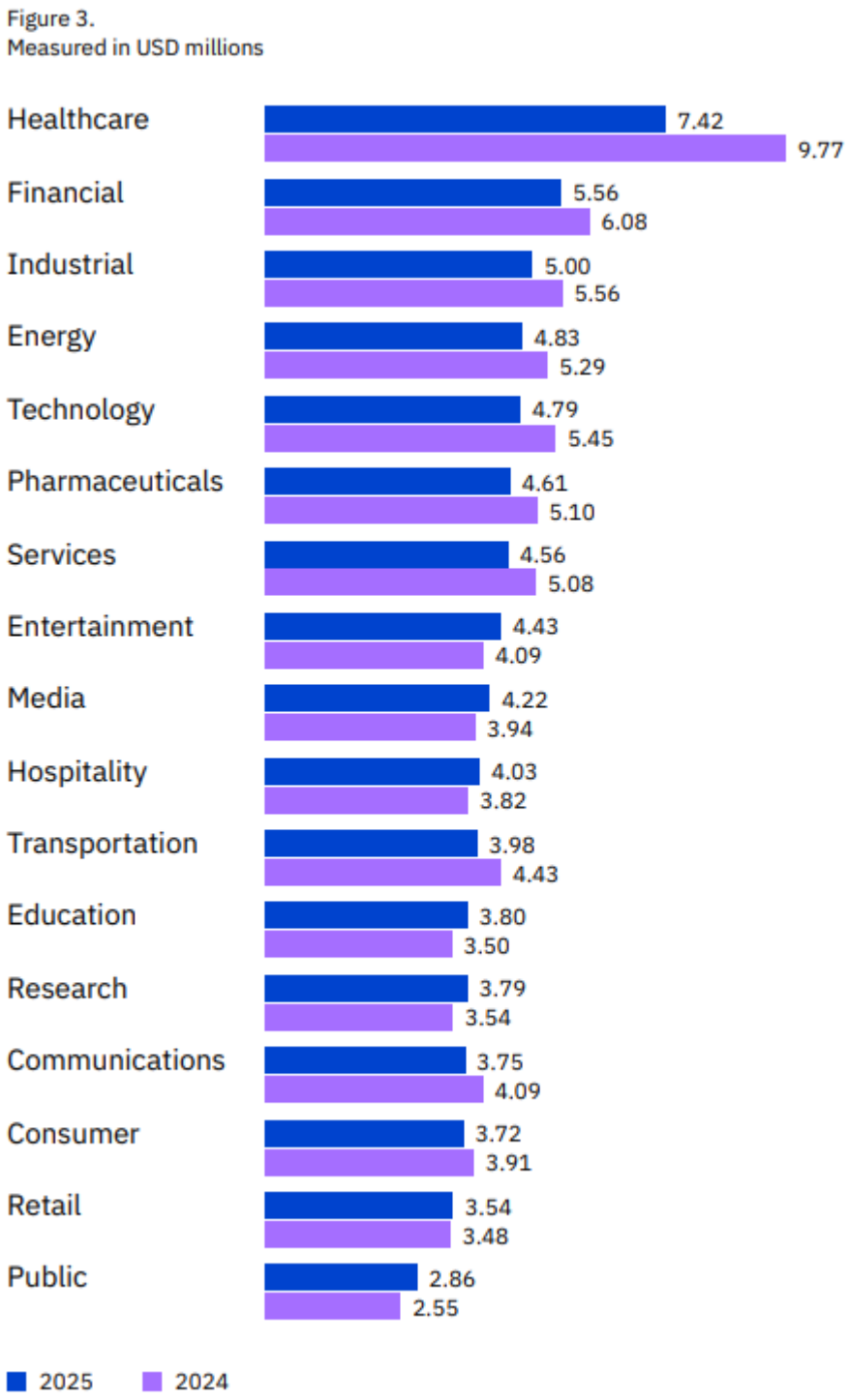
Nagrinėjami įvairūs problemos aspektai – nuo esamų 2FA standartų ir autentifikavimo metodų analizės iki jų pažeidžiamumo ir gedimų aptarimo. Taip pat nagrinėjamos atsarginių autentifikavimo metodų realizacijos galimybės, siekiant sumažinti grėsmes, kylančias dėl pirminių autentifikavimo metodų pažeidžiamumo.

1.1. Dviejų faktorių autentifikavimo sistemų saugos apžvalga

Nepaisant 2FA autentifikavimo efektyvumo, kibernetinių atakų mastas ir sukelti nuostoliai toliau auga. Pagal 2025 m. IBM „Cost of a Data Breach“ ataskaitą[2], vidutinė vieno duomenų pažeidimo kaina JAV pasiekė rekordinę 10,22 mln. JAV dolerių sumą. Sukčiavimas buvo dažniausiai pasitaikanti ataka (16 % visų pažeidimų), o trečiųjų šalių tiekimo grandinės kompromitavimas – antras pagal dažnumą (15 %). Straipsnyje „Countering Cybercrime Risks in Financial Institutions: Forecasting Information Trends“[8] taip pat minimas 2FA problemos aktualumas, ypač ryškus finansų sektoriuje, kuris yra vienas pagrindinių kibernetinių atakų taikinių.

Šie duomenys rodo, kad vien tik prevencinių autentifikavimo priemonių nepakanka – būtini ir atsarginiai mechanizmai, galintys užtikrinti autentifikavimo procesų tęstinumą gedimų ar atakų metu. Tačiau dauguma šiuolaikinių 2FA sprendimų tokių mechanizmų neįtraukia.

[2] IBM kibernetinių grėsmių tendencijų ataskaita: https://www.ibm.com/reports/data-breach



1 pav. Kibernetinių atakų tendencijos tarp industrijų, matuojamos žalos metrika[3]

Kaip matyti iš (1 pav.) IBM 2025 metų kibernetinių grėsmių ataskaitos - brangiausi duomenų pažeidimai fiksuojami sveikatos priežiūros sektoriuje (7,42 mln. JAV dolerių), o finansų sektoriaus organizacijos patiria antrą pagal dydį vidutinę pažeidimo kainą – 5,56 mln. JAV dolerių.

Nors šiuolaikiniai sprendimai padeda mažinti autentifikavimo riziką, dauguma jų orientuoti tik į prevenciją ir retai įtraukia atsarginius mechanizmus, galinčius užtikrinti autentifikavimo procesų veikimą gedimų ar atakų metu.

[3] IBM kibernetinių grėsmių tendencijų ataskaita: https://www.ibm.com/reports/data-breach

Iššūkis yra sukurti veiksmingą ir patikimą alternatyvų autentifikavimo metodą, kuris galėtų pakeisti pirminį 2FA procesą tuo atveju, jei jis taptų nepasiekiamas arba nepatikimas. Toks atsarginis metodas turi būti pakankamai lankstus, kad galėtų sklandžiai integruotis į esamą infrastruktūrą, išlaikant panašų saugumo ir naudojimo patogumo lygį kaip originalus 2FA.

1.2. Dviejų faktorių autentifikavimo standartai ir tipai

Paskyros, apsaugotos su dviejų faktorių autentifikacija, įprastai reikalauja dviejų nepriklausomų veiksnių. Įprastai šie veiksniai yra tai, ką vartotojas žino, pavyzdžiui, slaptažodis, ką jis turi: telefonas ar USB raktas, arba kas jis yra: biometriniai duomenys, tokie kaip piršto atspaudas ar veido atpažinimas [6].

Toliau nagrinėjami dažniausiai naudojami metodai dviejų faktorių autentifikacijos kontekste.

1.2.1. SMS kodai

SMS kodai, tai vienas populiariausių 2FA metodų, kai vartotojas gauna vienkartinį kodą į savo mobilųjį telefoną SMS žinute. Europos sąjungos skaitmeninės strategijos straipsnyje „Europiečių naudojimas ir požiūris į elektroninius ryšius ES“[4] teigiama, kad beveik visi europiečiai (96%) turi prieigą prie mobiliųjų telefonų, todėl šis autentifikavimo metodas yra plačiai naudojamas dėl paprasto diegimo ir vartotojų pažįstamumo su SMS.

Nors šis autentifikacijos tipas ir yra vienas populiariausių, jis turi saugumo spragų, kaip SIM kortelės keitimo pažeidžiamumas (angl.: SIM-swapping) ar tarpininko ataka (angl.: Man-In-The-Middle). [4,6]

1.2.2. TOTP – laiku pagrįsti vienkartiniai slaptažodžiai

Laiku pagrįsti vienkartiniai slaptažodžiai (angl.: Time-Based One-Time Passwords, TOTP) yra vienas iš populiariausių dviejų faktorių autentifikacijos metodų. Naudojantis įrankiais, kaip „Google Authenticator“ ar „Microsoft Authenticator“ sugeneruojamas laikinas kodas, paprastai sudarytas iš 6 arba 8 skaitmenų. Kodai įprastai keičiasi kas 30 sekundžių, ir generuojami remiantis laiko žyma bei slaptu raktu.

„Google Authenticator“ įrankis naudoja RFC 4226[5] aprašytą HOTP (angl.: HMAC-based One-Time Password, HOTP) algoritmą, šifruojant raktus HMAC-SHA-256 šifravimo funkcija. Šis metodas suteikia papildomą saugumo lygį, nes generuojami slaptažodžiai yra laikini, nereikalauja prieigos prie mobiliojo tinklo ir veikia tik nustatytą trumpą laiką. [4,9]

1.2.3. Iš anksto sugeneruoti kodai

Iš anksto sugeneruoti kodai, dažniausia naudojami kaip atsarginiai kodai, kuriuos vartotojas gali sugeneruoti iš anksto ir naudoti, jei kitos autentifikacijos priemonės tampa nepasiekiamos ar nepatikimos. Šie, dažniausia 8 skaitmenų ilgio kodai, sugeneruojami paslaugos tiekėjo ir naudotojas turi atsispausdinti, arba nusirašyti generuotus kodus.

Nepaisant autentifikacijos paprastumo, tokie kodai yra ypač pažeidžiami fizinių vagysčių atveju ar kopijavimo, nes tiek naudotojas, tiek paslaugos tiekėjas turi užtikrinti šių kodų saugumą. [6,9]

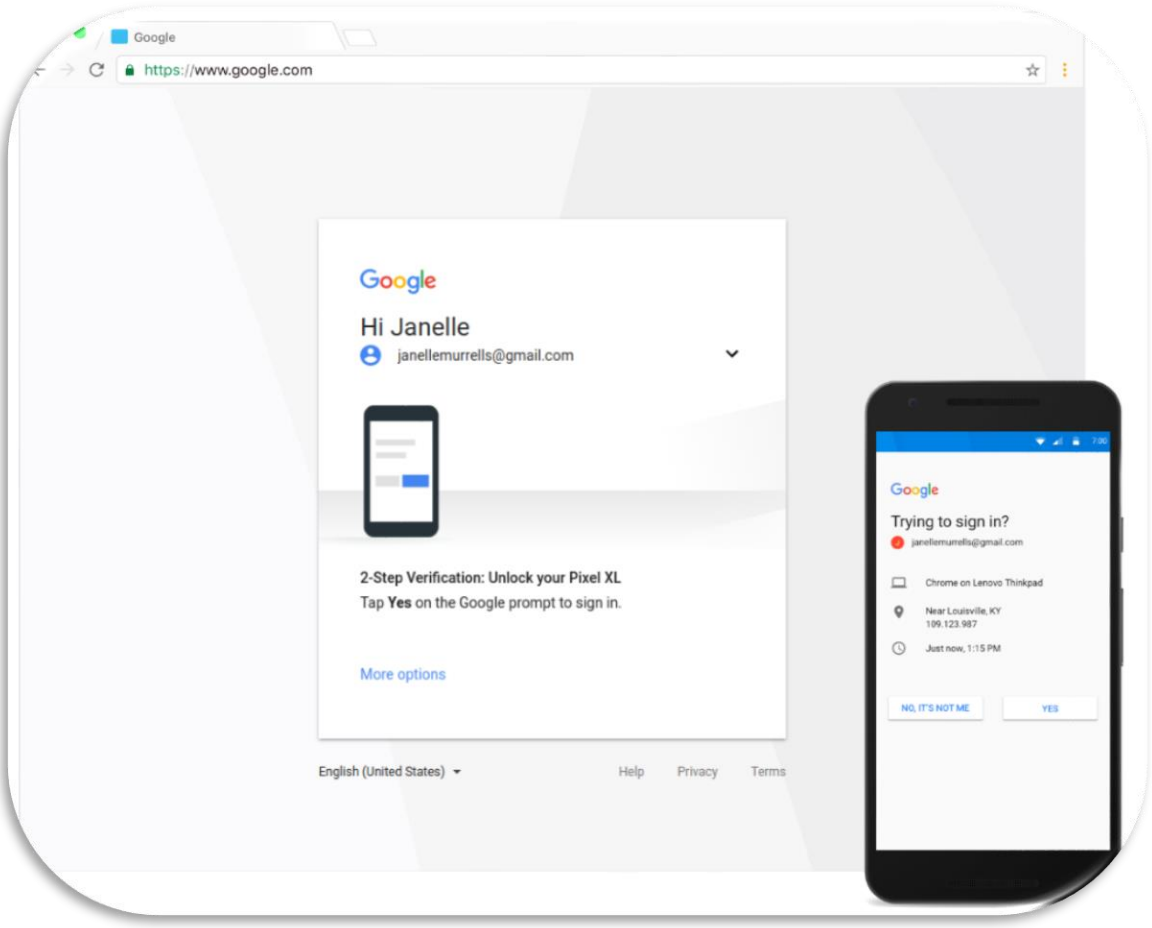
[4] Straipsnis „Europiečių naudojimas ir požiūris į elektroninius ryšius ES“: https://digital-strategy.ec.europa.eu/en/news/eurobarometer-europeans-use-and-views-electronic-communications-eu

[5] RFC 4226 (TOTP) algoritmo RFC aprašas: https://datatracker.ietf.org/doc/html/rfc6238

1.2.4. Paspaudimo autentifikacija

Paspaudimo autentifikacija yra autentifikacijos metodas, kai vartotojui siunčiamas pranešimas į išmanųjį mobilųjį įrenginį su galimybe patvirtinti arba atmesti prisijungimo bandymą (2 pav. „Google Prompt“ paspaudimo autentifikacijos metodas). Tai vienas iš patogiausių metodų, nes naudotojas neprivalo įvesti jokių kodų – autentifikacija vyksta vienu mygtuko paspaudimu.

Šis metodas populiarus dėl savo patogumo ir naudojimo paprastumo, ypač tarp organizacijų, diegiančių 2FA savo darbuotojų autentifikacijos procesuose. Sprendimai kaip „Duo Mobile“, „Google Prompt“ ar „Authy OneTouch“ yra plačiai naudojami dėl greito ir intuityvaus autentifikavimo proceso.



2 pav. „Google Prompt“ paspaudimo autentifikacijos metodas[6]

Paspaudimo autentifikacija turi kelis trūkumus: ji priklauso nuo interneto ryšio, todėl be tinklo nėra galimybės patvirtinti autentifikuotis, taip pat yra rizika prarasti įrenginį su aktyvia programėle, kas gali apsunkinti prisijungimą. Be to, nors duomenys ir yra šifruojami, senesnė įrenginio OS gali sudaryti sąlygas kenkėjiškiems asmenims bandyti nukreipti ar blokuoti pranešimus. [6,10,11]

1.2.5. U2F saugumo raktai

Universalus antrasis faktorius (angl.: Universal 2nd Factor, U2F) yra atviro standarto autentifikacijos metodas, kurį sukūrė „Google“ ir „Yubico“, o dabar jį remia FIDO aljansas (angl.: Fast IDentity Online). Šis metodas naudojamas kaip papildomas saugumo sluoksnis, kurį užtikrina specialūs

[6] GoogleBlog tinklaraščio straipsnis: https://workspaceupdates.googleblog.com/2017/10/making-google-prompt-primary-choice-for-2sv.html

fiziniai įrenginiai, tokie kaip USB, NFC arba „Bluetooth“ raktai. U2F autentifikacija pagrįsta viešojo ir privataus rakto kriptografijos metodu, kuris leidžia pasiekti aukštą apsaugos lygį.[12]

U2F autentifikacijos metode slaptasis raktas niekada nepalieka U2F įrenginio. Registracijos metu, viešasis raktas sugeneruojamas naudotojo ir saugomas serveryje. Tai reiškia, kad net jei užpuolikas bando apgauti naudotoją prisijungti prie netikro tinklalapio, U2F autentifikacija neveiks. Šie privalumai daro U2F ypač atsparų sukčiavimo atakom.

Nepaisant savo privalumų, U2F turi ir keletą trūkumų. Viena didžiausių rizikų yra įrenginio praradimas. Jei naudotojas netenka savo U2F įrenginio ir neturi nustatytų atsarginių autentifikacijos priemonių, jis gali prarasti prieigą prie savo paskyros. [6]

1.2.6. Biometriniai duomenys

Autentifikacija naudojant vartotojo biometrinius duomenis, tokius kaip piršto atspaudas, veido atpažinimas ar balso analizė, tampa vis labiau paplitusi dėl savo patogumo ir sunkiai atkuriamos unikalumo savybės.

Vienas iš pagrindinių biometrinių duomenų privalumų yra jų atsparumas tradicinėms slaptažodžių atakoms, tokioms kaip spėjimas ar sukčiavimas (angl.: phishing). Kadangi biometriniai duomenys yra unikalūs kiekvienam asmeniui ir negali būti lengvai kopijuojami, jie užtikrina aukštesnį saugumo lygį, lyginant su slaptažodžiais ar PIN kodais.

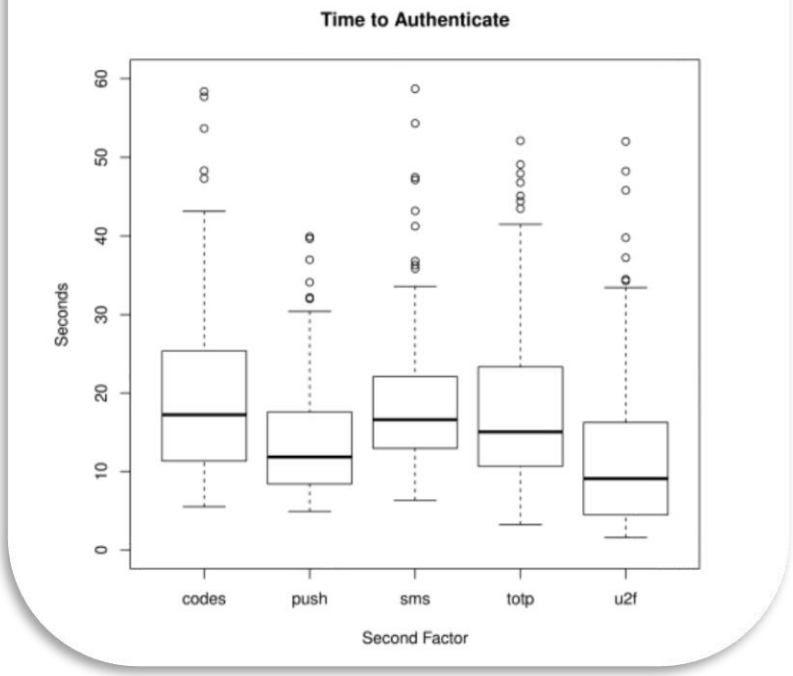
Viena didžiausių grėsmių naudojantis biomentriniais duomenimis pagrista autentifikacija yra duomenų nutekėjimas. Biometrinių duomenų šifravimo, perdavimo ir saugojimo metodai turi būti ypač saugūs, nes jei biometriniai duomenys būtų pavogti ar nutekinti – biometrinių duomenų neįmanoma pakeisti, kaip būtų galima pakeisti slaptažodį. [13,14]

1.2.7. Dviejų faktorių autentifikavimo standartų ir tipų apžvalga

Straipsnyje „A Usability Study of Five Two-Factor Authentication Methods“[6] aptariama kaip greitai veikia visi, **1.2** poskyriuje, išvardinti dviejų faktorių autentifikavimo standartai ir tipai, išskyrus biometrinių duomenų autentifikaciją.

Table 2: Authentication Time (seconds), Summary Statistics

Authentication Method	Q1	Median	Mean	Q3
Codes	11.3	17.2	28.0	25.4
Push	8.4	11.8	16.1	17.6
SMS	13.0	16.6	18.5	22.1
TOTP	10.7	15.1	23.9	23.3
U2F	4.5	9.1	13.0	16.3



3 pav. 2FA metodų autentifikacijos trukmė [6]

Atsižvelgiant į (3 pav. 2FA metodų autentifikacijos trukmė) lentelėje pateiktus medianos rezultatus, greičiausias iš visų dviejų faktorių autentifikacijos algoritmų yra TOTP (laiku pagrįsti vienkartiniai slaptažodžiai), kuriems vidutiniškai reikia tik 5,1 sekundės, kad naudotojas būtų autentifikuotas. Šis metodas yra ypač efektyvus dėl kodų generavimo naudotojo įrenginyje be poreikio išorinei komunikacijai.

Kiek lėtesnis, tačiau vis dar efektyvus autentifikavimo metodas, yra U2F (universalus antrasis faktorius), kuriam reikia 9,1 sekundės.

Paspaudimo autentifikacija užtrunka apie 11,8 sekundžių. Šis autentifikacijos metodas užtruka ilgiau, nes autentifikacijos procesas priklauso nuo interneto ryšio, sistemos apkrovos ir naudotojo, norint patvirtinti arba atmesti prisijungimo bandymą.

SMS ir iš anksto generuotų kodų autentifikacijos metodai yra lėčiausi. Naudojant šiuos metodus autentifikacija trunka apie 17 sekundžių. SMS dažniausiai užtrunka dėl mobiliojo tinklo veikimo greičio, o iš anksto sugeneruotų kodų autentifikacija gali būti lėtesnė dėl kodų saugojimo būdo, vietos, bei rankinio slaptažodžių vedimo.

1 lentelė. Populiariausių 2FA standartų naudojimo patirties (UX) palyginimas

2FA metodas	Naudojimo Patogumo lygis	Įdiegimo į sistemą patogumo lygis	Prieinamumo naudotojui lygis	Naudojimo greičio lygis	Bendro įvertinimo lygis (lygio vertės vidurkis)
SMS kodai	Aukštas	Aukštas	Aukštas	Žemas (~17 sek.)	Aukštas (2.5)
TOTP (laiku pagrįsti slaptažodžiai)	Vidutinis	Vidutinis	Vidutinis	Aukštas (~5,1 sek.)	Vidutinis (2.25)
Iš anksto sugeneruoti kodai	Žemas	Vidutinis	Žemas	Žemas (~17 sek.)	Žemas (1.25)
Paspaudimo autentifikacija	Aukštas	Aukštas	Vidutinis	Vidutinis (~11,8 sek.)	Aukštas (2.5)
U2F saugumo raktai	Vidutinis	Žemas	Žemas	Aukštas (~9,1 sek.)	Vidutinis (1.75)
Biometriniai duomenys	Aukštas	Vidutinis	Vidutinis	Aukštas	Aukštas (2.5)

Remiantis straipsniu: „A Usability Study of Five Two-Factor Authentication Methods“[6], lentelėje (1) vertinami šeši populiariausi dviejų faktorių autentifikacijos (2FA) metodai, atsižvelgiant į jų naudojimo patogumą, įdiegimo į sistemą paprastumą, prieinamumą naudotojams, naudojimo greitį. Kiekvienas metodas pasižymi savitais privalumais ir trūkumais.

SMS kodai yra vienas iš paprasčiausių ir plačiausiai naudojamų 2FA metodų. Šis 2FA metodas yra gerai žinomas ir lengvai naudojamas, taip pat nereikalauja sudėtingos infrastruktūros, o beveik visi vartotojai turi telefonus su SMS funkcionalumu, tačiau šio 2FA metodo naudojimo greičio įvertis yra žemas dėl SMS pristatymo trukmės ir būtinybės rankiniu būdu įvesti kodą.

TOTP 2FA metodas yra efektyvesnis greičio atžvilgiu, nes kodai pastoviai generuojami vartotojo įrenginyje, nepriklausomai nuo interneto ryšio kokybės. Tačiau šis metodas reikalauja išmaniojo telefono ir autentifikacijos programėlės, o įdiegimas į sistemą yra vidutiniškai sudėtingas. Naudojimo patogumas taip pat yra vidutinis, nes kodai turi būti įvedami rankiniu būdu.

Iš anksto sugeneruoti kodai yra labiausiai pažeidžiami ir nepatogūs naudotojui. Naudotojai turi užtikrinti, kad kodai būtų saugiai laikomi, nes jų praradimas gali lemti prieigos praradimą. Šio metodo įdiegimas yra nesudėtingas, tačiau lėtas naudojimo greitis ir priklausomybė nuo fizinio kodo laikymo daro jį mažiausiai efektyviu pasirinkimu.

Paspaudimo autentifikacija išsiskiria savo intuityvumu ir greitu autentifikavimo procesu, nes vartotojui tereikia patvirtinti užklausą vienu mygtuko paspaudimu. Šis metodas yra patogus naudotojui ir lengvai integruojamas su mobiliosiomis programėlėmis. Tačiau jo veikimas priklauso nuo stabilaus interneto ryšio, todėl jo prieinamumas vartotojui gali būti ribotas.

U2F saugumo raktai suteikia aukštą saugumo lygį ir greitą veikimą, tačiau reikalauja fizinio įrenginio, kurį vartotojas turi nuolat turėti su savimi. Šio metodo įdiegimas į sistemą yra sudėtingesnis, nes reikia kriptografinių raktų palaikymo serverio. Naudotojui šis metodas gali būti mažiau patogus dėl papildomo fizinio rakto poreikio.

Biometriniai duomenys, kaip pirštų atspaudai ar veido atpažinimas, yra patogiausi naudotojui. Šis metodas pasižymi aukštu naudojimo patogumu ir greičiu, kai naudojamas integruotas skaitytuvas. Vis dėlto, šio metodo įdiegimas yra vidutiniškai sudėtingas, nes reikalauja specialios įrangos, o prieinamumas vartotojui priklauso nuo įrenginio, turinčio biometrinių duomenų nuskaitymo funkciją.

Apibendrinant rezultatus, geriausiai įvertinti 2FA metodai yra SMS kodai, paspaudimo autentifikacija ir biometriniai duomenys. SMS kodai yra vertinami už paprastą diegimą ir aukštą prieinamumą. Paspaudimo autentifikacija išsiskiria savo intuityvumu. Biometriniai duomenys suteikia aukštą patogumo ir saugumo lygį, ypač naudojant integruotus skaitytuvus. Šie metodai yra rekomenduojami naudoti, kai reikia subalansuoti saugumą, greitį ir naudotojo patogumą.

1.3. Atsarginiai autentifikavimo metodai

Dabartinėje skaitmeninėje erdvėje dviejų faktorių autentifikavimas yra reikšmingas sistemų saugumo patobulinimas, lyginant su tradiciniu vieno veiksnio autentifikavimu (SFA) paremtomis sistemomis.

Kaip aptarta poskyryje: „1.2. Dviejų faktorių autentifikavimo standartai ir tipai“, šiuo metu populiariausi 2FA metodai, kaip laiku pagrįsti vienkartiniai slaptažodžiai (TOTP), SMS žinutės, paspaudimo autentifikacija, bei biometrinių duomenų autentifikacija yra plačiai pritaikomi ir pakankamai saugūs, tačiau pasitaiko situacijų, kai pagrindinis autentifikavimo metodas tampa neprieinamas, dėl pamesto telefono, laikinų ryšio trikdžių ar kitų techninių gedimų.

Siekiant apsiaugoti nuo atvejų, kai pirminis autentifikacijos metodas yra neprieinamas arba nepatikimas, reikia taikyti atsarginius antro faktoriaus autentifikavimo metodus, kurie gali užtikrinti vartotojo prieigą prie sistemų net ir nesėkmingai veikiant pagrindiniam autentifikavimo mechanizmui.

Dviejų faktorių autentifikacijos kontekste, dažniausiai naudojami atsarginiai autentifikavimo metodai yra: autentifikacija per el. paštą, iš anksto generuoti kodai, biometriniai duomenys ir fiziniai U2F raktai.

1.3.1. Autentifikacija per el. paštą

Tai yra vienas iš populiariausių atsarginių autentifikacijos metodų. El. pašto autentifikacija yra plačiai naudojamas atsarginis būdas, kuris suteikia praktinį sprendimą, kai pagrindiniai autentifikavimo metodai tampa neprieinami ar nepatikimi. Naudojantis šiuo metodu vartotojas gauna vienkartinį kodą arba patvirtinimo nuorodą į savo registruotą el. pašto adresą, kurį jis gali panaudoti autentifikavimo procesui užbaigti.

Nors autentifikacija El. paštu yra lengvai prieinamas ir universalus autentifikacijos būdas – daugelis vartotojų gali pasiekti savo el. pašto paskyras iš įvairių įrenginių, tačiau šis metodas gali kelti sistemos saugumo rizikų. Jeigu užpuolikas įsilaužia į naudotojo įrenginį ar patį el. paštą, jis galės apeiti daugumą pagrindinių 2FA saugos metodų. Be to, el. laiško pristatymo vėlavimai arba tinklo problemos gali apsunkinti šio metodo naudojimą. Naudojant el. pašto autentifikaciją, taip pat kyla rizika tapti sukčiavimo (angl.: phishing) atakų auka, kai naudotojai gauna apgaulingus laiškus, imituojančius teisėtas autentifikavimo užklausas.

Autentifikacija el. paštu yra lengvai prieinamas atsarginės autentifikacijos metodas, bet jis gali tapti kritine saugumo spraga sistemoje, jeigu naudotojai nesilaiko el. pašto saugumo gerųjų praktikų.[1,5]

1.3.2. Iš anksto generuoti kodai

Iš anksto sugeneruoti atsarginiai kodai yra vienas paprasčiausių alternatyvių autentifikavimo būdų. Tai iš anksto sugeneruoti vienkartiniai kodai, kurie naudotojams suteikiami pradinio dviejų faktorių autentifikavimo nustatymo metu. Šie kodai yra naudojami kaip atsarginis autentifikavimo metodas, leidžiantis vartotojams pasiekti savo paskyras tais atvejais, kai pagrindiniai autentifikavimo metodai, pavyzdžiui, mobilieji pranešimai, tampa neprieinami.

Atsarginiai kodai yra itin naudingi situacijose, kai vartotojas praranda prieigą prie savo mobiliojo įrenginio arba kai tinklo problemos trukdo gauti vienkartinius kodus per SMS ar pasiekti autentifikavimo programėles.

Šis metodas, iš pirmo žvilgsnio, gali atrodyti saugesnė 2FA alternatyva, lyginant su autentifikacija el. paštą, tačiau kyla tos pačios grėsmės kaip minėta el. pašto autentifikacijos kontekste. Jeigu atsarginiai kodai saugomi skaitmeninėje laikmenoje, užpuolikai gali pasinaudoti naudotojo aplaidumu ir įsilaužti į naudotojų įrenginius, siekdami perimti svarbią informaciją, įskaitant atsarginius kodus. Tokiu atveju atsarginiai kodai, nors ir sukurti kaip apsaugos priemonė, gali tapti sistemos saugumo spraga, jei naudotojas netinkamai juos saugo ar duoda galimybę užpuolikams juos pasiekti.[1,2,5]

1.3.3. Biometriniai duomenys

Autentifikacija naudojantis naudotojo biometriniais duomenimis, kaip pirštų atspaudais ar veido bruožais, yra itin saugi ir patogi atsarginės autentifikacijos alternatyva. Šis metodas užtikrina aukštą saugumo lygį, nes autentifikacijos metu naudojami unikalūs žmogaus fiziologiniai bruožai, kuriuos sunku klastoti ar nukopijuoti. [3]

Šis metodas ypač naudingas situacijose, kai kitos pagrindinės ar atsarginės autentifikavimo priemonės, kaip slaptažodžiai ar atsarginiai kodai, yra neprieinami. Autentifikacijos piršto atspaudu greitis ir paprastumas padeda užtikrinti sklandų naudotojo patirties procesą. Tačiau, situacijose, kai piršto antspaudo autentifikavimo sistema yra nepatikima, ar nepakankamai saugi, taikomosios programos, kaip „CityBee“7 ar „Bolt Drive“8 naudoja vidines arba trečiųjų šalių veido ir paso atpažinimo funkcijas, kurių autentifikacijos laikotarpis gali siekti nuo 3 iki 15 minučių, kas sistemose, kaip sveikatos priežiūros, aviacijos ar finansų, gali būti nepriimtinas laiko tarpas naudotojo autentifikacijai, dėl operatyvumo reikalavimų. Šiose sistemose ilgas laukimo laikas gali sukelti paslaugų teikimo vėlavimus, ar net pavojų žmonių gyvybėms, pavyzdžiui medicininių duomenų prieigos atidėjimas skubios pagalbos atveju.

1.3.4. Fiziniai U2F raktai

Fiziniai universalūs antrojo faktoriaus (U2F) įrenginiai, kaip „YubiKey“9, suteikia patikimą atsarginės autentifikacijos apsaugą, kuri yra nepriklausoma nuo tinklo ryšio ar mobiliojo įrenginio kai dauguma kitų atsarginių autentifikavimo metodų priklauso nuo vieno pagrindinio naudotojo įrenginio kaip išmanusis telefonas.

[7] CityBee programėlė: https://citybee.lt/
[8] Bolt Drive programėlė: https://bolt.eu/en-lt/drive/
[9] YubiKey įrenginio tinklapis: https://www.yubico.com/

U2F įrenginiai veikia naudodami viešojo rakto kriptografiją, kuri užtikrina, kad tik teisėtas naudotojas gali prisijungti prie sistemos. Dėl savo atsparumo įprastoms grėsmėms, tokioms kaip sukčiavimo atakos, U2F įrenginiai laikomi vienu saugiausių autentifikavimo sprendimų.

Šis atsarginis autentifikavimo metodas taip pat turi ir tam tikrų apribojimų. U2F įrenginiai yra fiziniai objektai, todėl jie gali būti pamesti, pavogti arba sugadinti, o tai kelia riziką naudotojo prieigai prie paskyros. Be to, U2F raktų naudojimas dažniausia reikalauja suderinamos aparatinės ir programinės įrangos, todėl ne visos sistemos gali palaikyti šiuos įrenginius. Taip pat, dar vienas svarbus aspektas yra pradinis įsigijimo kaina, kuri gali būti ženkliai didesnė nei kitų autentifikavimo metodų, kaip el. pašto autentifikavimas. [1,3]

1.4. Esamų dviejų faktorių autentifikavimo sistemų pažeidžiamumas

Dviejų faktorių autentifikavimas (2FA) yra plačiai naudojamas kaip papildomas saugumo sluoksnis apsaugoti naudotojų paskyras bei sistemas nuo neteisėtos prieigos. Tačiau net ir populiariausi, plačiausiai naudojami dviejų faktorių autentifikavimo metodai turi žinomų pažeidžiamumų, kurie gali kelti grėsmę naudotojų duomenų ir sistemų saugumui.

1.4.1. SMS kodai

Nepaisant SMS kodų autentifikacijos paprastumo ir plačios prieigos, šis metodas pasižymi rimtais saugumo trūkumais. Viena dažniausių atakų yra SIM kortelės keitimo ataka (angl.: SIM swapping), kai užpuolikai, pasinaudodami socialine inžinerija, įtikina mobiliojo ryšio operatorius perkelti vartotojo telefono numerį į naują SIM kortelę. Tai leidžia užpuolikams gauti autentifikavimo kodus ir pasiekti 2FA apsaugotas paskyras.[15]

„Efani“, Jungtinių Amerikos Valstijų mobiliojo ryšio paslaugų teikėjas, pateikė statistiką[10], kad SIM kortelių perėmimas JAV pakilo net 400% nuo 2023 iki 2024 m., bei JAV Federalinis tyrimų biuras, nustatė[11], kad 2023 metais, beveik 50 milijonų JAV dolerių buvo pavogti naudojant tik SIM keitimo atakas. Remiantis šiomis statistikomis, galima teigti, kad autentifikacija SIM kodais yra nepatikimas būdas autentifikuoti jautrių sistemų naudotojus.

Be to, SMS žinutės dažnai keliauja nešifruotu būdu, todėl gali būti perimtos vykdant tarpininko atakas. Dar viena grėsmė naudojant SMS autentifikaciją - sukčiavimo atakos, kai vartotojai apgaulės būdu pateikia SMS kodus užpuolikams.[6]

1.4.2. TOTP - laiku pagrįsti vienkartiniai slaptažodžiai

TOTP metodas yra saugesnis autentifikacijos metodas nei SMS kodai, tačiau jis taip pat turi savo trūkumų. Jei slaptasis raktas, naudojamas vienkartinių kodų generavimui, yra nutekintas, užpuolikai gali generuoti galiojančius kodus ir apeiti 2FA apsaugotas sistemas.[4,16]

TOTP autentifikacijos įrankių atsarginiai mechanizmai dažnai pasikliauja mažiau saugiais metodais, tokiais kaip el. paštas ar SMS, kas sukelia papildomų saugumo spragų TOTP įrankiuose.

1.4.3. Iš anksto sugeneruoti kodai

Šis metodas įprastai yra naudojamas kaip atsarginis 2FA metodas, kai pagrindinis autentifikavimo metodas nėra prieinamas. Nors šis metodas, iš implementacijos pusės, yra nesudėtingas įgyvendinti

[10] „Efani“ mobiliojo ryšio paslaugų teikėjo statistika: https://www.efani.com/blog/sim-swapping-statistics
[11] JAV Federalinio tyrimų biuro (FBI) statistika: https://www.ic3.gov/AnnualReport/Reports/2023_IC3Report.pdf

ir naudoti, tačiau iš anksto sugeneruoti kodai gali būti lengvai prarasti arba pavogti, jei naudotojas juos neatsargiai saugo, ar laiko nesaugiame įrenginyje.[6]

Nesaugus šių kodų laikymas gali būti išnaudotas brutalios jėgos atakų (angl.: brute-force attacks)[17], kai užpuolikas bando įvesti visus galimus slaptažodžių arba kodų variantus, kol suranda tinkamą. Kadangi iš anksto sugeneruotų kodų skaičius paprastai yra ribotas, pavyzdžiui, 8 ar 10 kodų, ši ataka yra labai efektyvi, ypač kai užpuolikas sugeba perimti kodų sąrašą.

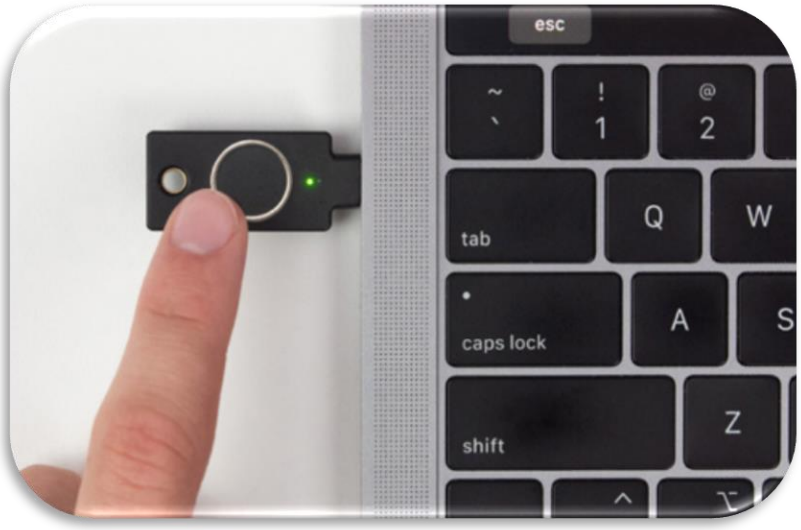
1.4.4. Paspaudimo autentifikacija

Šis metodas, kai naudotojas patvirtina autentifikavimo užklausą savo išmaniajame įrenginyje, yra patogus ir greitas. Tačiau naudotojai, kasdien gaudami daugybę tokių autentifikavimo pranešimų, gali patirti patvirtinimo nuovargį (angl.: confirmation fatigue), dėl kurio naudotojai gali neapdairiai patvirtinti neteisėtą užklausą. Užpuolikai dažnai simuliuoja autentifikavimo užklausas, tikėdamiesi, kad naudotojas jas patvirtins iš įpročio. [18]

1.4.5. U2F saugumo raktai

Fizinai autentifikavimui skirti raktai laikomi viena saugiausių autentifikavimo priemonių, tačiau U2F raktai ne visom sistemom tinka, nes ne visos platformos juos palaiko. Didžiausia U2F raktų saugumo grėsmė - prarasti fizinį raktą, ypač jei nėra įdiegtų atsarginių autentifikavimo būdų. [6]

Pametus U2F raktą, atsiranda grėsmė piktavaliui panaudoti jį prisijungiant prie sistemos. Tačiau šios rizikos galima išvengti perkant išmanius U2F prietaisus kaip „Yubico“ įmonės „Bio“ kartos raktus[12].



4 pav. „Yubico“ „Bio“ kartos U2F saugumo raktas[13]

„Yubico“ įmonės „Bio“ kartos U2F raktai (4) naudoja piršto antspaudų biometrinius duomenis ir taip užtikrina, kad fiziniai kriptografiniai raktai negalės būti panaudoti piktavalių, kai naudotojo raktas yra pamestas ar pavogtas. [19]

[12] „Yubico“ „Bio“ kartos U2F saugos raktai: https://www.yubico.com/products/yubikey-bio-series/
[13] „Yubico“ U2F raktas „YubiKey C Bio - FIDO Edition“: https://resources.yubico.com/53ZDUYE6/at/wf5hf9jbbnchtbw97vhn6/YubiKey_Bio_Series_Product_Brief.pdf

1.4.6. Biometriniai duomenys

Autentifikavimas biometriniais duomenimis, kaip pirštų atspaudai ar veido atpažinimas, yra patogus ir intuityvus autentifikavimo metodas, tačiau ir šis metodas turi rimtų saugumo trūkumų. Biometriniai duomenys yra nekintantys, todėl, jei jie yra nutekinti ar suklastoti, jie tampa ilgalaike grėsme naudotojui. Be to, užpuolikai gali pasinaudoti įvairiomis klastojimo technologijomis, pavyzdžiui, 3D modeliais, ar dirbtiniu intelektu, kad apeitų šiuos biometrinio autentifikavimo metodus. [6,13,20]

1.4.7. El. pašto autentifikacija

El. paštas dažnai naudojamas kaip atsarginis autentifikavimo metodas, tačiau jis taip pat turi savo rizikų. Jei el. pašto paskyra yra nulaužta, užpuolikas gali pasinaudoti šiuo kanalu, kad apeitų autentifikaciją ir pasiektų apsaugotas sistemas, ar net kitas, susietas, el. pašto paskyras.

Didžiausia grėsmė šiam autentifikacijos metodui yra sukčiavimo (angl.: phishing) ataka, kai naudotojas gauna apgaulingus laiškus, kuriuos užpuolikai naudoja siekdami išgauti prisijungimo duomenis ar kitą jautrią naudotojo informaciją. [13,15,21]

1.4.8. Esamų dviejų faktorių autentifikavimo sistemų pažeidžiamumų apžvalga

Remiantis atlikta 2FA metodų analize, toliau pateikiama apibendrinta pažeidžiamumų apžvalga. Kiekvienas metodas vertinamas pagal du kriterijus: saugumo lygį (atsparumą kibernetinėms atakoms) ir žmogaus pažeidžiamumo lygį (tikimybę, kad naudotojo klaida ar socialinės inžinerijos ataka pakenks autentifikacijos procesui). Bendras įvertis apskaičiuojamas kaip šių dviejų kriterijų vidurkis.

2 lentelė. Populiariausių 2FA standartų pažeidžiamumo palyginimas

2FA metodas	**Saugumo lygis** (žemas: 1, vidutinis: 2, aukštas: 3)	**Žmogaus pažeidžiamumo lygis** (aukštas: 1, vidutinis: 2, žemas: 3)	**Bendro įverčio lygis** (lygio vertės vidurkis: žemas (1-1,49), vidutinis (1,5-2,49), aukštas (2,5-3))
SMS kodai	Žemas (1)	Vidutinis (2)	Vidutinis (1.5)
TOTP (laiku pagrįsti slaptažodžiai)	Vidutinis (2)	Vidutinis (2)	Vidutinis (2)
Iš anksto sugeneruoti kodai	Žemas (1)	Aukštas (1)	Žemas (1)
Paspaudimo autentifikacija	Aukštas (3)	Vidutinis (2)	Aukštas (2.5)
U2F saugumo raktai	Aukštas (3)	Žemas (3)	Aukštas (3)
Biometriniai duomenys	Vidutinis (2)	Vidutinis (2) .	Vidutinis (2)
El. pašto autentifikacija	Vidutinis (2)	Aukštas (1)	Vidutinis (1.5)

Lentelėje (2) vertinami nagrinėti šeši dviejų faktorių autentifikacijos (2FA) metodai, atsižvelgiant į jų saugumo lygį ir žmogaus pažeidžiamumo lygį.

SMS kodai turi žemą saugumo vertinimą, nes jie yra pažeidžiami sukčiavimo, tarpininko ir SIM kortelės keitimo atakų. Be to, naudotojai gali tapti sukčiavimo aukomis, pateikdami kodus apgaulingose svetainėse. Dėl šių priežasčių bendras įvertinimas yra vidutinis.

TOTP (laiku pagrįsti slaptažodžiai) siūlo vidutinį saugumo lygį, nes kodai generuojami naudotojo įrenginyje ir nepriklauso nuo tinklo ryšio. Tačiau slapto rakto nutekėjimas išlieka grėsme. Šis metodas yra mažiau pažeidžiamas sukčiavimo atakoms, todėl jo bendras įvertinimas yra vidutinis.

Iš anksto sugeneruoti kodai yra mažiausiai saugūs, nes jie gali būti lengvai prarasti arba pavogti, o naudotojai dažnai jų tinkamai nesaugo. Tai daro šį metodą pažeidžiamu ir mažiau patikimu, todėl bendras įvertis yra žemas.

Paspaudimo autentifikacija yra vienas saugiausių metodų, nes naudojami užšifruoti ryšio kanalai, tačiau ji gali būti paveikta patvirtinimo nuovargio grėsmės. Naudotojai gali netyčia patvirtinti neteisėtas užklausas, tačiau bendras įvertinimas išlieka aukštas dėl atsparumo daugeliui atakų.

U2F saugumo raktai yra labai saugūs, nes naudoja viešojo ir privataus rakto kriptografiją. Šis metodas yra mažai pažeidžiamas dėl papildomų apsaugos priemonių, todėl jo bendras įvertinimas yra aukštas.

Biometriniai duomenys pasižymi vidutiniu saugumo lygiu, nes jų nutekėjimas gali sukelti ilgalaikę grėsmę. Naudotojai yra mažiau pažeidžiami dėl biometrinių duomenų klastojimo sudėtingumo. Bendras šio metodo įvertinimas yra vidutinis.

El. pašto autentifikacija yra vidutiniškai saugi, tačiau ji yra pažeidžiama sukčiavimo atakoms. Naudotojams neatpažinus tokios atakos, užpuolikas gali lengvai gauti prieigą prie sistemos. Todėl bendras įvertinimas yra tik vidutinis.

Remiantis lentelės (2) bendro įverčio duomenimis, geriausiai įvertinti 2FA metodai yra U2F saugumo raktai ir paspaudimo autentifikacija. U2F saugumo raktai išsiskiria savo saugumo stiprumu, nes naudoja viešojo ir privataus rakto kriptografiją, o papildomos apsaugos priemonės mažina naudotojo pažeidžiamumą. Paspaudimo autentifikacija yra aukštai įvertinta dėl užšifruotų ryšio kanalų, kurie užtikrina atsparumą daugeliui kibernetinių atakų. Šie metodai yra rekomenduojami naudoti, kai sistemai reikia aukšto autentifikacijos saugumo lygio.

1.5. Dviejų faktorių autentifikavimo atsarginių metodų realizacijos

Naudotojai gali netekti prieigos prie vieno iš dviejų faktorių autentifikacijos (2FA) elementų, pavyzdžiui, pamesti telefoną arba pamiršti slaptažodį. Tokiais atvejais ypač svarbu turėti atsarginius autentifikavimo metodus, kurie ne tik užtikrintų prieigos atkūrimą, bet ir apsaugotų nuo situacijų, kai pagrindinis autentifikacijos būdas tampa neprieinamas ar nepatikimas. Tokie atsarginiai sprendimai leidžia naudotojams išlaikyti prieigą prie sistemų net ir esant techniniams trikdžiams ar kitoms nenumatytoms aplinkybėms, užtikrinant nenutrūkstamą saugumą ir sistemos pasiekiamumą.

„**Google**“ siūlo kelis 2FA metodus, įskaitant SMS žinutes, autentifikavimo programas ir U2F saugos raktus[14]. Naudotojai, naudojantys „Google“ ekosistemos teikiamomis paslaugomis taip pat gali sugeneruoti atsarginius kodus, kuriuos galima naudoti praradus prieigą prie pagrindinio 2FA metodo.

[14] „Google“ U2F saugos raktas: „Titan Security Key“. Nuoroda: https://store.google.com/us/product/titan_security_key?hl=en-US

Šie kodai turėtų būti saugiai saugomi, kad būtų galima atkurti prieigą prie paskyros kritiniais atvejais. Nuo 2023 m. „Google Authenticator“, TOTP dviejų faktorių autentifikavimo mobilioji programėlė, taip pat sinchronizuoja TOTP slaptažodžius tarp „Google“ paskyrų [15], tad net pametus pirminį įrenginį, galima pasiekti 2FA programėlę net iš kito įrenginio, prisijungus su savo „Google“ paskyra. Nors šis funkcionalumas ir patogus naudotojui, tačiau jeigu užpuolikas įsilaužtų į naudotojo paskyrą, jis galėtų pasiekti ir kitas 2FA apsaugotas naudotojo sistemas.

„**Microsoft**“ taip pat siūlo įvairius 2FA metodus, tokius kaip autentifikavimo programėlės, SMS kodai ir el. pašto patvirtinimai. Be to, vartotojai gali nustatyti saugos klausimus, kurie padeda atkurti prieigą prie paskyros praradus prieigą prie pagrindinio autentifikavimo metodo.[16]

„**Cisco Duo**“[17] siūlo įvairius kelių faktorių autentifikavimo (MFA) metodus, įskaitant „Duo Push“ paspaudimo autentifikaciją, SMS kodus, telefono skambučius ir TOTP kodus. Be to, „Duo“ palaiko U2F saugos raktus, tokius kaip „YubiKey“, kurie gali būti naudojami kaip atsarginiai autentifikavimo metodai. Tai suteikia naudotojams lankstumo pasirinkti tinkamiausią autentifikavimo būdą, atsižvelgiant į jų poreikius ir situaciją.

„**Okta**“[18] siūlo daugybę MFA metodų, tokių kaip „Okta Verify“ programėlė, SMS kodai, el. pašto patvirtinimai ir saugos klausimai. Be to, „Okta“ palaiko trečiųjų šalių autentifikavimo priemones, tokias kaip „Google Authenticator“ ir „Duo Security“. Tai leidžia organizacijoms pritaikyti autentifikavimo procesą pagal savo saugumo reikalavimus ir vartotojų poreikius.

1.6. Išvados ir tolimesni 2FA atsarginių algoritmų tyrimo uždaviniai

Atlikta analizė atskleidė, kad dviejų faktorių autentifikavimo metodai, nors ir reikšmingai pagerina bendrą sistemų saugumą, nėra pakankami kritinėms sistemoms apsaugoti, ypač kai pagrindinis autentifikacijos kanalas tampa nepasiekiamas arba nepatikimas. Todėl būtina užtikrinti geresnius 2FA alternatyvius sprendimus, kurie:

1. Užpildo dabartines 2FA atsarginių metodų spragas.

   Įprasti pirminiai 2FA metodai, tokie kaip SMS kodai, TOTP, biometrika ar U2F raktai, pasižymi įvairiomis pažeidžiamumo formomis – nuo socialinės inžinerijos iki prarasto fizinio įrenginio. Tokių trūkumų pasekmės ypač didelės ten, kur autentifikavimo sutrikimai gali sukelti finansinius nuostolius ar kelti pavojų žmonių sveikatai (pvz., sveikatos priežiūros, aviacijos ar finansų srityse). Standartiniai atsarginiai 2FA metodai, kaip SMS kodai, el. paštas, iš anksto sugeneruoti kodai, neužtikrina tokio paties saugumo lygio kaip pirminiai 2FA sprendimai (TOTP, paspaudimo autentifikacija) arba smarkiai apriboja vartotojo patogumą dėl lėtesnio, neintuityvaus veikimo (ypač naudojant el. pašto žinutes). Šie atsarginiai 2FA metodai neatitinka saugumo ir patogumo lygio, kaip pirminis 2FA, situacijose, kai pirminis 2FA metodas yra nepatikimas, reikšmingai sumažina bendrą sistemos patikimumą bei faktiškai panaikina 2FA teikiamą pridėtinę vertę, todėl būtina ieškoti pažangesnių ir labiau atsparių atsarginių autentifikacijos algoritmų.Padengia 2FA tiekėjų serverių nepasiekiamumo problemą.

[15] „Google Authenticator“ straipsnis, apie TOTP sinchronizavimą: https://security.googleblog.com/2023/04/google-authenticator-now-supports.html
[16] „Microsoft“ 2FA: https://www.microsoft.com/lt-lt/security/business/security-101/what-is-two-factor-authentication-2fa
[17] „Cisco Duo“ MFA įrankiai: https://duo.com/product/multi-factor-authentication-mfa/authentication-methods/tokens-and-passcodes
[18] „Okta“ MFA įrankiai: https://help.okta.com/oie/en-us/content/topics/identity-engine/authenticators/about-authenticators.htm

2. Padengia 2FA tiekėjų serverių nepasiekiamumo problemą.

   Pirminiai industriniai 2FA sprendimai, kaip „Okta“, „Cisco Duo“, nenumato, ką daryti, jei paties 2FA tiekėjo paslauga sutrinka ar tampa nepatikima. Atsarginiai 2FA algoritmai turėtų veikti net ir tuo atveju, kai išoriniai autentifikavimo serveriai tampa nepasiekiami dėl gedimų ar kibernetinių atakų.

3. Yra lengvai pritaikomi prie esamų sistemų ir patogūs vartotojui.

   Inovatyvūs 2FA atsarginių algoritmų sprendimai turi būti lengvai integruojami, kad nepakenktų kasdieniam sistemos procesų veikimui. Situacijose, kai pirminis 2FA metodas yra nepasiekiamas ar nepatikimas ypač svarbu palaikyti nenutrūkstamą paslaugų teikimą bei išsaugoti aukštą vartotojo patirties (UX) lygį.

Toliau darbe orientuojamasi į šių atsarginių 2FA algoritmų kūrimą bei validavimą realioje aplinkoje. Vienas iš svarbiausių uždavinių – integruoti prevencinius mechanizmus, užtikrinančius autentifikacijos veikimo tęstinumą net ir atsiradus techniniams ar saugumo trikdžiams (pvz., DDoS atakoms). Taip pat būtina išlaikyti paprastumą ir patogumą, kad atsarginiai metodai nekeltų papildomų kliūčių naudotojams. Šiais reikalavimais grįstas atsarginis autentifikavimo algoritmas leistų reikšmingai sustiprinti kritinių sistemų patikimumą ir atitikti nuolat augančius kibernetinio saugumo reikalavimus.

2. Atsarginio dviejų faktorių autentifikavimo projektas

Atlikus analizę paaiškėjo būtinybė sukurti atsarginį, patikimą ir lengvai integruojamą sprendimą, kuriame bus naudojama vartotojo USB laikmena kaip atsarginis antras faktorius. Pasirinktas metodas yra patikimas 2FA, nes su USB laikmena kaip atsarginis 2FA naudotojui leidžia išlikti mobilam, autentifikuotis be išmaniojo telefono, o sistemų administratoriams suteikia papildomą saugumo sluoksnį nereikalaujant trečiųjų šalių paslaugų. Pagrindinis projekto tikslas – suprojektuoti atsparų, patogų ir lengvai integruojamą dviejų faktorių autentifikavimo (2FA) mechanizmą, kuriame antrąjį faktorių atlieka naudotojo USB laikmena. Sprendimas skirtas kritinėms informacinėms sistemoms, kurioms būtinas nepertraukiamas prieigos patikimumas net ir esant paslaugų teikėjų ar tinklo sutrikimams arba pametus įprastą antrąjį faktorių – pavyzdžiui, išmanųjį telefoną, tokiais atvejais autentifikaciją galima atlikti su USB laikmena.

2.1. Atsarginis 2FA algoritmas naudojant USB laikmeną

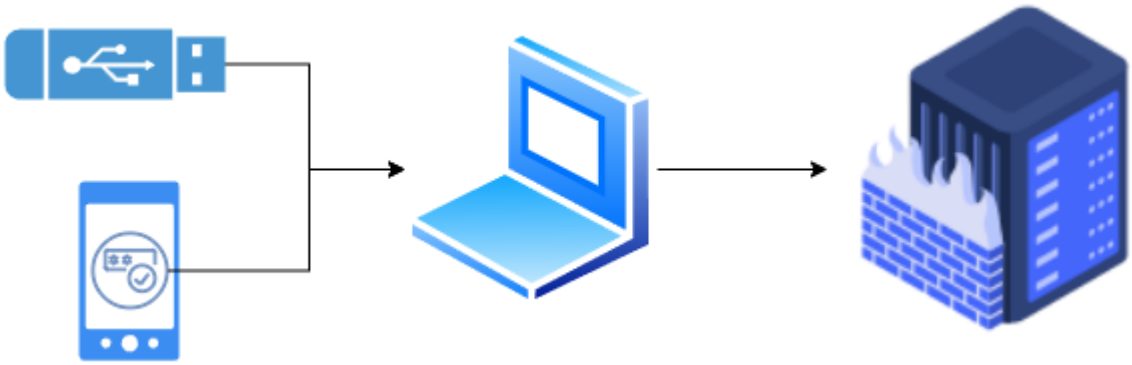
Patikimo 2FA su USB laikmena produkto tikslas - suteikti organizacijoms atsarginį, nuo trečiųjų šalių nepriklausomą tapatybės patvirtinimo kanalą, kuris veiktų net ištikus mobiliųjų įrenginių ar tinklo sutrikimams. Sprendimas turi būti intuityvus galutiniam vartotojui, lengvai diegiamas IT padaliniams ir atitinkantis aukštus informacijos saugos standartus, kad infrastruktūra liktų pasiekiama tik teisėtiems naudotojams.

Patikimo antro faktoriaus autentifikavimo sistema sudaryta iš trijų loginių sluoksnių (5), kurių kiekvienas atlieka atskirą funkciją autentifikavimo procese.

Pirmasis sluoksnis apima naudotojo turimus autentifikavimo įrankius: išmanųjį telefoną su TOTP programėle, pavyzdžiui, „Google Authenticator“, bei USB laikmeną, kurioje saugoma pasirašyta autentifikavimo programa. Šis sluoksnis atsakingas už vienkartinių kodų generavimą, atsarginio autentifikavimo funkcionalumą ir jautrių autentifikavimo duomenų (kriptografinių raktų) saugojimą. Išmanusis telefonas naudojamas pagrindiniam antro faktoriaus autentifikavimui, o USB laikmena veikia kaip atsarginė priemonė tais atvejais, kai pagrindinis autentifikavimo metodas tampa nepasiekiamas arba nepatikimas.

Antrasis sluoksnis yra naudotojo kompiuteris, kuris veikia kaip tarpinė vykdymo aplinka tarp naudotojo 2FA įrenginių ir nuotolinės sistemos. Naudotojo kompiuteryje paleidžiama USB laikmenoje saugoma autentifikavimo programa, apdorojami naudotojo įvesti duomenys, siunčiamos užklausos serveriui ir perduodami autentifikavimo metu apskaičiuoti duomenys.

Trečiasis loginis sluoksnis yra nuotolinis autentifikacijos serveris, atsakingas už autentifikavimo proceso valdymą. Šis sluoksnis tikrina naudotojo prisijungimo duomenis, validuoja vienkartinius kodus ar kitus pateiktus autentifikavimo artefaktus ir išduoda prieigos žetonus



5 pav. Sistemos loginiai sluoksniai

Toks sluoksnių išdėstymas (5) leidžia aiškiai paskirstyti atsakomybes tarp naudotojo įrenginių, vykdymo aplinkos ir centrinės autentifikacijos infrastruktūros. Dėl to sistema tampa lengviau prižiūrima, saugesnė ir atsparesnė gedimams, nes pagrindinio ir atsarginio autentifikavimo funkcijos gali būti įgyvendinamos nepriklausomai, tačiau išlaikant bendrą veikimo vientisumą.

Visą pranešimų eigą saugo abipusio autentifikavimo ir nulinio pasitikėjimo (angl.: Zero-Trust) modelis, kuris garantuoja, kad nė viena komponentė negali kompromituoti visos grandinės.

2.1.1. Nulinio pasitikėjimo modelis

Nulinio pasitikėjimo (angl.: Zero-Trust) modelis remiasi nuostata, kad kiekviena užklausa nepriklausomai nuo jos kilmės, tinklo segmento ar naudotojo padėties turi būti autentifikuota, autorizuota ir tik po to įleista prie reikiamų išteklių. Todėl net ir lokaliai veikiančios darbo vietos, prijungtos prie universiteto LAN, laikomos potencialiai nepatikimomis, kol jos neįrodo savo tapatybės ir saugos būsenos.

Šiuo atveju, USB laikmena, naudojama kaip atsarginis antrasis faktorius: laikinas vienkartinis kodas sugeneruojamas uždaroje, nuo išorinių paslaugų nepriklausomoje aplinkoje – paties vartotojo kompiuteryje, tačiau sugeneruoto slaptažodžio ir USB laikmenos identiteto patikra atliekama organizacijos autentifikacijos serveryje per šifruotą HTTPS kanalą. Tokiu būdu pašalinamas poreikis pasitikėti mobiliojo ryšio operatoriais ar trečiųjų šalių 2FA tiekėjais, o galutinis autorizacijos sprendimas priimamas tik IAM serveryje.

Galiausiai nulinio pasitikėjimo modelis turėtų veikti visą laiką: jei rizikos valdymo sistema pastebi keistą elgesį - pavyzdžiui, kelias iš eilės neteisingas OTP, sistema turėtų paprašyti papildomo patvirtinimo arba laikinai apriboti prieigą.

2.1.2. Kriptografiniai algoritmai

USB autentifikatoriaus saugumo modelis paremtas trimis kriptografiniais komponentais:

- Simetriniu šifravimu, raktų konteinerio apsaugai;
- Raktų išvedimo funkcija, slaptažodžio pavertimui šifravimo raktu;
- Skaitmeniniu parašu, autentifikacijos iššūkio pasirašymui.

USB laikmenos raktų konteineriui šifruoti pasirinktas AES-GCM (angl.: Galois/Counter Mode) algoritmas, apibrėžtas NIST SP 800-38D publikacijoje [22]. AES-GCM yra autentifikuoto šifravimo režimas, kuris vienu žingsniu užtikrina tiek duomenų konfidencialumą, tiek integralumą.

3 lentelė. Šifravimo algoritmų palyginimas

Algoritmas	Šifravimas ir autentifikacija vienu žingsniu	Integralumo tikrinimas	Aparatūrinis pagreitinimas (AES-NI)	Turi NIST standartizaciją	Palaikomas paralelizavimas
AES-GCM	Taip	Integruotas (GHASH)	Taip	NIST SP 800-38D	Taip
AES-CBC	Ne (reikia atskiro HMAC)	Reikia papildomo HMAC	Taip	NIST SP 800-38A	Ne
ChaCha20-Poly1305	Taip	Taip	Ne	Ne	Taip

AES-GCM pasirinktas dėl šių priežasčių:

1. Vientisumo garantija. GCM režimas automatiškai sugeneruoja 128 autentifikacijos žymę (angl.: authentication tag), kuri leidžia aptikti bet kokį šifruotų duomenų pakeitimą. Jei USB laikmenoje esantis raktų konteineris būtų modifikuotas (pvz., kenkėjiško kodo), iššifravimo metu tai bus aptikta ir operacija bus atmesta. AES-CBC[23] režimas šios savybės neturi – be papildomo HMAC mechanizmo, modifikuoti duomenys gali būti iššifruoti, tačiau jų turinys bus sugadintas, kas gali sukelti saugumo spragas;

2. Aparatūrinis pagreitinimas. Šiuolaikiniai procesoriai (Intel, AMD) turi AES-NI instrukcijų rinkinį, kuris žymiai pagreitina AES operacijas. GCM režimas efektyviai pasinaudoja šiomis instrukcijomis, užtikrindamas greitą šifravimo ir iššifravimo procesą;

3. NIST standartizacija. AES-GCM yra NIST standartizuotas algoritmas ir plačiai naudojamas. ChaCha20-Poly1305, nors ir saugus, nėra NIST standartizuotas ir .NET 8 aplinkoje nėra tiesiogiai palaikomas.

AES-GCM šifravimo raktą būtina išvesti iš naudotojo slaptažodžio naudojant tam skirtą raktų išvedimo funkciją. Šiam tikslui pasirinkta PBKDF2 (angl.: Password-Based Key Derivation Function 2), apibrėžta RFC 8018 rekomendacijose [25]. Šiuo metu egzistuoja keletas alternatyvių raktų išvedimo funkcijų: bcrypt, scrypt ir Argon2id. Pastarosios, ypač Argon2id (2015 m. slaptažodžių maišos konkurso (PHC) nugalėtojas), pasižymi didesniu atsparumu GPU pagrįstoms atakoms [24]. Tačiau šiam projektui pasirinktas PBKDF2 dėl kelių esminių priežasčių.

Pirma, PBKDF2 yra vienintelė NIST standartizuota ir FIPS-140 sertifikuota raktų išvedimo funkcija [25]. Tai yra svarbus kriterijus, nes prototipas orientuotas į organizacijas, kurioms gali būti taikomi federaliniai ar tarptautiniai informacijos saugos reikalavimai (pvz., viešojo sektoriaus institucijos, finansų ar sveikatos priežiūros sektoriai). Nei bcrypt, nei scrypt, nei Argon2id šiuo metu neturi NIST sertifikacijos, todėl jų naudojimas tokiose aplinkose gali reikalauti papildomų pagrindimų arba išimčių.

Antra, PBKDF2 yra tiesiogiai palaikomas .NET platformos per „System.Security.Cryptography.Rfc2898DeriveBytes“ klasę, nereikalaujant papildomų trečiųjų šalių bibliotekų. Tai mažina priklausomybių skaičių, supaprastina saugumo auditą ir sumažina tiekimo grandinės atakų (angl.: supply chain attack) riziką

Be simetrinio šifravimo, USB autentifikatoriui reikalingas ir asimetrinis skaitmeninio parašo algoritmas - juo pasirašomas serverio pateiktas iššūkis autentifikacijos metu. Šiam tikslui pasirinktas ECDSA P-256, standartizuotas NIST FIPS 186-5 specifikacijoje. ECDSA P-256 pasirinktas dėl kelių esminių privalumų:

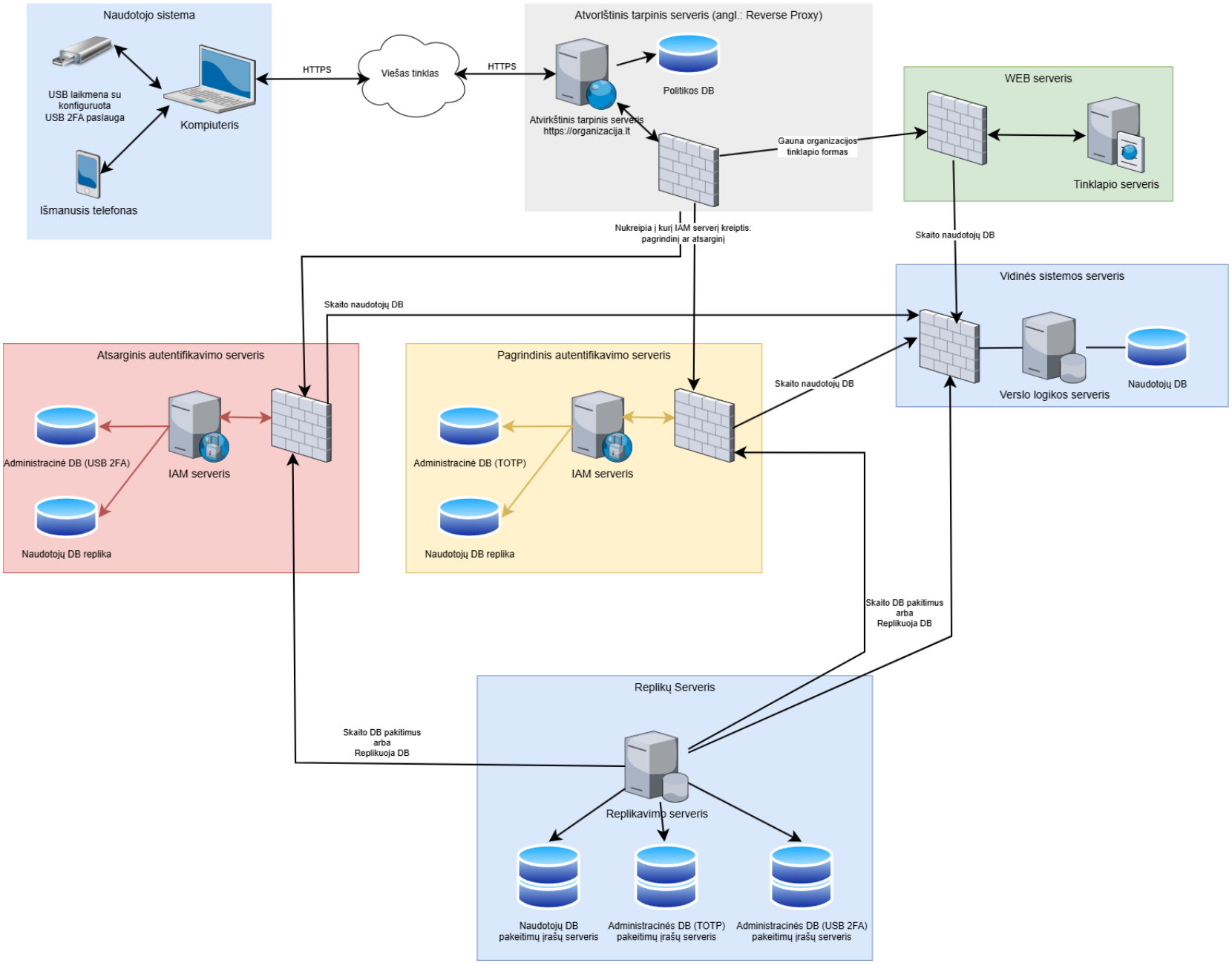
- Saugumas - užtikrina saugumo lygį, atitinkantį AES-128[19] standartą. Šis lygis laikomas pakankamu apsaugai nuo šiuolaikinių atakų, įskaitant brutalios jėgos bandymus;
- Kompaktiškumas - ECDSA raktai yra žymiai trumpesni nei RSA raktai esant analogiškam saugumo lygiui. P-256 privatus raktas užima tik 32 baitus (256 bitus), o viešasis raktas – 64 baitus, palyginti su RSA-3072, kurio raktai siekia 384 baitus ir daugiau;
- Našumas - elipsinių kreivių operacijos yra greitesnės nei RSA, todėl parašų generavimas ir tikrinimas vyksta sparčiau, kas ypač svarbu autentifikacijos procesui;
- Platus palaikymas - P-256 yra NIST standartizuota kreivė, palaikoma visose pagrindinėse kriptografinėse bibliotekose, įskaitant .NET System.Security.Cryptography, bei atitinka WebAuthn/FIDO2 specifikacijas.

Šie trys kriptografiniai komponentai: AES-GCM šifravimas, PBKDF2 raktų išvedimas ir ECDSA P-256 parašai, sudaro USB autentifikatoriaus saugumo pagrindą. Jų detali realizacija ir integravimas į prototipą aprašomi trečiame skyriuje.

2.2. Patikimo atsarginio 2FA algoritmo topologija

Norint užtikrinti, kad autentifikavimo procesas būtų patikimas, atsparus trikdžiams ir palaikytų nenutrūkstamą sistemos veikimą kritinėmis situacijomis, buvo suprojektuota architektūra (6), paremta dviem nepriklausomais autentifikavimo serveriais, valdomais per centralizuotas prieigos vartus – atvirkštinį tarpinį serverį (angl.: reverse proxy).

[19] AES-128 NIST saugumo standarto aprašymas: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf



6 pav. Patikimo atsarginio 2FA algoritmo topologija

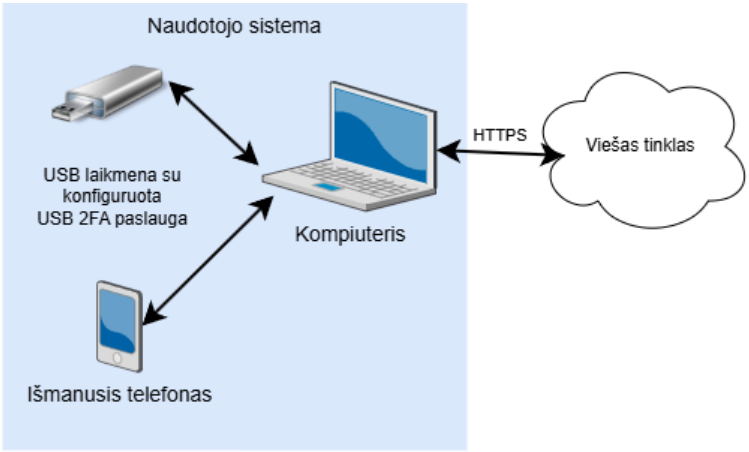
Sistema papildyta atskiru replikavimo komponentu, kuris užtikrina nuoseklų naudotojų duomenų sinchronizavimą tarp pagrindinio ir atsarginio autentifikavimo serverių.

Pateikta sistemos architektūra (6) parodo pagrindinius autentifikavimo procese dalyvaujančius komponentus. Toliau, kiekvieno iš komponentų paskirtis bus pristatyta atskirai nagrinėjant: naudotojo autentifikavimo įrangą, atvirkštinį tarpinį serverį, autentifikavimo, bei replikavimo serverius, taip pat web ir vidinės sistemos serverius.

2.2.1. Naudotojo autentifikavimo įranga

Naudotojai šioje sistemoje naudoja du skirtingus įrenginius (7):

1. Išmanųjį telefoną su TOTP autentifikavimo programėle (pvz., „Google Authenticator“), skirtą autentifikuotis pagal numatytąjį scenarijų;
2. USB laikmeną, kurioje įdiegta specialiai sukonfigūruota atsarginė 2FA paslauga, naudojama tuomet, kai pagrindinis TOTP metodas tampa neprieinamas arba nepatikimas.

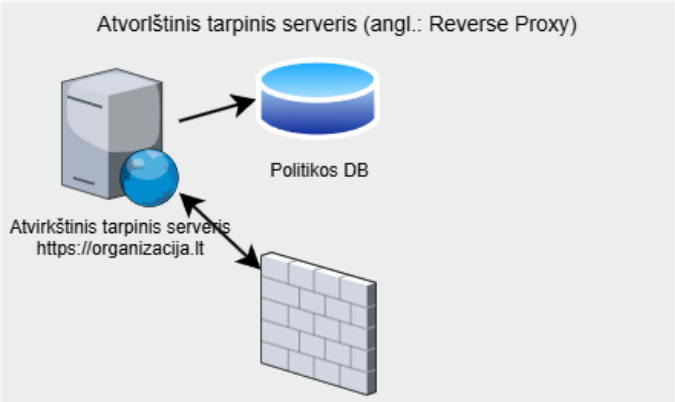


7 pav. Patikimo atsarginio 2FA algoritmo topologija. Naudotojo sistema

Tai leidžia užtikrinti lankstų, dvigubą autentifikacijos mechanizmą, apsaugotą tiek nuo tinklo trikdžių (pvz., telefono praradimo), tiek nuo sisteminių gedimų (pvz., TOTP autentifikavimo serverio nepasiekiamumo).

2.2.2. Atvirkštinis tarpinis serveris

Pirmasis kontaktinis taškas tarp naudotojo ir organizacijos sistemų yra atvirkštinis tarpinis serveris (angl.: Reverse proxy), pasiekiamas per HTTPS protokolą. Šis serveris priima autentifikavimo užklausas iš išorinio tinklo, analizuoja ir pritaiko autentifikacijos politiką, saugomą atskiroje politikų duomenų bazėje. O įvertinus politikas bei autentifikacijos užklausą - nusprendžia į kurį IAM serverį (pagrindinį ar atsarginį) nukreipti autentifikacijos srautą, arba užklausą atmesti, jei serveris pažymėtas kaip neprieinamas.



8 pav. Patikimo atsarginio 2FA algoritmo topologija. Atvirkštinis tarpinis serveris

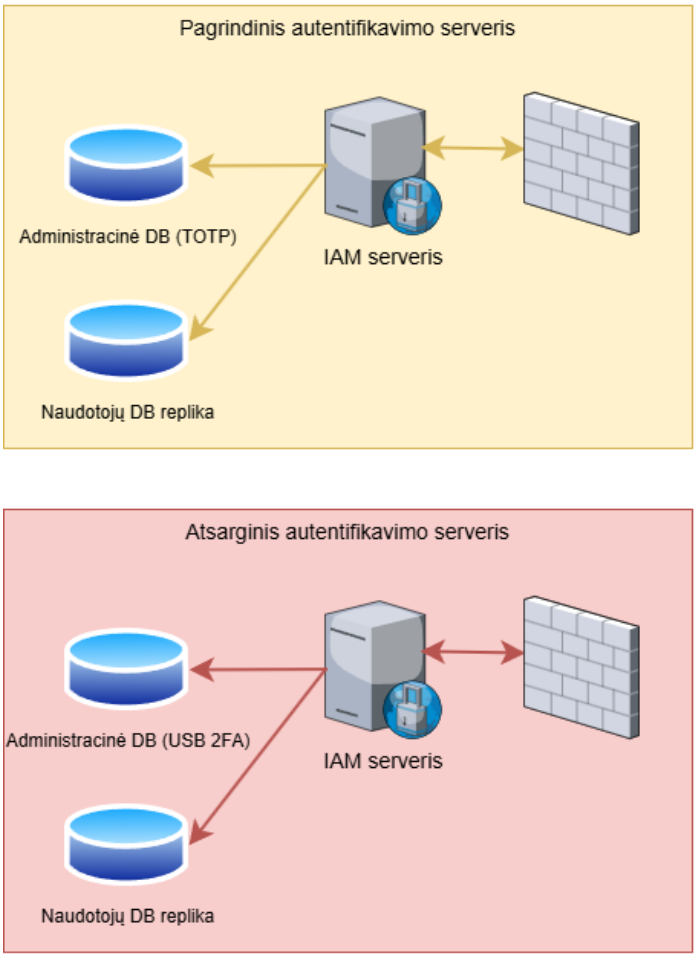
Šis komponentas (8) veikia kaip autentifikavimo vartai, užtikrinantys, kad visa logika dėl autentifikacijos pasirinkimo būtų centralizuota ir valdoma administratorių, be būtinybės modifikuoti atskirų komponentų.

2.2.3. Pagrindinis ir atsarginis autentifikavimo (IAM) serveriai

Pagrindinis IAM serveris yra atsakingas už pirmojo faktoriaus (kas tu esi: naudotojo slapyvardis, slaptažodis) autentifikavimą, bei antrojo faktoriaus autentifikavimą (ką tu turi: naudotojo išmanusis telefonas) naudojant TOTP algoritmą. Šį serverį sudaro administracinė duomenų bazė (9), kuri saugo kiekvieno naudotojo šifruotus prisijungimo raktus, bei pagrindinio 2FA (TOTP) raktus, kurie yra susieti su unikaliu naudotojo ID.

Pagrindiniame IAM serveryje taip pat yra naudotojų duomenų bazės replika, kuri leidžia serveriui atlikti autentifikaciją net ir tuo atveju, kai centrinė naudotojų duomenų bazė yra nepasiekiama dėl vykstančios replikacijos ar kitos techninės priežasties.

Šis serveris atlieka visus veiksmus, susijusius su autentifikacijos per mobilųjį įrenginį patvirtinimu.



9 pav. Patikimo atsarginio 2FA algoritmo topologija. IAM serveriai

Atsarginis IAM serveris yra analogiškas pagrindiniam, tačiau skirtas autentifikuoti naudotojus, naudojančius ne TOTP 2FA kodus, o USB 2FA laikmenas. Šis serveris:

- Naudoja atskirą administracinę duomenų bazę, kurioje saugomi kiekvieno naudotojo šifruoti prisijungimo raktai, bei USB įrenginių identifikatoriai ir su jais susieti naudotojų raktai;
- Taip pat turi naudotojų DB repliką, užtikrinančią veikimą ir galimybę autentifikuoti net centrinei naudotojų duomenų bazei tapus laikinai nepasiekiamai.

Šifruotų raktų saugojimas atskiroje administracinėje duomenų bazėje ir naudotojų DB replikos naudojimas užtikrina, kad atsarginė autentifikacija gali būti vykdoma lokaliai ir nepriklausomai nuo mobiliojo ryšio ar trečiųjų šalių paslaugų.

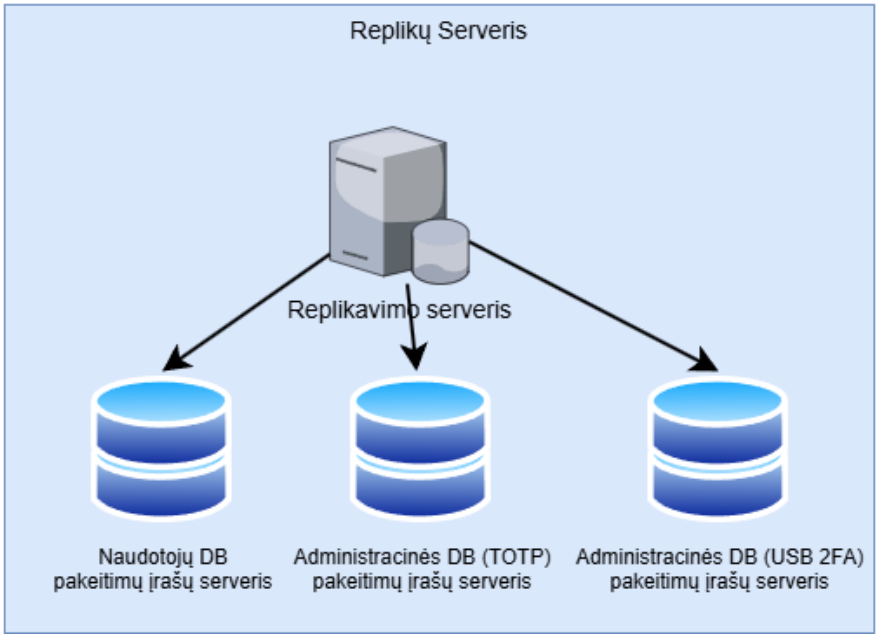
2.2.4. Replikų serveris

Duomenų replikacija tarp skirtingų IAM komponentų yra patikėta replikavimo serveriui (10), kuris užtikrina nuoseklų duomenų srautą, remdamasis galutinio nuoseklumo (angl.: eventual consistency) modeliu. Tai reiškia, kad duomenų pakeitimai neatsinaujina visose sistemose akimirksniu, tačiau per

nustatytą laiką išsilygina tarp visų duomenų bazių, užtikrinant jų nuoseklumą net ir esant tinklo ar serverių apkrovoms.

Replikų serveris atlieka šias funkcijas:

- Kaupia visus naudotojų duomenų, prisijungimo raktų, TOTP raktų ir USB 2FA įrašų pakeitimus;
- Užtikrina, kad pakeitimai būtų sinchronizuojami tarp pagrindinio ir atsarginio autentifikavimo serverių;
- Leidžia atkurti bet kurios DB repliką nuo bet kurio istorinės būsenos taško, esant poreikiui (pvz., po kibernetinio incidento ar DB korupcijos).



10 pav. Patikimo atsarginio 2FA algoritmo topologija. Replikų serveris

Replikavimo serveris nėra prieinamas jokioms kitoms sistemoms – jis veikia autonomiškai ir tik pats vykdo skaitymo ar rašymo operacijas į replikacinius duomenų šaltinius. Serveris nesaugo visos duomenų bazės atsarginės kopijos, bet kaupia tik atliktų pakeitimų (angl.: change logs) istoriją, kuri leidžia efektyviai sinchronizuoti ar atkurti duomenis nuo bet kurio laiko tarpo.

2.2.5. Web ir vidinės sistemos serveriai

Po sėkmingos autentifikacijos naudotojas yra nukreipiamas į organizacijos vidinį tinklapį, WEB serveris priima naudotojo seanso informaciją (pvz., OAuth2 žetoną), sąveikauja su vidinės sistemos serveriu, kad būtų galima patikrinti naudotojo teises ir leidimus, bei prieiti prie naudotojui autorizuotos informacijos.

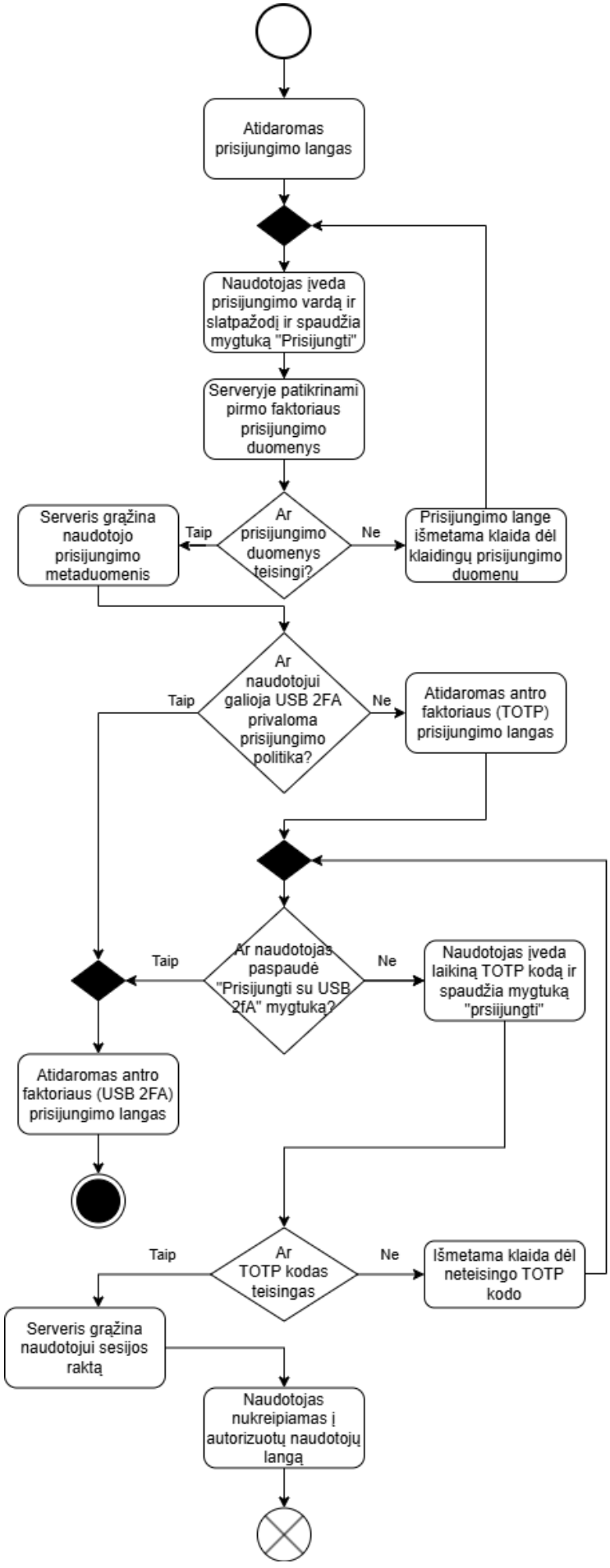
Šie serveriai yra paskutinė autentifikacijos grandinės dalis ir atsakingi už naudotojo sesijos autorizavimą bei sisteminių funkcijų teikimą.

2.3. Pagrindinis autentifikavimo procesas

Pagrindinis autentifikacijos procesas, naudojamas projekte yra grindžiamas TOTP (angl.: Time-based One-Time Password) standartu. Šis metodas užtikrina patikimą, nuo trečiųjų šalių nepriklausomą autentifikavimą, nes laikinieji slaptažodžiai generuojami lokaliai naudotojo išmaniajame įrenginyje. Procesas prasideda naudotojui atidarius prisijungimo langą sistemoje arba organizacijos internetiniame portale.

Prisijungimo lange naudotojas įveda savo prisijungimo vardą bei slaptažodį ir patvirtina tapatybę paspausdamas mygtuką „Prisijungti“. Ši informacija persiunčiama aktyviam IAM serveriui, kuris vykdo pirmojo autentifikavimo faktoriaus: slapyvardžio ir slaptažodžio patikrą. Jei prisijungimo duomenys yra neteisingi, naudotojui išmetama klaidos žinutė. Tuo atveju, jei naudotojo slaptažodis ir vardas yra teisingi, IAM serveris grąžina laikiną autentifikacijos kodą bei naudotojo metaduomenis, kuriuose aprašoma jam taikoma autentifikavimo politika.

Metaduomenyse galima nustatyti, ar konkrečiam naudotojui leidžiama autentifikuotis naudojant TOTP, USB 2FA, ar abu būdus. Naudojant gautus duomenis, tinklapio serveris gali patikrinti, ar naudotojui privalomai taikoma USB 2FA autentifikavimo politika. Jei tokia politika aktyvuota, naudotojas automatiškai nukreipiamas į atskirą USB 2FA autentifikacijos langą ir negali tęsti prisijungimo per TOTP.



11 pav. Pagrindinis autentifikavimo procesas, srauto diagrama

Jeigu naudotojas lieka pagrindiniame autentifikacijos sraute ir įveda TOTP kodą, sistema jį patikrina IAM serveryje. Jei kodas teisingas – naudotojui išduodamas galiojantis sesijos raktas (angl.: session token), kuris leidžia naudotojui prisijungti prie vidinių organizacijos sistemų. Tuo atveju, kai įvestas TOTP kodas yra neteisingas, naudotojui pateikiamas klaidos pranešimas ir suteikiama galimybė bandyti dar kartą.

2.4. Atsarginis autentifikavimo procesas

Atsarginis autentifikavimo procesas naudojamas tada, kai naudotojas negali pasinaudoti pagrindiniu antrojo faktoriaus autentifikacijos metodu (TOTP). Tokia situacija susidaro, pavyzdžiui, pametus mobilųjį telefoną arba praradus prieigą prie autentifikatoriaus programėlės

Atsarginio autentifikavimo atveju antrojo faktoriaus patikrinimas vykdomas per naudotojo kompiuterį, pasinaudojant specialiai sukonfigūruota USB laikmena lokaliai kompiuteryje paleidžia USB 2FA paslaugą (12). Autentifikacija paremta iššūkio-atsako mechanizmu [27], taikomu WebAuthn/FIDO2 specifikacijose. Serveris sugeneruoja kriptografiškai atsitiktinį vienkartinį iššūkį, kurį USB laikmenoje saugomas privatus raktas pasirašo skaitmeniniu parašu. Serveris patikrina parašą pagal registracijos metu išsaugotą viešąjį raktą. Kadangi iššūkis kiekvieną kartą yra naujas, ankstesnio atsako pakartojimas nebeveikia, taip užtikrinamas atsparumas pakartojimo atakoms.

Procesas prasideda atidarius bandant prisijungti prie organizacijos, atsidarius antrojo faktoriaus prisijungimo langą. Pirmiausia organizacijos tinklalapis patikrina pirmojo faktoriaus autentifikacijos metu grąžintus metaduomenis ir nustato, ar naudotojui leidžiamas USB 2FA metodas. Jei politika leidžia USB autentifikaciją, sistema pradeda atsarginio srauto žingsnius. Naudotojas prijungia USB laikmeną ir paleidžia joje įdiegtą autentifikatoriaus paslaugą, kuri atrakinama slaptažodžiu.

Atrakinus laikmeną, prisijungimo tinklalapis pastoviai tikrina, ar lokali USB 2FA paslauga pasiekiama iš naršyklės. Nustačius, kad USB autentifikatorius pasiekiamas - tinklalapis siunčia užklausą į organizacijos autentifikacijos serverį, prašydamas vienkartinio iššūkio. Serveris sugeneruoja 32 baitų atsitiktinę iššūkio reikšmę, užregistruoja ją „challenges“ lentelėje kartu su 120 sekundžių galiojimo laiku[20] ir grąžina ją klientui.

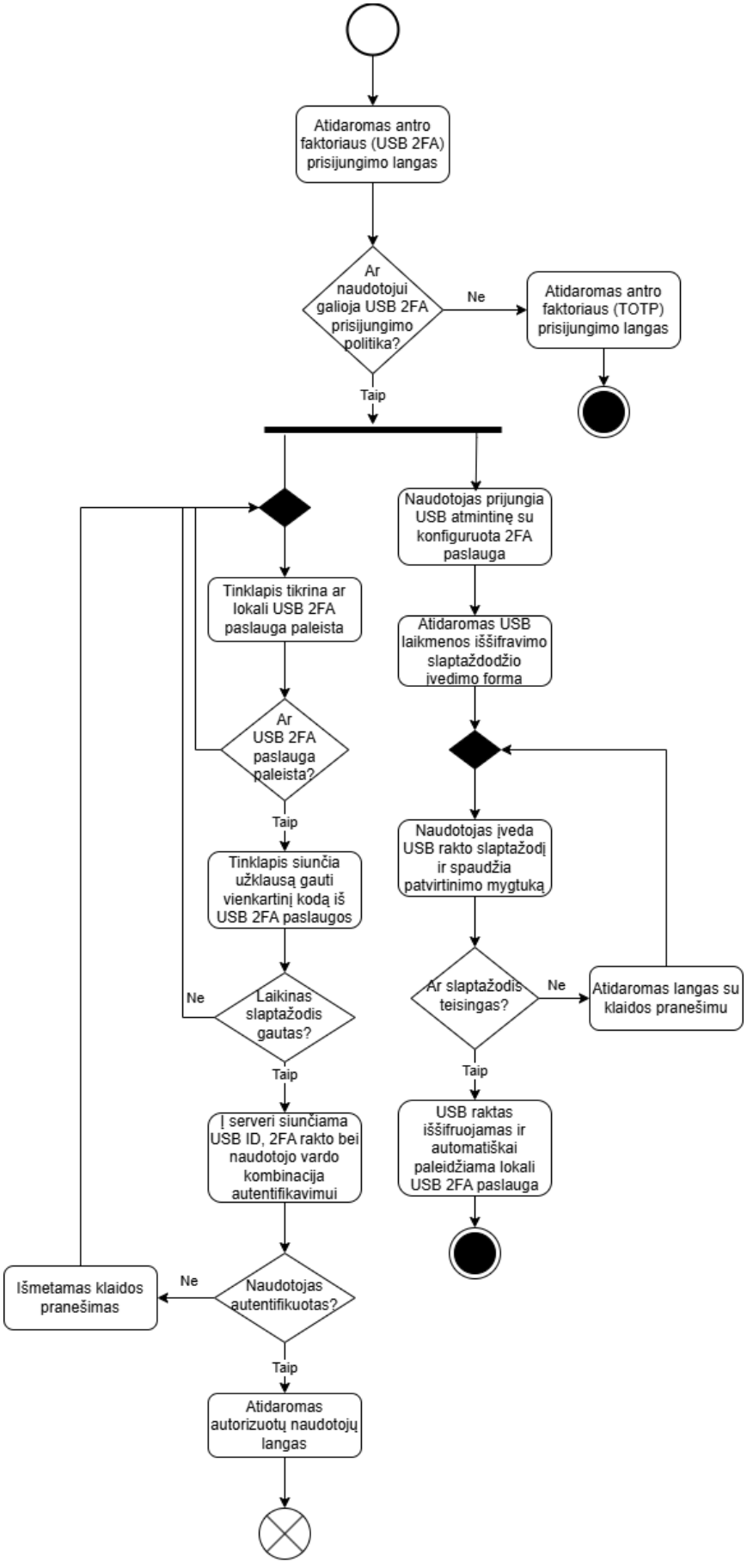
Gauto iššūkį naršyklė persiunčia USB autentifikatoriaus paslaugai. Servisas iššūkį pasirašo ECDSA P-256 privačiu raktu, laikomu tik šifruotame „keystore.enc“ faile USB laikmenoje, ir grąžina skaitmeninį parašą bei pasirašymo metaduomenis (rpId maišos reikšmę, parašų skaitliuką, kontrolės vėliavas) pagal WebAuthn autentifikacinių duomenų (angl.: authenticator data) struktūrą. Naršyklė šį paketą kartu su USB laikmenos identifikatoriumi persiunčia į organizacijos autentifikacijos serverį.

Serveris patikrina parašo teisingumą pagal naudotojo viešąjį raktą, užregistruotą USB laikmenos konfigūravimo metu, įsitikina, kad iššūkis dar nebuvo panaudotas ir nėra pasibaigęs jo galiojimo laikas, bei patvirtina, jog laikmenos identifikatorius sutampa su kredencialu.

Jei autentifikacija nepavyksta (pavyzdžiui, netinkamas parašas, iššūkis jau panaudotas ar nebegalioja, laikmena neatitinka registracijos įrašų), naudotojui pateikiamas klaidos pranešimas.

Jei autentifikacija sėkminga, naudotojui grąžinamas pilnas sesijos žetonas ir atidaromas organizacijos

[20] WebAuthn/FIDO2 specifikacija ir autentifikacijos rekomendacijos: https://www.w3.org/TR/webauthn-2/



12 pav. Atsarginis autentifikavimo procesas, srauto diagrama

Integruojant USB autentifikatorių kaip atsarginį antro faktoriaus algoritmą - organizacija pasiekia tiek techninį atsparumą, tiek aukštą autentifikacijos pasiekiamumo lygį, net ir sutrikus pagrindiniam antrojo faktoriui.

2.5. USB laikmenos 2FA paslaugos konfigūravimo procesas

USB laikmenos konfigūravimo metu USB laikmena yra paverčiama patikimu atsarginiu 2FA raktu.

Konfigūravimo procesas prasideda, kai sistemos administratorius įdeda tuščią arba svarbių duomenų nebeturinčią USB laikmeną į kompiuterį. Tuomet vykdomas pilnas laikmenos formatavimas ir sukuriama nauja FAT32 particija, kuri yra plačiai palaikoma (Windows, Linux, macOS). Pati particija paliekama nešifruota, nes USB laikmenoje visa jautri informacija bus laikoma tik viename faile - „keystore.enc“, kuris bus šifruotas AES-256-GCM algoritmu su PBKDF2-HMAC-SHA256 išvestu raktu (600 000 iteracijų, pagal OWASP rekomendaciją21). Kiti laikmenos failai: pats autentifikatoriaus vykdomasis failas ir konfigūracijos, nėra priskiriami konfidencialiai informacijai, todėl papildomas particijos lygmens šifravimo sluoksnis nesuteiktų papildomos saugumo naudos, bet tik pablogintų palaikomumą ir diegimo paprastumą. Tik jautrios informacijos šifravimas atitinka „Zero-Trust“ principą: apsaugoma tik tai, kas iš tikrųjų yra paslaptis.

Į suformatuotą diską kopijuojamas USB autentifikatoriaus paslaugos paketas, kuriame yra:

- Pasirašytas savarankiškas .NET vykdomasis failas („Authenticator.exe“), atsakingas už lokalų HTTPS servisą ir iššūkių pasirašymą;
- Užšifruotas raktų konteineris (keystore.enc) su naudotojo ECDSA P-256 privačiu raktu;
- Konfigūracijos failas (config.json) su organizacijos domeno, prievado ir leistinų šaltinių (angl.: allowed origins) nustatymais;
- Pagalbinės .NET 8 vykdomosios bibliotekos, užtikrinančios programos veikimą be papildomų priklausomybių diegimo.

Įkėlus paslaugos failus, administratorius siunčia užklausą į IAM serverį, su USB identifikacijos, bei naudotojo identifikacijos kodais, siekiant užregistruoti naują USB laikmeną kaip atsarginį 2FA. Registracijos sėkmės atveju serveris išsaugo laikmenos identifikatorių IAM serveryje ir grąžina registracijos raktą, kuris saugomas USB laikmenoje.

Registracijai nepavykus, IAM serverio paslauga grąžina klaidos žinutę, kuria remiantis, administratorius gali imtis tolimesnių veiksmų, siekiant sutvarkyti klaidą, pavyzdžiui: naudotojui jau yra priskirtas galiojantis USB 2FA raktas, administratorius turi blokuoti seną registruotą USB 2FA raktą, kad galėtų pakeisti nauju.

Sekmingai įvykdžius konfiguracijos žingsnius, naudotojas galės autentifikuotis bet kurioje organizacijos sistemoje, palaikančioje USB 2FA schemą.

2.6. Atsarginio dviejų faktorių projekto išvados

Atsarginio dviejų faktorių autentifikavimo projektas, naudojant USB laikmeną, kaip atsarginį antrą autentifikacijos faktorių yra nuo trečiųjų šalių nepriklausomas 2FA metodas ir gali būti sėkmingai integruotas į organizacijos infrastruktūrą. Projekte suformuota architektūra, leidžia sklandžiai perjungti tarp pagrindinio (TOTP) ir atsarginio (USB 2FA) antro faktoriaus autentifikavimo režimų, padeda užtikrinti autentifikacijos tęstinumą net esant techniniams sutrikimams (pvz.: kai vyksta sistemos duomenų bazių atstatymas po kibernetinės atakos).

[21] OWASP slaptažodžių saugojimo rekomendacijos: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

Suprojektuota sistema pasižymėtų aukštu lankstumo ir saugumo lygiu, nes apjungia Zero-Trust modelio principus, decentralizuotą autentifikavimo sprendimus bei lokalią OTP generaciją, nepriklausančią nuo mobiliojo ryšio ar interneto paslaugų teikėjų. Papildomas replikavimo serveris suteikia galimybę sinchronizuoti naudotojų duomenis tarp IAM serverių net ir esant tinklo trikdžiams.

Naudojant šį sprendimą, naudotojo autentifikacijos procesas išliktų nesudėtingas ir pakankamai greitas autentifikuojantis tiek pagrindiniu, tiek atsarginiu antro faktorio rėžimu. Toks sprendimas ypač aktualus organizacijoms, kurios negali sau leisti autentifikacijos paslaugų prastovų (angl.: downtime).

3. Atsarginio dviejų faktorių autentifikavimo prototipas

Remiantis antrame skyriuje suprojektuota atsarginio dviejų faktorių autentifikavimo sistema, šiame skyriuje aprašoma jos praktinė realizacija ir veikiančio prototipo kūrimas.

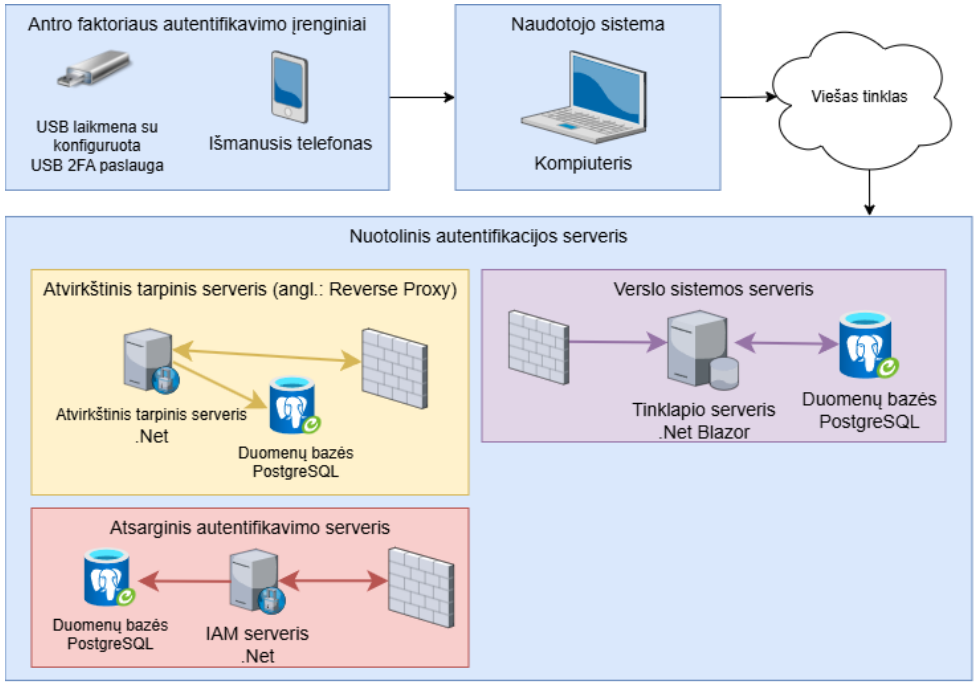
Prototipo tikslas - suteikti galimybę vartotojui prisijungti prie organizacijos informacinių sistemų naudojant alternatyvų autentifikavimo kanalą, kuris veiktų kaip patikimas atsarginis antrojo faktoriaus sprendimas šalia tradicinio TOTP metodo.

3.1. Atsarginio USB 2FA prototipo realizacija

Atsarginio dviejų faktorių autentifikavimo sistema sukurta taip, kad užtikrintų patikimą, nuo trečiųjų šalių nepriklausomą tapatybės patvirtinimo būdą, kuris galėtų veikti net esant tinklo ar mobiliosios infrastruktūros sutrikimams.

Sistemos architektūra paremta trijų loginių komponentų sąveika (13):

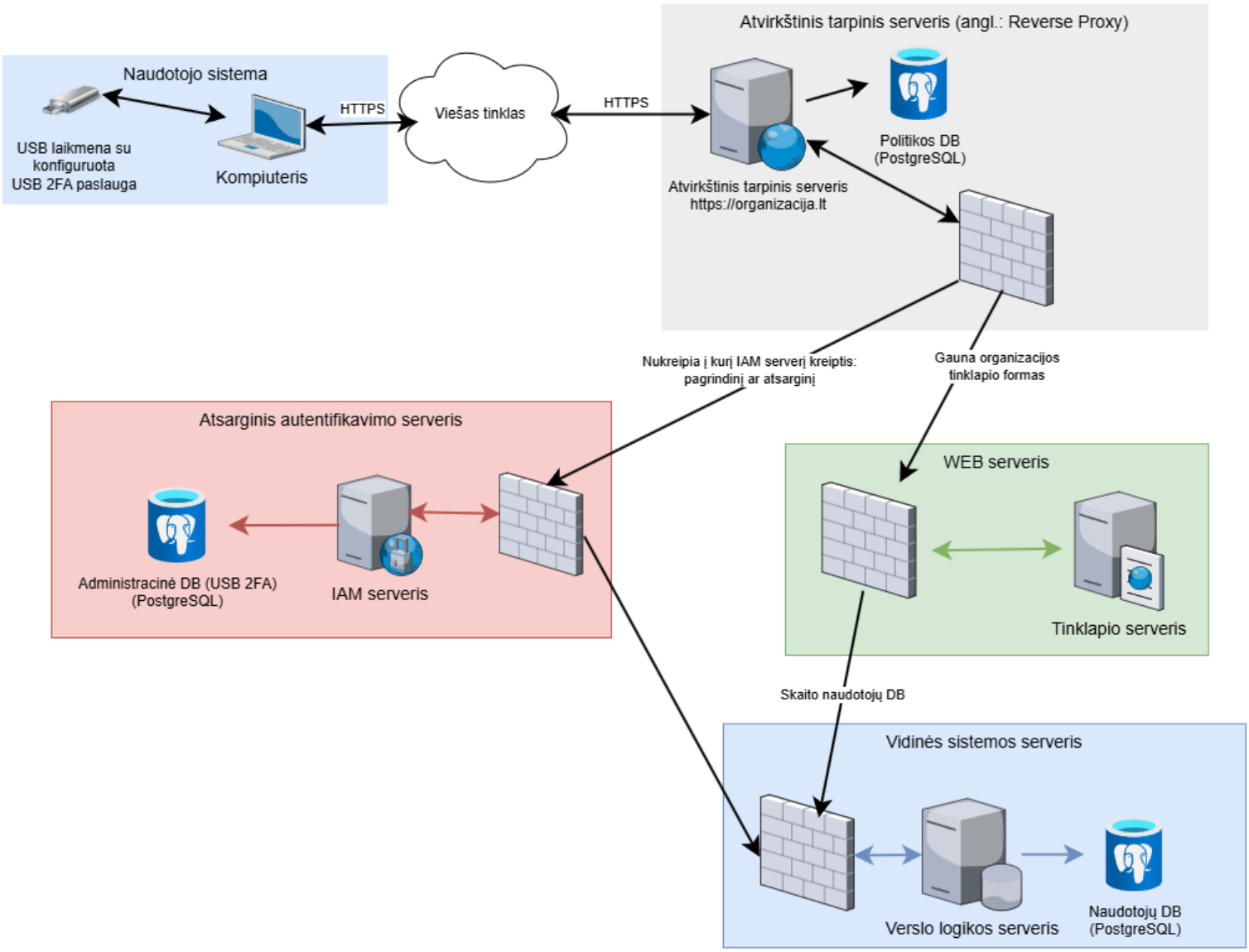
1. Naudotojo USB įrenginys su įdiegtu autentifikatoriumi – fizinis antras autentifikacijos faktorius, kuriame saugomi užšifruoti raktai ir autentifikavimo programa;
2. Naudotojo kompiuteris – tarpinis elementas tarp USB autentifikatoriaus ir nuotolinio autentifikacijos serverio;
3. Nuotolinis autentifikacijos serveris – centrinis sistemos komponentas, kuris tikrina gautus autentifikacijos duomenis, validuoja parašus ir išduoda prieigos žetonus naudotojui.



13 pav. Sistemos realizacijos loginiai sluoksniai

Pagal antrame skyriuje suprojektuotą topologiją (6), patikimas atsarginis 2FA algoritmas papildomai apima replikų serverio sluoksnį, skirtą duomenų sinchronizavimui tarp kelių IAM serverių aukšto pasiekiamumo reikalaujančiose sistemose. Šis sluoksnis prototipe neįgyvendintas, nes nėra reikalingas atsarginio 2FA algoritmo funkcionalumui demonstruoti vieno autentifikacijos serverio mazgo aplinkoje. Replikų sluoksnio realizacija paliekama kaip ateities tobulinimas, tolesniam darbo vystymui diegiant sistemą produkcinėje aplinkoje.

Realizacijos architektūra (14) paremta nulinio pasitikėjimo (angl.: Zero-Trust) principais, užtikrinančiais, kad nė vienas sistemos komponentas – nei naudotojo autentifikavimo įrenginys, nei tarpinis kompiuteris, nei nuotolinis serveris neturi besąlygiško pasitikėjimo kitu. Visi duomenų mainai tarp komponentų yra autentiški ir kriptografiškai apsaugoti, o kiekviena autentifikavimo operacija vykdoma tik gavus abipusį patvirtinimą. Tokiu būdu užtikrinama, kad kompromituotas ar netinkamai sukonfigūruotas elementas negalėtų pažeisti visos autentifikacijos grandinės.



14 pav. Prototipo architektūra

Ši architektūra užtikrina decentralizuotą tapatybės patvirtinimo procesą ir leidžia naudoti tiek įprastą TOTP kodų generatorių mobiliajame įrenginyje, tiek atsarginį USB autentifikatorių, veikiantį be ryšio su išmaniuoju telefonu ar trečiųjų šalių paslaugomis.

3.1.1. Atsarginio USB 2FA prototipe naudojami įrankiai

Prototipo kūrimas atliktas naudojant šiuolaikines atviro kodo technologijas ir Microsoft .NET ekosistemą. Atviro kodo sprendimai pasirinkti dėl jų laisvo prieinamumo, nemokamo naudojimo bei

plačių pritaikymo galimybių tiek akademinėje, tiek pramoninėje aplinkoje. Šių technologijų atvirumas leidžia laisvai analizuoti ir tobulinti programinio kodo struktūrą, o tai ypač svarbu kuriant saugumo sistemas, kuriose būtinas skaidrumas ir galimybė audituoti realizaciją.

4 lentelė. Prototipe naudojami įrankiai ir technologijos

Įrankis / Technologija	**Paskirtis**	**Versija**	**Licencija**
.NET Core	Serverio ir autentifikatoriaus programinė aplinka (C#)	8.0	MIT
Blazor WebAssembly	Vartotojo sąsajos sluoksnis	8.0	MIT
PostgreSQL	Reliacinė duomenų bazės valdymo sistema	16	PostgreSQL
NGNIX	Atvirkštinis tarpinis serveris, apkrovos balansavimas	1.28	BSD-2
Bruno	API testavimo įrankis	3.1.4	MIT
NUnit	Vienetų ir integracinių testų karkasas (.NET)	4.3	MIT

Sukurta sistema susideda iš trijų pagrindinių loginių sluoksnių: naudotojo įrenginio su USB autentifikatoriumi, naudotojo kompiuterio ir nuotolinio autentifikacijos serverio.

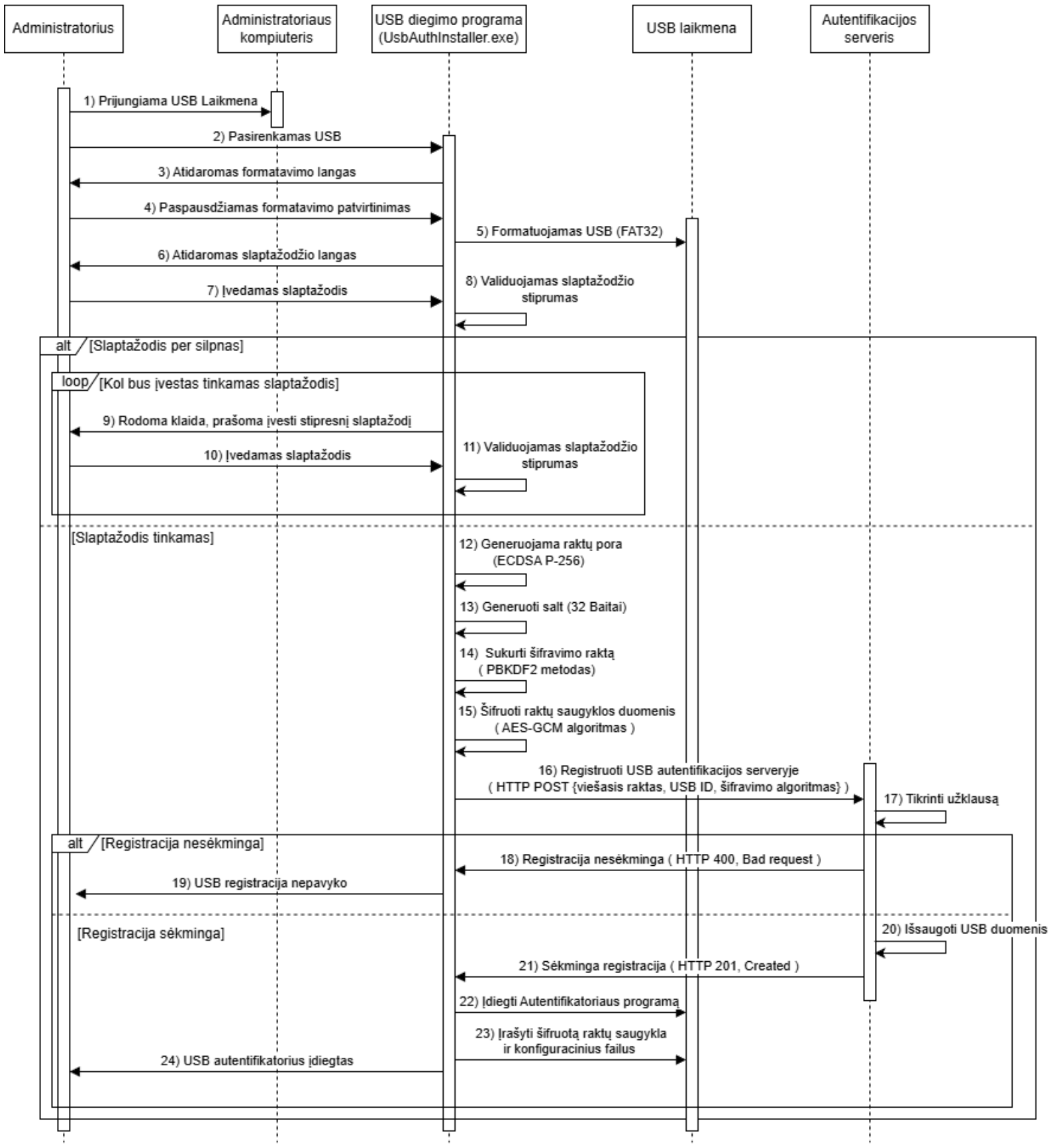
3.2. USB autentifikatoriaus diegimo prototipas

USB autentifikatorius, tai pirmasis iš trijų loginių sluoksnių: naudotojo įrenginys ir USB laikmena – veikia kaip fizinis antrojo faktoriaus įrodymas.

Atsarginiam antro faktoriaus autentifikavimui pasitelkiamas paprastas USB atminties įrenginys. Šioje laikmenoje saugoma pasirašyta autentifikavimo programa „Authenticator.exe“, parašyta C# kalba .NET 8 aplinkoje, bei užšifruota kriptografinių raktų saugykla.

USB autentifikatorius nėra specialus FIDO2 pagrįstas įrenginys – tai programinė autentifikatoriaus realizacija, veikianti paprastoje USB laikmenoje. Tokiu būdu užtikrinamas suderinamumas ir prieinamumas: autentifikavimo procesas nepriklauso nuo specialios įrangos, o laikmeną galima naudoti bet kuriame kompiuteryje su „Windows 10“ ar naujesne, „macOS 12“ ar naujesne, arba šiuolaikine „Linux“ operacine sistema (pvz., „Ubuntu 20.04+“).

Autentifikatorius gali veikti kaip atsarginis faktorius tuo atveju, kai pagrindinio TOTP kodo sugeneruoti negalima, pavyzdžiui, praradus mobilųjį telefoną.



15 pav. USB autentifikatoriaus diegimo sekų diagrama

USB autentifikatoriaus diegimo procesas susideda iš trijų pagrindinių etapų (15): laikmenos paruošimo, paslaugos failų šifravimo ir įkėlimo, bei registracijos IAM serveryje.

3.2.1. USB autentifikatoriaus laikmenos paruošimas

Pirmiausia tuščia USB laikmena formatuojama į FAT32 failų sistema. Ši failų sistema pasirinkta dėl universalaus suderinamumo – FAT32 palaiko visos pagrindinės operacinės sistemos (Windows, macOS, Linux) be papildomų tvarkyklių ar programinės įrangos diegimo. Tai užtikrina, kad naudotojas galės naudoti USB autentifikatorių bet kuriame kompiuteryje, nepriklausomai nuo jo platformos. Alternatyviai gali būti naudojama exFAT failų sistema, kuri pašalina FAT32 4 GB failo dydžio apribojimą, tačiau autentifikatoriaus atveju šis apribojimas nėra aktualus, kadangi visi failai yra mažesni nei 50 MB.

3.2.2. USB autentifikatoriaus laikmenos šifravimas ir failų įkėlimas

Šifravimui naudojamas naudotojo pasirinktas slaptažodis, kurio stiprumas tikrinamas pagal organizacijos nustatytas taisykles. Minimalūs slaptažodžio reikalavimai:

- Ilgis: ne mažiau kaip 12 simbolių;
- Didžiosios raidės: bent viena (A-Z);
- Mažosios raidės: bent viena (a-z);
- Skaitmenys: bent vienas (0-9);
- Specialūs simboliai: bent vienas (!@#$%^&* ir pan.).

Slaptažodžio stiprumo reikalavimai suformuoti atsižvelgiant į NIST SP 800-63B-4 gaires [26]. Remiantis šiuo NIST straipsniu USB laikmenos slaptažodis prototipe yra skaitomas kaip aktyvacijos paslaptis (angl.: activation secret) - jis nėra siunčiamas į serverį ir naudojamas tik raktų konteinerio iššifravimui, todėl slaptažodžiui taikomi griežtesni reikalavimai nei centralizuotiems. Pagal „Hive Systems“ 2025 m. slaptažodžių laužimo tyrimą22, 12 simbolių slaptažodis su visų tipų simboliais brutalios jėgos ataka užtruktų apie 3 milijardus metų, naudojant 12 „RTX 5090“ vaizdo plokščių su bcrypt maišos funkcija.

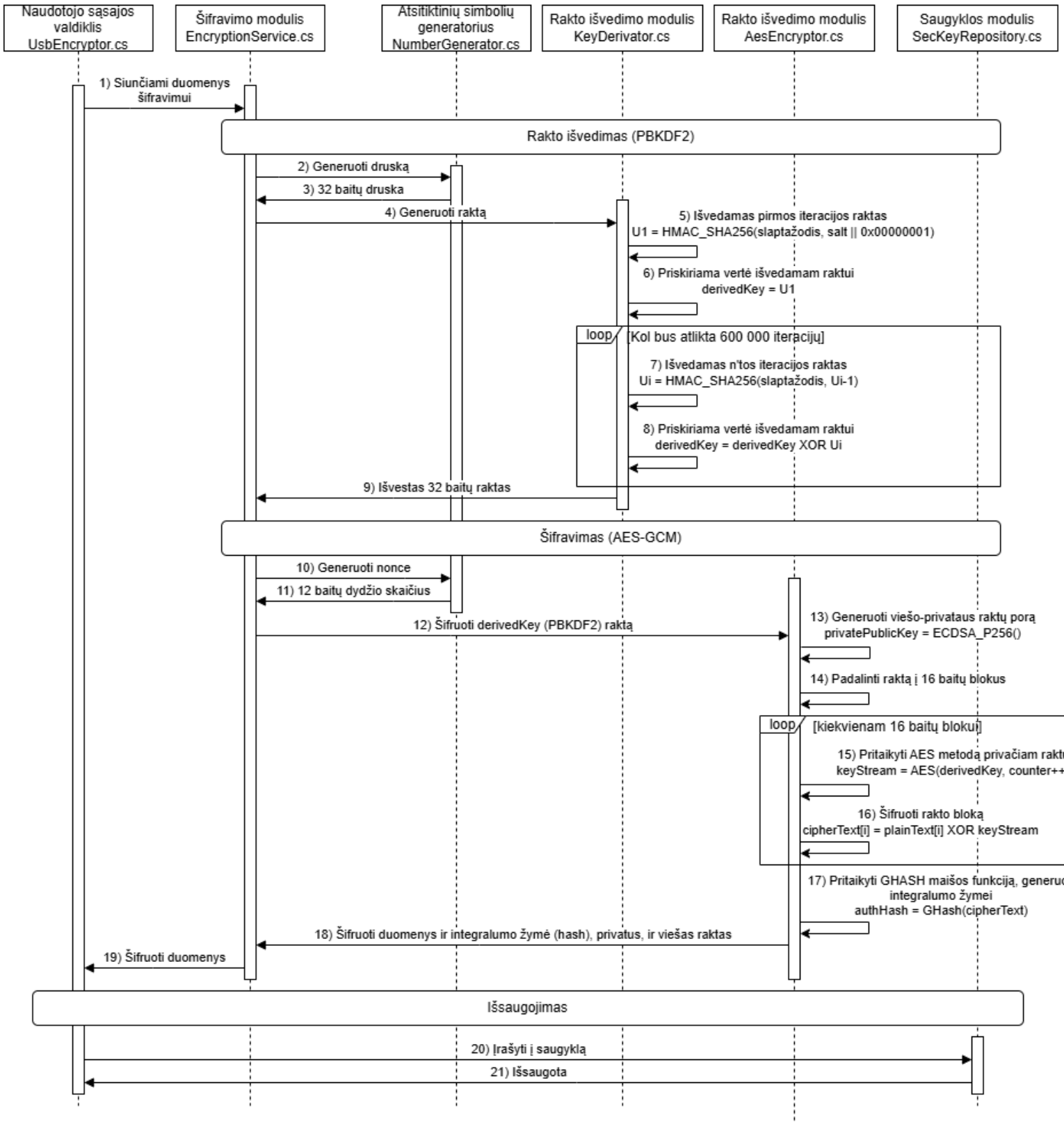
Į paruoštą laikmeną kopijuojamas autentifikatoriaus paketas, kurį sudaro:

- Authenticator.exe savarankiškas .NET 8 vykdomasis failas, nereikalaujantis papildomų bibliotekų ar diegimo;
- keystore.enc užšifruotas raktų konteineris, kuriame saugomas naudotojo privatus raktas;
- config.json užšifruotas konfigūracijos failas su organizacijos domeno ir servisų nustatymais;
- Kiti .NET vykdomieji failai, kurie užtikrina programos paleidimą be papildomų priklausomybių diegimo.

Šifravimo raktas išvedamas iš naudotojo slaptažodžio naudojant PBKDF2 (angl.: Password-Based Key Derivation Function 2) funkciją (2.1.2). PBKDF2 yra standartizuota raktų išvedimo funkcija, specialiai sukurta slaptažodžių apdorojimui, pagal RFC 8018[23] rekomendacijas [25].

[22] https://www.hivesystems.com/password-table

[23] RFC 8018 rekomendacijos PBKDF2 funkcijai: https://datatracker.ietf.org/doc/html/rfc8018



16 pav. USB autentifikatoriaus raktų šifravimo-saugojimo sekų diagrama

Prototipe naudojamos 600 000 PBKDF2-HMAC-SHA256 iteracijų, atitinkančios 2023 m. OWASP (angl.: Open Worldwide Application Security Project) rekomendacijas[24]. Pagrindinė PBKDF2 metodo paskirtis - dirbtinai sulėtinti rakto išvedimo procesą, atliekant daugybę iteracijų. Prototipe naudojamos 600 000 iteracijų - tai reiškia, kad kiekvienas bandymas atspėti slaptažodį užtrunka žymiai ilgiau nei įprasta maišos funkcija. Pavyzdžiui, jei viena SHA-256 operacija trunka mikrosekundę, tai 600 000 PBKDF2 iteracijų užtruks apie 0,3 sekundės. Tokiu būdu brutalios jėgos ataka, bandanti milijonus slaptažodžių kombinacijų, tampa nepraktiškai ilga - vietoj kelių minučių ji užtruktų metus ar dešimtmečius.

Kartu su PBKDF2 naudojama unikali 32 baitų (256 bitų) druska (angl.: *salt*) - atsitiktinai sugeneruota reikšmė, kuri pridedama prie slaptažodžio prieš raktų išvedimą. Druska išsaugoma kartu su užšifruotu raktų konteineriu ir nėra slapta. Jos paskirtis - užtikrinti, kad net du naudotojai, pasirinkę identišką slaptažodį, turėtų skirtingus šifravimo raktus. Tai apsaugo nuo iš anksto apskaičiuotų vaivorykštės lentelių (angl.: *rainbow tables*) atakų, kuriose užpuolikas naudoja didžiules duomenų bazes su iš anksto apskaičiuotomis slaptažodžių maišomis. Kadangi kiekvienas raktų konteineris turi unikalią druską, užpuolikas negali pakartotinai naudoti anksčiau atliktų skaičiavimų ir turi kiekvieną USB laikmeną atakuoti atskirai.

Tokia konfigūracija, AES-GCM šifravimas su PBKDF2 raktų išvedimu ir unikalia druska, užtikrina atsparumą tiek žodyno, tiek brutalios jėgos atakoms, net jei USB laikmena būtų fiziškai prarasta ar pavogta.

3.2.3. USB autentifikatoriaus registracija serveryje

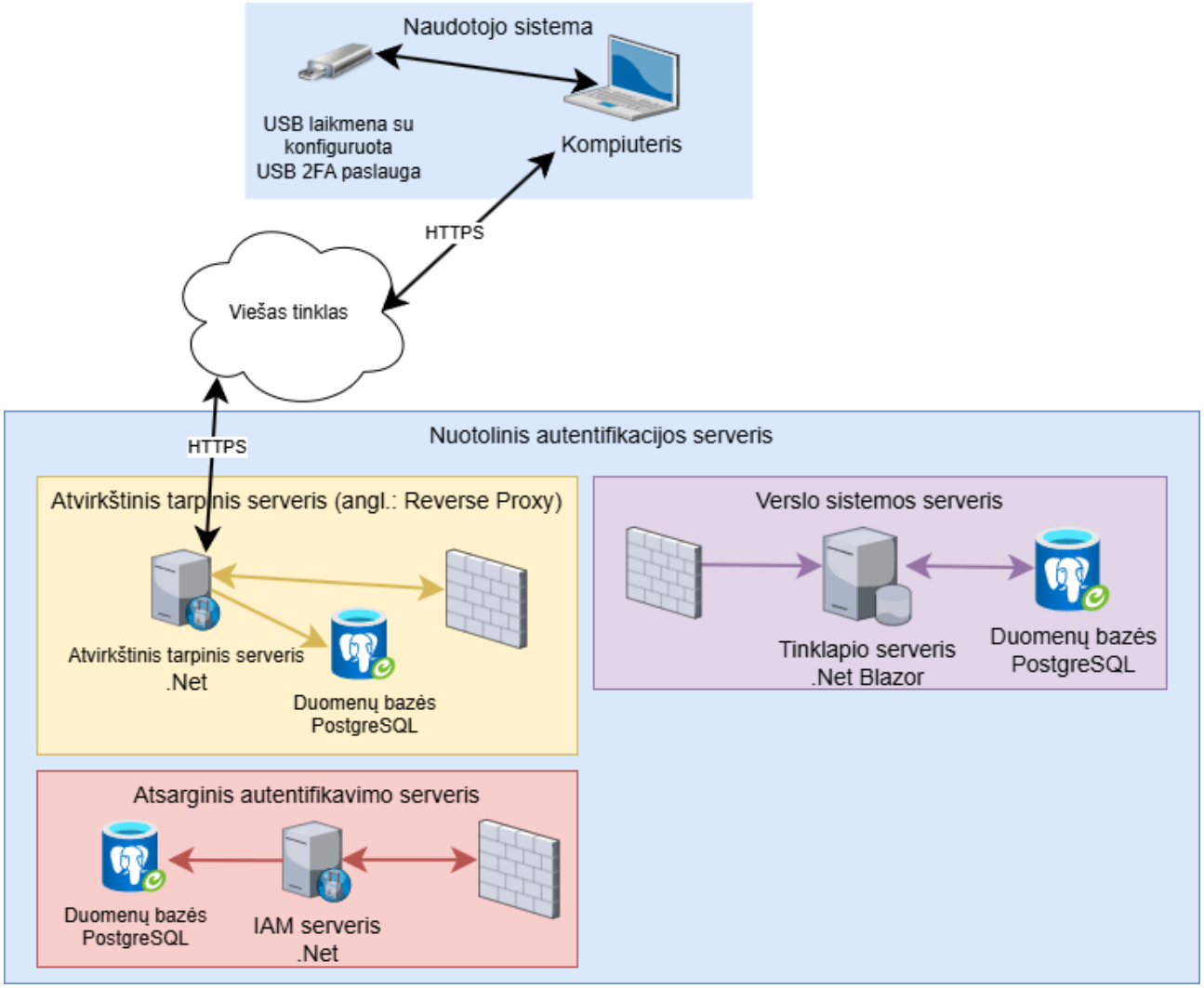
Sėkmingai įkėlus failus, administratorius inicijuoja USB laikmenos registraciją IAM serveryje. Registracijos metu sugeneruojama unikali viešojo ir privataus rakto pora naudojant ECDSA P-256 (angl.: Elliptic Curve Digital Signature Algorithm) algoritmą.

Viešasis raktas perduodamas ir išsaugomas serveryje, kur jis naudojamas autentifikacijos metu gautiems parašams tikrinti. Privatus raktas niekada nepalieka USB laikmenos - jis iš karto užšifruojamas AES-GCM algoritmu ir saugomas raktų konteineryje (keystore.enc).

3.3. Autentifikavimo proceso realizacija naudotojo kompiuteryje

Antrasis sistemos sluoksnis - naudotojo kompiuteris (17), kuris veikia kaip tarpinis elementas tarp USB laikmenos ir autentifikacijos serverio.

[24] OWASP slaptažodžių rekomendacijų atmintinė: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html



17 pav. Prototipo autentifikavimo proceso topologija

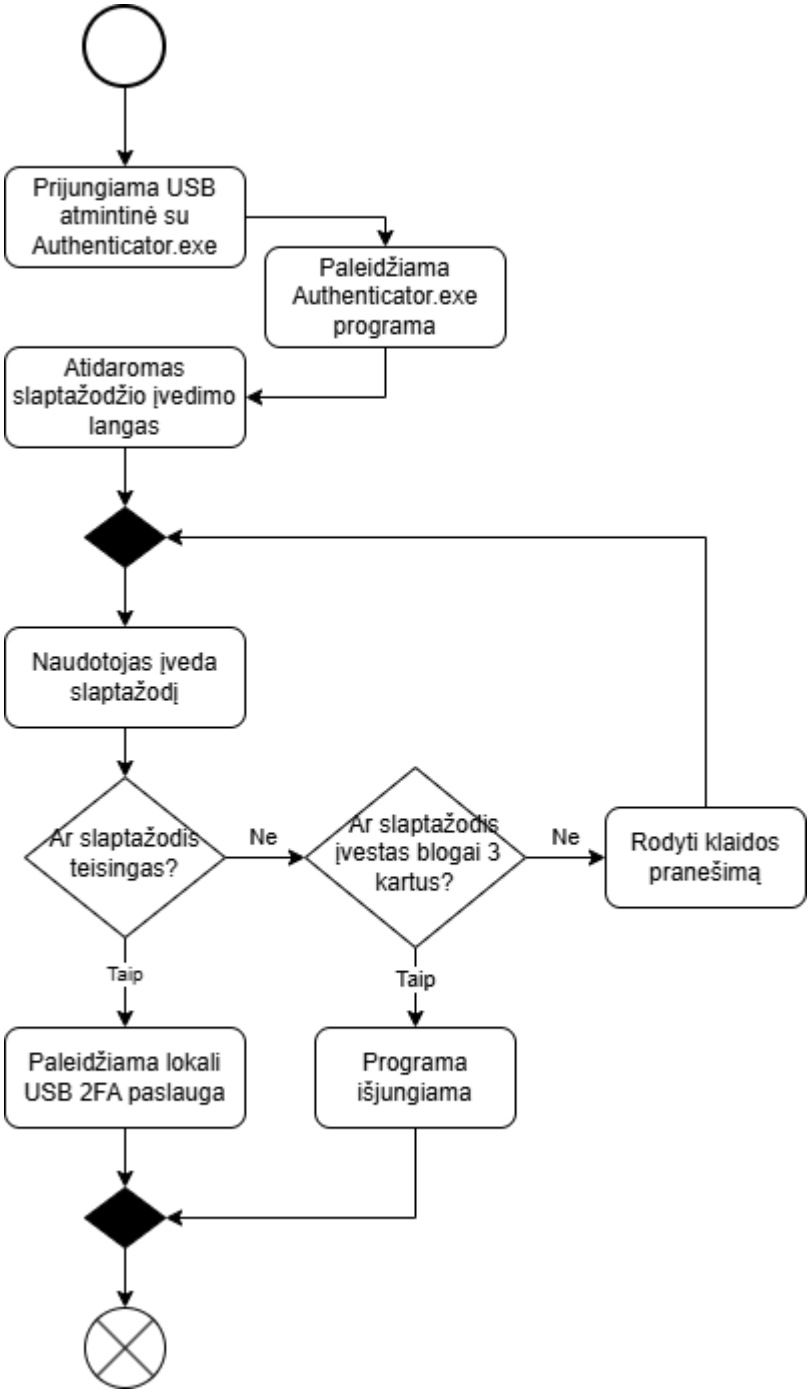
Prieš aptariant USB 2FA realizacijos detales, svarbu paminėti, kaip prototipe elgiamasi su pirminiu antrojo faktoriaus metodu, tai yra TOTP. Kadangi šio darbo tyrimo objektas yra atsarginio 2FA srauto elgsena, pirminis TOTP srautas realizuotas tik kliento pusėje kaip imituotas gedimas. Kliento puslapis „/2fa/totp“ priima bet kokį kodą, imituoja apie 1,5 sekundės tinklo vėlinimą ir visada grąžina klaidą „TOTP verification service is currently unavailable“. Toks sprendimas leidžia vienodai ir pakartotinai atkartoti „pirminis 2FA neprieinamas“ scenarijų kiekvieno eksperimento metu, nereikalaujant tikros TOTP infrastruktūros.

3.3.1. Autentifikavimo procesas. USB autentifikatoriaus paleidimas

Prijungus USB laikmeną ir paleidus Authenticator.exe, naudotojui pateikiamas slaptažodžio įvedimo langas (18). Programa įvesti slaptažodį. Po trijų nesėkmingų bandymų programa automatiškai užveriama.

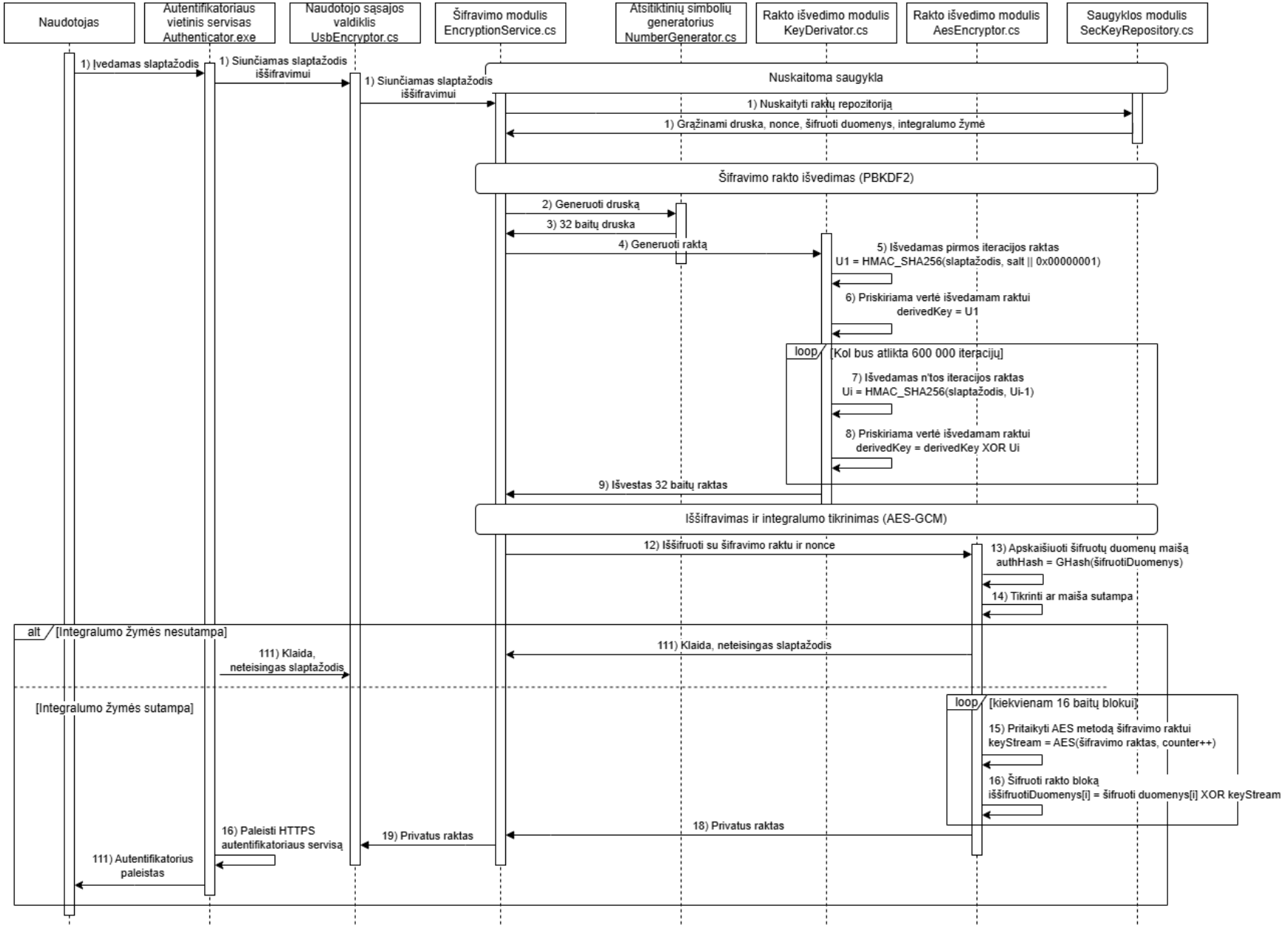
Prototipe nėra įgyvendinta programinio slaptažodžio įvedimo bandymų ribojimo (angl.: lockout), kadangi ši apsauga nėra kritinė šio darbo kontekste. Brutalios jėgos atakų atsparumas užtikrinamas pasirinktu PBKDF2-HMAC-SHA256 iteracijų skaičiumi (600 000, OWASP 2023 rekomendacija), dėl kurio vieno slaptažodžio bandymo skaičiavimo kaina yra pakankamai didelė, kad didelio masto žodyno ar brutalios jėgos atakos taptų nepraktiškos.

Tolesniuose sistemos tobulinimo etapuose rekomenduojama integruoti bandymų ribojimo mechanizmą - idealiai, programinį, ne aparatūrinį.



18 pav. USB 2FA programos Authenticator.exe paleidimo srauto diagrama

Įvedus teisingą slaptažodį, vykdomas raktų konteinerio iššifravimo procesas (19 pav.). Pirmiausia programa nuskaito keystore.enc failą, kuriame saugomi šifravimo metu išsaugoti duomenys: druska, vienkartinis skaičius (angl.: nonce: number that only can be used once), užšifruoti duomenys ir integralumo žymė. Tuomet iš naudotojo slaptažodžio ir druskos, naudojant PBKDF2 funkciją su 600 000 iteracijų, išvedamas šifravimo raktas. Galiausiai AES-GCM algoritmas patikrina integralumo žymę ir, jei ji sutampa, iššifruoja privatų raktą. Jei slaptažodis neteisingas - iššifravimas nepavyksta ir naudotojui rodomas klaidos pranešimas



19 pav. USB slaptažodžio iššifravimo sekų diagrama

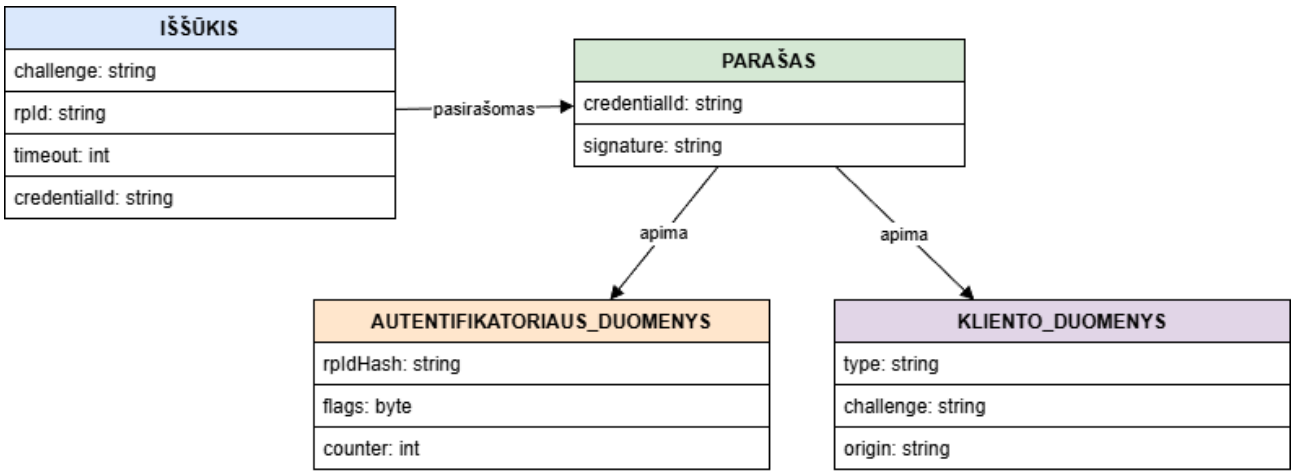
Sėkmingai iššifravus privatų raktą, jis saugomas tik programos atmintyje – niekada neįrašomas į diską. Tuomet programa automatiškai paleidžia vietinį HTTPS servisą, kuris klausosi prievado 53242. Šis servisas geba priimti užklausas tik iš organizacijos domeno (per CORS apribojimus) ir pasirašyti serverio pateiktą iššūkį naudodamas privatų raktą bei grąžinti pasirašytą atsaką.

3.3.2. Autentifikavimo procesas. Autentifikacijos serveris ir USB autentifikatorius

Siekiant užtikrinti, kad autentifikavimo paslauga būtų pasiekiama tik iš įgalioto organizacijos tinklalapio, vietiniame Authenticator.exe servise įdiegtas CORS apribojimas, leidžiantis priimti užklausas tik iš konkretaus domeno (pavyzdžiui: https://auth.organizacija.lt). Tokiu būdu, net jei kitos svetainės ar programos bandytų inicijuoti HTTPS užklausas į vietinę autentifikatoriaus paslaugą, jos būtų atmestos naršyklės lygmeny.

Šis apribojimas užkerta kelią kenkėjiško kodo injekcijos (angl.: cross-site scripting) ir sukčiavimo atakoms, kai kenkėjiškas puslapis bandytų gauti prieigą prie vartotojo autentifikatoriaus. Tokia realizacija atitinka „Zero-Trust“ principus, pagal kuriuos pasitikėjimas suteikiamas tik aiškiai apibrėžtiems subjektams.

Authenticator.exe servisas naudoja System.Security.Cryptography bibliotekas. Pasirašymo operacijai taikomi šiuolaikiniai asimetriniai algoritmai – ECDSA P-256 (16) ir Ed25519, užtikrinantys aukštą saugumo lygį bei gerą veikimo spartą. Parašai formuojami pagal WebAuthn/FIDO2 principus: pasirašomas ne tik serverio sugeneruotas iššūkis, bet ir svetainės tapatybę patvirtinantis identifikatorius (rpIdHash) (20).



20 pav. USB autentifikavimo esybių diagrama

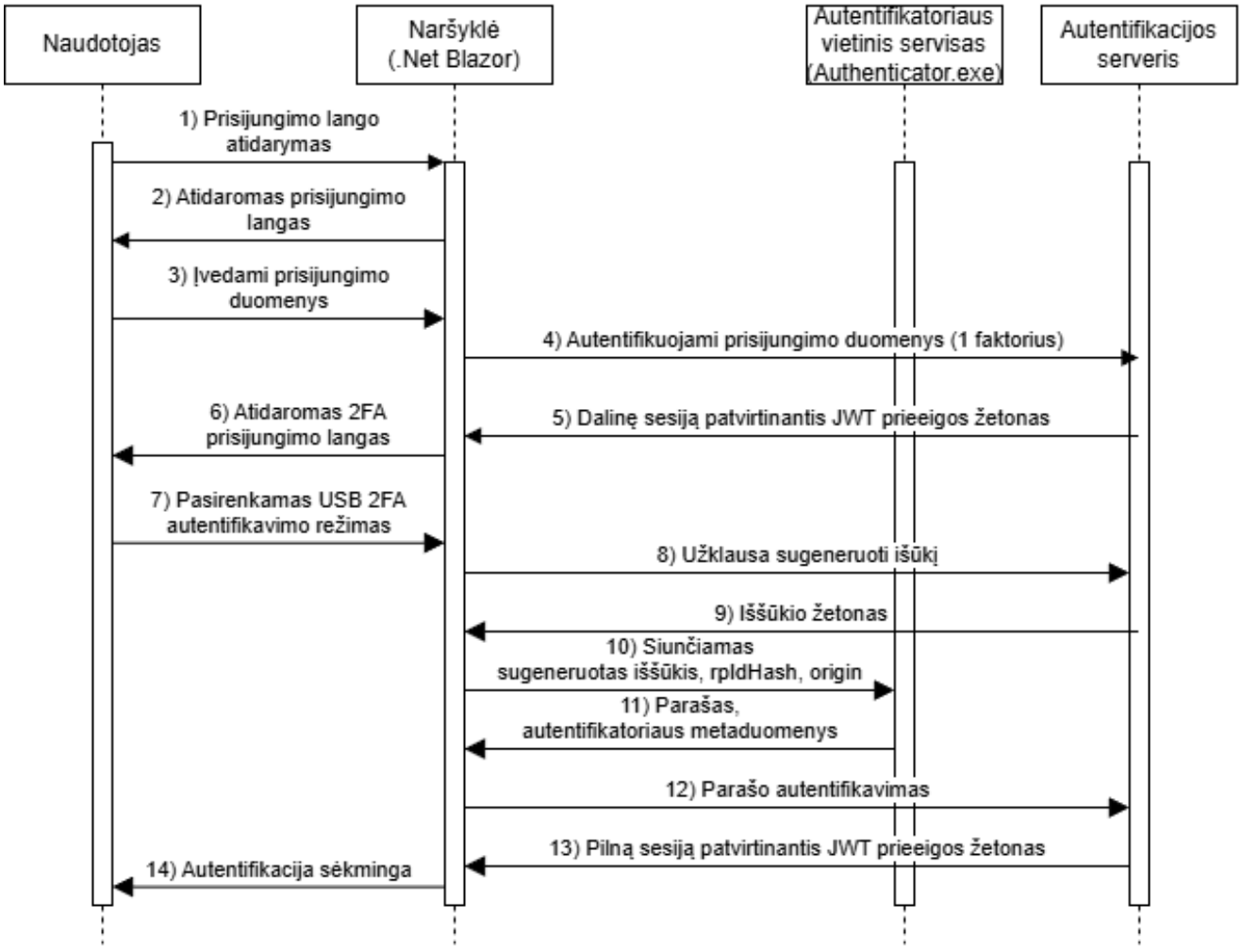
Autentifikacijos procesas (21) pagrįstas iššūkio–atsakymo mechanizmu (angl.: challenge–response), kai nuotolinis autentifikacijos serveris sukuria unikalų vienkartinį iššūkį, o autentifikatorius jį pasirašo naudodamas savo privatų raktą. Kadangi iššūkis kiekvieną kartą generuojamas naujai, pakartotinis anksčiau naudoto atsakymo pateikimas tampa bevertis, taip užkertamas kelias paartojimo ir tarpininko tipo atakoms.

Autentifikacijos procesas apima šiuos pagrindinius etapus:

1. Naudotojas per organizacijos autentifikacijos puslapį inicijuoja prisijungimą prie sistemos;

2. Serveris sugeneruoja iššūkį ir išsiunčia jį į naudotojo kompiuteryje paleistą USB autentifikatoriaus servisą;
3. Autentifikatorius pasirašo iššūkį savo privačiu raktu ir grąžina atsakymą su parašu bei metaduomenimis;
4. Serveris patikrina parašo teisingumą pagal naudotojo viešąjį raktą, įrašytą USB autentifikatoriaus registracijos metu;
5. Jei autentifikacija sėkminga, serveris išduoda JWT prieigos žetoną, suteikiantį naudotojui prieigą prie sistemos.

Šio proceso veikimą iliustruoja sekų diagrama (21) kurioje pavaizduotas duomenų apsikeitimas tarp keturių dalyvių: naudotojo, naršyklės, USB autentifikatoriaus ir nuotolinio autentifikacijos serverio.



21 pav. USB 2FA prototipo autentifikavimo sekų diagrama

Toks autentifikavimo modelis užtikrina abipusį pasitikėjimą tarp naudotojo ir serverio bei pilnai atitinka Zero-Trust architektūros principus.

USB autentifikatorius šiuo atveju veikia kaip nepriklausomas, fizinis saugumo komponentas, kuris gali būti naudojamas net praradus ar laikinai neturint prieigos prie mobiliojo 2FA.

3.4. Nuotolinio autentifikacijos serverio prototipas

Trečiasis sluoksnis - nuotolinis autentifikacijos serveris - realizuotas naudojant .NET 8 karkasą bei .NET Blazor. Serveris yra atsakingas už vartotojų tapatybės patvirtinimą, prieigos žetonų (JWT) išdavimą ir sesijų valdymą. Prototipo realizacijoje demonstruojamas atsarginis autentifikacijos modulis, įgyvendinantis USB 2FA iššūkio-atsakymo mechanizmą.

3.4.1. Autentifikacijos serverio struktūra

Autentifikacijos serveris realizuotas kaip RESTful HTTPS API servisas. Šis servisas projektuotas laikantis REST architektūros principų, kur kiekvienas galutinis taškas atitinka konkrečią autentifikacijos operaciją (5).

5 lentelė. Autentifikacijos API galutinių taškų specifikacija

Galutinis taškas (angl.: endpoint)	**Metodas**	**Paskirtis**	**Įvestis**	**Išvestis**
/api/auth/login	POST	Pirmojo faktoriaus autentifikacija	username, password	partialToken, userMeta
/api/auth/usb/challenge	POST	Iššūkio generavimas	partialToken, credentialId	challenge, rpId, timeout
/api/auth/usb/verify	POST	USB parašo patikra	challenge, signature, authenticatorData	fullToken
/api/auth/token/refresh	POST	Žetono atnaujinimas	refreshToken	newAccessToken
/api/auth/logout	POST	Sesijos užbaigimas	accessToken	status

Pirmojo faktoriaus autentifikavimui skirtas galutinis taškas „/api/auth/login“ priima POST užklausą su naudotojo prisijungimo vardu ir slaptažodžiu. Sėkmingos autentifikacijos atveju serveris grąžina dalinį sesijos žetoną bei naudotojo metaduomenis, nurodančius, kokie antrojo faktoriaus metodai jam leidžiami. Šis dalinis žetonas galioja tik 5 minutes ir suteikia teisę tęsti autentifikacijos procesą, tačiau nesuteikia prieigos prie apsaugotų sistemos resursų.

Antrojo faktoriaus autentifikavimui realizuoti du atskiri galutiniai taškai. Prototipe TOTP autentifikacija realizuota tik kliento pusėje kaip imituotas gedimas. USB 2FA autentifikacijai sukurti USB 2FA autentifikacijai sukurti du susiję galutiniai taškai: „/api/auth/usb/challenge“ generuoja ir grąžina iššūkį, o „/api/auth/usb/verify“ priima pasirašytą atsaką ir atlieka parašo validaciją.

Sėkmingai užbaigus abiejų faktorių autentifikaciją, serveris išduoda pilną JWT prieigos žetoną per „/api/auth/token“ galutinį tašką. Šis žetonas naudojamas visoms tolesnėms užklausoms į apsaugotus sistemos resursus.

3.4.2. Naudotojų slaptažodių saugojimas

Naudotojų slaptažodžiai autentifikacijos serverio „users“ lentelėje saugomi ne atviru tekstu, o maišos pavidalu, naudojant Argon2id algoritmą. Argon2id yra 2015 m. „Password Hashing Competition“ konkurso laimėtojas ir nuo 2021 m. yra laikomas standartizuota funkcija (RFC 9106[25]).

Prototipe naudojami šie Argon2id parametrai: atminties sąnauda m = 64 MiB (65 536 KiB), iteracijų skaičius t = 3, lygiagretumas p = 4, druskos ilgis 16 baitų, maišos ilgis 32 baitai. Parametrai atitinka „OWASP Password Storage Cheat Sheet“[26] rekomendacijas. Kiekvienam slaptažodžiui generuojama atsitiktinė druska, o rezultatas saugomas PHC eilutės formatu („$argon2id$v=19$m=65536,t=3,p=4$<salt>$<hash>„).

3.4.3. JWT žetonų struktūra ir pasirašymas

Prototipo autentifikacijos serveris naudoja RS256 algoritmą (RSA su SHA-256) JWT žetonų pasirašymui, raktų pora yra RSA 2048 bitų. Naudojami trys žetonų tipai ir jų galiojimo laikai: dalinis, prieigos ir atnaujinimo žetonai.

Dalinis žetonas (angl.: partial token), galioja 5 minutes. Išduodamas po sėkmingo pirmojo faktoriaus (slaptažodžio) tikrinimo, turi „partial=true“ reikalavimą (angl.: claim). Naudojamas antrajam faktoriui (USB) inicijuoti.

Prieigos žetonas (angl.: access token), galioja 60 minučių. Išduodamas po sėkmingos dviejų faktorių autentifikacijos.

Atnaujinimo žetonas (angl.: refresh token), galioja 7 dienas. Saugomas serverio „sessions“ lentelėje. Tai ne JWT, o 32 baitų atsitiktinė reikšmė, leidžianti serveriui anuliuoti sesiją (JWT žetonus atšaukti sudėtinga, atnaujinimo žetonas šią problemą apeina).

3.4.4. Greičio ribojimas USB autentifikatoriuje

Naudotojo kompiuteryje paleistas USB autentifikatoriaus servisas taiko užklausų ribojimą pagal kliento IP adresą: ne daugiau kaip 10 užklausių per 60 sekundžių vienam IP. Viršijus ribą, grąžinamas HTTP 429 („Too Many Requests“) statusas. Šis mechanizmas apsaugo nuo pernelyg dažnų užklausų (pavyzdžiui, piktavališko naršyklės plėtinio).

3.4.5. Duomenų bazės architektūra

Serverio pusėje naudojama atviro kodo PostgreSQL 16 duomenų bazė. Duomenų bazė suprojektuota atsižvelgiant į saugumo reikalavimus ir efektyvų duomenų atskyrimą tarp skirtingų autentifikacijos metodų.

[25] RFC 9106 Argon2id funkcija: https://doi.org/10.17487/RFC9106

[26] OWASP slaptažodžių talpinimo rekomendacijos: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

6 lentelė. Autentifikacijos serverio duomenų bazės lentelės

Lentelė	Paskirtis	Pagrindiniai laukai
users	Naudotojų duomenys	id, username, password_hash, created_at, updated_at
credentials	2FA kredencialai	id, user_id, type, public_key, status, created_at
sessions	Aktyvios sesijos	id, user_id, token_hash, expires_at, ip_address
audit_logs	Audito įrašai	id, user_id, action, ip_address, result, timestamp
challenges	Autentifikacijos iššūkiai	id, credential_id, challenge, is_used, created_at, expires_at, used_at

Pagrindinė „users“ lentelė saugo esminius naudotojų duomenis: unikalų identifikatorių, prisijungimo vardą, slaptažodžio maišą (angl.: hash) bei sukūrimo ir atnaujinimo laiko žymes.

Atskira „credentials“ lentelė saugo antrojo faktoriaus duomenis. Lentelėje saugomas kredencialo tipas, viešasis raktas, bei kredencialo būsena (pvz.: aktyvus, atšauktas, laikinai užblokuotas).

Sesijų valdymui skirta „sessions“ lentelė registruoja visas aktyvias naudotojų sesijas, įskaitant išduotų žetonų maišas, galiojimo laiką, IP adresą ir naudotojo agento informaciją. Tai leidžia administratoriams stebėti aktyvias sesijas ir prireikus jas atšaukti.

Audito tikslams sukurta „audit_logs“ lentelė, kurioje fiksuojami visi autentifikacijos bandymai - tiek sėkmingi, tiek nesėkmingi. Kiekvienas įrašas apima laiko žymę, naudotojo identifikatorių, veiksmo tipą, IP adresą, naudotojo agentą bei rezultatą. Ši informacija būtina saugumo incidentų tyrimui ir atitikties reikalavimų užtikrinimui.

USB autentifikatoriams papildomai sukurta „usb_devices“ lentelė, sauganti įrenginio identifikatorių, parašų skaitliuką ir paskutinio naudojimo laiką. Parašų skaitliukas yra esminis saugumo elementas, apsaugantis nuo pakartojimo atakų - kiekvienas naujas parašas turi turėti didesnę skaitliuko reikšmę nei ankstesnis.

Prototipas papildomai naudoja „challenges“ lentelę, kurioje laikinai saugomi serverio sugeneruoti vienkartiniai iššūkiai USB 2FA autentifikacijai. Kiekvienas įrašas turi 120 sekundžių galiojimo laiką („expires_at“), atitinkantį „W3C WebAuthn Level 2“ specifikacijos[27] rekomenduojamą iššūkio galiojimo laiką, vienkartinio naudojimo žymą („is_used“) bei pasirašymo momento žymą („used_at“). Parašo tikrinimo metu iššūkio įrašas žymimas kaip panaudotas ir toliau nebegali būti pakartotinai naudojamas. Taip apsisaugoma nuo pakartojimo atakų. Alternatyvus sprendimas būtų saugoti iššūkius operatyviojoje atmintyje, tačiau toks variantas yra sunkiau audituojamas, todėl prototipe pasirinkta naudoti duomenų bazės įrašus.

Svarbu paminėti, kad serveris nesaugo naudotojo privataus rakto ar kitų jautrių kriptografinių paslapčių. Privatus raktas niekada nepalieka USB laikmenos „keystore.enc“ failo, o serverio pusėje

[27] W3C WebAuthn Level 2 specifikacija: https://www.w3.org/TR/webauthn-2

saugomas tik kredencialo viešasis raktas „credentials.public_key“. Toks sprendimas atitinka nulinio pasitikėjimo (angl.: Zero-Trust) modelio principus: net kompromitavus autentifikacijos serverį, užpuolikas negalėtų suklastoti naudotojo parašų be tiesioginės prieigos prie USB laikmenos ir ją atrakinančio slaptažodžio.

3.5. Prototipo realizacijos išvados

Šiame skyriuje aprašytas ir sukurtas atsarginio dviejų faktorių autentifikavimo prototipas, kurio struktūra ir techniniai sprendimai realizuoti pagal antrajame skyriuje apibrėžtą projektą.

Atlikti darbai ir jų rezultatai:

- Realizuota trijų sluoksnių architektūra, apjungianti USB autentifikatorių, naudotojo kompuerį ir nuotolinį autentifikacijos serverį. Kiekvienas komponentas veikia savarankiškai, o tarpusavio sąveika paremta nulinio pasitikėjimo (Zero-Trust) principais.
- Įdiegtas atsarginis 2FA mechanizmas pagal WebAuthn Level 2 iššūkio-atsako protokolą. Iššūkiai generuojami kriptografiškai saugiu būdu, galioja 120 sekundžių pagal W3C rekomendacijas ir saugomi „challenges“ lentelėje su vienkartinio naudojimo apsauga nuo pakartojimo atakų.
- USB autentifikatoriaus raktų saugyklai („keystore.enc“) naudojamas AES-256-GCM algoritmas su PBKDF2-HMAC-SHA256 raktų išvedimu (600 000 iteracijų pagal OWASP rekomendaciją). Privatus raktas niekada nepalieka USB laikmenos, o serveryje saugomas tik viešasis raktas.
- USB autentifikatoriaus servisas užtikrina užklausų ribojimą (10 užklausų per minutę vienam IP), apsaugantį nuo pernelyg dažnų užklausų.
- Serveryje naudotojų slaptažodžių maišos saugomos Argon2id pavidalu (RFC 9106, OWASP 2023 rekomendacija). Sesijos valdomos JWT žetonais (RS256, RSA-2048) su trimis žetonų tipais: dalinis (5 min), prieigos (60 min) ir atnaujinimo (7 d.).

Toliau, ketvirtajame skyriuje bus eksperimentiškai įvertinta, ar sukurtas prototipas iš tikrųjų tenkina pagrindinį darbo uždavinį – leisti naudotojui sėkmingai užbaigti prisijungimą, kai pirminis 2FA metodas tampa nepasiekiamas, ir ar autentifikavimo trukmė bei patikimumas atitinka praktinius naudotojų reikalavimus.

4. Eksperimentinis atsarginio 2FA algoritmo tyrimas

Sukurto prototipo vertinimas grindžiamas pagrindiniu darbo tikslu - patikrinti, ar USB laikmenos pagrindu realizuotas atsarginis 2FA metodas gali pakeisti pagrindinį autentifikavimo kanalą tais atvejais, kai TOTP autentifikavimas tampa nepasiekiamas arba nepatikimas.

Vertinant prototipą svarbiausia nustatyti, ar sistema teisingai realizuoja pagrindinius autentifikavimo scenarijus, tinkamai reaguoja į klaidingus arba piktavališkus bandymus ir ar atsarginio prisijungimo procesas išlieka prieinamas eiliniam organizacijos naudotojui.

Eksperimente TOTP komponentas neįtrauktas į vertinimo apimtį, nes prototipe jis realizuotas kaip imituotas gedimas, atkartojantis „pirminis 2FA neprieinamas“ scenarijų. Eksperimentas matuoja tik tuos autentifikacijos kelius, kurie apima USB 2FA atsarginio algoritmą, nes būtent atsarginis 2FA algoritmas yra šio darbo tyrimo objektas.

4.1. Testavimo aplinka ir vertinimo kriterijai

Vertinimas atliekamas naudojant 3 skyriuje aprašytą prototipą, sudarytą iš trijų loginių komponentų: USB autentifikatoriaus, naudotojo kompiuterio ir nuotolinio autentifikacijos serverio. USB laikmenoje saugoma autentifikavimo programa ir užšifruotas raktų konteineris, naudotojo kompiuteris veikia kaip tarpinė autentifikatoriaus vykdymo aplinka, o serveris tikrina gautus duomenis, validuoja parašus ir išduoda prieigos žetonus.

Eksperimentas atliktas viename darbo vietos kompiuteryje. Tiek nuotolinis autentifikacijos serveris, tiek naudotojo dalis veikė tame pačiame įrenginyje, kad būtų galima atkartoti pilną autentifikavimo srautą be papildomos infrastruktūros. Aparatinės ir programinės įrangos parametrai pateikiami lentelėje.

7 lentelė. Testavimo aplinkos parametrai

Parametras	**Reikšmė**
Operacinė sistema	Windows 10 Pro (versija 10.0.19045)
Procesorius	Intel Core i5-9600K
Operatyvioji atmintis	32 GiB
USB laikmena	Kingston DataTraveler 3.0, 64 GiB
Laikmenos failų sistema	FAT32 (suformatuota diegimo metu)
Vykdymo aplinka	.NET 8 (USB autentifikatorius, USB diegimo įrankis, autentifikacijos serveris), .NET 9 (žiniatinklio aplikacija)
Naršyklė	Google Chrome (stabilus kanalas)
Konteinerių aplinka	Rancher Desktop (Windows, WSL2)
Duomenų bazė	PostgreSQL 16 (paleidžiama per docker compose)

Serveris paleistas standartine komandine eilute „docker compose up -d“, tinklapio aplikacija ir autentifikacijos serveris paleisti kaip lokalūs .NET procesai. USB autentifikatorius paleidžiamas

tiesiogiai iš USB laikmenos, be administratoriaus teisių. Prieš kiekvieną patikimumo scenarijų duomenų bazė atkuriama į pradinę būseną komandomis „docker compose down -v“ ir „docker compose up -d“, taip užtikrinant, kad ankstesnių bandymų artefaktai (sesijos, iššūkiai, audito įrašai) netrikdytų sekančio scenarijaus.

Vertinimui taikomi trys kriterijai:

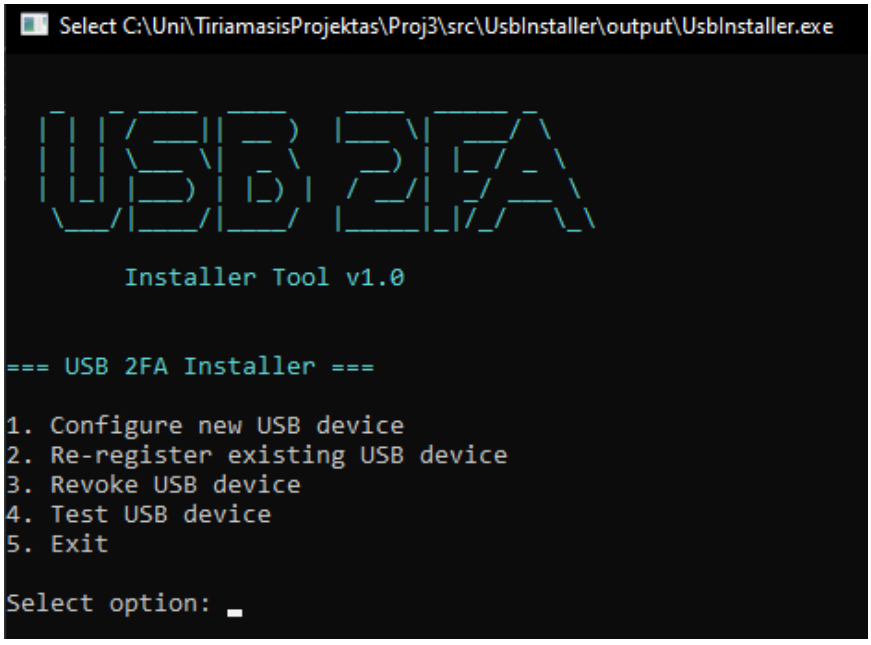
- Funkcionalumas - ar sistema visuose žingsniuose elgiasi taip, kaip aprašyta 2 ir 3 skyriuose.
- Patikimumas - ar sistema atmeta aiškiai apibrėžtus piktavališkus arba klaidingus bandymus su iš anksto numatytu rezultatu.
- Pritaikomumas - ar prisijungimą atsarginiu metodu gali atlikti įprastas naudotojas savo darbo vietos kompiuteryje be specializuotos įrangos ar administratoriaus teisių.

Vertinant prototipą šie kriterijai atsakys ir pagrindinius eksperimento klausimus: ar prototipas veikia taip, kaip suprojektuota, ar jis atmeta klaidingus arba piktavališkus bandymus ir ar jį gali panaudoti eilinis naudotojas savo darbo vietoje. Tolesniuose skyriuose aprašoma eksperimento eiga pagal kiekvieną išvardintą kriterijų.

4.2. Funkcinis prototipo patikrinimas

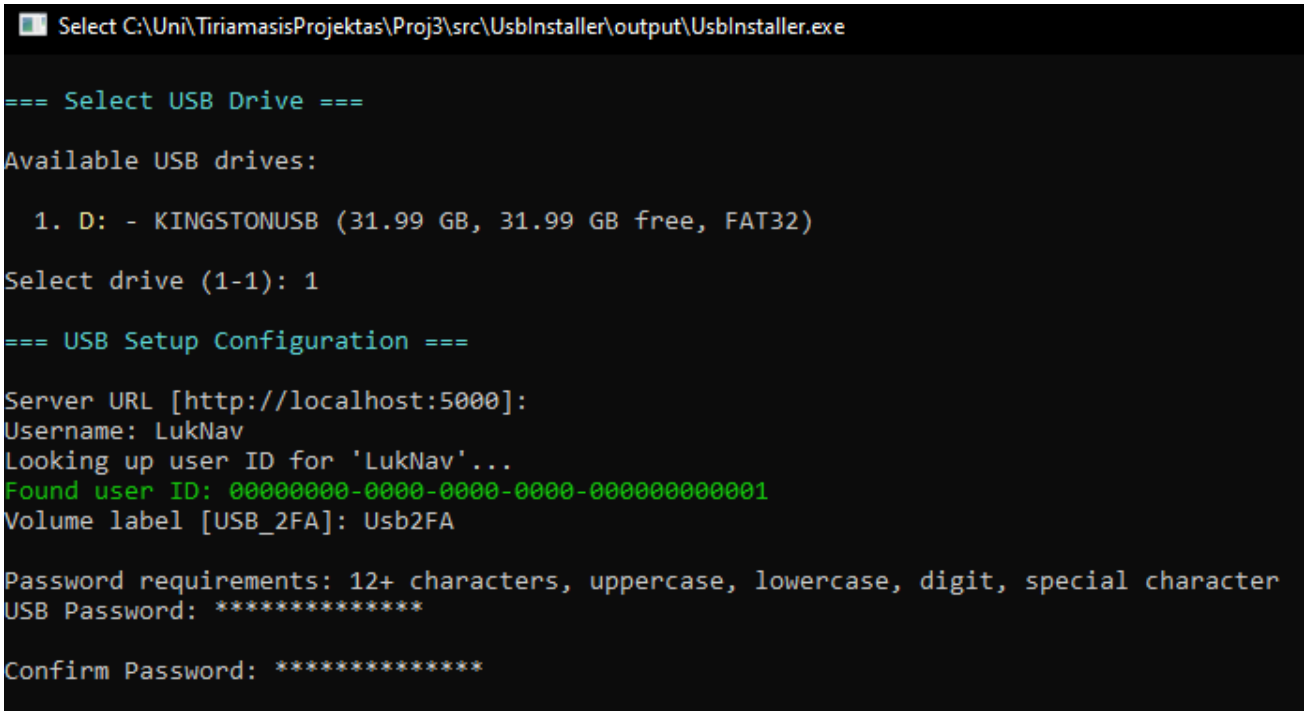
Funkcinis patikrinimas atliekamas vienu nuosekliu scenarijumi: naudotojui administratorius pasiruošia atsarginį antrojo faktoriaus USB įrenginį, o vėliau naudotojas juo pasinaudoja, kai pirminis TOTP 2FA metodas yra nepasiekiamas. Pradinės sąlygos: nuotolinis autentifikacijos serveris paleistas, duomenų bazė pradinės būsenos, USB laikmena prijungta prie kompiuterio, o organizacijos administratorius turi prieigą prie USB diegimo įrankio.

Pirmiausia administratorius per autentifikacijos serverio API sukuria naują naudotoją. Tuomet paleidžiamas USB diegimo įrankis (22).



22 pav. USB diegimo įrankio pradinis langas.

USB diegimo įrankiui atsidarius administratorius pasirenka konfiguruoti naują USB laikmeną (23), tuomet įveda naudotojo numatytą slaptažodį, naudotojo slapyvardį, naują USB pavadinimą, bei pakeičia ar patvirtina kitus numatytus parametrus (pvz.: serverio nuoroda).



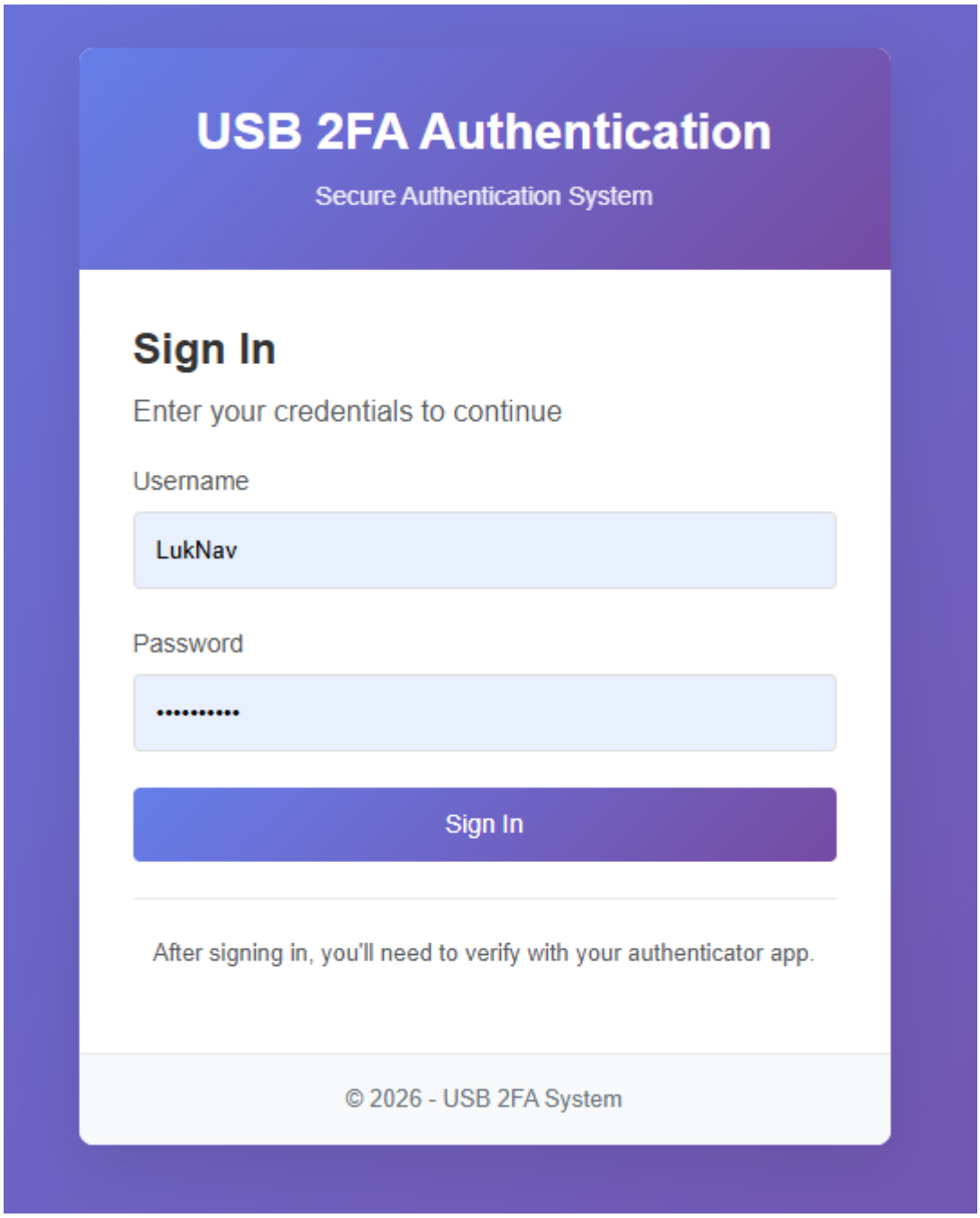
23 pav. USB 2FA konfiguracijos langas

Įrankis suformatuoja laikmeną į FAT32, sugeneruoja ECDSA P-256 raktų porą, per Windows valdymo įrankių sąsają (WMI) nuskaito laikmenos fizinį identifikatorių (PNPDeviceID), užregistruoja viešąjį raktą drauge su nuskaitytu fiziniu identifikatoriumi serveryje, sukuria ir užšifruoja raktų konteinerį „keystore.enc“ ir nukopijuoja į laikmenos pagrindinį katalogą autentifikatoriaus paketą: vykdomąjį failą, raktų konteinerį, paslaugos konfigūracijos failą bei trumpą naudotojo instrukcijų failą.

24 pav. USB konfiguracijos patvirtinimo langas

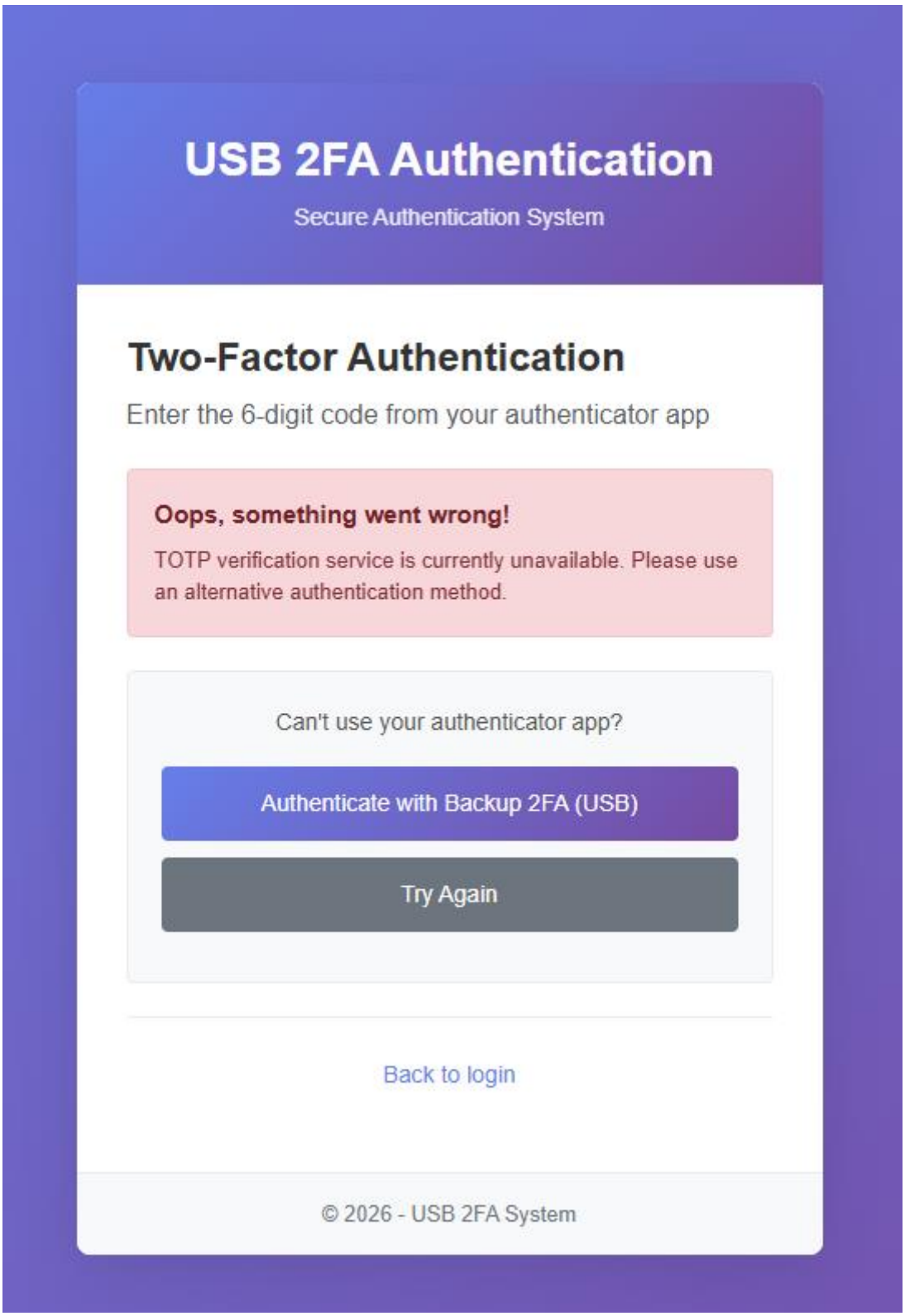
USB 2FA įrankio diegimo pabaigoje (24), patikrinama, kad duomenų bazės „usb_devices“ lentelėje atsiranda įrašas su užpildytu naudotojo identifikatoriaus, viešojo rakto ir įrenginio identifikatoriaus laukais.

Toliau naudotojas naršyklėje atidaro organizacijos tinklapį (25), įveda prisijungimo vardą ir slaptažodį.



25 pav. Prisijungimo puslapis

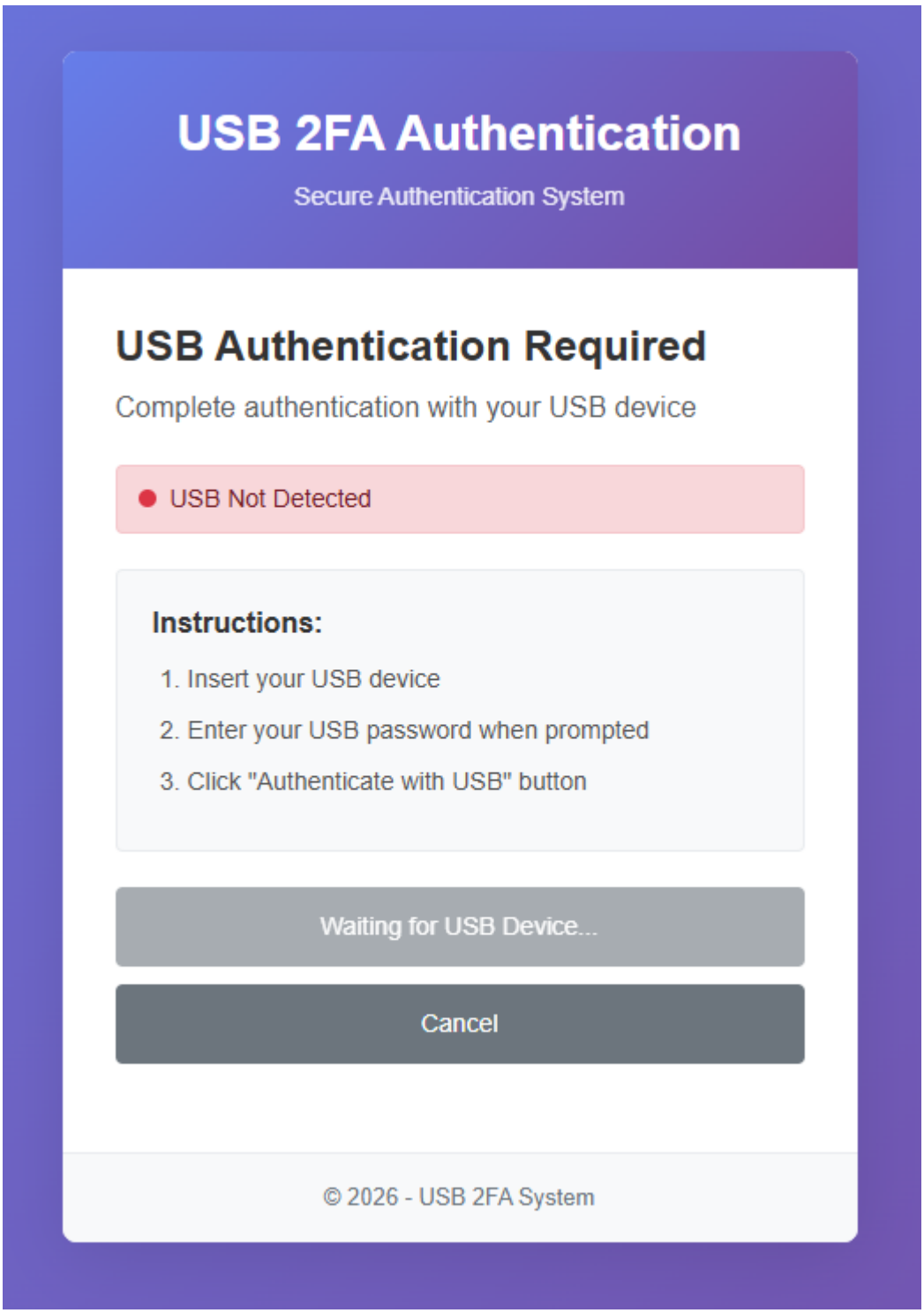
Serveris patikrina naudotojo pirmo faktoriaus prisijungimo duomenis ir grąžina dalinį žetoną, o naršyklėje atsidaro pirminio antro faktoriaus TOTP įvedimo langas. Naudotojas įveda bet kokį šešių skaitmenų kodą - grąžinama imituota klaida, kad pagrindinis 2FA nepasiekiamas (26).



26 pav. Pagrindinio (TOTP) neveikiančio 2FA langas

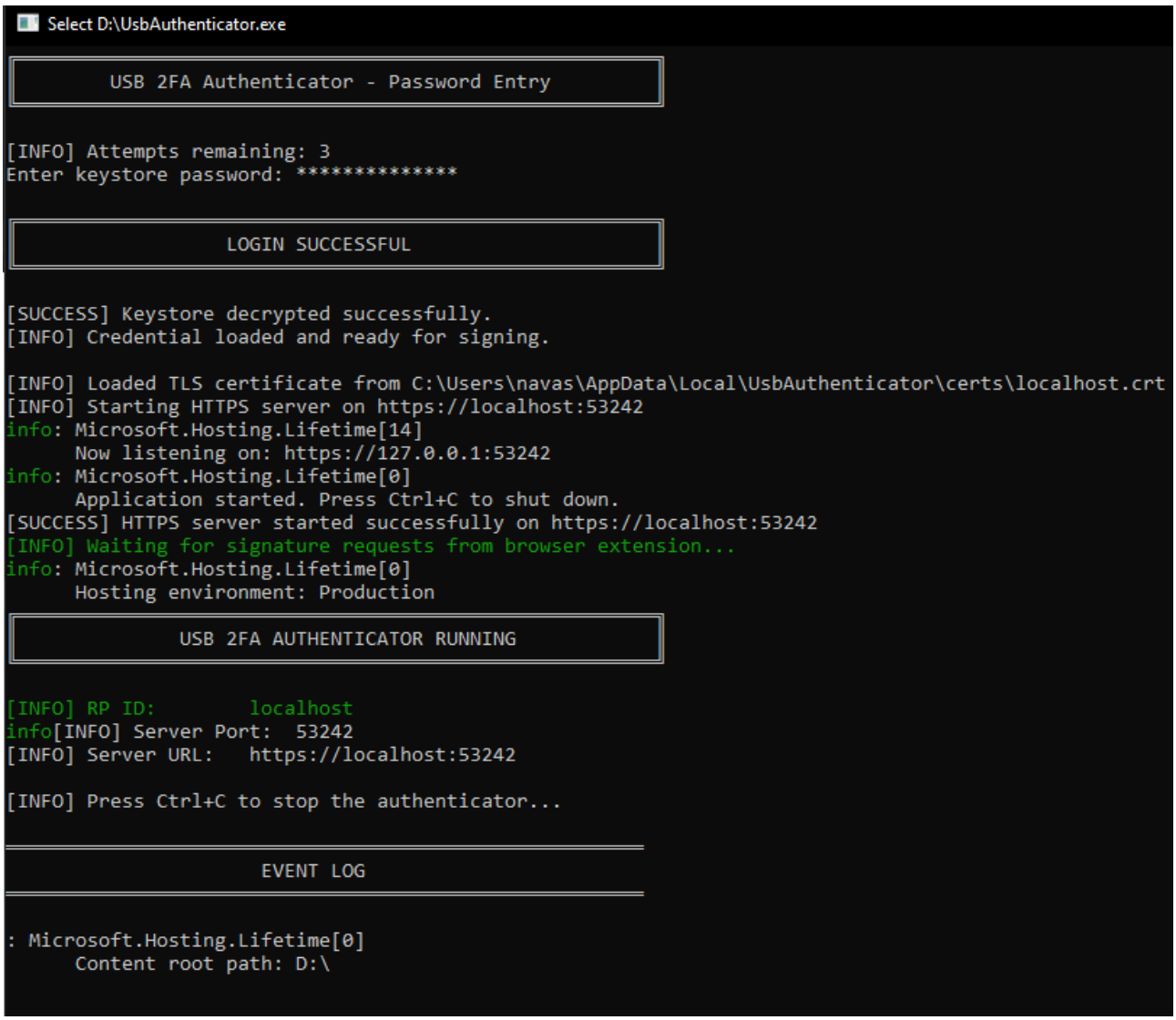
Po klaida atsiranda mygtukas, leidžiantis pereiti į atsarginį USB 2FA autentifikavimą. Naudotojas paspaudžia mygtuką, kuris atsidaro USB 2FA langą.

USB 2FA autentifikavimo lange, fone vykdomas USB laikmenos aptikimas. Jeigu USB 2FA lokalus servsias nerastas ar nepasiekiamas - po 2-3 sekundžių rodomas pranešimas (27), kad nepavyko rasti USB 2FA įrenginio („USB device detected“).



27 pav. Atsarginio USB 2FA langas

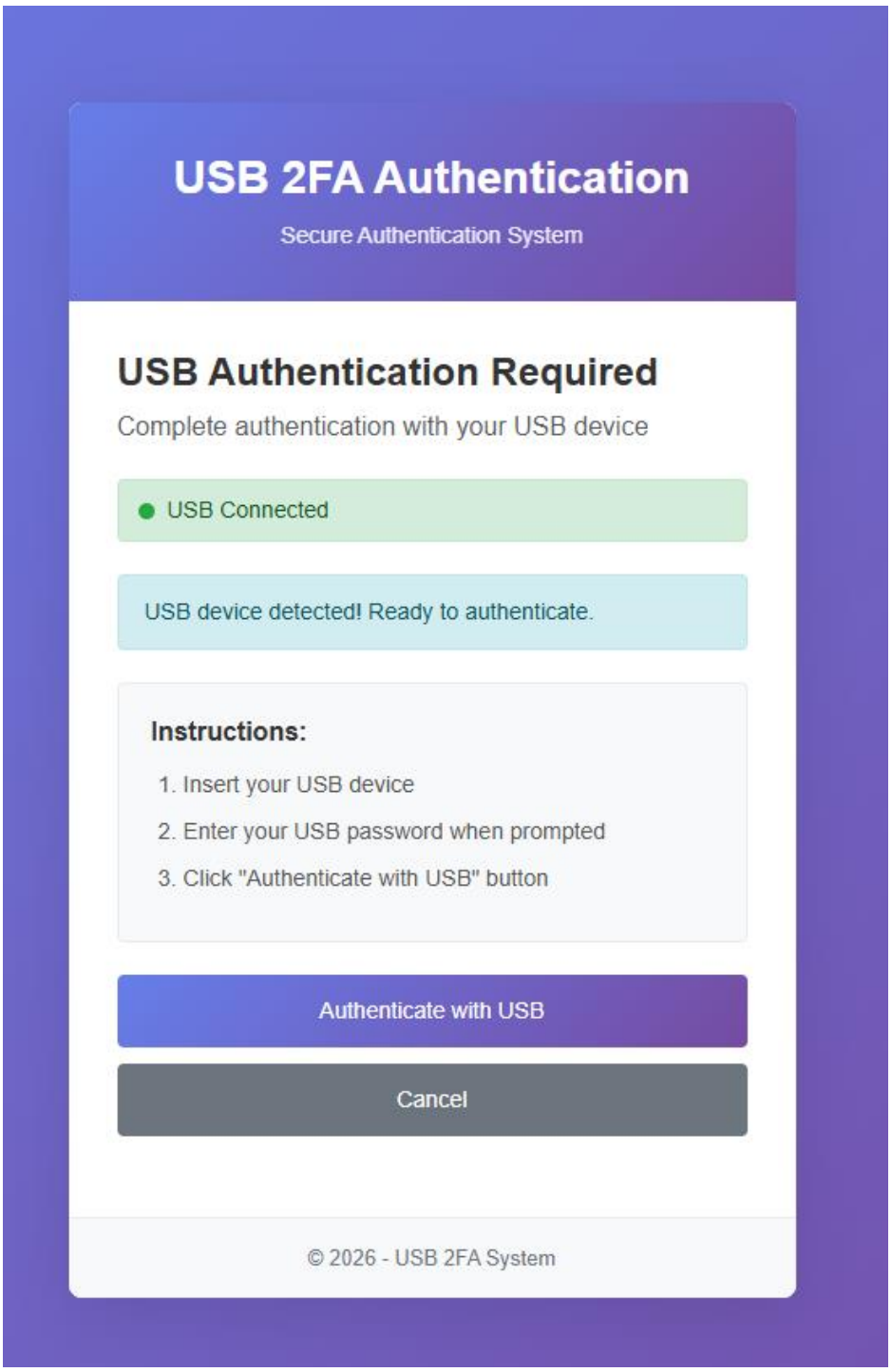
Naudotojas paleidžia autentifikatoriaus vykdomąjį failą tiesiai iš USB laikmenos (28). Programa parodo slaptažodžio įvedimo dialogą, įvedęs teisingą slaptažodį, programa iššifruoja raktų konteinerį.



28 pav. USB autentifikatoriaus programos langas

Programos atmintyje atsiranda privatus ECDSA P-256 raktas, per WMI nuskaitomas USB laikmenos PNPDeviceID identifikatorius, ir paleidžiamas vietinis HTTPS servisas prievade 53242. Fizinio USB identifikatoriaus patikra atliekama serverio pusėje.

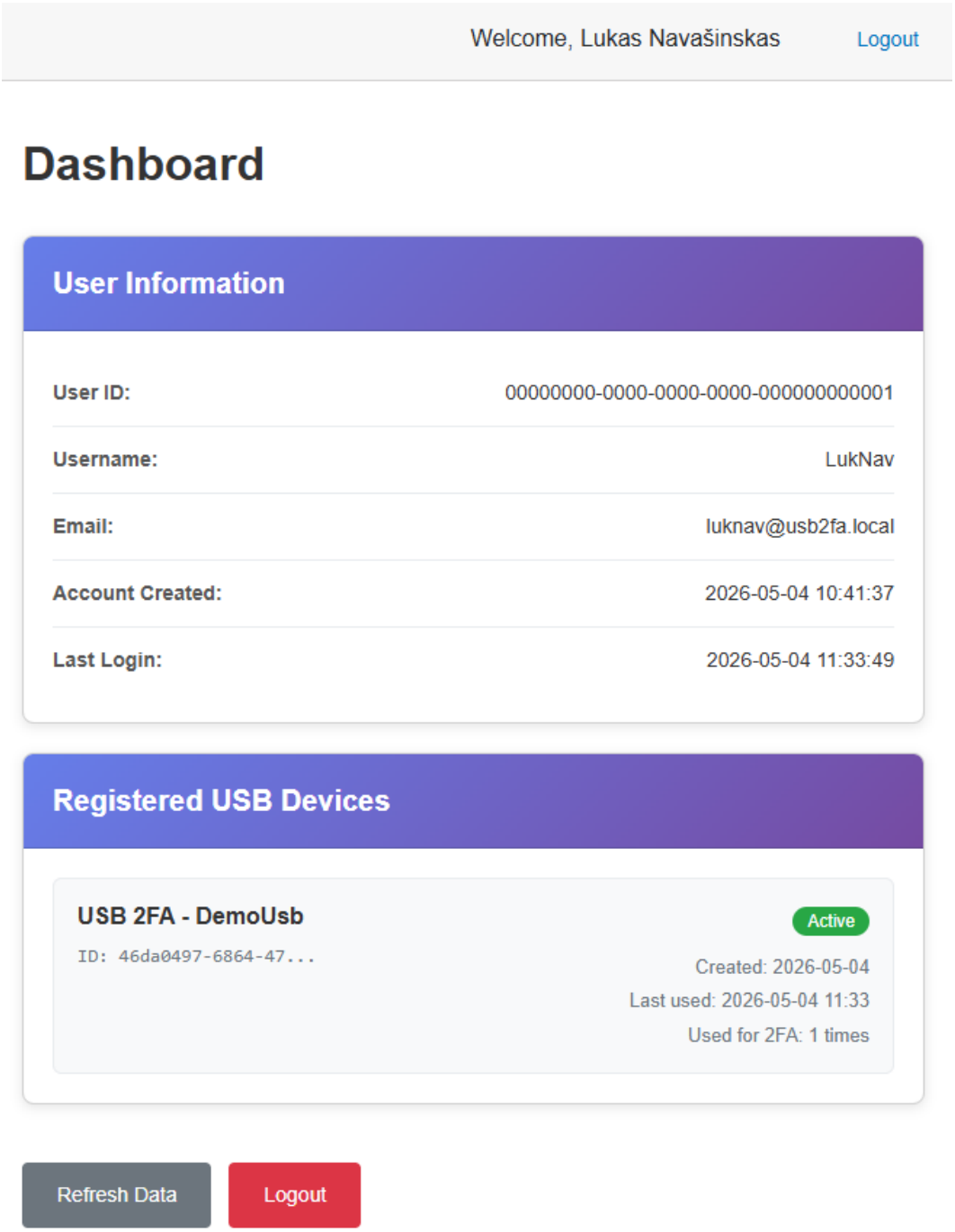
Organizacijos puslapis, paspaudus mygtuką „Authenticate with USB“ (29), gauna iššūkį iš nuotolinio serverio, perduoda jį USB autentifikatoriaus lokaliam servisui ir grąžina atgal pasirašytą atsakymą, kuriame yra parašas, autentifikatoriaus duomenys, kliento duomenys ir per WMI gyvai nuskaitytas USB laikmenos fizinis identifikatorius.



29 pav. USB autentifikacijos puslapis

Tinklapis išsiunčia USB 2FA prisijungimo duomenis į autentifikacijos serverį, kuris patikrina parašą pagal registracijos metu išsaugotą viešąjį raktą, palygina iššūkio identifikatorių su nepanaudotų iššūkių lentele, palygina gautą fizinio USB identifikatoriaus reikšmę su „usb_devices“ lentelėje saugoma diegimo metu užregistruota reikšme bei patikrina parašo skaitliuko vertę.

Bet kuriam iš šių patikrinimų nepavykus, serveris grąžina apibendrintą atsakymą „INVALID_SIGNATURE“ (31), dėl saugumo neatskleidžiantį tikslios autentifikacijos užklausos atmetimo priežasties. Sėkmingu autentifikacijos atveju serveris išduoda prieigos ir atnaujinimo žetonus, o naudotojas nukreipiamas į apsaugoto turinio organizacijos puslapį (30).



30 pav. Apsaugotas organizacijos puslapis

Visi devyni žingsniai (naudotojo sukūrimas, USB laikmenos paruošimas, pirmojo faktoriaus patikrinimas, TOTP gedimo imitacija, perėjimas į atsarginį srautą, USB autentifikatoriaus paleidimas, raktų konteinerio iššifravimas, iššūkio-atsakymo apsikeitimas ir prieigos žetono išdavimas) buvo sėkmingai atlikti.

Funkcinis patikrinimas laikomas tenkinančiu pagrindinį projekto reikalavimą - leisti naudotojui užbaigti prisijungimą, kai pirminis 2FA metodas tampa nepasiekiamas.

4.3. Patikimumo scenarijai

Patikimumo scenarijuose bandomos konkrečios atakos arba klaidingi bandymai ir tikrinama, ar prototipo sistema juos atmeta. Kiekvienas scenarijus turi iš anksto suformuluotą prielaidą, eigą bei

numatomą rezultatą. Scenarijai parinkti taip, kad padengtų abu apsaugos sluoksnius - kliento pusės patikrą USB autentifikatoriuje ir serverio pusės patikrą tikrinant užklausas.

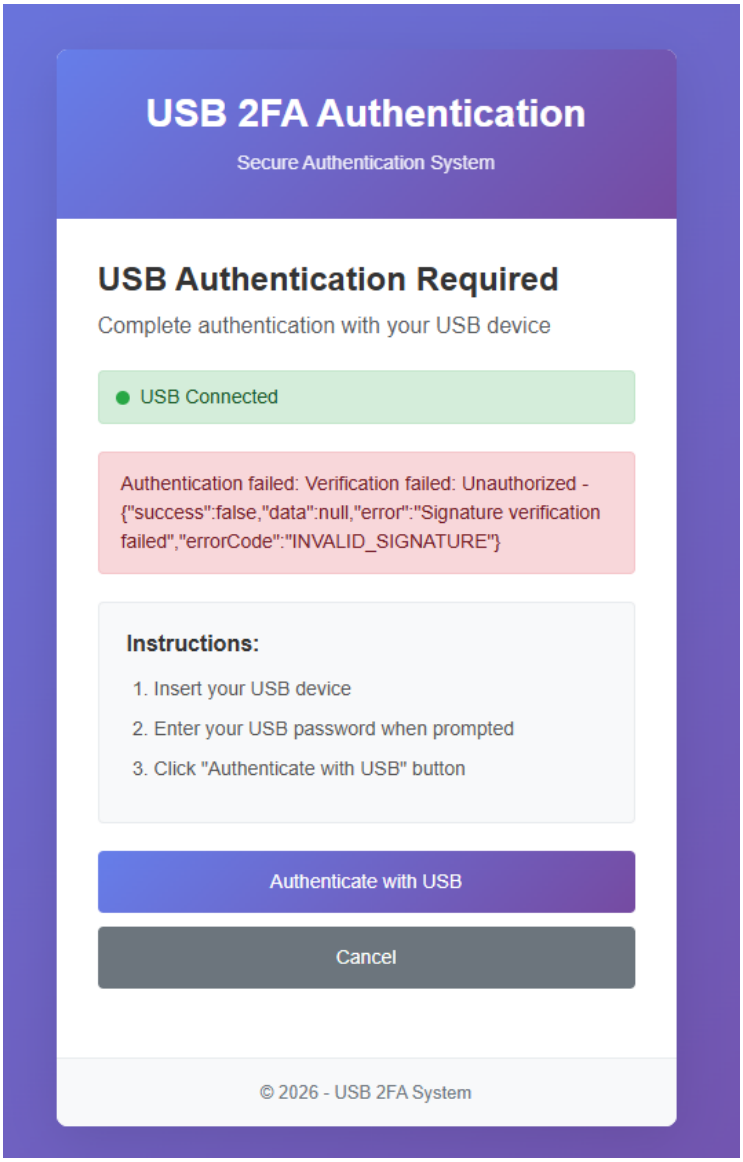
4.3.1. USB kopijos scenarijus

Pirmasis scenarijus apima raktų konteinerio kopijavimą į kitą USB laikmeną.

Šiame scenarijuje piktavalis turi fizinę prieigą prie užregistruotos USB laikmenos, žino naudotojo bei USB autentifikatoriaus slaptažodį ir turi savo paruoštą FAT32 laikmeną. Visi failai (vykdomasis failas, „keystore.enc“, konfigūracijos failas) nukopijuojami iš originalios laikmenos į piktavalio laikmeną, originali USB laikmena atjungiama, o nauja prijungiama ir paleidžiama.

Piktavalis įveda teisingą slaptažodį, sėkmingai iššifruoja raktų konteinerį ir bando užbaigti autentifikavimą: autentifikatorius per WMI nuskaito naujosios laikmenos PNPDeviceID ir kartu su tinkamu parašu siunčia jį serveriui.

Tikimasi, kad serveris atmes užklausą, palyginęs gautą reikšmę su „usb_devices“ lentelėje saugoma originalo reikšme, ir grąžins apibendrintą HTTP 401 atsakymą be tikslios atmetimo priežasties.



31 pav. USB autentifikacijos klaida

Stebėtas rezultatas (31) atitiko patikimumo testo prielaidą: audito žurnale fiksuojamas „verify_signature_failed“ įrašas su priežastimi „Physical USB identifier mismatch“, o naršyklėje rodomas tas pats apibendrintas „INVALID_SIGNATURE“ pranešimas kaip ir bet kurios kitos nepavykusio autentifikavimo užklausos patikrinimo atveju. Dėl bendrinės klaidos žinutės piktavalis negali atskirti, kas tiksliai nepavyko, kodėl atmetė 2FA autentifikaciją.

4.3.2. Iššūkio pakartojimo atakos scenarijus

Antrasis scenarijus apima iššūkio pakartojimo ataką. Piktavalis tinklo srauto stebėjimo priemonėmis perima sėkmingą „/api/auth/usb/verify“ užklausą ir bando ją pakartoti.

Tikimasi, kad pirmoji užklausa grąžins prieigos žetonus, o pakartotinė bus atmesta, nes iššūkio įrašas duomenų bazės „challenges“ lentelėje pažymėtas kaip panaudotas („is_used = true“).

Stebėtas rezultatas atitiko pakartojimo atakos testo prielaidą: serveris grąžino HTTP 401 „Signature verification failed“, o audito žurnale fiksuotas įrašas su priežastimi „challenge already used“.

4.3.3. Pasibaigęs iššūkio galiojimo scenarijus

Trečiasis scenarijus tikrina pasibaigusio iššūkio atmetimą. Iššūkis gaunamas per „/api/auth/usb/challenge“ galutinį tašką, tačiau prieš pasirašymą laukiama ilgiau nei 120 sekundžių. Po laukimo pasirašytas atsakymas siunčiamas į serverį.

Numatomas rezultatas - užklausa atmetama, nes lentelės įrašo „expires_at“ reikšmė yra mažesnė nei einamasis laikas.

Stebėtas rezultatas atitiko iššūkio galiojimo testo prielaidą: serveris grąžino HTTP 401 „Signature verification failed“, o audito žurnale fiksuotas įrašas su priežastimi „Challenge expired“.

4.3.4. Neteisingo USB slaptažodžio scenarijus

Ketvirtasis scenarijus tikrina neteisingą USB slaptažodį. Piktavalis turi fizinę užregistruotą laikmeną, įrenginio identifikatoriaus patikra praeina, tačiau slaptažodis įvedamas neteisingai.

Tikimasi, kad po trijų neteisingų bandymų programa nutrauks darbą ir nepaleis vietinio HTTPS serviso.

Stebėtas rezultatas atitiko USB slaptažodžio testo prielaidą. Verta paminėti, kad USB laikmenoje nesaugoma bandymų skaitiklio būklė, todėl programą galima paleisti iš naujo - teorinis brutalios jėgos bandymas išlieka įmanomas. Tačiau PBKDF2-HMAC-SHA256 su 600 000 iteracijų vienam bandymui suteikia maždaug 300 milisekundžių uždelsimą, o tai užtikrina, kad brutalios jėgos atakos būtų nerealistiškos.

4.3.5. Patikimumo scenarijų išvados

Patikimumo scenarijai parodo kad sistema atmeta visus numatytus piktavališkus arba klaidingus bandymus neteisėtai prisijungti į organizacijos sistemą per atsarginį 2FA.

8 lentelė. Patikimumo scenarijų rezultatų santrauka

Scenarijus	Apsaugos sluoksnis	Rezultatas
USB įrenginio kopija	Serveris	Atmesta
Pakartotinis iššūkio panaudojimas	Serveris	Atmesta
Pasibaigęs iššūkis	Serveris	Atmesta
Neteisingas USB įrenginio slaptažodis	Klientas	Atmesta su pastaba
Modifikuotas autentifikatoriaus duomenų blokas	Serveris	Atmesta

Visi lentelėje pateikti scenarijai baigėsi taip, kaip buvo ir tikėtasi: kiekvienu atveju autentifikacijos užklausa atmesta. Naudotojas visada gauna tą patį apibendrintą atsakymą („Signature verification failed“, HTTP 401) be konkrečios atmetimo priežasties, todėl iš serverio pusės piktavalis negali atskirti, kuri autentifikacijos užklausų dalis buvo atmesta - tikslus paaiškinimas lieka tik audito žurnale. Atlikti patikimumo testų scenarijai patvirtina, kad sistema atlaiko numatytus piktavališkus ir klaidingus bandymus.

4.4. Pritaikomumo patikrinimas

Pritaikomumo patikrinimas atliekamas supaprastintu naudojimo scenarijumi, kuris atliekamas naudojantis 4.1 skirsnio lentelėje aprašyta aplinką. Demonstracija nebuvo atliekama su išorės dalyviais, todėl tai nėra formalus naudotojų patirties tyrimas.

Pritaikomumo testo scenarijus: naudotojui sugedo telefonas, kuriame buvo TOTP programėlė ir jam reikia greitai prisijungti prie organizacijos paskyros pasinaudojant iš anksto paruoštu atsarginu antru faktoriumi - USB laikmena. Naudotojas atveria prisijungimo puslapį, įveda prisijungimo vardą ir slaptažodį (25). Po pirmojo faktoriaus rodomas TOTP įvedimo puslapis, bandymas su bet kokiu kodu rodo klaidos pranešimą ir mygtuką, leidžiantį pereiti į atsarginį USB autentifikavimą (26). USB 2FA puslapis, fone aptikęs prijungtą laikmeną, puslapis rodo pranešimą apie sėkmingą USB autentifikatoriaus įrankio aptikimą (29). Paleidžiamas autentifikatoriaus vykdomasis failas, atsidaro slaptažodžio įvedimo dialogas (28). Įvedęs slaptažodį naudotojas grįžta į naršyklę, paspaudžia parašo prašymo mygtuką ir patenka į apsaugoto turinio puslapį (30).

Visą eigą nuo TOTP klaidos iki apsaugoto turinio puslapio pavyko atlikti per maždaug 45 sekundes. Didžiausia laiko dalis - USB slaptažodžio įvedimas ranka.

4.5. Eksperimento išvados

Šiame skyriuje atliktas trijų kriterijų eksperimentinis prototipo vertinimas patvirtino pagrindinį projekto reikalavimą - leisti naudotojui užbaigti prisijungimą, kai pirminis 2FA metodas tampa nepasiekiamas.

Funkcinis testas parodė, kad sistema teisingai įgyvendina visą atsarginio prisijungimo srautą nuo naudotojo sukūrimo serveryje iki prieigos žetono išdavimo po sėkmingos USB 2FA autentifikacijos.

Patikimumo scenarijai parodė, kad sistema atmeta visus numatytus piktavališkus arba klaidingus bandymus, o nepavykusio patikrinimo atveju serveris grąžina bendrinį HTTP 401 atsakymą be tikslios atmetimo priežasties.

Pritaikomumo demonstracija parodė, kad atsarginio prisijungimo eigą galima atlikti per maždaug 45 sekundes, be administratoriaus teisių ir be papildomos programinės įrangos diegimo.

5. Išvados ir rekomendacijos

Šiame darbe buvo išanalizuotos dviejų faktorių autentifikacijos (2FA) sistemos, jų pažeidžiamumai ir atsarginių metodų trūkumai. Remiantis atlikta analize, suprojektuotas ir realizuotas patikimas atsarginis 2FA sprendimas, naudojantis USB laikmeną kaip fizinį antrąjį faktorių.

Atlikta esamų dviejų faktorių autentifikavimo metodų analizė parodė, kad TOTP ir SMS sprendimai gali būti nepasiekiami praradus telefoną, o aparatūriniai FIDO2 raktai reikalauja papildomos įrangos pirkimo bei dažnai - administratoriaus teisių diegiant programas, kurių reikia palaikyti aparatūrinius raktus. Todėl USB laikmenos pagrindu veikiantis atsarginis 2FA sprendimas yra pagrįstas pasirinkimas atsarginės prieigos uždaviniui spręsti:

1. Suprojektuotas ir įgyvendintas keturių komponentų sprendimas (autentifikavimo serveris, žiniatinklio programa, USB diegimo įrankis ir nešiojama USB autentifikatoriaus programa), kuriame raktas niekada nepalieka USB laikmenos, o serveryje saugomas tik viešasis raktas. Toks suskaidymas leidžia naudotojui autentifikuotis bet kuriame kompiuteryje su Windows 10 ir naujesne operacine sistema, neturint administratoriaus teisių ir nediegiant papildomos programinės įrangos.

2. Sprendime panaudotos tik atviro kodo technologijos ir standartai leidžia sprendimą pritaikyti tiek komerciniams, tiek nekomerciniams poreikiams be licencinių apribojimų.

3. Eksperimentinio tikrinimo metu visi atakų bandymai buvo atmesti, o teisėtas autentifikavimas užtruko apie 45 sekundes. Tai parodo, kad sukurtas sprendimas tinkamas praktiniam naudojimui.

4. Sukurta sistema atitinka Zero-Trust architektūros principus: kiekvienas autentifikavimo veiksmas yra atskirai patikrinamas, privatusis raktas neperduodamas tinklu, o iššūkis galioja tik 120 sekundžių. Dėl to užpuolikas, gavęs tinklo srauto kopiją, jos panaudoti pakartotiniam autentifikavimui negali.

Rekomendacijos:

1. Tolimesniems tyrimams siūloma ištirti biometrinių duomenų (pirštų atspaudo) integravimo galimybes į USB autentifikatorių, kas užtikrintų papildomą apsaugos sluoksnį praradus USB įrenginį;

2. Praktiniam diegimui rekomenduojama atlikti išsamesnius apkrovos ir saugumo testavimus su didesniu naudotojų skaičiumi bei įvertinti sprendimo suderinamumą su įvairiomis operacinėmis sistemomis;

3. Šiuo metu aplikacija palaikoma Windows operacinėje sistemoje. Vystant projektą toliau rekomenduojama išplėsti palaikymą Linux ir macOS operacinėms sistemoms sukuriant papildomas integracijas USB įrenginio metaduomenų skaitymui ir valdymui.

Sukurtas prototipas įrodo, kad paprasta USB laikmena gali būti paversta patikimu antruoju autentifikacijos faktoriumi, nepriklausomu nuo išmaniojo telefono ir trečiųjų šalių paslaugų, kartu užtikrinant aukštą saugumo lygį ir atitikimą šiuolaikiniams kibernetinio saugumo standartams.

3. Literatūros sąrašas

1. LAMBA, M. ir kt. Is Two Factor Authentication Enough to Secure your System: A Study. In *Journal of Management* . 2024. Vol. 13, no. 01. .

2. ALI, G. ir kt. Two-Factor Authentication Scheme for Mobile Money: A Review of Threat Models and Countermeasures. In *Future Internet* . 2020. Vol. 12, no. 10, p. 160. .

3. BRUZGIENE, R. - JURGILAS, K. Securing Remote Access to Information Systems of Critical Infrastructure Using Two-Factor Authentication. In *Electronics* . 2021. Vol. 10, no. 15, p. 18. .

4. GILSENAN, C. - EGELMAN, S. Security and Privacy Failures in Popular 2FA Apps . 2023.

5. KUZIOR, A. ir kt. Cybersecurity and cybercrime: Current trends and threats. 2024.

6. REESE, K. ir kt. A Usability Study of Five Two-Factor Authentication Methods. 2019.

7. MEYER, L.A. ir kt. [interaktyvus]. .[s.l.]: arXiv, 2023. Prieiga per internetą: <https://arxiv.org/abs/2305.00945>.

8. KUZIOR, A. ir kt. Countering Cybercrime Risks in Financial Institutions: Forecasting Information Trends. In *Journal of Risk and Financial Management* . 2022. Vol. 15, no. 12, p. 613. .

9. BERRIOS, J. ir kt. Factorizing 2FA: Forensic analysis of two-factor authentication applications. In *Forensic Science International: Digital Investigation* . 2023. Vol. 45, p. 301569. .

10. SZCZYGIEŁ, I. ir kt. Two-factor authentication (2FA) comparison of methods and applications. In *Advances in Web Development Journal* [interaktyvus]. 2023. Vol. 1, no. 1. Prieiga per internetą: <https://journals.edu.pl/index.php/awdj/article/view/9>.

11. JUBUR, M. ir kt. An In-Depth Analysis of Password Managers and Two-Factor Authentication Tools. In *ACM Comput. Surv.* [interaktyvus]. 2025. Prieiga per internetą: <https://doi.org/10.1145/3711117>.

12. PETROV, P. ir kt. USING THE UNIVERSAL TWO FACTOR AUTHENTICATION METHOD IN WEB APPLICATIONS BY SOFTWARE EMULATED DEVICE. In *20th International Multidisciplinary Scientific GeoConference Proceedings SGEM 2020* [interaktyvus]. 2020. p. 403–410. Prieiga per internetą: <https://epslibrary.at/sgem_jresearch_publication_view.php?page=view&editid1=7013>.

13. WEE, A.K. ir kt. Excavating Vulnerabilities Lurking in Multi-Factor Authentication Protocols. In [interaktyvus]. 2024. Prieiga per internetą: <http://arxiv.org/abs/2407.20459>.

14. ILGIZAROVNA, Z.R. TWO-FACTOR BIOMETRIC AUTHENTICATION SYSTEM. In . 2024. Vol. 46, no. 5, p. 150–162. .

15. WANG, D. ir kt. Understanding security failures of multi-factor authentication schemes for multi-server environments. In *Computers & Security* . 2020. Vol. 88, p. 101619. .

16. KALASH ir kt. Enhancing Security in Smart Contract Wallets : An OTP Based 2-Factor Authentication Approach. In *Proceedings of the 26th International Conference on Distributed Computing and Networking* [interaktyvus]. New York, NY, USA: Association for Computing Machinery, 2025. p. 211–220. Prieiga per internetą: <https://doi.org/10.1145/3700838.3700868>.

17. ALOTAIBI, H. ir kt. Usability Testing of Memorable Word in Security Enhancing in e-Government and e-Financial Systems. In *International Journal of Advanced Computer Science and Applications* [interaktyvus]. 2023. Vol. 14, no. 9. [žiūrėta 2025-01-19]. . Prieiga per internetą: <http://thesai.org/Publications/ViewPaper?Volume=14&Issue=9&Code=IJACSA&SerialNo=28>.

18. HESS, E.M. ir kt. Vulnerabilities of Multi-factor Authentication in Modern Computer Networks. 2021.

19. PATAT, G. - SABT, M. Please Remember Me: Security Analysis of U2F Remember Me Implementations in The Wild. In *Actes SSTIC 2020, 18ème Symposium sur la sécurité des technologies de l'information et des communications (SSTIC 2020)* [interaktyvus]. Rennes, France, 2020. Prieiga per internetą: <https://inria.hal.science/hal-02865105>.

20. JIANG, M. ir kt. Biometric-based two-factor authentication scheme under database leakage. In *Theoretical Computer Science* . 2024. Vol. 1000, p. 114552. .

21. GERKEN, J.F. - WANG, Z. Phishing Susceptibility and Mitigation in the 2FA Context : An Investigation of How the Interplay of Psychological and Individual Factors and UX Design Can Influence Users' Decisions to Login to a Suspicious Website. In . 2024. .

22. DWORKIN, M.J. [interaktyvus]. .Gaithersburg, MD: National Institute of Standards and Technology, 2007. [žiūrėta 2026-02-25]. Prieiga per internetą: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf>.

23. DWORKIN, M.J. [interaktyvus]. .Gaithersburg, MD: National Institute of Standards and Technology, 2001. [žiūrėta 2026-02-25]. Prieiga per internetą: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>.

24. WETZELS, J. [interaktyvus]. .[s.l.]: arXiv, 2016. [žiūrėta 2026-02-22]. Prieiga per internetą: <https://arxiv.org/abs/1602.03097>.

25. TURAN, M.S. ir kt. [interaktyvus]. .Gaithersburg, MD: National Institute of Standards and Technology, 2010. [žiūrėta 2026-02-22]. Prieiga per internetą: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf>.

26. TEMOSHOK, D. ir kt. [interaktyvus]. .Gaithersburg, MD: National Institute of Standards and Technology (U.S.), 2025. [žiūrėta 2026-03-17]. Prieiga per internetą: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63B-4.pdf>.

27. BINDEL, N. ir kt. FIDO2, CTAP 2.1, and WebAuthn 2: Provable Security and Post-Quantum Instantiation. In *2023 IEEE Symposium on Security and Privacy (SP)* [interaktyvus]. San Francisco, CA, USA: IEEE, 2023. p. 1471–1490. [žiūrėta 2026-04-19]. Prieiga per internetą: <https://ieeexplore.ieee.org/document/10179454/>.