KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Arvydas Nauckūnas

MOBILIOJI VPN IDENTIFIKACIJA PAREMTA PKCS11 KRIPTOPROCESORIUMI IR ARTIMOJO LAUKO KOMUNIKACIJA

Baigiamasis magistro darbas

Vadovas
Prof. dr. Eligijus Sakalauskas

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

MOBILIOJI VPN IDENTIFIKACIJA PAREMTA PKCS11 KRIPTOPROCESORIUMI IR ARTIMOJO LAUKO KOMUNIKACIJA

Baigiamasis magistro darbas
Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Vadovas
(parašas) Prof. dr. Eligijus Sakalauskas
(data)

Recenzentas
(parašas) Lekt. dr. Aleksejus Michalkovič
(data)

Projektą atliko
(parašas) Arvydas Nauckūnas
(data)

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Arvydas Nauckūnas

(Studento vardas, pavardė)

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

(Studijų programos pavadinimas, kodas)

„Mobilioji VPN identifikacija paremta PKCS11 kriptoprocesoriumi ir artimojo lauko komunikacija“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 16 m. gegužės 20 d.

Kaunas

Patvirtinu, kad mano **Arvydo Nauckūno** baigiamasis projektas tema „Mobilioji VPN identifikacija paremta PKCS11 kriptoprocesoriumi ir artimojo lauko komunikacija“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

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

Nauckūnas, A. „Mobilioji VPN identifikacija paremta PKCS11 kriptoprocesoriumi ir artimojo lauko komunikacija“. Magistro baigiamasis projektas / vadovas prof. dr. Eligijus Sakalauskas; Kauno technologijos universitetas, Informatikos fakultetas, Kompiuterių katedra.

Kaunas, 2016. 58 p.

SANTRAUKA

Magistro darbo tikslas – išanalizuoti ir parinkti tinkamiausius komponentus mobiliosios VPN identifikacijos sistemos prototipo sukūrimui. Pagal analizės metu gautas išvadas parinkti labiausiai tinkamus komponentus kriptografiškai stiprios, mobiliosios VPN identifikacijos sistemos projektavimui ir prototipo sukūrimui. Pasinaudojant sukurtu prototipu atlikti sistemos charakteristikų ir bendrą sistemos saugos tyrimą. Atlikti siūlomos sistemos kokybinį palyginimą su esamomis sistemomis.

Išanalizavus jau naudojamus vietinių kompiuterių VPN identifikacijos būdus ir sistemas, buvo pastebėta, kad nei vienas iš esamų sprendimų nesuteikia pakankamai mobilumo ir negarantuoja aukščiausio lygio vartotojo identifikacijos raktų saugumo. Analizės metu sistemos prototipo projektavimui ir realizavimui buvo pasirinkti labiausiai paplitę ir aukščiausią saugumą galintys užtikrinti komponentai. Projektuojant mobiliosios VPN identifikacijos sistemą privačiųjų raktų saugojimui buvo pasirinktas fizinis įrenginys suteikiantis FIPS140-2 3 lygio apsaugą ir galintis integruotis su mobiliuoju telefonu, palaikančiu microSD kortelę.

Nauckūnas, Arvydas. *Mobile Identification To VPN System Using PKCS11 Crypto Processor And NFC*: *Master's* thesis in field of informatics / supervisor assoc. prof. Eligijus Sakalauskas. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Informatics

Key words: VPN, NFC, OpenVPN, Cryptoki, PKI, Crypto processor.

Kaunas, 2016. 58 p.

SUMMARY

The purpose of this work is to make a research in mobile VPN identification system, analyze and select best suited components for its prototype realization. Based on analysis results select most suited components that enable design and realization of mobile VPN identification system capable of using strong cryptographic methods. Make of use of the prototype to gather system characteristics and study its overall security. Compare proposed system to already existing solutions.

Already existing local computer VPN identification methods and systems were analyzed. It was noticed that none of the existing solutions provide mobility benefit along with highest level of private key security. During analysis stage components that are most widely spread and offer best security were selected for realization of proposed system prototype. When designing mobile VPN identification

system a physical security measure was selected for private key storage which provides FIS140-2 level 3 security and is able to integrate with mobile phones that support microSD slot.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

TERMINŲ IR SANTRUMPŲ ŽODYNAS

1. VPN (angl. *Virtual private network*) – virtualus privatus tinklas.
2. ALK – artimojo lauko komunikacija.
3. NFC (angl. *Near field communication*) – artimo lauko komunikacija (ALK).
4. RSA (angl. *Rivest-Shamir-Adleman cryptosystem*) – viešojo rakto kriptosistema.
5. PKCS#11 (angl. *Public-key cryptography standards*) – kriptografinių raktų programavimo prievadas.
6. OpenVPN (angl. *open-source virtual private network*) – atviro kodo programa skirta sukurti virtualius privačius tinklus.
7. Cryptoki – terminas apibūdinantis PKCS#11 standartą.
8. X.509 – standartas apibrėžiantis viešojo rakto sertifikatus ir jų valdymą.
9. 3DES (DES/Triple) (angl. *Triple Data Encryption Algorithm*) – simetrinio rakto blokinis šifras, kuris pritaiko DES šifro algoritmą tris kartus kiekvienam duomenų blokui.
10. OpenSSL – atviro kodo biblioteka realizuojanti SSL ir TLS protokolus.
11. ISO (angl. *International Organization for Standardization*) – tarptautinė standartizacijos organizacija, kurianti tarptautinius standartus ir glaudžiai bendradarbiaujanti su IEC.
12. IEC (angl. *International Electrotechnical Commission*) – tarptautinė standartų organizacija, kuri parengia ir paskelbia tarptautinius standartus visoms elektrinėms, elektroninėms ir susijusioms technologijoms.
13. Peer-to-peer (P2P) – komunikacijos tipas, vykstantis tik tarp dviejų vienodo lygio įrenginių.
14. RFID (angl. *Radio-frequency identification*) – bevielio ryšio technologija, naudojama nuskaityti palaikomus pasyvius objektus.
15. GSMA (angl. *GSM Association*) – organizacija, kurianti ir palaikanti GSM mobiliųjų telefonų standartus.
16. IrDa (angl. *Infrared Data Association*) – komunikacijos technologija tarp įrenginių, naudojanti infraraudonuosius spindulius.
17. Bluetooth – bevielės technologijos standartas, skirtas apsikeisti duomenimis nedideliu atstumu naudojant radio dažnius nuo 2.4 iki 2.485 GHz.
18. ECMA (angl. *European Computer Manufacturers Association*) – tarptautinė informacijos ir komunikacijos sistemų standartizavimo organizacija.
19. ETSI (angl. *European Telecommunications Standards Institute*) – nepriklausoma Europos telekomunikacijų pramonės standartizavimo organizacija.
20. WiFi – bevielio vietinio tinklo technologija.
21. SSL (angl. *Secure Sockets Layer*) – kriptografinis protokolas, sukurtas saugiai komunikacijai per internetą.

22. TLS (angl. *Transport Layer Security*) – kriptografinis protokolas, sukurtas saugiai komunikacijai per internetą.
23. GNU GPL (angl. *GNU General Public License*) – plačiausiai naudojama nemokamos programinės įrangos licencija, suteikianti galutiniam vartotojui teisę naudoti, analizuoti, dalintis ir keisti programinę įrangą.
24. NAT (angl. *Network address translation*) – metodas kai interneto protokolo paketų antraščių tinklo adresai yra pakeičiami siekiant peradresuoti paketus į kitą internetinių adresų erdvę.
25. OSI (angl. *Open Systems Interconnection model*) – modelis, apibrėžiantis komunikacijos sistemos funkcijas abstrakcijos lygiais.
26. LZO (angl. *Lempel–Ziv–Oberhumer*) – duomenų suspaudimo algoritmas.
27. IANA (angl. *Internet Assigned Numbers Authority*) – Amerikos korporacija, kuri prižiūri internetinio protokolo viešosios adresų erdvės paskyrimų ir kitų susijusių paslaugų valdymą.
28. ISP (angl. *Internet service provider*) – organizacija, suteikianti prieigą prie interneto paslaugų.
29. RFC (angl. *Request for Comments*) – publikacija, apibūdinanti metodus, elgseną, tyrimus arba inovacijas tinkamas darbui su internetu arba prie interneto prijungtomis sistemomis.
30. IETF (angl. *Internet Engineering Task Force*) – atvirų standartų organizacija, kuri kuria ir skatina interneto standartų naudojimą.
31. L2TP (angl. *Layer 2 Tunneling Protocol*) – tuneliavimo protokolas, skirtas palaikyti virtualius privačius tinklus. Naudojant vien šį protokolą, duomenys nėra šifruojami.

ĮVADAS

Darbas priklauso Informacijos ir informacinių technologijų saugos studijų programai.

Šiame darbe yra projektuojama ir analizuojama paprastai, ir greitai įvykdoma, bet tuo pačiu saugi VPN identifikacija pasinaudojant prie kompiuterio neprijungta laikmena saugoti slaptiems autentifikavimo raktams.

Mobilusis telefonas tapo kiekvieno žmogaus gyvenimo dalimi. Šiai laikais mobilieji telefonai gali pakeisti daugelį įrenginių ir pasiūlyti dar daugiau galimybių vartotojui. Jie gali pakeisti fotokamerą, skaičiuotuvą, vaizdo kamerą ir panašius įrenginius.

Šiame informacijos amžiuje, kai yra daug įvairių sistemų, vartotojams, norint dirbti su sistemomis, tenka atsiminti vis daugiau vartotojo prisijungimo vardų ir slaptažodžių. Dėl didėjančio informacijos kiekio ir jos slaptumo, į saugumą kreipiama vis daugiau dėmesio. Slaptažodžiai tampa vis ilgesni ir sudėtingesni, atsiranda reikalavimas juos keisti po tam tikro laiko [1].

Autentifikavimui vis plačiau yra naudojamos viešojo ir privačiojo rakto kriptosistemos. Tai padidina saugumą, nes dažnai naudojamas raktų ilgis yra pakankamai didelis, kad būtų neįmanoma jo atspėti ir iššifruoti duomenų. Tiesa, iškyla problema, kur saugiai laikyti šiuos slaptus raktus, kad net ir pametus raktų laikmeną, autentifikavimo raktai išliktų apsaugoti. Ir kaip įgyvendinti, kad šie raktai būtų paprastai, bet saugiai pasiekiami.

Darbo problematika ir aktualumas

Kompiuteriai su privačiais saugiais tinklais, komunikuojantys per nesaugų kanalą yra dažnai prijungiami pasinaudojant virtualiuoju privačiuoju kanalu. Taip yra užtikrinama, kad duomenys nebus perskaityti, kai keliauja nesaugiu kanalu. Šiam kanalui sukurti yra naudojami slaptažodžiai arba didesnį saugumą suteikianti viešojo privačiojo rakto kriptosistema. Taigi iškyla problema, kaip pasinaudojant tik telefonu kaip privačiojo rakto laikmena, paprastai ir greitai sukurti saugų kanalą iš kompiuterio į privačius tinklus [2].

Darbo tikslas ir uždaviniai

Tikslas: kriptografiškai stiprios, mobiliosios VPN identifikacijos sistemos prototipo sukūrimas. Atlikti kokybinį palyginimą su esamomis sistemomis.

Uždaviniai:

- Atlikti projektuojamos sistemos saugumo analizę;
- Atlikti sistemos prototipo charakteristikų tyrimą;
- Atlikti RSA kriptosistemos funkcijų analizę;
- Atlikti PKCS#11 standarto panaudojimo analizę.

Darbo rezultatai ir jų svarba

Darbo rezultatai parodys analizuojamų komponentų panaudojimo galimybę mobiliajai VPN identifikacijos sistemos prototipui sukurti. Iš analizuojamų komponentų bus parinkti labiausiai tinkantys komponentai, atsižvelgiant į suteikiamą saugumą, vartojimo paprastumą ir komponentų tarpusavio suderinamumą.

Darbo struktūra

Dokumente apžvelgiamos labiausiai paplitusios ir šiuo metu sparčiai plintančios technologijos, kurios galėtų padėti realizuoti mobiliąją VPN identifikacijos sistemą.

Pirmiausia yra apžvelgiamos technologijos ir jų funkcijos, kurios atitinka įsivaizduojamą mobiliosios VPN identifikacijos schemą. Palyginus panašias technologijas yra pasirenkamos geriausios pagal suteikiamą saugumą, vartojimo paprastumą ir komponentų tarpusavio suderinamumą.

Atsirinkus technologijas realizuoti mobiliajai VPN identifikacijai yra palyginami jau egzistuojantys panašūs sprendimai.

Darbo dokumentą sudaro:

- Įvadas, kuriame analizuojama darbo problematika, tikslai ir uždaviniai;
- Analizė, kurioje svarstomas aukšto lygio galimas sistemos veikimo principas, nagrinėjami galimi komponentai mobiliosios VPN identifikacijos sukūrimui, bei aptariamos panašios sistemos, palyginami jų privalumai ir trūkumai;
- Projektavimas, kuriame aprašoma, kaip pagal analizės išvadas pasirinkti komponentai bus panaudojami mobiliosios VPN sistemos sukūrimui. Kaip sistemos komponentai sąveikaus tarpusavyje ir kur sistemos duomenys bus laikomi;
- Realizacija, kurioje aprašoma kaip buvo realizuotas sistemos prototipas, identifikacijos funkcijų įgyvendinimas ir jo veikimas;
- Eksperimentas, kuriame išbandyti sistemos charakteristikos ir realizuoti metodai, atliekant jų analizę;
- Rezultatų apibendrinimas ir išvados;
- Literatūros sąrašas, kuriame pateikiami naudoti literatūros šaltiniai;
- Priedai, kur yra pateikiamas darbo tema parašytas straipsnis.

1. VPN IDENTIFIKACIJOS SISTEMŲ IR KOMPONENTŲ ANALIZĖ

Analizės skyriuje yra analizuojamas aukšto lygio galimas sistemos veikimo principas, nagrinėjami galimi komponentai mobiliosios VPN identifikacijos sukūrimui, bei aptariamos panašios sistemos, palyginami jų privalumai ir trūkumai

1.1. Analizės tikslas

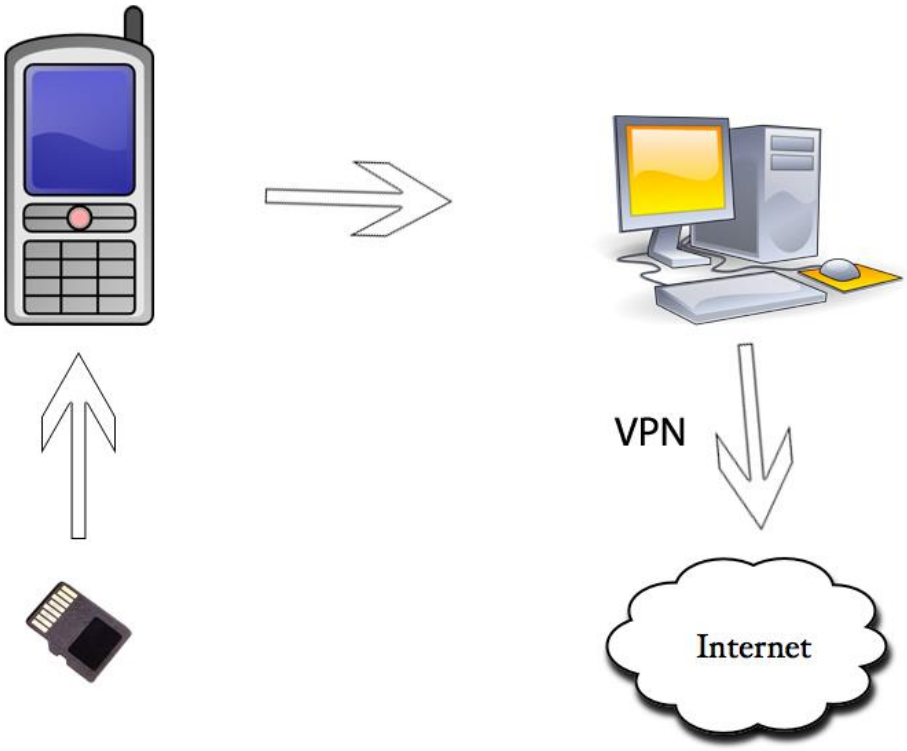
Analizės tikslas yra išanalizuoti tinkamiausius komponentus lengvai panaudojamai ir greitai įvykdomai, bet tuo pačiu saugiai VPN identifikacijos sistemai sukurti pasinaudojant prie kompiuterio neprijungta laikmena saugoti slaptiems autentifikavimo raktams.

1.2. Tyrimo objektas

Tyrimo objektas yra mobilioji VPN identifikacijos sistema, sistemos prototipo charakteristinės savybės ir saugumo aspektai.

1.3. Siūlomos sistemos analizė

Pagal darbe iškeltus tikslus ir uždavinius buvo nustatytas galimas siūlomos sistemos aukšto lygio veikimo principas pateiktas 1.1 paveiksle.



1.1 pav. Mobiliosios VPN identifikacijos sistemos koncepcija

1.4. Kriptografinė microSD

RSA viešojo rakto kriptosistemos funkcijos yra aprašytos PKCS#11 standartu. Šis standartas taip pat yra realizuotas specializuotose microSD kortelėse, kurios palaiko kriptografines funkcijas. Šios kortelės gali būti naudojamos privatiesiems raktams laikyti telefone, o raktai yra pasiekiami tik per gamintojo pateiktą biblioteką.

Kriptografijoje PKCS#11 yra vienas viešojo rakto kriptografijos standartų taip pat apibrėžiantis programavimo prievadą, kuris turi galimybę sukurti ir manipuliuoti kriptografiniais raktais bei realizuoti kriptografines funkcijas [3]. Šis standartas apibrėžia nuo platformos nepriklausomus kriptografinius įrenginius, tokius kaip fizinius saugumo modulius ir išmaniąsias korteles. Programavimo prievadas dažniausiai sutrumpintai yra vadinamas „Cryptoki“ [4].

Programinis „Cryptoki“ prievado standartas apibrėžia plačiausiai naudojamus kriptografinius objektus, tokius kaip RSA raktai, X.509 sertifikatai [5], simetriniai šifravimo algoritmai, pvz., AES ir panašūs. Taip pat PKCS#11 apibrėžia visas funkcijas, naudojamas šiems kriptografiniams objektams kurti, keisti ir ištrinti [3].

Dauguma komercinių sertifikatų pasirašymo centrų naudoja PKCS#11 standartu paremtas programas, kad per jas būtų pasiekiamas sertifikatų pasirašymo centro privatusis raktas, skirtas pasirašymui, arba pasirašytas vartotojo sertifikatas. Tarp platformų bendraujančios programos, kurios naudoja išmaniąsias korteles, remiasi PKCS#11 standartu, pavyzdžiui, „Mozilla Firefox“ ir *OpenSSL* (pasinaudodama papildoma biblioteka). „Cryptoki“ taip pat yra naudojamas išmaniosioms kortelėms ir fiziniams saugumo moduliams nuskaityti [6].

1.5. Artimojo lauko komunikacija

Artimojo lauko komunikacija (ALK) (angl. NFC, *Near Field Communication*) yra standartų rinkinys, skirtas išmaniesiems telefonams ir panašiems įrenginiams, kai yra norima sukurti radijo ryšį tarp dviejų įrenginių juos suliečiant ar suartinant pakankamai mažu atstumu. Dažniausiai atstumas, kuriame vyksta šio tipo komunikacija, yra 10 centimetrų ar mažesnis [7].

Kaip ir su kortelių, veikiančių per atstumą technologija, ALK naudoja elektromagnetinę indukciją tarp dviejų arti viena kitos esančių žiedinių antenų. Remiantis ISO/IEC 18000-3 standartu ALK veikia globaliai prieinamame ir nelicencijuotame radijo 13,56 MHz dažnyje. Greičiai gali būti nuo 106 Kbit/s iki 424 Kbit/s (1.1 lentelė). Ryšio metu bendrauja tik 2 įrenginiai, iš kurių vienas visada būna komunikacijos iniciatorius, turintis savo energijos šaltinį ir aktyviai generuojantis radijo dažnio lauką. Įrenginys, su kuriuo yra bendraujama, gali būti pasyvus, neturėti savo energijos šaltinio, kuris energiją gauną iš komunikaciją inicijavusio aktyvaus įrenginio radijo dažnio lauko. Tokie pasyvūs įrenginiai gali būti žymekliai, lipdukai, fizinio rakto dalis ar kortelės, kurios turi integruotas schemas turinčias antenas, per kurias gauna energiją iš sukurto radijo lauko, ir sugebančias perduoti informaciją [7].

Komunikacija taip pat gali vykti tarp dviejų aktyvių įrenginių (angl. *Peer-to-peer*), jeigu šie komunikacijoje dalyvaujantys įrenginiai turi savo atskirus energijos šaltinius [8].

ALK standartas apibrėžia komunikacijos protokolus ir duomenų apsikeitimo formatus, ir yra paremtas egzistuojančiais radijo ryšio identifikacijos (angl. RFID, *radio frequency identification*) standartais, įskaitant ISO/IEC 14443 ir *FeliCa*. Standartai taip pat palaiko ISO/IEC 18092 ir tuos, kurie yra apibrėžti „NFC Forum“. Papildomai GSMA apibrėžė platformą GSMA ALK standartams (angl. *GSMA NFC Standards*) mobiliuosiuose įrenginiuose palaikyti. Šis standartas papildomai apibrėžia patikimų paslaugų valdiklį (angl. *Trusted Services Manager*), vienos terpės protokolą (angl. *Single Write Protocol*), testavimą ir sertifikavimą, saugumo elementą (angl. SE, *secure element*).

Dabartinės ir planuojamos ALK pritaikymo galimybės gali būti bekontakčiai perdavimai (angl. *contactless transactions*), apsikeitimas duomenimis ir paprastas bei greitas sudėtingesnių komunikavimo protokolų sujungimas.

1.1 lentelė. ALK greitis ir naudojamas kodavimas [9]

Greitis	**Aktyvus įrenginys**	**Pasyvus įrenginys**
424 Kbit/s	Man, 10 proc. ASK	Man, 10 proc. ASK
212 Kbit/s	Man, 10 proc. ASK	Man, 10 proc. ASK
106 Kbit/s	Modified Miller, 100 proc. ASK	Man, 10 proc. ASK

ALK pagrindiniai privalumai yra mažas komunikacijos nuotolis, kai yra reikalinga kuo saugesnė komunikacija, ir greitesnis ryšio užmezgimas. Dėl indukcinio suporavimo ir todėl, kad nereikia papildomos įvesties iš vartotojo pusės, komunikacijos kanalo tarp dviejų įrenginių sukūrimas užtrunka mažiau nei dešimtadalį sekundės [10].

ALK suteikia privalumų vartotojams ir verslui dėl savo protokolo savybių:

- Lengvai naudojamas: komunikacijai užtenka paprasto įrenginių sulietimo;
- Įvairiapusiškas: platus sričių, aplinkų ir panaudojimų pritaikymas;
- Atviras ir paremtas standartais: ALK technologija yra paremta globaliai naudojamais ISO, ECMA ir ETSI standartais;
- Įgalina paprastesnį technologijų panaudojimą: ALK yra naudojamas sparčiai ir paprastai sudėtingesnių komunikacijos protokolų, tokių kaip „Bluetooth“, belaidžio ryšio ir pan., ryšiui užmegzti;
- Paveldimai saugus: ALK veikimas yra labai artimo nuotolio, todėl laikomas gana saugiu;
- Suderinamumas: veikia su egzistuojančiomis be kontaktinių kortelių technologijomis;
- Paruoštas saugumui: turi numatytas galimybes palaikyti saugias programas.

Vienodo lygio įrenginių režime (angl. *Peer-to-Peer mode*) du ALK įgalinti įrenginiai gali apsikeisti duomenimis tarpusavyje. Pavyzdžiui, sulietus du aktyvius įrenginius su įjungtu ALK, galima

apsikeisti vizitinėmis kortelėmis. Taip pat galima paprasčiau sujungti du įrenginius „Bluetooth“ ryšiu pasinaudojant ALK [10].

1.5.1. ALK palyginimas su kitomis komunikacijos technologijomis

Pasinaudojus 1.2 lentele galima greitai palyginti ALK privalumus ir trūkumus su kitomis mažo atstumo komunikacijos technologijomis.

1.2 lentelė. Mažo atstumo komunikacijos technologijų palyginimas [11]

	ALK (angl. *NFC*)	**RFID**	**IrDa**	**„Bluetooth”**
Ryšio sukūrimo laikas	<0,1 ms	<0,1 ms	~0,5 s	~6 s
Atstumas	Iki 10 cm	Iki 3 m	Iki 5 m	Iki 30 m
Panaudojamumas	Orientuotas į žmogų, paprastas, intuityvus, greitas	Orientuotas į daiktus, paprastas	Orientuotas į duomenis, paprastas	Orientuotas į duomenis, Vidutiniškas
Atsirenkamumas	Aukštas, suteiktas, saugumas	Dalinai suteiktas	Regėjimo linijos	Kas esi tu?
Panaudojimo atvejai	Mokėjimai, prieigos suteikimas, duomenų dalinimasis, paslaugų inicializavimas, lengvas ryšio sukūrimas	Daiktams sekti, identifikuoti	Valdyti ir apsikeisti duomenimis	Tinklas apsikeisti duomenimis ir ausinėms prijungti
Vartotojo patirtis	Priliesti, tiesiog prisijungti	Gauti informacijai	Paprastas	Reikalingas konfigūravimas

1.5.2. *OpenVPN*

OpenVPN yra atvirojo kodo programinė įranga, kuri realizuoja virtualiojo privačiojo tinklo technologijas, kad sukurtų saugų įrenginio su įrenginiu (angl. *point-to-point*) arba tinklo su tinklu (angl. *site-to-site*) susijungimus maršruto parinkimo ar tuneliavimo režimuose. Ši programinė įranga naudoja protokolą, kuris pasitelkia SSL / TLS raktams apsikeisti. *OpenVPN* gali komunikuoti per tinklų adresų pakeitimus (angl. NAT, *network adress translators*) ir ugniasienes. Programinė įranga buvo parašyta James Yonan ir yra publikuota pagal *GNU General Public License* (GPL) [12].

OpenVPN leidžia įrenginiams autentifikuoti vienas kitą pasinaudojant iš anksto pasidalintu slaptu raktu (angl. *pre-shared secret key*), sertifikatais arba vartotojo vardu su slaptažodžiu. Naudojant konfigūraciją su daugiau nei vienu klientu įgalina serverį suteikti kiekvienam vartotojui autentifikavimo sertifikatą pasinaudojant parašu ir sertifikatų pasirašymo centru. *OpenVPN* naudoja *OpenSSL* kriptografinę biblioteką ir SSL / TLS protokolą, taip palaikydamas daugelį saugumo ir valdymo konfigūracijų [12].

OpenVPN buvo pritaikytas ir daugiau platformų, tokių kaip *DD-WRT*, *OpenWRT*, *SoftEther VPN*.

OpenVPN gali komunikacijai naudoti du skirtingus prievadų tipus. Tai gali būti OSI 3 lygio IP arba OSI 2 lygio tunelis. Taip pat galima panaudoti duomenų suspaudimo biblioteką LZO. IANA priskirtas prievadas *OpenVPN* programai yra 1194. Jis yra numatytasis prievadas, kai yra naudojama *OpenVPN*, jeigu nustatymuose nėra nurodyta kitaip [12].

1.5.3. *OpenVPN*, L2TP / IPSec ir PPTP palyginimas

Kompiuterių tinkluose OSI 2 lygio tuneliavimo protokolas (angl. L2TP, *Layer 2 Tunelling Protocol*) yra naudojamas palaikyti virtualiesiems privatiesiems tinklams arba kaip dalis pristatymo paslaugų, kurias naudoja interneto tiekėjai (angl. ISP, *internet service provider*). Naudojant vien L2TP nėra suteikiamas duomenų šifravimas ar konfidencialumas. Tam būna panaudojamas kitas protokolas, kuriuo būna apgaubti L2TP paketai [13].

Šiuo atveju L2TP paketai yra apgaubiami IPSec protokolu. Kadangi L2TP paketai yra paslėpti IPSec paketuose, jokia informacija apie vidinį privatųjį tinklą negali būti išgauta iš užšifruoto paketo. Kadangi dažniausiai IPSec yra iššifruojami už ugniasienės, todėl nereikia ugniasienėje atverti prievado UDP 1701, kuris yra naudojamas L2TP protokolo [13].

L2TP / IPSec komunikacijoje galime išskirti tunelio panaudojimą ir saugaus kanalo sukūrimą. Tunelis leidžia nepakeistiems paketams būti perduotiems iš vieno tinklo į kitą, o saugus kanalas užtikrina duomenų konfidencialumą. Šiuo atveju IPSec sukuria saugų kanalą, ir L2TP sukuria tunelį. Detalesnis protokolų palyginimas 1.3 lentelėje.

1.3 lentelė. Virtualių privačių tinklų protokolų palyginimas [14]

	PPTP	**L2TP / IPSec**	***OpenVPN***
Apie protokolą	Labai paprastas VPN paremtas PPP. PPTP specifikacija neapibrėžia šifravimo ar autentifikavimo mechanizmų ir remiasi PPP protokolo tuneliu realizuoti saugumo funkciją.	Pažangus protokolas standartizuotas IETF RFC 3193 ir yra naudojamas vietoj PPTP Microsoft platformose kur saugus duomenų šifravimas yra reikalingas.	Pažangus atviro kodo VPN sprendimas palaikomas „OpenVPN technologies“, kuris dabar yra plačiausiai naudojamas atviro kodo tinklų srityje. Naudoja patikrintą SSL/TLS šifravimo protokolą.
Duomenų šifravimas	PPP duomenys yra užšifruoti pasinaudojant Microsoft Point-to-Porint Encryption protokolu (MPPE). MPPE realizuoja RSA RC4 šifravimo algoritmą su 128 bitų maksimalaus dydžio sensų raktais.	L2TP duomenys yra užšifruoti pasinaudojant standartizuotu IPSec protokolu. RFC 4835 nurodo kad 3DES arba AES šifravimo algoritmas yra naudojamas konfidencialumui užtikrinti.	*OpenVPN* naudoja *OpenSSL* biblioteką šifruojant duomenis. *OpenSSL* palaiko daug skirtingų kriptografinių algoritmų tokių kaip 3DES, AES, RC5, Blowfish.
Saugumo pažeidžiamumai	Microsoft PPTP realizacija turi rimtų pažeidžiamumų. MSCHAP-v2 yra pažeidžiamas žodyno atakos, o RC4 algoritmas yra pažeidžiamas bitų sukeitimo (angl. *bit-flipping*) atakos.	IPSec neturi jokių didelių pažeidžiamumų ir yra laikomas ganėtinai saugiau kai naudojamas su saugiau šifravimo algoritmu tokiu kaip AES.	*OpenVPN* neturi jokių didelių pažeidžiamumų ir yra laikomas labai saugiu kai yra naudojamas su saugiu šifravimo algoritmų tokiu kaip AES.

	PPTP	**L2TP / IPSec**	***OpenVPN***
Greitis	Su RC4 ir 128 bitų raktais, šifravimo sąnaudos yra mažiausios iš visų 3jų šioje lentelėje paminėtų algoritmų, todėl jis yra greičiausias.	L2TP / IPSEC turi šiek tiek didesnį pertekliškumą dėl naudojamos dvigubo apgaubimo.	Kai yra naudojamas numatytame UDP režime patikimame tinkle *OpenVPN* turėtų veikti greičiau nei L2TP / IPSec.
Prievadai	PPTP naudoja TCP prievadą 1723 ir GRE (protokolą 47). Galima lengvai užblokuoti uždraudžiant GRE protokolą.	L2TP / IPSEC naudoja UDP 500 prievadą pirminiam raktų apsikeitimui, protokolą 50 IPSec šifruotiems duomenims (ESP), UDP 1701 prievadą pirminiam L2TP konfigūravimui ir UDP 4500 kai paketai keliauja per NAT. L2TP / IPSec yra lengviau užblokuoti nei *OpenVPN*.	*OpenVPN* gali būti lengvai sukonfigūruotas veikti su bet kokiu UDP ar TCP prievadu. Kad apeiti labai apribojančias ugniasienes *OpenVPN* gali būti sukonfigūruotas naudoti TCP 443 prievadą.
Įdiegimas / Konfigūracija	Visos *Windows* versijos ir dauguma kitų operacinių sistemų, įskaitant mobiliuosius telefonus) standartiškai palaiko PPTP. PPTP reikia tik vartotojo vardo, slaptažodžio ir serverio adreso norint jį sukonfigūruoti.	Visos *Windows* versijos nuo 2000/XP ir Mac OSX 10.3+ ir dauguma mobiliųjų telefonų operacinių sistemų standartiškai palaiko L2TP / IPSec.	*OpenVPN* nėra standartiškai įtrauktas į operacines sistemas ir reikalauja programos įdiegimo. Programos įdiegimas yra paprastas ir ilgai neužtrunka.
Stabilumas / Suderinamumas	PPTP nėra toks patikimas ir taip greitai neatsistato po nestabilios tinklo veiklos kaip *OpenVPN*. Turi smulkių suderinamumo problemų su GRE protokolu ir kai kuriais maršrutizatoriais.	L2TP / IPSec yra sudėtingesnis nei *OpenVPN* yra gali būti sunkiau sukonfigūruoti kad patikimai veiktų tarp įrenginių esančių u NAT maršrutizatorių.	Labai stabilus ir greitas per bevielius, mobilius ir kitus nepatikimus tinklus kur paketų pametimas ir grūstis yra dažnas. *OpenVPN* palaiko TCP režimas kurį geriausia naudoti nepatikimuose susijungimuose, bet šis režimas paaukoja dalį greičio dėl TCP protokolo savybių.
Platformų suderinamumas	• *Windows* • *Mac OSX* • *Linux* • *Apple iOS* • *Android* • *DD-WRT* • *OpenWRT*	• *Windows* • *Mac OSX* • *Linux* • *Apple iOS* • *Android*	• *Windows* • *Mac OSX* • *Linux* • *Android* • *Apple iOS* • *DD-WRT* • *OpenWRT*
Išvados	Dėl didelių saugumo spragų nėra tikslinga naudoti PPTP išskyrus kai įrenginiai palaiko tik PPTP.	L2TP / IPSec yra geras pasirinkimas, bet šiek tiek atsilieka nuo *OpenVPN* greito veikimo ir puikaus stabilumo. Naudojant mobiliuosius įrenginius yra	*OpenVPN* yra geriausias pasirinkimas visoms platformoms. Jis veikia greitai, saugiai ir patikimai. Vienintelis trūkumas tai, kad

	PPTP	L2TP / IPSec	*OpenVPN*
		greičiausiai sukonfigūruojamas, nes yra standartiškai palaikomas.	reikia instaliuoti programą, bet tai ilgai neužtrunka.

OpenVPN naudoja standartinius tinkle protokolus (TCP ir UDP) ir gali būti užkonfigūruotas naudoti nestandartinį prievado numerį. Tai yra gera alternatyva IPSec situacijose, kuriuose virtualaus privataus tinklo standartiniai prievadų numeriai gali būti blokuojami interneto paslaugų tiekėjo. Interneto paslaugų tiekėjas už standartinio VPN tinklo prievado atidarymą gali paprašyti papildomo mokesčio ir tai pristatyti kaip papildomą paslaugą [14].

1.5.4. RSA kriptosistema

RSA yra viena iš praktiškai panaudojamų ir plačiai paplitusių viešojo rakto kriptosistemų, naudojamų saugiam duomenų perdavimui. Tokioje kriptosistemoje užšifravimo raktas yra viešai prieinamas ir skiriasi nuo iššifravimo rakto, kuris yra slaptas ir nėra viešai skelbiamas. Ši asimetrija yra paremta praktiškai sunkia matematine skaidymo problema, kur dviejų didelių pirminių skaičių sandaugos rezultatas yra bandomas išskaidyti ir sužinoti tuos pradinius didelius pirminius skaičius. RSA kriptosistemos pavadinimas yra sudarytas iš pavardžių žmonių, kurie pirmieji viešai apibūdino algoritmą, – Rono Rivesto, Adi Shamiro ir Leonardo Adlemano pirmųjų raidžių [15].

RSA kriptosistemos vartotojas sukuria ir tada viešai paskelbia savo viešąjį raktą, kuris yra paremtas dviem dideliais pirminiais skaičiais, kartu su papildoma reikšme. Pirminiai skaičiai turi būti laikomi paslaptyje. Bet kas, pasinaudojęs sukurtu viešuoju raktu, gali užkoduoti pranešimą, bet jeigu viešasis raktas yra pakankamai didelis, tik žinant pirminius skaičius, kurie yra slapti, yra įmanoma iškoduoti pranešimą. RSA algoritmas apibrėžia tris žingsnius: raktų sukūrimą, užšifravimą ir iššifravimą [16].

Sistemos, kuriose yra naudojama privačiojo ir viešojo raktų, identifikacijai užtenka tik pasirašyti serverio pateiktą arba komunikacijos metu išskaičiuotą reikšmė, kad įvyktų vartotojo identifikacija. Mobiliojoje VPN identifikacijos sistemoje bus naudojama RSA pasirašymo funkcija. Raktų generavimo funkcija bus atliekama atskirai nuo mobiliosios VPN identifikacijos sistemos.

1.6. Mobiliosios VPN identifikacijos naudotojų analizė

Mobiliosios VPN identifikacijos vartotojai gali būti visi žmonės, kurie turi *Android* operacinės sistemos mobiliuosius telefonus ir naudojasi kompiuteryje esančiu *OpenVPN* programa autentifikavimui naudodami RSA viešojo rakto kriptosistemą [17].

1.7. Esamų problemos sprendimų analizė

Esami sprendimai padengia tik dalį mobiliosios VPN identifikacijos sistemos (1.4 lentelė). Juos būtų galima išskirti į dvi sritis:

- ALK autentifikavimas tik su tiesiogiai komunikuojančiu įrenginiu;
- VPN programos tiesioginis naudojamas, kai autentifikavimo raktai yra kompiuteryje arba prie jo tiesiogiai prijungtoje laikmenoje.

1.4 lentelė. Esamų sprendimų palyginimas

Palyginimo kriterijus	***OpenVPN***	**L2TP / IPSec**	***Android Beam***	***S beam***
Paskirtis	Sukurti VPN tunelį tarp įrenginių		Perduoti duomenis tarp mobiliųjų įrenginių	
Konfigūravimas	Būtina papildoma išankstinė konfigūracija ir reikalingas autentifikavimas		Mygtuko paspaudimu įjungti ALK (angl. *NFC*) ir suliesti įrenginius	
Naudojamos pagrindinės technologijos	SSL, TLS	L2TP, IPSec	ALK (angl. *NFC*), „Bluetooth“	ALK (angl. *NFC*), „Wi-Fi Direct“
Veikimo aprašymas	Sukuriamas VPN tunelis tarp įrenginių ir/arba tinklų		Duomenys kuriuos norima persiųsti yra persiunčiami „Bluetooth“ ryšiu	Duomenys kuriuos norima persiųsti yra persiunčiami „Wi-Fi Direct“ ryšiu

Esamų VPN identifikacijos sprendimų palyginimas yra pateiktas 1.5 lentelėje pagal kokybinius parametrus.

1.5 lentelė. Esamų VPN identifikacijos sistemų palyginimas

Palyginimo kriterijus	**Identifikacija slaptažodžiu**	**Identifikacija privačiu raktu esančiu vietiniame kompiuteryje**	**Identifikacija telefonu per identifikacijos paslaugą**
Maža žodyno ar bute-force atakos rizika	-	+	+
FIPS 140-2 lygio raktų apsauga	-	-	-
OS neturi tiesioginės prieiga prie identifikacijos rakto/slaptažodžio	-	-	+
Perkeliamumas	+	-	+
Nereikia prieigos prie interneto	+	+	-
Nereikia papildomos programinės įrangos prieigai prie raktų	+	+	-
Nereikia papildomos fizinės įrangos	+	+	+

1.8. Darbo tikslas, uždaviniai ir siekiami privalumai

Darbo tikslas yra realizuoti saugią, paprastą bei mobiliąją VPN identifikacijos sistemą, kai raktai yra saugomi specializuotoje microSD kortelėje su kriptografinėmis funkcijomis. Taip visi autentifikavimo raktai yra kartu su vartotoju. Raktai, esantys telefone, nėra tiesiogiai prieinami, nes yra saugomi specializuotai saugioje microSD kortelės atmintyje.

Mobiliosios autentifikavimo metu per ALK (angl. *NFC*) skaitytuvą kompiuteryje esančiai agentinei programai yra perduodama autentifikavimo raktu pasirašytą maišos reikšmė, kuri yra saugiai pašalinama įvykus VPN autentifikavimui ir saugaus virtualaus tinklo sukūrimui.

1.9. Siekiamo sprendimo apibrėžimas

Norint išspręsti šia problemą, galima panaudoti telefone esančią atmintį. kuri palaiko kriptografines funkcijas, ir gali apsaugoti slaptus raktus nuo piktavališkų programų [4]. Taip pat, pasinaudojant paprastu ir saugiu komunikacijos protokolu, perduoti privačiuoju raktu pasirašytą maišos reikšmę į kompiuterį, kuriame esanti programa, pasinaudojusi šios reikšmės parašu, sukurs virtualųjį privatų kanalą į kitus privačius tinklus. Programa, esanti kompiuteryje, taip pat pasirūpins, kad maišos reikšmė būtų pašalinta iš kompiuterio po virtualaus privataus tunelio autentifikavimo.

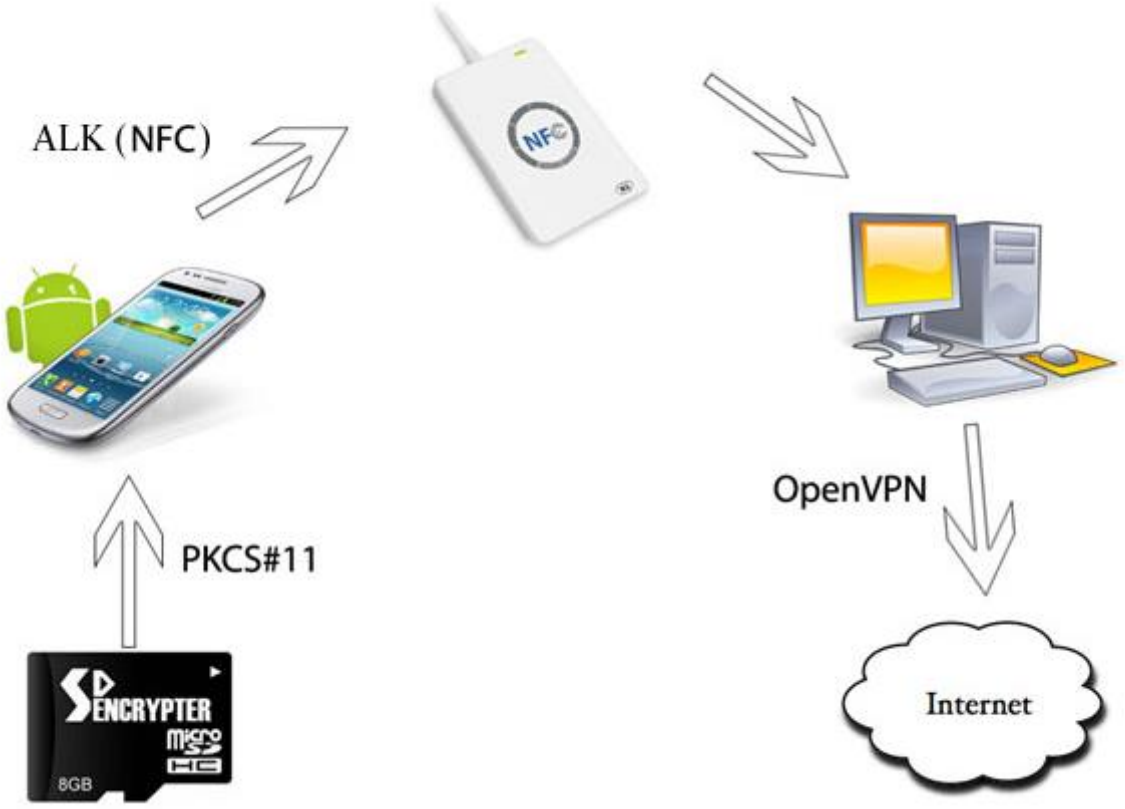
1.10. Analizės išvados

Atlikus galimų mobiliosios VPN identifikacijos sistemos komponentų analizę buvo atsižvelgiama ir palyginama komponentų suteikiamas saugumo lygis, vartojimo paprastumas ir tarpusavio suderinamumas.

Pagal gautus analizės rezultatus buvo nuspręsta mobiliajai VPN identifikacijai panaudoti šiuos esminius komponentus:

- PKCS#11 standartą palaikanti microSD kortelė su kriptografinėmis funkcijomis, nes privatūs raktai bus saugomi specializuotoje ir tam pritaikytoje išorinėje laikmenoje. Jeigu raktai būtų saugomi telefono atmintyje arba paprastoje microSD kortelėje, padidėja tikimybė, kad raktai gali būti sukompromituoti kenkėjiškos programos ar kitų išorinių veiksnių;
- ALK bevielio ryšio technologija yra skirta autentifikavimui reikalingos informacijos perdavimui tarp mobilaus telefono ir kompiuterio. Ši technologija pasirinkta, nes jos nereikia specialiai konfigūruoti, pakanka tik įjungti ALK ryšį;
- Mobilusis telefonas su *Android* operacine sistema yra pasirinktas dėl šių mobiliųjų telefonų paplitimo. *Android* telefone bus programėlė, kuri pasirūpins autentifikavimo informacijos perdavimu tarp mobilaus telefono ir kompiuterio naudojant ALK ryšį. Taip pat ši programėlė turės prieigą prie microSD kortelės su kriptografinėmis funkcijomis;
- Kompiuteris su *Windows* operacine sistema pasirinktas dėl šios operacinės sistemos didelio paplitimo tarp vartotojų. Kompiuteryje veiks agentinė programa, kuri pasirūpins VPN ryšio sukūrimu, naudojant pasirašytą maišos reikšmę, gautą iš mobilaus telefono. Po autentifikavimo agentinė programa pasirūpins, kad konfigūracija būtų pašalinta iš kompiuterio;
- Virtualiam privačiam tinklui sukurti buvo pasirinktas *OpenVPN* programinis paketas. Pagrindiniai šio programinio paketo privalumai yra paprasta konfigūracija ir konfigūracijos lankstumas.

Atsižvelgus į šias technologijas, realizuosime 1.2 paveiksle pavaizduotos mobiliosios VPN identifikacijos sistemos prototipą.



1.2 pav. Mobiliosios VPN identifikacijos sistema

2. SIŪLOMOS MOBILIOSIOS VPN IDENTIFIKACIJOS APRAŠYMAS

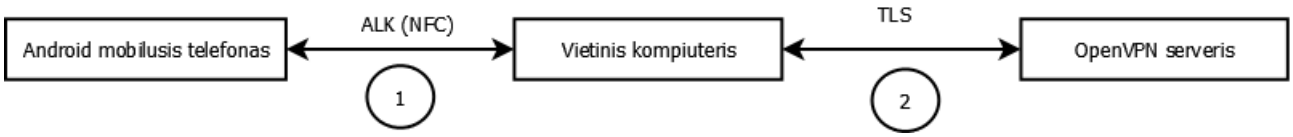
Mobiliąją VPN identifikaciją galime išskirti į dvi dalis (2.1 paveikslas). Paveiksle identifikacija, pažymėta Nr. 1, vyksta tarp mobiliojo telefono ir kompiuterio, pasinaudojant ALK ryšiu. Identifikacijos metu šie įrenginiai apsikeičia savo sertifikatais, identifikuojančius įrenginį, taip yra patikrinama, ar komunikacija iš tiesų vyksta tarp patikimų įrenginių. Šie įrenginių sertifikatai yra pasirašyti bendros sertifikatų tarnybos (angl. *Certificate Authority*).

Jeigu sertifikatai nėra patikimi, tai komunikacija yra nutraukiama. Kitu atveju, *Android* mobiliajame įrenginyje yra pasirenkamas norimas *OpenVPN* susijungimo profilis. Šiame profilyje *OpenVPN* konfigūracija yra susieta kartu su atitinkamais autentifikavimo raktais, kuriais galima autorizuotis konfigūracijoje nurodytame profilyje. Pasirinkto profilio konfigūracijos informacija yra perduodama agentinei programai, veikiančiai vietiniame kompiuteryje ALK ryšiu.

Agentinė programa inicijuoja antrą identifikacijos etapą, pažymėta Nr. 2, su *OpenVPN* serveriu pagal gautą konfigūraciją iš *Android* mobiliojo įrenginio. Sėkmingai įvykus autentifikavimui tarp kompiuterio ir *OpenVPN* serverio, yra sukuriamas saugus TLS kanalas.

Kai tik įvyksta identifikacija, pažymėta Nr. 2, visa perduota informacija iš *Android* mobiliojo įrenginio į vietinį kompiuterį yra saugiai pašalinama.

2.1. Bendrinis siūlomos sistemos veikimas



2.1 pav. Principinė siūlomos mobiliosios VPN identifikacijos schema

Siūlomos sistemos veikimas, kai visos programos bus įdiegtos, pavaizduotas (2.1 paveikslas). Vartotojui, prisijungusiam ir dirbančiam su vietiniu kompiuteriu, reikės savo *Android* telefone paleisti specializuotą programėlę, ir suvedus savo programėlės PIN kodą, pasirinkti norimą *OpenVPN* susijungimo profilį. Pasirinkus profilį, užteks priliesti telefoną prie vietinio kompiuterio ALK prievado ir identifikacijos žingsniai bus atliekami automatiškai. Jeigu visi duomenys pateikti teisingi ir nebuvo jokių ryšio sutrikimų, tai vartotojas vietiniame kompiuteryje po kelių sekundžių turėtų pasiekti įrenginį ar tinklus, pasiekiamus per sukurtą VPN kanalą.

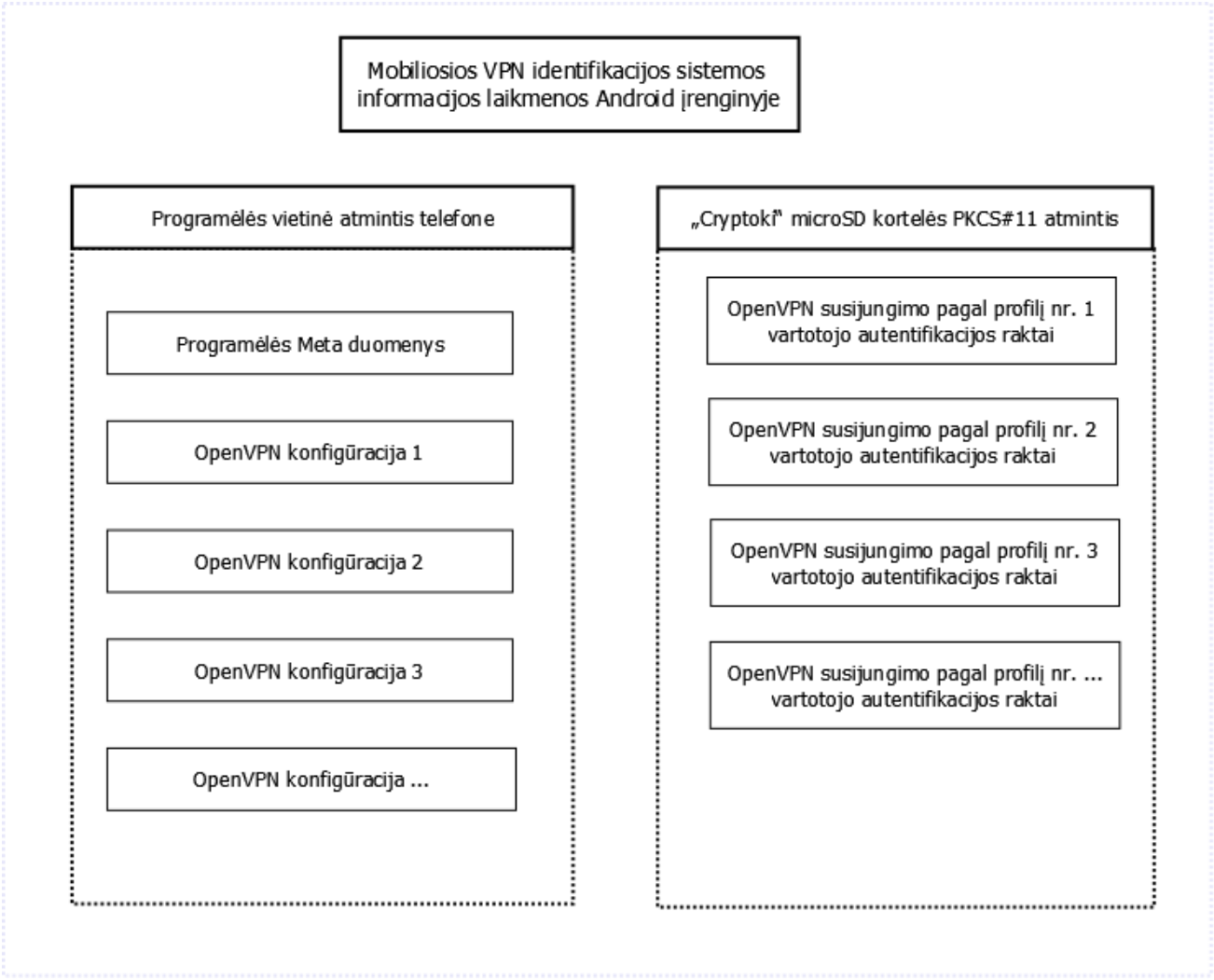
2.2. Informacijos saugojimas

Visa pagrindinė identifikacijos informacija yra saugoma mobiliajame telefone (2.2 paveikslas). Jautri informacija, tokia kaip privatūs raktai, yra saugomi specializuotoje microSD kortelėje su kriptografinėmis funkcijomis. Prie informacijos šioje kortelėje galima prieiti tik per specializuotai sukurtą programėlę, kuri naudojasi specialia PKCS#11 standarto microSD kortelei pritaikyta biblioteka. Tačiau prieigai prie šios PKCS#11 standarto kortelės papildomai reikia papildomai autentifikuotis per *Android* programėlę, perduodant kortelės slaptažodį.

Kiekvieną kartą vartotojas, naujai paleisdamas programėlę, turės suvesti savo nustatytą microSD kortelės PIN kodą, taip identifikuodamas save autorizuotu mobiliojo įrenginio vartotoju, kurio autentifikavimo raktai ir yra saugomi kortelėje.

Mažiau svarbi informacija yra saugoma programėlei išskirtoje vietinėje telefono atmintyje. Tai bus tokie duomenys kaip, pavyzdžiui, pačios programėlės meta duomenys ir konfigūraciniai failai, kurie ALK ryšiu bus perduodami vietiniam kompiuteriui.

2.2.1. *Android* mobiliajame telefone



2.2 pav. Mobiliosios *Android* sistemos informacijos laikmenos turinys

Mobiliajame įrenginyje saugomą informaciją galime išskirti į dvi dalis (2.2 paveikslas). Telefone programėlei skirtoje atmintyje saugomi duomenys yra mažiau svarbūs. Jeigu piktavalis sugebėtų šią informaciją išgauti iš vartotojo mobiliojo įrenginio, tai nepadarytų jokios žalos vartotojui, nes šioje atmintyje yra saugomi programėlės meta duomenys, *OpenVPN* konfigūracijos sąryšiai su saugioje atmintyje, specializuotoje PKCS#11 microSD kortelėje, saugomais raktų indeksais. Dar yra saugoma *OpenVPN* konfigūracijų informacija, tokia kaip *OpenVPN* serverio IP ar DNS adresai ir naudojamas protokolas TCP ar UDP, bei kita pagalbinė informacija. Šią informaciją galima ir taip sužinoti atliekant

prievadų skanavimą (angl. *port scanning*), nukreiptą į *OpenVPN* serverį. Todėl, jeigu ši informacija būtų sužinota piktavalio, tai nepakenktų vartotojo prisijungimo saugumui.

Jautri informacija yra saugoma specialioje microSD kortelės saugioje atmintyje. Ši informacija yra privatūs, viešieji raktai ir sertifikatai, kurie yra naudojami identifikacijai tarp mobiliojo *Android* įrenginio ir vietinio kompiuterio, bei vietinio kompiuterio ir *OpenVPN* serverio. Kadangi šie raktai yra pasiekiami tik per specialią programėlės sąsają ir vartotojas autentifikuojasi programėlėje savo PIN kodu, kuris taip pat yra panaudojamas autentifikuotis prieigai per sąsają į microSD kortelės saugią sritį ir įgalina pasinaudoti realizuotomis kriptografinėmis funkcijomis.

2.2.2. Vietiniame kompiuteryje

Vietiniame kompiuteryje nebus saugoma jokia ilgalaikė informacija, *OpenVPN* kliento programoje bus nustatyta nesaugoti žurnale (angl. *log*) informacijos apie susijungimus.

Konfigūracijos informacija bus laikoma kompiuteryje tik tam tikru laiko momentu. Pavykus ar nepavykus sukurti saugaus kanalo, ši konfigūracinė informacija bus pašalinta.

2.2.3. *OpenVPN* serveryje

OpenVPN serveryje jokia vartotojo informacija nebus saugoma. Identifikacijai pakanka, kad vartotojo sertifikatas, kuriuo yra autentifikuojamasi, yra pasirašytas tos pačios sertifikatų tarnybos kaip ir *OpenVPN* serverio sertifikatas.

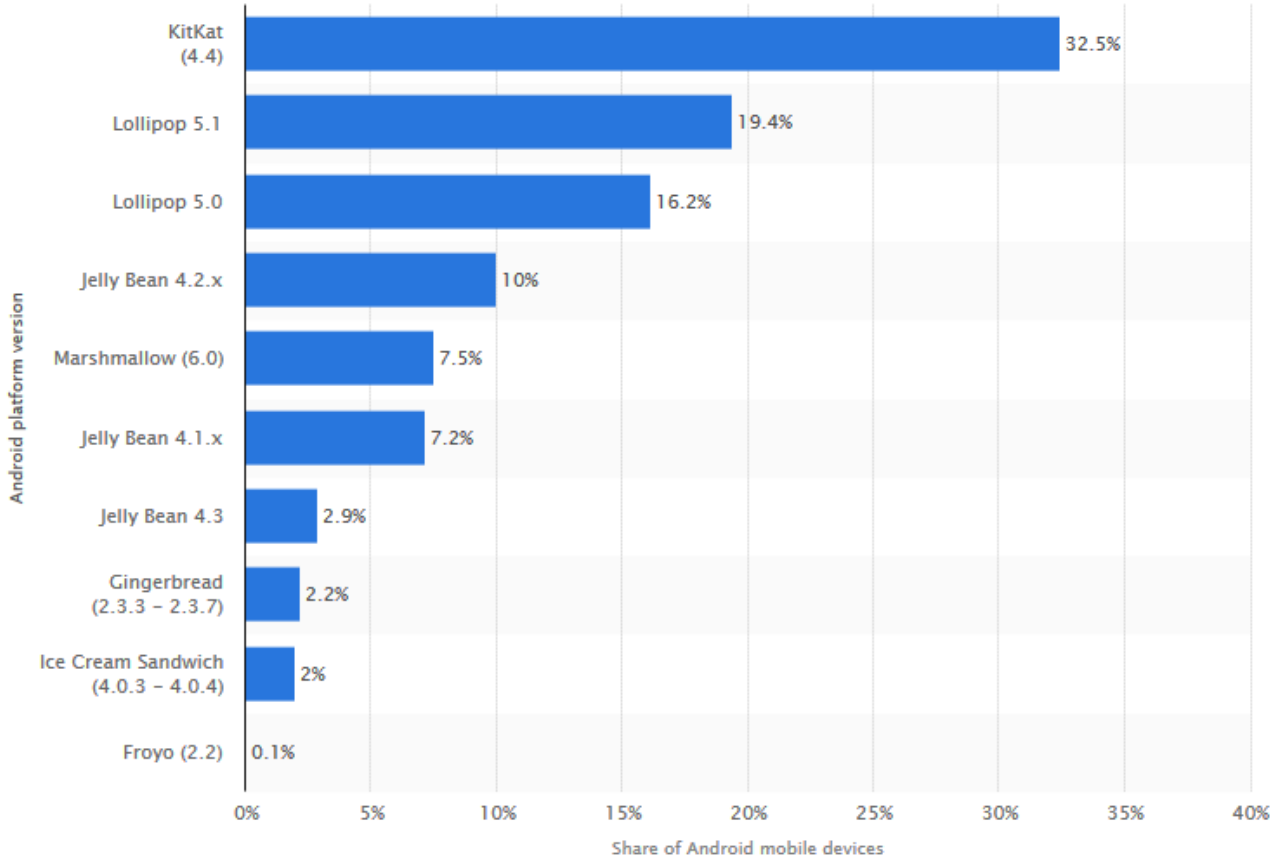
Vienintelė informacija, kuri yra saugoma apie vartotojus, yra aktyvūs seansai (angl. *sessions*). Ir *OpenVPN* serverio žurnale saugoma informacija apie pavykusius ir nepavykusius susijungimus.

2.3. Sistemų reikalavimai

Visose 2.1 paveiksle pažymėtuose įrenginiuose reikia įdiegti papildomą programinę įrangą, kuri dažniausiai nėra įdiegiama kartu su įrenginių operacinėmis sistemomis.

2.3.1. *Android* mobilusis įrenginys

Mobilusis įrenginys turi turėti įdiegtą į įrenginį ALK sąsają. Šis įrenginys privalo būti *Android* versijos 4.0.3 arba aukštesnės, nes to reikalauja suderinamumas su pasirinkta Go-Trust PKCS#11 standarto microSD kortele. Kaip matome iš 2.3 paveikslo (pagal 2016 metų pirmo gegužės 2 dienos duomenis), įrenginiai su minimalia reikiama versija šiuo metu sudaro virš 90 proc. *Android* mobiliųjų įrenginių rinkos, todėl šis siūlomas autentifikavimo metodas yra pritaikomas daugumoje *Android* mobiliųjų įrenginių.



2.3 pav. *Android* versijų pasiskirstymas [18]

2.3.2. Vietinis kompiuteris

Vietiniame kompiuteryje turi būti įdiegta šiai identifikacijai sukurta agentinė programa ir kompiuterio operacinė sistema turi būti „Microsoft Windows 7“ ar aukštesnė. Norint programą paleisti su kitomis operacinėmis sistemomis ar jų versijomis, gali reikėti papildomai įdiegti specializuotas tvarkykles (angl. *drivers*), kurios įgalintų operacinę sistemą atpažinti ALK sąsają (angl. *interface*).

Kartu su agentine programa turi būti įdiegtas ir *OpenVPN* kliento programa. Vietinio kompiuterio vartotojas, kuris naudosis saugiu kanalu, turi turėti pakankamas prieigos teises prie vietinio kompiuterio, kad *OpenVPN* klientas galėtų pridėti naujus maršrutus į maršrutų lentelę, kurie bus nukreipiami per sukurtą saugų kanalą. Vietinis kompiuteris turi būti prijungtas prie vietinio tinklo arba interneto, priklausomai nuo to, kurioje vietoje yra įdiegtas profilyje nurodytas *OpenVPN* serveris.

2.3.3. *OpenVPN* serveris

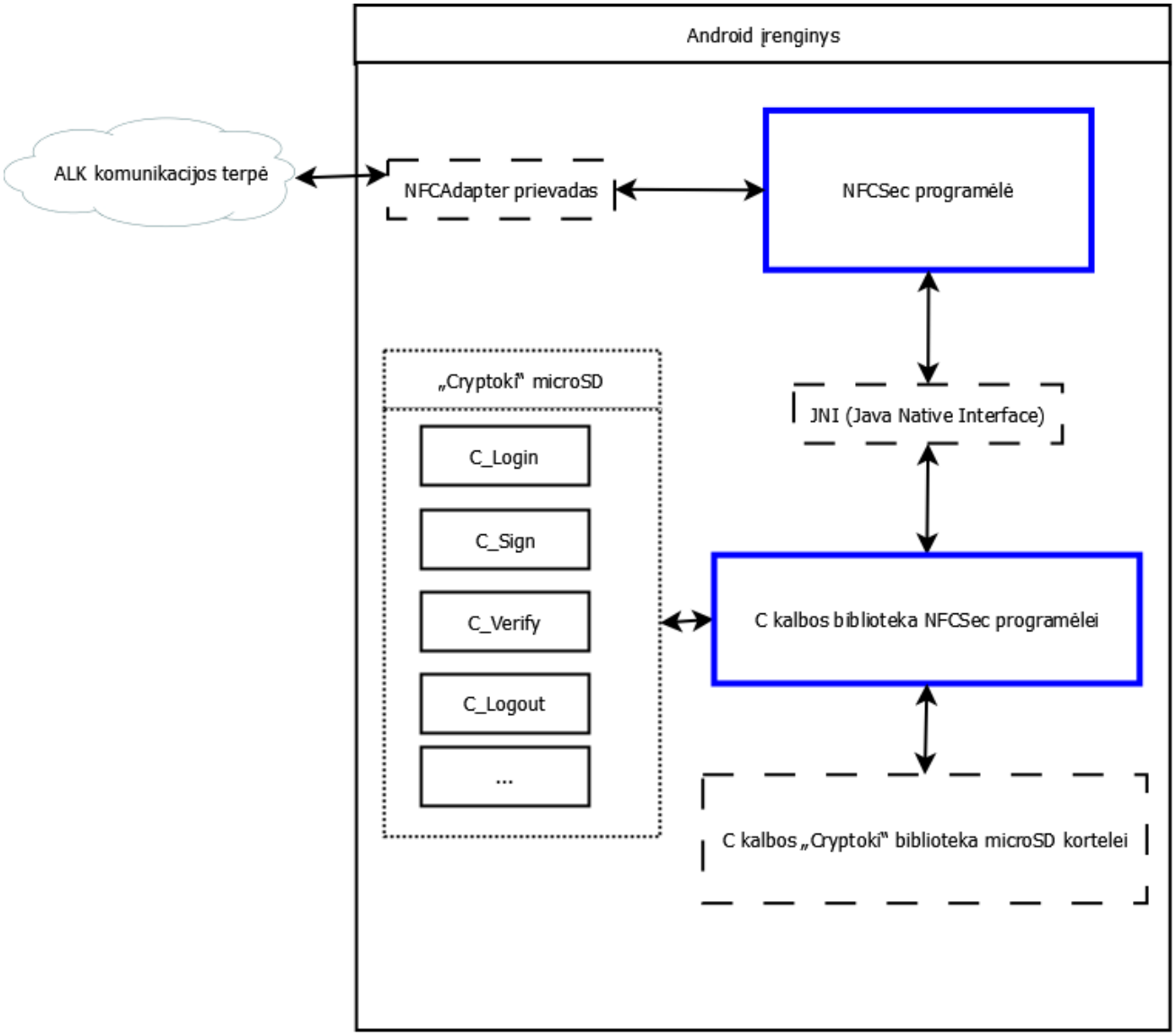
OpenVPN serveriu apribojimų beveik nėra, nes šis programinis paketas gali būti įdiegtas į daugelį operacinių sistemų. Svarbu, kad *OpenVPN* serverio prievadas būtų pasiekiamas per internetą arba vietinį tinklą, priklausomai nuo to, kur yra klientai, kurie jungsis į sukonfigūruotą *OpenVPN* serverį. Ugniasienėje (angl. *firewall*) gali reikėti atverti atitinkamus prievadus.

2.4. Detalizuotas sistemos komponentų veikimas ir sandara

Projektuojamos pilnos sistemos veikime galima išskirti tris pagrindinius sistemos veikimui būtinus įrenginius ir dvi komunikacijos terpes.

2.4.1. *Android* mobilusis įrenginys

Android mobiliajame įrenginyje, kuris turi ALK komunikacijos galimybes ir yra pakankamo lygio versijos, bus įdiegta sukurta programėlė pavadinimu *NFCSec*. Šios programėlės tikslas yra vykdyti kriptografines pasirašymo komandas specializuotoje „Cryptoki“ microSD kortelėje ir atlikti komunikaciją ALK ryšiu su vietiniu kompiuteriu.



2.4 pav. Detalizuota *Android* įrenginio komponentų sąveika sistemoje

Pagal 2.4 paveikslą galime išskirti dvi pagrindines sritis, su kuriomis veikia *NFCSec* programėlė *Android* mobiliajame įrenginyje.

Viena iš sričių yra sisteminės *Android* bibliotekos *NFCAdapter* panaudojimas komunikacijos kanalui sukurti su kompiuteriu, kuris palaiko ALK ryšį. Kita sritis yra skirta bendravimui su specializuota „Cryptoki“ microSD kortele, pasinaudojant kortelės gamintojo pateikta *C* kalbos biblioteka, per kurią yra realizuojamos šios PKCS#11 standarto kriptografinės funkcijos.

Gamintojo pateikta standartinė biblioteka yra parašyta *C* kalba *armeabi* architektūros procesoriams, bet ji tinka ir *armeabi-v7a* architektūros procesoriams. Kadangi *Android* programavimo

kalba yra *Java*, tai, norint pasinaudoti gamintojo pateikta biblioteka, reikia pasinaudoti *Android* sistemos galimybe vykdyti *C* kalbos programinį kodą pasitelkiant JNI (angl. *Java Native Interface*) prievadu. Tam reikia sukurti papildomą biblioteką, jos funkcijas apibrėžti *Android* programėlės programiniame kode, o šių funkcijų realizaciją įgyvendinti *C* kalbos kode.

Realizuotą kodą reikia sukompiliuoti *armeabi* arba *armeabi-v7a* architektūros procesoriams, kuriuos naudoja *Android* mobilieji įrenginiai, pasinaudojant *Android* NDK (angl. *Android Native Development Kit*). Tada turint pateiktą kortelės gamintojo ir mūsų sukurtą bibliotekas reikia įtraukti į programėlę. Bibliotekos privalo būti užkrautos prieš panaudojant šių bibliotekų funkcijas.

Programėlėje bus naudojamos 2.1 lentelėje išvadintos „Cryptoki“ (PKCS#11) standarto funkcijos.

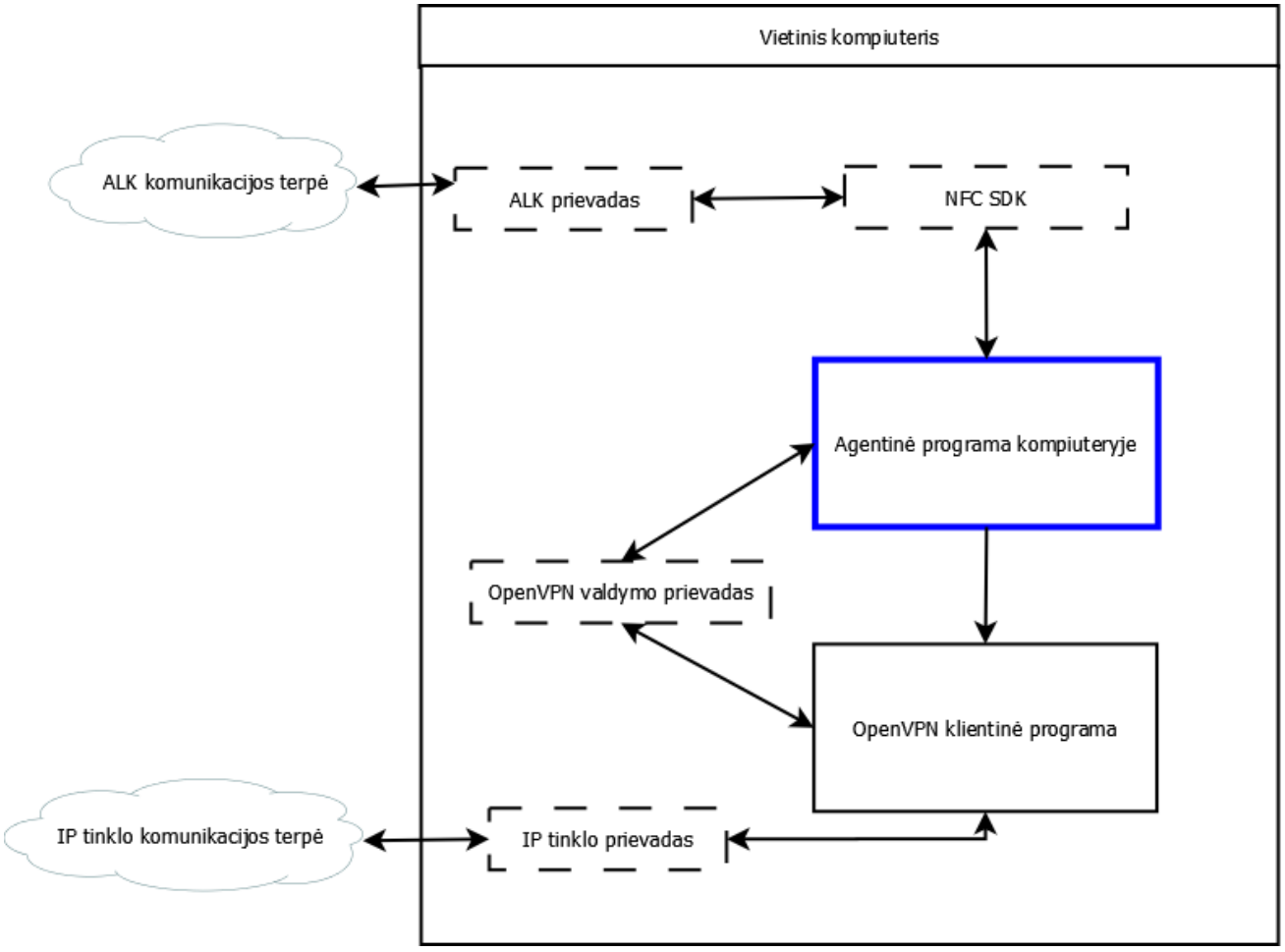
2.1 lentelė. Naudojamos PKCS#11 funkcijos ir jų paskirtis

Funkcija	**Paskirtis**
C_Initialize	Paruošia PKCS#11 standarto bibliotekos funkcijas panaudojimui ir įkrauna į atmintį
C_Finalize	Užbaigia PKCS#11 standarto bibliotekos funkcijų naudojimą ir atlaisviną atmintį
C_GetSlotList	Sukuria sąrašą microSD kortelių prijungtų prie įrenginio ir suranda koks laikmenos identifikatorius yra priskirtas gamintojo PKCS#11 standarto kortelei
C_OpenSession	Sukuria sesiją su „Cryptoki“ kortele
C_CloseSession	Nutraukia sesiją su „Cryptoki“ kortele
C_Login	Autorizuojasi vartotojo teisėmis į kortelę perduodant nustatytą PIN kodą
C_Logout	Atsijungia nuo kortelės kaip vartotojas
C_FindObjectsInit	Inicializuoja objektų paiešką kortelėje
C_FindObjects	Suranda kortelėje esančius objektus pagal nustatytus paieškos filtrus
C_FindObjectsFinal	Užbaigia objektų paiešką
C_Sign	Atlieka perduotos informacijos pasirašymą pasirinktu algoritmu ir nurodytu privačiuoju raktu, kuris yra saugomas kortelėje

Aprašytos PKCS#11 standarto funkcijos (2.1 lentelėje) yra naudojamos informacijos bloko, kuris yra gautas iš *OpenVPN* serverio, pasirašymui su vartotojo privačiuoju raktu laikomu kortelės saugioje atmintyje. Pagal *OpenVPN* reikalavimus blokas turi būti pasirašytas naudojant PKCS#11 standarto mechanizmą *CKM_RSA_PKCS* [19].

2.4.2. Vietinis kompiuteris

Vietinis kompiuteris projektuojamoje sistemoje yra įrenginys, kuriam norima užtikrinti prieigą prie nutolusių resursų, pasinaudojant *OpenVPN* saugaus ryšio kanalu. VPN kanalas yra sukuriamas tarp vietinio kompiuterio ir *OpenVPN* serverio. Šie įrenginiai gali būti tame pačiame tinkle arba nutolę vienas nuo kito per Internetą. Reikia užtikrinti, kad vietinis kompiuteris galėtų pasiekti *OpenVPN* serverį IP protokolu.



2.5 pav. Detalizuota vietinio kompiuterio komponentų sąveika sistemoje

Kaip pavaizduota 2.5 paveiksle, veikianti agentinė programa kompiuteryje klausosi ALK prievado ir laukia ALK ryšio sukūrimo iš *Android* mobiliojo įrenginio su tam skirta *NFCSec* programėle ir PKCS#11 kriptografines funkcijas palaikančia microSD kortele.

Kai gaunama komanda iš mobiliojo įrenginio apie norimą VPN ryšio sukūrimą, yra apsikeičiama įrenginių sertifikatai, pasirašytai tos pačios sertifikatų valdymo tarnybos. Tada mobilusis įrenginys perduoda *OpenVPN* konfigūracijos failą ALK ryšiu agentinei programai su susijungimo parametrais, tokiais kaip ryšio protokolas TCP ar UDP, nutolusio *OpenVPN* serverio IP adresas, naudojami duomenų suspaudimo būdai ir kitais būsimą sesiją apibūdinančiais parametrais.

Agentinė programa konfigūracijos failą ir vartotojo sertifikatą perduoda *OpenVPN* klientinei programai jos paleidimo metu, kartu nurodant, kad bus naudojamas *OpenVPN* valdymo prievadas.

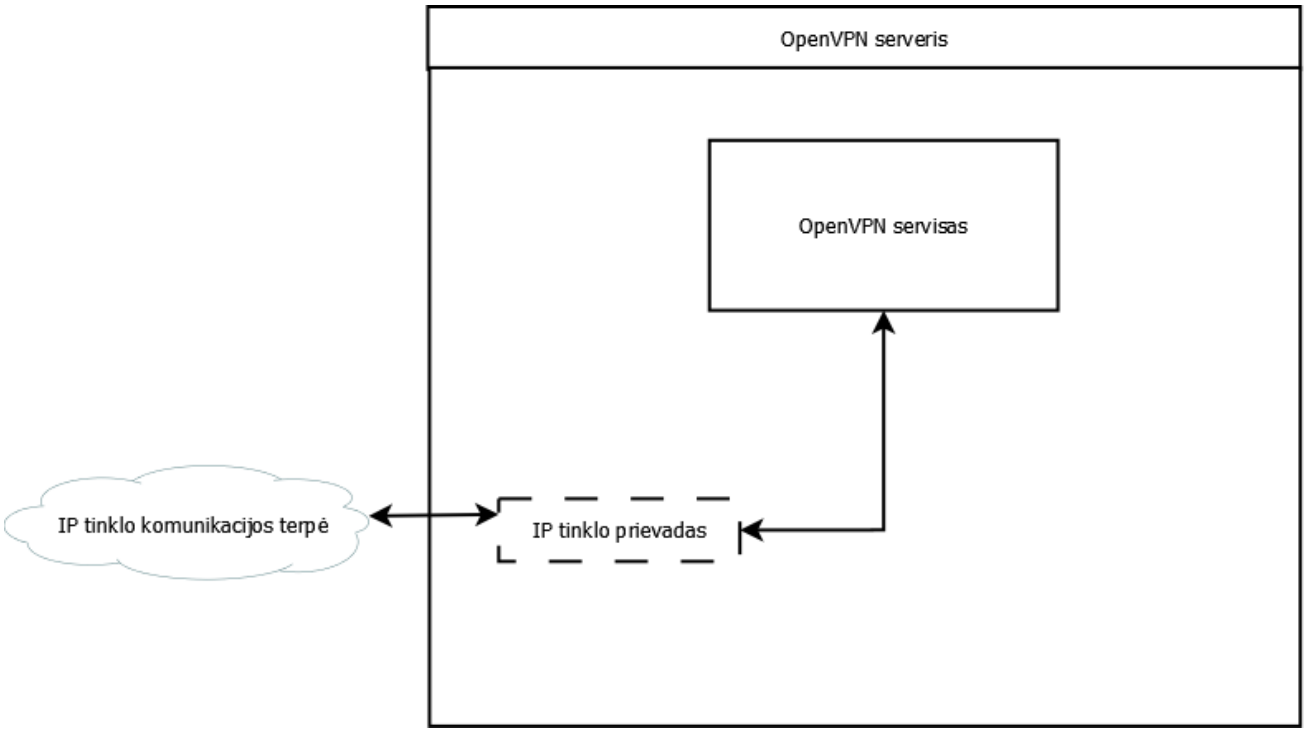
Agentinė programa prisijungia prie *OpenVPN* valdymo prievado ir gali stebėti visą susijungimo procesą. Jeigu susijungimas nepavyksta dėl nepasiekiamo serverio ar kitų priežasčių, tai bandymas sukurti saugaus ryšio kanalą yra nutraukiamas. Jeigu susijungimas pavyksta, *OpenVPN* valdymo kanalu yra gaunamas informacijos blokas, kuris turi būti pasirašytas privačiuoju vartotojo raktu, saugomu *Android* mobiliojo įrenginio PKCS#11 microSD kortelės atmintyje. Pagal *OpenVPN* reikalavimus, blokas turi būti pasirašytas naudojant PKCS#11 standarto mechanizmą *CKM_RSA_PKCS* [19]. Kadangi šis informacijos blokas yra gaunamas ASCII formatu, jam reikia

atlikti Base64 iškodavimo operaciją [20]. Taip gaunamas galutinis 36 baitų ilgio informacijos blokas, kuris turi būti pasirašytas. Atsakymas į perduotą informacijos bloką turi būti grąžintas trumpiau nei per 1 minutę. Jeigu parašas yra grąžinamas per ilgiau nei 1 minutę, *OpenVPN* serveris pasirašytą bloką laiko negaliojančiu ir sugeneruoja naują informacijos bloką pasirašymui.

Informacijos blokas, gautas iš *OpenVPN* serverio pasinaudojant agentinės programos ALK prievadu, yra persiunčiamas *Android* mobiliojo įrenginio programėlei pasirašymui. Kai programėlė pasirašo informacijos bloką, parašas yra grąžinamas agentinei programai. Tada parašas, per valdymo prievadą, yra perduodamas *OpenVPN* serveriui. Jeigu parašas yra teisingas ir buvo perduotas laiku, įvyksta *OpenVPN* tunelio sukūrimas pagal konfigūracijos parametrus. Siekiant apsaugoti *OpenVPN* konfigūracinius parametrus, po sėkmingo VPN tunelio sukūrimo, agentinė programa pašalina konfigūracijos failą iš vietinio kompiuterio [21].

2.4.3. *OpenVPN* serveris

OpenVPN serverio sertifikatas yra pasirašytas tos pačios sertifikatų tarnybos. Vartotojai, sukurdami tunelį su *OpenVPN* serveriu, yra atpažįstami pagal sertifikatų tarnybos sertifikatą ir tikrinama, ar sertifikatas yra galiojantis.



2.6 pav. Detalizuota *OpenVPN* serverio komponentų sąveika sistemoje

OpenVPN serverio (pagal 2.6 paveikslą) konfigūracija yra paprasta, palyginus su kitais pagrindiniais sistemos komponentais. Tai yra todėl, kad *OpenVPN* serveris veikia kaip vartai (angl. *gateway*) norint, kad autorizuoti vartotojai per saugų kanalą pasiektų sukonfigūruotus tinklo resursus.

2.4.4. ALK ryšio kanalas

ALK ryšio kanalas dėl naudojimo paprastumo ir ryšio sukūrimo spartos yra naudojamas greitai perduoti mažus informacijos kiekius tarp dviejų sistemos komponentų: *Android* mobiliojo įrenginio ir vietinio kompiuterio.

Kadangi šiuo kanalu perduodami duomenys nėra privatūs, tai nėra būtina juos šifruoti ir, jeigu piktavalis šiuos duomenis nuskaitytų, tai nepadarytų jokios žalos vartotojui. Todėl komunikacijos metu persiunčiamus duomenis užtenka tik pasirašyti užtikrinant jų vientisumą.

Pagal ALK standartą komunikacijoje gali dalyvauti tik du įrenginiai. Jeigu komunikacijoje dalyvautų daugiau nei 2 įrenginiai perduodama, informacija taptų iškraipyta ir niekas jos negalėtų perskaityti. Tai galima būtų prilyginti DoS atakai, bet, dėl trumpo ALK ryšio komunikavimo atstumo, tai yra mažai tikėtina ir piktavalis turėtų būti labai arti komunikacijoje dalyvaujančių įrenginių.

Jeigu piktavaliui ALK ryšiu pavyktų pasiklausyti perduodamos informacijos, tai vartotojas nenukentėtų, nes piktavalis gautų tik viešuosius raktus ir parašus, kurie yra aktualūs tik perduodamos informacijos vientisumui užtikrinti.

Komunikacijos greitis naudojant ALK nėra didelis, tačiau perduodama realizuojamos sistemos informacija taip neužima daug vietos. Visas duomenų apsikeitimas turėtų trukti mažiau nei 1 sekundę.

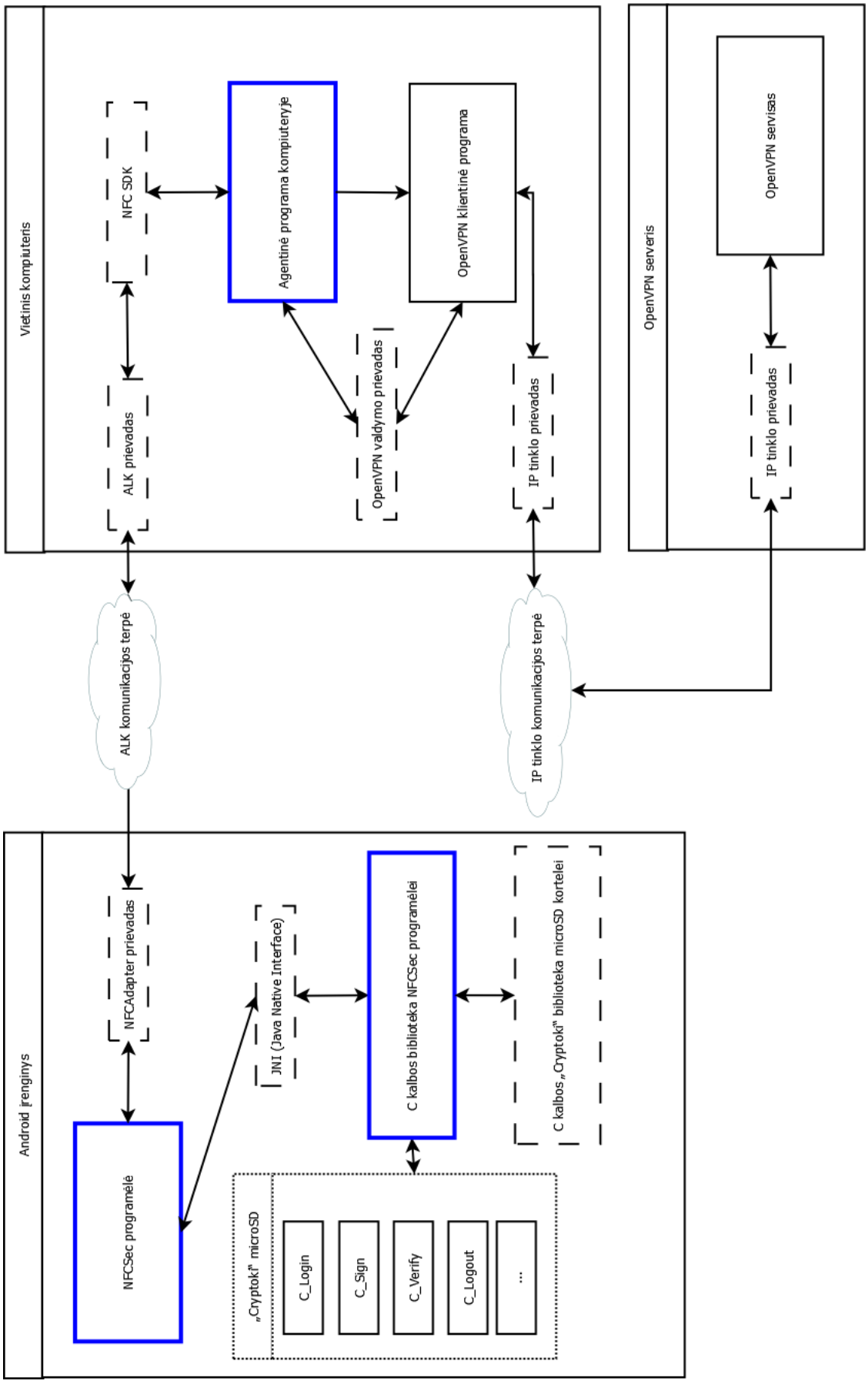
2.4.5. IP tinklo ryšio kanalas

IP tinklo ryšio kanalas realizuojamai sistemai yra reikalingas tik pagrindiniams dviems įrenginiams: vietiniam kompiuteriui ir *OpenVPN* serveriui.

Vietinis kompiuteris naudoją šį kanalą VPN tunelio sukūrimui su *OpenVPN* serveriu. Susijungimas su *OpenVPN* serveriu yra realizuojamas pagal pateiktą vartotojo konfigūracijos failą. Ryšys gali būti užmegztas TCP arba UDP protokolu, bet *OpenVPN* serveris turi būti sukonfigūruotas klausytis atitinkamai TCP arba UDP prievado. Norint sumažinti tinklo apkrovimą, galima naudoti duomenų suspaudimą, bet tai rekomenduojama tik tada, kai vietiniame kompiuteryje ir *OpenVPN* serveryje yra pakankamai laisvų skaičiavimo resursų, nes suspaudimas apkrauna įrenginių procesorius.

Saugaus ryšio sukūrimui yra naudojamas TLS. Autentifikavimui yra naudojama viešojo rakto kriptosistema (angl. *public key infrastructure*). Simetrinio rakto autentifikavimas šioje realizuojamoje sistemoje yra negalimas [22].

2.5. Detalizuotas visos sistemos veikimas



2.7 pav. Detalizuotas visos sistemos veikimas

Mobiliosios VPN identifikacijos sistemos prototipas bus realizuotas pagal 2.7 paveiksle pateiktą schemą, išskiriant 3 pagrindinius sistemos komponentus: *Android* mobilųjį įrenginį, vietinį kompiuterį ir *OpenVPN* serverį.

Android įrenginiui reikės sukurti programėlę ir ją įdiegti į įrenginį, kartu su microSD kortele, palaikančia PKCS#11 standarto funkcijas. Programėlės sukūrimui bus naudojama plačiausiai šiuo metu naudojama *Android* programavimo aplinka „Android Studio“, taip pat *Java* bei *C* programavimo kalbos.

Vietiniame kompiuteryje bus sukurta ir įdiegta agentinė programa, valdanti ALK komunikaciją tarp mobiliojo įrenginio, vietinio kompiuterio ir *OpenVPN* klientinės programos. *OpenVPN* klientinė programa bus įdiegta kompiuteryje.

Serveryje bus įdiegtas ir sukonfigūruotas *OpenVPN* serverio paketas.

2.6. Projektavimo išvados

Mobiliosios VPN identifikacijos sprendime siūloma panaudoti FIPS 140-2 lygio apsaugą privatiems identifikacijos raktams saugoti. Buvo pasiūlyta PKCS#11 standarto microSD kortelė su kriptoprocesoriumi, kuris užtikrina FIPS 140-2 3 lygio apsaugą. Taip vartotojams nereikės nešiotis papildomo įrenginio, nes microSD kortelė bus naudojama kartu su mobiliuoju telefonu.

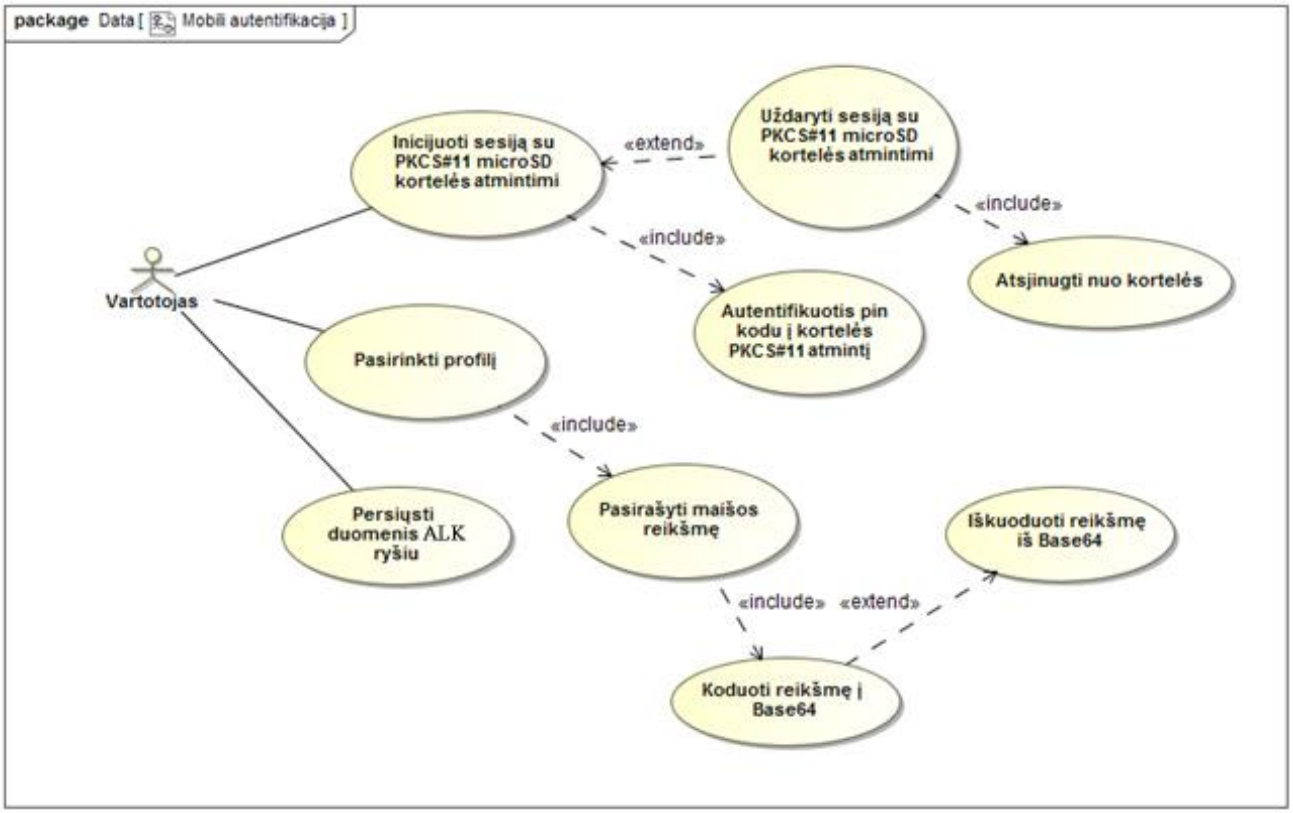
Programinė įranga bus įdiegiama į vietinį kompiuterį kartu su VPN klientu. Mobiliajame telefone taip pat bus įdiegiama programėlė, skirta prieigai prie PKCS#11 microSD kortelės privačiojo rakto objekto. *OpenVPN* serveryje bus naudojama standartinė konfigūracija. Identifikacijos sistemos panaudojimui būtina turėti PKCS#11 standarto microSD kortelę telefone, įdiegtas programas telefone ir vietiniame kompiuteryje.

3. MOBILIOSIOS VPN IDENTIFIKACIJOS SISTEMOS PROTOTIPAS

Kuriant mobiliosios VPN identifikacijos sistemą, reikia realizuoti jos prototipą. Naudojantis prototipu, galima įvertinti sistemos autentifikavimo spartą, atskirų sistemos komponentų spartą, jų patikimumą ir kitus esminius faktorius.

3.1. Sistemos prototipo reikalavimai

Sukurto prototipo vartotojas, tai gali būti ir prototipo testuotojas, naudosis 3.1 paveiksle aprašytomis funkcijomis. Kadangi sistema buvo projektuojama perkeliamumui ir vartojimo paprastumui, todėl didžioji dalis vartotojo naudojamų funkcijų yra automatizuotos.



3.1 pav. Kuriamos sistemos prototipo panaudos atvejų diagrama

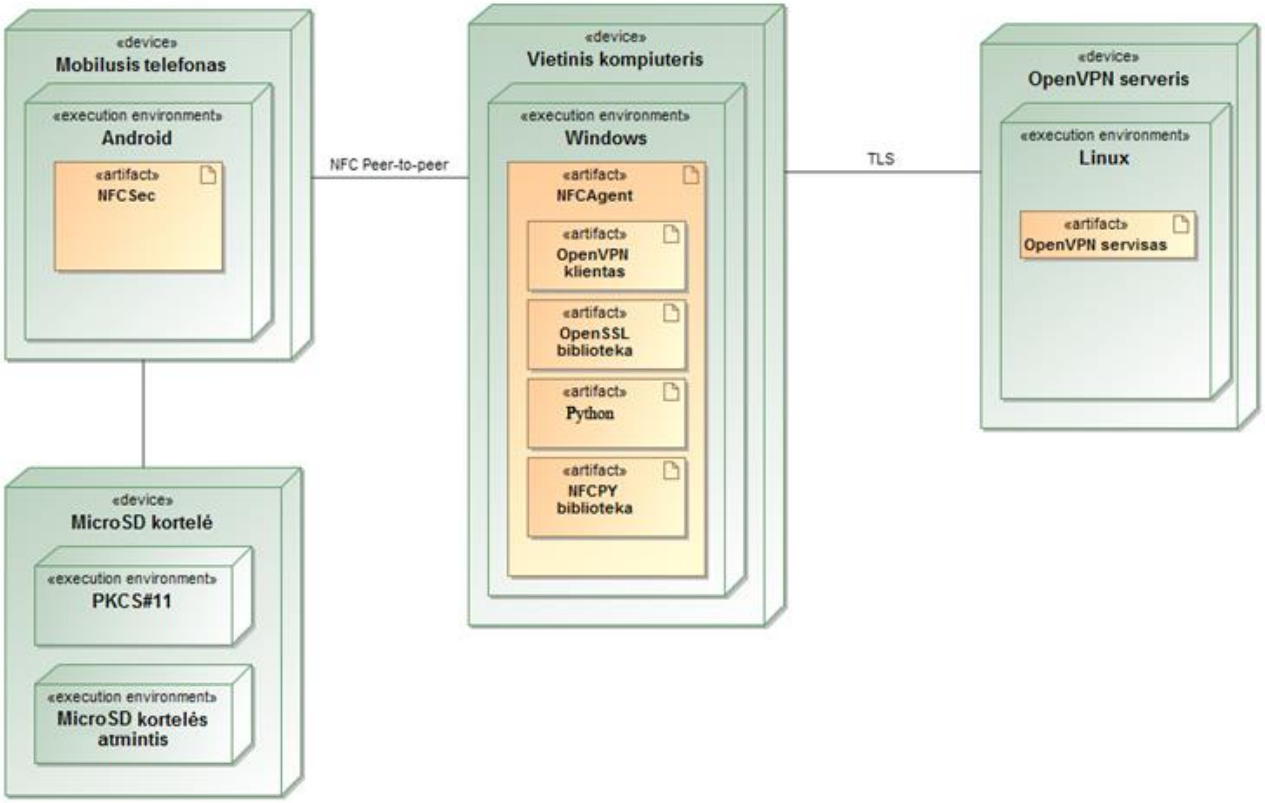
3.2. Mobiliosios VPN identifikacijos sistemos prototipo struktūrinė schema

Realizuoto sistemos prototipo struktūrine schema yra pavaizduota 3.2 paveiksle. Nors prototipas yra realizuota su plačiausiai paplitusiais komponentais, bet dėl naudojamų technologijų perkeliamumo galima realizuoti ir su kitais komponentais.

Android mobilųjį įrenginį galima pakeisti į *iOS* įrenginį su ALK palaikymu. Tačiau dėl to, kad *iOS* fiziškai neturi išorinio microSD kortelės reikia papildomai prijungti įrenginį, kuris įgalina microSD kortelės palaikymą [23].

Vietinio kompiuterio komponentas gali būti realizuotas ir su kita operacine sistema, pavyzdžiui *Linux*, kuri palaiko *Python* ir *OpenVPN* bei sugeba bendrauti per ALK prievadą.

OpenVPN servisas gali veikti, bet kurioje aplinkoje, kurioje jis yra palaikomas. Taip yra, nes siūlomoje sistemoje yra atliekama tik konfigūracija, be jokių kitų sistemos pakeitimų.



3.2 pav. Kuriamos sistemos prototipo komponentų diagrama

3.2.1. *OpenVPN* serverio komponentai

Serverio dalyje yra atliekamas tik *OpenVPN* paslaugos konfigūravimas. Prototipo testavimui yra naudojama 3.1 lentelėje nurodyta įranga ir programiniai komponentai.

3.1 lentelė. *OpenVPN* serverio įranga ir naudojami komponentai

Kompiuterinė įranga	Synology DS212 NAS
CPU	1x 1.6Ghz
Operatyvioji atmintis (RAM)	256MB
Operacinė sistema	*Linux*
Operacinės sistemos branduolio versija	2.6.32.12
Komponentai	*OpenVPN* 2.3.6 *OpenSSL* 1.0.1q-fips

Konfigūracinio *OpenVPN* paslaugos failo „openvpn.conf“ turinys su paaiškintais pagrindiniais parametrais failo komentaruose yra pavaizduotas 3.3 paveiksle.

```
dev tun # VPN tinklo tipas

server 10.9.0.0 255.255.255.0   # Serverio naudojamų adresų potinklis

dh /var/packages/VPNCenter/target/etc/OpenVPN/keys/dh1024.pem   # Diffie-Hellman parametrai
ca /var/packages/VPNCenter/target/etc/OpenVPN/keys/ca.crt       # CA sertifikatas
cert /var/packages/VPNCenter/target/etc/OpenVPN/keys/server.crt # Serverio sertifikatas
key /var/packages/VPNCenter/target/etc/OpenVPN/keys/server.key  # Serverio privatusis raktas

max-clients 5   # Vienu metu gali būti prisijungę iki 5 klientų

persist-tun
persist-key

verb 3

keepalive 10 60
reneg-sec 0

push „route 10.9.0.0 255.255.255.0" # Klientui perduodamas pasiekiamas tinklas

status /tmp/ovpn_status_2_result 30
status-version 2
proto udp   # Naudojamas protokolas
port 1194   # Naudojamas standartinis prievadas
```

3.3 pav. *OpenVPN* serverio konfigūracija

3.2.2. Vietinio kompiuterio komponentai

Vietiniame kompiuteryje realizuotas agentinės programos prototipas, iš esmės veikiantis duomenų perdavimui tarp *OpenVPN* klientinės programos ir mobiliajame telefone realizuotos programėlės *NFCSec* prototipo. Su *NFCSec* programėle agentinė programa *NFCAgent* bendrauja ALK ryšiu su SNEP palaikymu. Su *OpenVPN* programa yra bendraujama naudojantis *OpenVPN* programos valdymo prievadu (angl. *management interface*).

Vietiniame kompiuteryje bus įdiegta sukurtas *NFCAgent* programos prototipas ir 3.2 lentelėje nurodyta programinė įranga.

3.2 lentelė. *NFCAgent* naudojami komponentai

Komponentas	**Versija**
OpenVPN	2.3.10
OpenSSL	1.0.1j

Kartu įdiegiamas patikimos sertifikatų tarnybos (angl. *Certificate Authority*) sertifikatas, kuris bus naudojamas *OpenVPN* konfigūracijoje. *NFCAgent* programėlės veikimui reikia papildomai įdiegti „nfcpy“ biblioteką kartu su *Python* 2 versija.

NFCpy – *Python* biblioteka realizuoja P2P (angl. *peer-to-peer*) komunikaciją per ALK (angl. *NFC*) skaitytuvą. *Python* versija 2 yra reikalinga „nfcpy“ bibliotekos panaudojimui. Prototipo testavimui bus naudojama *Python* versija 2.7.

3.2.3. Mobiliojo telefono komponentas

Sukurtas mobiliojo *Android* telefono programėlės *NFCSec* prototipas pavaizduotas 3.5 paveiksle. Prototipas naudoja NFC SNEP protokolą bendravimui su ALK skaitytuvu, prijungtu prie vietinio

kompiuterio. Su PKCS#11 standarto microSD kortelės atmintimi programėlė bendrauja per standartinį prievadą, kaip ir su paprasta microSD kortele. Komunikacija su saugia atmintimi vyksta ne tiesiogiai, o per kortelės gamintojo pateiktą biblioteką, šiuo atveju microSD kortelė yra „GoTrust“ gamintojo, o naudojama biblioteka „GTTknP11“.

Norint, kad sukurtas programėlės prototipas korektiškai veiktų, reikia suteikti prieigą prie tam tikrų *Android* sistemos komponentų. Kadangi yra naudojama microSD kortelė, reikia prieigos skaityti ir rašyti į kortelę. Taip pat komunikacijai naudojant ALK, reikia prieigos prie ALK sisteminio modulio. NFC SNEP protokolo palaikymui yra būtinas minimalus *Android* operacinės sistemos API lygis 16. Taigi, programėlės naudojimas galimas tik turint *Android* versiją 4.1, dar kitaip vadinamą *JellyBean*, arba aukštesnę. Siekiant, kad programėle būtų galima naudotis užtikrinant šiuos reikalavimus, konfigūraciniame faile „AndroidManifest.xml“, turi būti nurodyti 3.4 paveiksle esantys apribojimai.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.nauckunas.arvydas.nfcsec" >

    <uses-permission android:name="android.permission.NFC"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-feature android:name="android.nfc" android:required="true" />
    <uses-sdk android:minSdkVersion="16"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

3.4 pav. *NFCSec* programėlės informacinio failo turinys

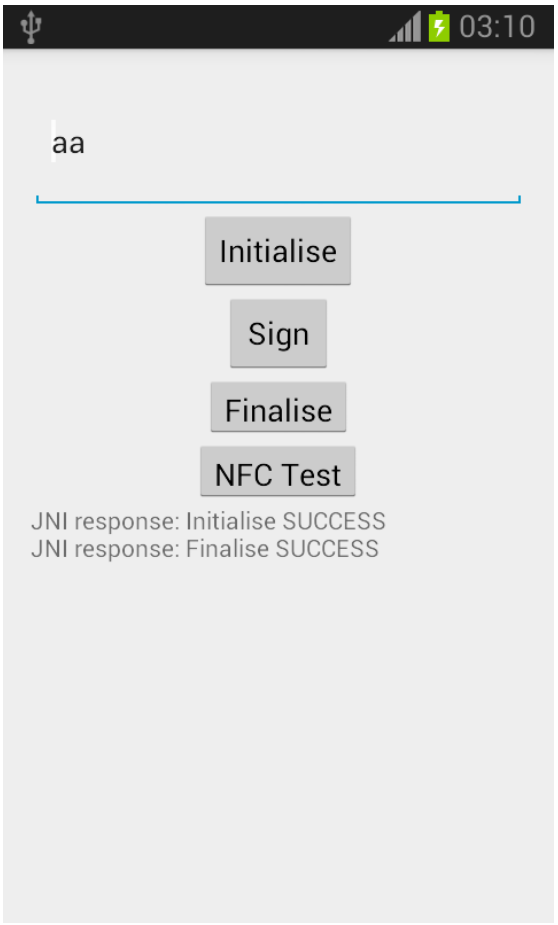
Prototipo testavimui bus naudojamas 3.5 paveiksle pavaizduota grafinė *NFCSec* programėlės sąsaja. Grafinėje sąsajoje tekstinis laukas naudojamas pasirašymo testavimui, o tekstinis išvesties laukas parodo kiekvieno iš realizuotų funkcijų išvesties rezultatą ir apdorojimo trukmę. Norint panaudoti gamintojo biblioteką, kuri yra realizuota per JNI (angl. *Java Native interface*), buvo sukurta

programėlei skirta JNI *C* biblioteka, pavadinta „*NFCSec*_cryptoLib“. Kai *NFCSec* yra įjungiama, šios bibliotekos yra iškart įkraunamos į atmintį.

Visos realizuotos funkcijos testavimui, išskyrus „NFC Test“, sąveikauja per JNI su *C* programos kodu. „NFC Test“ nesąveikauja su microSD kortelės saugia atmintimi, o tik realizuoja duomenų apsikeitimą NFC SNEP protokolu.

Grafinės sąsajos mygtukų veiksmai:

- Initialise – panaudoja „*NFCSec*_cryptoLib“ biblioteką, suranda microSD kortelės prievadą, jeigu jų būtų daugiau nei vienas, patikrina ar įdėta kortelė yra palaikoma, palaiko PKCS#11 standartą. Po patikrinimų inicijuoja sesiją su microSD kortelės saugia atmintimi bei autentifikuoja vartotoją su PIN kodu;
- Sign – atlieka privačiojo rakto paiešką saugioje atmintyje pagal profilį, atlieka pasirašymą pagal pasirinktą funkciją ir grąžiną parašą, užkoduotą Base64 formatu. Pasirašymo funkcija yra naudojama tokia, kokią naudoja *OpenVPN* vartotojo autentifikavimui. Pagal PKCS#11 standartą, ši funkcija yra vadinama *CKM_RSA_PKCS*.
- Finalise – atjungia vartotojo prieigą prie kortelės saugios atminties ir užveria sesiją.
- NFC Test – naudojama duomenų perdavimui tarp *NFCSec* programėlės ir ALK skaitytuvo, su kuriuo bendrauja *NFCAgent*. Taip pat naudojama informacijos greitaveikos tyrimui.



3.5 pav. Kuriamos *Android* programėlės *NFCSec* prototipo ekranvaizdis

3.2.4. Kriptografinės microSD kortelės komponentas

Programėlės testavimui į inicializuotą microSD kortelės saugią atmintį (3.6 paveikslas), *Windows* aplinkoje, yra įkeliamas privatusis raktas, gali būti poroje su sertifikatu. Tam yra panaudojama gamintojo pateikta programinė įranga. Norint įkelti privatų raktą, reikia žinoti vartotojo PIN kodą (3.7 paveikslas), be šio kodo operacijos su kortelės saugia atmintimi yra negalimos. Galimi įkeliamų objektų formatai yra „.p12“ ir „.pfx“.

3.6 pav. „GoTrust“ microSD kortelės duomenų valdymo įrankis

3.7 pav. Privačiojo rakto ir sertifikato įkėlimas „GoTrust“ įrankiu

Nors 3.8 paveiksle yra rodomi tik įkelti sertifikatai, jei įkeliamame faile yra privatusis raktas, jis irgi bus įkeliamas į programą, tik nebus pavaizduojamas šioje programoje. Norint panaudoti privatų

raktą mobiliajai *OpenVPN* autentifikacijai, yra būtina kartu turėti sertifikatą, pasirašytą patikimo sertifikatų centro. Tuo pačiu sertifikatų centru turi pasitikėti ir *OpenVPN* serveris.

3.8 pav. Įkeltas sertifikatas ir privatusis raktas į kortelės PKCS#11 atmintį

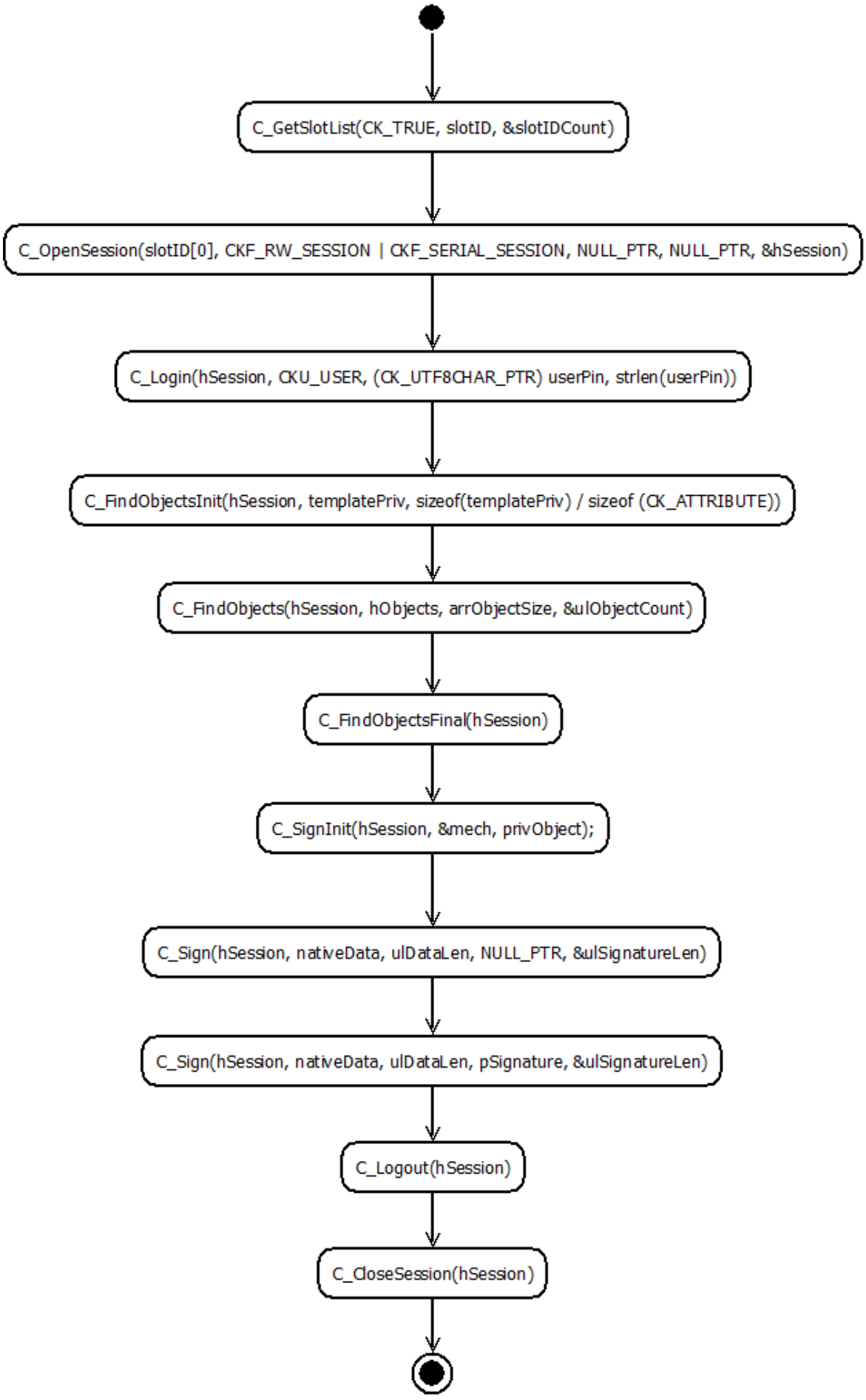
3.3. Maišos reikšmės pasirašymo procesas kriptoprocesoriuje

Maišos reikšmės pasirašymo procesas yra aprašytas sukurtoje *C* kalbos bibliotekoje ir yra pavaizduotas 3.9 paveiksle. Visos 3.9 paveiksle išvardintos funkcijos yra atliekamos PKCS#11 kriptoprocesoriuje eilės tvarka. Kintamieji, kurie aprašyti vien didžiosiomis raidėmis, yra standartiniai PKCS#11 kintamieji visada perduodami pavaizduotoms funkcijomis.

Maišos reikšmės pasirašymo proceso eiga:

1. *C_GetSlotList* gauna PKCS#11 prijungtų kortelių prievado identifikatorius per *slotID* masyvą ir jų kiekį per *slotIDCount*;
2. Prototipe naudojama tik vienas microSD kortelės prievadas, todėl atidarydami sesiją su kriptoprocesoriumi pateikiame pirmosios rastos PKCS#11 kortelės prievado identfikatorių per *slotID[0]*. *hSession* gaunamas sesijos identifikatorius;
3. Prisijungiame prie kortelės autentifikuodami vartotoją PIN kodu (*userPin*) kartu nurodydami jo įlgį;
4. Pradedame rakto objektų paiešką perduodami sesijos (*hSession*) ir privataus rakto paieškos šabloną (*templatePriv*) bei šablono dydį;
5. Su *C_FindObjects* funkcija yra gaunamas objektų masyvas (*hObjects*) ir jų kiekis (*ulObjectCount*);
6. Kriptoprocesoriui yra nurodoma, kad daugiau paieškos operacijų nebus atliekama perduodant *hSessions* funkcijai *C_FindObjectsFinal*;

7. Pasirašymo pradžiai funkcijai *C_SignInit* yra pateikiamas surasto privataus objekto identifikatorius *privObject* ir parašymo algoritmo parametras (*mech*). Šios sistemos identifikacijai naudojamas parašo algoritmas yra *CKM_RSA_PKCS*;
8. Pasirašymas vyksta perduodant *C_Sign* parametrus, tokius kaip maišos reikšmę pasirašymui (*nativeData*), jos ilgį (*ulDataLen*), parašo kintamąjį (*pSignature*) ir pirmuoju *C_Sign* kvietimu gautą parašo ilgį. Po šios funkcijos įvykdymo yra gautas maišos reikšmės parašas;
9. Vartotojas yra atjungiamas nuo PKCS#11 kriptoprocesoriaus (*C_Logout*) perduodant sesijos kintamojo reikšmę *hSession*;
10. Maišos pasirašymo procesas yra užbaigiamas uždarant sesiją su microSD kortele (*C_CloseSession*).



3.9 pav. Maišos reikšmės pasirašymas PKCS#11 kriptoprocesoriuje

3.4. Projektuojamos sistemos veikimas

Pagrindinis projektuojamos sistemos veikimas, sprendimas į aprašytą probleminę sritį, yra pavaizduotas 3.10 paveiksle esančioje diagramoje. Išskirti visi pagrindiniai komponentai, parodyta, kaip jie sąveikauja tarpusavyje ir kokius duomenis jie perduoda.

Komponentų tarpusavio sąveika yra išskirta į 4 pagrindinius etapus:

1. Duomenų perdavimas į kompiuterį – iš mobiliojo įrenginio *NFCSec* programėlės ALK ryšiu į vietiniame kompiuteryje įdiegtą programą *NFCAgent* perduodami šie duomenys:
    a. *OpenVPN* konfigūracinis failas – pagal konfigūracinį failą *OpenVPN* klientinė programa užmezga ryšį su nurodytu nuotoliniu serveriu kartu su nustatytais parametrais;
    b. Sertifikato failas – kartu su konfigūraciniu failu yra perduodamas su konfigūracijoje aprašytu ryšiu susijęs vartotojo sertifikatas, kuris yra pateikiamas *OpenVPN* serveriui.
2. TLS rankos paspaudimo (angl. *TLS hanshake*) pradžia – *OpenVPN* klientas pagal pirmame etape gautus duomenis ir parametrus inicijuoja TLS ryšio užmezgimo pradžią:
    a. Perduodami standartiniai TLS ryšio parametrai, tokie kaip versija, atsitiktinis skaičius, palaikomi šifrai, suspaudimas ir kita informacija;
    b. Serveris perduoda palaikomus ryšio parametrus. Serveris klientui taip pat perduoda palaikomus ryšio parametrus ir atsitiktinį skaičių;
    c. Klientas serveriui pateikia savo sertifikatą, kuris pirmame etape buvo perkeltas į vietinį kompiuterį;
3. Kliento autentifikavimas – šioje stadijoje vyksta kliento autentifikavimas ir patikrinama, ar pateiktas kliento sertifikatas iš tikrųjų priklauso ryšio iniciatoriui. Patikrinama, ar vartotojas pradėjęs TLS susijungimą, turi sertifikatą atitinkantį privatų raktą.
    a. Pagal 2 etape siųstus pranešimus yra sugeneruojama maišos reikšmė;
    b. Sugeneruota maišos reikšmė yra perduodama į *Android* mobilųjį telefoną, į *NFCSec* programėlę, nes kliento privačiojo rakto nėra kompiuteryje;
    c. *Android* telefone ar specialioje programėlės atmintyje taip pat nėra privataus kliento rakto. Kadangi privatusis raktas yra saugomas microSD kortelės, palaikančios PKCS#11 standartą, specialioje atmintyje, maišos reikšmė yra perduodama į microSD kortelės specialią atmintį pasirašymui. Prieš perduodant reikšmę, vartotojas yra autentifikuojamas į kortelę savo PIN kodu;
    d. Pagal profilį susietu vartotojo privačiu raktu yra pasirašoma maišos reikšmė;

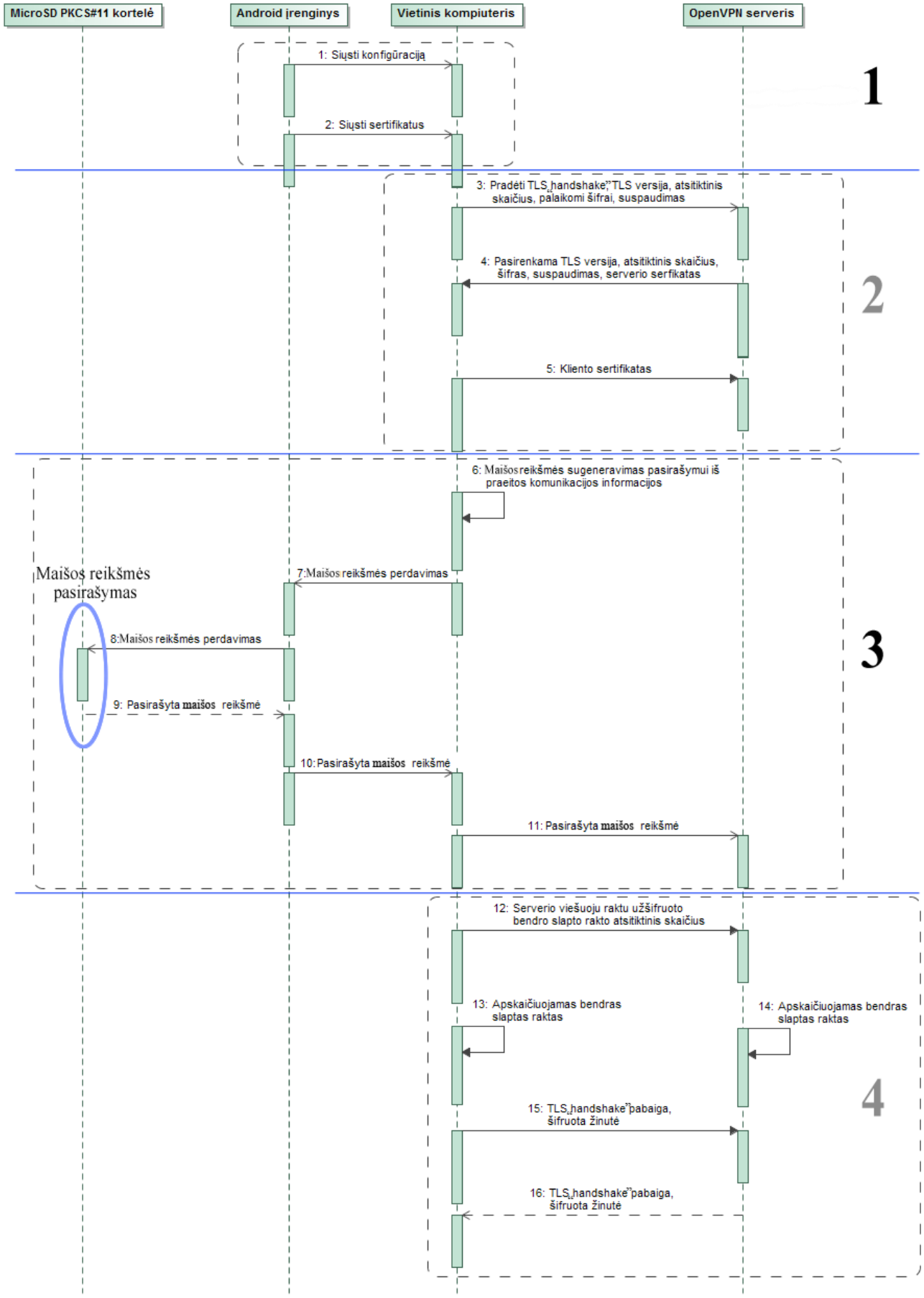
e. Pasirašyta reikšmė yra gražinama *NFCSec* programėlei tolesniam perdavimui;
f. Programėlė pasirašytą maišos reikšmę perduoda į vietinį kompiuterį ALK ryšiu agentinei programai *NFCAgent* apdoroti;
g. *NFCAgent* perduoda pasirašytą maišos reikšmę *OpenVPN* serveriui autentifikuoti;

4. Kai vartotojas yra sėkmingai autentifikuojamas, pasirašyta maišos reikšmė yra sėkmingai patikrinama vartotojo sertifikatu, įvyktas paskutinis autentifikavimo etapas.
   a. *OpenVPN* klientas serverio sertifikatu užšifruoja atsitiktinį skaičių ir nusiunčia serveriui;
   b. Klientas ir serveris apskaičiuoja bendrą slaptą raktą iš savo siųstų atsitiktinių skaičių ir kliento siųsto atsitiktinio skaičiaus;
   c. Klientas ir serveris bendru slaptu raktu šifruota žinute užbaigia TLS kanalo sudarymą.

Atlikus šiuos etapus *OpenVPN* serveris suteikia prieigą prie tinklo resursų, priklausomai pagal vartotojo konfigūraciją ir prieigos prie tinklo resursų teises.

Vartotojui programėlėje reikia įvesti į *NFCSec* programėlę PIN kodą, pasirinkti norimą susijungimo profilį ir priliesti telefoną prie vietinio kompiuterio ALK skaitytuvo. Kiti žingsniai atliekami automatiškai *NFCSec* ir *NFCAgent* programų pagalba.

Prototipo testavimo tikslais kai kurie etapai yra išskaidyti detaliau norint ištestuoti atskirus sistemos komponentus.



3.10 pav. Detalizuota duomenų srauto diagrama

3.5. Realizacijos išvados

Mobiliosios VPN identifikacijos komponentai buvo pasirinkti pagal užimamą rinkos dalį, bet ši sistema taip pat gali būti realizuojama ir su kai kuriais kitais komponentais, taip praplečiant siūlomos sistemos pritaikymo galimybes.

Android operacinėje sistemoje įdiegta programinė įranga taip pat gali būti realizuota *iOS* operacinėje sistemoje. Tam reikia papildomo priedo, kuris suteiktų prieigą prie microSD kortelės. Agentinė programa, veikianti vietiniame kompiuteryje su *Windows* operacine sistema, gali būti realizuota ir *Linux* operacinėje sistemoje. Kadangi *OpenVPN* serveryje yra naudojama standartinė konfigūracija ir naudojama standartinė paslauga, tai *OpenVPN* serverio paslauga gali veikti visose aplinkose, kurias ji numatytai palaiko.

Pasiūlytos mobiliosios identifikacijos sistema yra paremta galimybe atlikti identifikaciją pasirašant maišos reikšmę ir galimybe tai padaryti per 60 sekundžių laikotarpį. Pagrindiniai identifikacijos sistemos žingsniai yra perduoti informaciją iš mobiliojo telefono į kompiuterį VPN ryšio užmezgimui pradėti ir pasirašyti su konkrečia sesija susijusią maišos reikšmę.

4. SISTEMOS CHARAKTERISTIKŲ IR SAUGUMO TYRIMAS

4.1. Sistemos saugumas

Visas sistemos saugumas yra paremtas kiekvieno komponento saugumo įvertinimu ir komunikacijos protokolų saugumo įvertinimu. Kaip jautri informacija sistemoje yra saugoma bei naudojama [24].

Sistema yra sudaryta iš trijų pagrindinių komunikuojančių įrenginių:

- *Android* mobilusis telefonas – naudojamas *OpenVPN* kliento sertifikatų tarnybos ir kliento sertifikato laikymui. Privatusis raktas yra saugomas kaip objektas PKCS#11 funkcijų palaikymą turinčioje microSD kortelėje;
- Vietinis kompiuteris – įrenginys sistemoje, kuriam reikia sukurti VPN ryšį ir taip gauti prieigą prie nuotolinių tinklų ar resursų;
- Nuotolinis *OpenVPN* serveris – naudojamas suteikti prieigą prie nutolusių tinklų ar resursų. Gali būti įdiegtas vidiniame tinkle, neturinčiame prieigos prie interneto, arba gali būti pasiekiamas per internetą, užtikrinant klientui saugią prieigą prie tinklo resursų.

4.1.1. PKCS#11 microSD kortelės komponento sauga

PKCS#11 microSD kortelės kriptografinės funkcijos sistemoje yra pasiekiamos tik per gamintojo pateiktą *Android* biblioteką, kuri naudoja microSD skaitymo ir rašymo komandas, kad iškviestų specialias kriptoprocesoriaus funkcijas.

Su objektais, laikomais kortelėje, gali būti manipuliuojama tik tada, kai API autentifikuojasi į kortelę su iš anksto nustatytu PIN kodu. Jei nėra sėkmingai autentifikuojamasi, tai visos PKCS#11 operacijos su kortelės objektais (privačiais raktais, sertifikatais) yra uždraustos.

Pagrindinės funkcijos, naudojamos šiam sprendimui, yra RSA PKCS #1 v1.5 pasirašymo funkcija RSASP1 (*K*, *m*), kur įvesties parametrai yra [25]:

K – RSA privatusis raktas, kur *K* yra apibrėžtas viena iš šių formų:

- pora (*n*, *d*);
    - kur *n* – modulis (angl. *modulus*);
    - *d* – privatusis rodiklis (angl. *private exponent*);
- penketas (*p*, *q*, *dP*, *dQ*, *qInv*);
    - kur *p, q* – pirminiai modulio daugikliai (angl. *prime factors of the modulus*);
    - *e* – viešasis rodiklis (angl. *public exponent*);
    - *dP* – apskaičiuojama pagal formulę:

$$dP = \left(\frac{1}{e}\right) \bmod (p - 1) \qquad (1);$$

    - *dQ* – apskaičiuojamas pagal formulę:

$$dQ = \left(\frac{1}{e}\right) mod(q-1) \qquad (2);$$

- *qInv* – apskaičiuojamas pagal formulę:

$$qInv = \left(\frac{1}{q}\right) \bmod p \qquad (3).$$

m – žinutės maišos reikšmė, skaičius tarp 0 ir *n*-1

Gauta funkcijos reikšmė yra parašas *s* , skaičius tarp 0 ir *n*-1. Yra daroma prielaida, kad privatusis raktas K yra teisingas.

1. Jeigu žinutės maišos reikšmė *m* nėra tarp 0 ir *n*-1 tada žinutės maišos reikšmė yra per ilga ir negali būti pasirašyta

   1.1. Jeigu naudojama pirma *K* forma (*n, d*) skaičiuojama,

$$s = m^d \bmod n \qquad (4)$$

   1.2. Kitu atveju jei yra naudojama antra *K* forma (*p, q, dP, dQ, qInv*),

$$s_1 = m^{dP} \bmod p \qquad (5)$$

$$s_2 = m^{dQ} \bmod p \qquad (6)$$

$$h = qInv(s_1 - s_2) \bmod p \qquad (7)$$

$$s = s_2 + h \times q \qquad (8)$$

2. Rezultatas *s* – maišos reikšmės parašas [25]. s_1, s_2 ir *h* yra tarpinės skaičiavimo reikšmės.

RSA parašo ilgis priklauso nuo privačiojo rakto dydžio. Privačiojo rakto, kurio dydis yra 1024 bitai, parašo dydis yra 128 baitai, o rakto su 2048 bitais – 256 baitai.

Pasirašymo laikas buvo matuojamas su skirtingais raktų dydžiais, pasirašymą atliekant kriptoprocesoriuje. Pasirašymas, įskaitant vartotojo autentifikavimą į microSD kortelę ir kitos susijusios operacijos, užtruko apie 2 sekundes. Šioje kriptoprocesoriaus realizacijoje nebuvo pastebėtas laiko skirtumas pasirašant su skirtingo dydžio raktais.

4.1.2. *Android* mobiliojo telefono sauga

Sistemos sauga mobiliajame telefone remiasi operacinės sistemos sauga. Kadangi privatus rakto neįmanoma išgauti iš microSD kortelės, o nežinant PIN kodo nėra įmanoma panaudoti jokios PKCS#11 funkcijos su privačiojo rakto objektu.

Vienintelis būdas gauti microSD kortelės PIN kodą yra tuomet, jeigu vartotojas pasako šią informaciją arba kai yra stebimos mobiliojo telefono klaviatūros įvestys. Sudėtingas būdas sužinoti PIN kodą yra stebėti visas operacinės sistemos operacijas su microSD kortele.

Prevencija nuo PIN kodo informacijos sužinojimo iš technologinės pusės gali būti suvaldyta vartotojui atidžiai renkantis kokias programėles įrašyti į įrenginį ir kokių prieigos teisių programėlės reikalauja. Taip pat naudotis tik tokiu telefonu, kuriame vartotojui nėra suteiktos padidintos prieigos, prie operacinės sistemos, teisės. Telefonas, kurio vartotojas turi pilnas prieigos prie OS teises,

praplečia mobiliojo telefono atakos paviršių ir sukuria papildomų saugumo grėsmių su didesniu poveikiu dėl pilnos prieigos prie operacinės sistemos ir jos procesų.

4.1.3. ALK kanalo sauga

ALK ryšys yra įmanomas tik tarp dviejų, bet ne daugiau, įrenginių. Šiuo atveju ryšys yra tarp mobiliojo telefono ir *Windows* kompiuterio su ALK skaitytuvu. Skaitytuvas gali būti įmontuotas arba prijungtas kaip periferinis įrenginys. Nors teorinis ALK ryšio veikimo nuotolis yra iki 10 cm, bet, po atliktų testų su keliais mobiliaisiais telefonais ir skaitytuvais, buvo nustatyta, kad realiomis sąlygomis atstumas nėra didesnis nei 1,5 cm. Taip pat šis atstumas buvo pasiektas, kai telefonas buvo laikomas tam tikru kampu, lygiagrečiai skaitytuvui. Tai apriboja informacijos pavogimo tikimybę ar „man-in-the-middle“ ataką, nebent skaitytuvas, su kuriuo yra sukurtas ryšys, buvo fiziškai modifikuotas.

Jei ALK komunikacijos būtų pasiklausoma, šioje pasiūlytoje sistemoje nėra perduodama jokia jautri informacija, kurią būtų galima panaudoti. Perduodama komunikacijos galioja tik esamai sesijai. Duomenys perduodami ALK ryšiu:

- CA ir kliento sertifikatai – viešai prieinami;
- *OpenVPN* parametrai – failas, saugantis nuotolinio *OpenVPN* serverio IP adresą ir prievado numerį. Šią informaciją galima gauti atliekant prievadų skanavimą;
- Maišos reikšmės – susijusios tik su esamu TLS ryšiu ir nėra panaudojamos kitur. Net ir su esamu ryšiu, maišos reikšmės tampa negaliojančios po 60 sekundžių;
- Parašas – susijęs tik su esamu TLS ryšiu ir nėra panaudojamas kitur. Net ir su esamu ryšiu, parašas tampa negaliojantis po 60 sekundžių.

Duomenų perdavimo greitis buvo testuojamas su skirtingais mobiliaisiais telefonais ir ALK skaitytuvais. Kai telefonai buvo pakankamai arti, kad būtų sukuriamas ALK ryšys, perdavimo greitis visada atitikdavo standartą apibrėžiantį 424 Kbit/s greitaveiką. Pamatuotas ne valdymo informacijos perdavimo greitis buvo apie 40 KB/s. Duomenų kiekis, kurį reikia perduoti ALK ryšiu per VPN identifikacijos procesą, iš esmės priklauso nuo sertifikatų ir privačiojo rakto dydžių. *OpenVPN* konfigūracijos parametrų failas užima iki 1 KB vietos. Testavimo metu su rakto, kurio dydis yra 2048 bitai, bendras ALK duomenų persiuntimas užtruko mažiau nei 0,2 sekundės.

4.1.4. Vietinio kompiuterio sauga

Vietinis kompiuteris turi būti pakankamai saugus konkrečiam VPN ryšiui, kurį bandoma yra užmegzti. Jokia jautri informacija, susijusi su VPN ryšiu, nėra laikoma vietiniame kompiuteryje. Per VPN tunelio sukūrimo tarpsnį vietinis kompiuteris tvarko tik ALK ryšiu perduodamą informaciją. Bendru atveju, jeigu daugiau nei vienas vartotojas naudojasi konkrečiu vietiniu kompiuteriu kai kurie failai gali būti peržiūrimi kito vartotojo. Po sėkmingo VPN tunelio sukūrimo, kompiuteryje veikianti agentinė programa pašalina visą konfigūracinė informaciją. Jautri informacija, tokia kaip privatusis

raktas, niekada nėra perduodama į vietinį kompiuterį. Jeigu neautorizuotas vartotojas gaus prieigą prie kompiuterio, jis negalės pasiekti nuotolinio tinklo resursų per VPN ryšį.

Agentinė programa leidžia inicijuoti VPN susijungimą tik tada, kai vartotojas yra prisijungęs prie savo darbinės aplinkos. Nėra leidžiama inicijuoti VPN ryšį, jeigu vartotojas yra užrakinęs ekraną. Siekiant išvengti situacijos kai kitas žmogus bandys sukurti VPN susijungimą be tuo metu dirbančio vartotojo žinios, ekrane agentinė programa parodo perspėjimą, kai yra bandoma sukurti VPN ryšį. Tik viena *OpenVPN* sesija yra leidžiama konkrečiu laiko momentu.

4.2. Sistemos prototipo charakteristikos

Sistemos prototipas buvo testuojamas su keletu skirtingų *Android* mobiliųjų telefonų ir ALK skaitytuvų. Buvo matuojama esminių sistemos komponentų greitaveika ir bendra vienos VPN mobiliosios identifikacijos trukmė.

Buvo pastebėta, kad didžiausia dalis mobiliosios VPN identifikacijos laiko užtrunka maišos reikšmės pasirašymo procesui. Buvo stebėta, kad pasirašymo procesas kriptoprocesioriuje nepriklauso nuo pasirinkto *Android* telefono ir jo spartos. Šis testavimas buvo atliekamas *Android* operacinėje sistemoje iš vartotojo programėlių veikė tik viena suprojektuota ir realizuota *NFCSec* programėlė. Pasirašymo proceso trukmė microSD PKCS#11 kriptoprocesoriuje nepriklausė nuo privačiųjų raktų ilgio. Kadangi microSD kortelės prievado mobiliajame telefone sparta yra pakankamai greitai, todėl pasirašymo proceso trukmės vienodumas atsiranda dėl FIPS140-2 3 lygio [26] PKCS#11 kriptoprocesoriaus realizacijos. Stebėta pasirašymo proceso trukmė yra apie 2 sekundes.

ALK ryšio realus didžiausias veikimo nuotolis buvo matuojamas. Buvo pastebėta, kad didžiausias veikimo nuotolis yra iki 1,5 cm tarp *Android* įrenginių ir ALK skaitytuvų. Teorinis nuotolis yra iki 10 cm. Visų testavimo metu buvo stebėta vien tik didžiausia įrenginių palaikoma greitaveika, apie 40 KB/s.

Bendras sistemos identifikacijos laikas nuo vartotojo VPN profilio pasirinkimo ir PIN kodo įvedimo yra iki 3 sekundžių. Didžiausią dalį mobiliosios VPN identifikacijos laiko yra pasirašoma maišos reikšmė (apie 2 sekundes) ir perduodama informacija ALK ryšiu (iki 1 sekundės).

4.3. Siūlomos sistemos palyginimas su esamais sprendimais

Siūloma sistema pagal kokybinius parametrus yra palyginama 4.1 lentelėje su kitais esamais, panašiais, sprendimais. Kaip matome iš palyginimų, projektuojama sistema suteikia vartotojui aukščiausio lygio apsaugą ir perkeliamumą. Kaip matome pagrindiniai šio darbo problemos sprendimo tikslai buvo įvykdyti. Esami sprendimai nusileidžia siūlomai sistemai pagal perkeliamumo ir saugumo kriterijus. Siūlomos sistemos sprendimas reikalauja papildomos fizinės įrangos ir programų įdiegimo.

4.1 lentelė. VPN identifikacijos sistemų palyginimas

Palyginimo kriterijus	Siūlomas sprendimas	Identifikacija slaptažodžiu	Identifikacija privačiuoju raktu esančiu vietiniame kompiuteryje	Identifikacija telefonu per identifikacijos paslaugą
Maža žodyno ar bute-force atakos rizika	+	-	+	+
FIPS 140-2 lygio raktų apsauga	+	-	-	-
OS neturi tiesioginės prieiga prie identifikacijos rakto/slaptažodžio	+	-	-	+
Perkeliamumas	+	+	-	+
Nereikia prieigos prie interneto	+	+	+	-
Nereikia papildomos programinės įrangos prieigai prie raktų	-	+	+	-
Nereikia papildomos fizinės įrangos	-	+	+	+

4.4. Bendras saugumo įvertinimas

Pagrindinis pasiūlytos sistemos prototipo aspektas yra kaip apsaugoti privačiojo rakto prieigą ir galimybę naudoti kriptografines funkcijas su privačiojo rakto objektu. Kadangi išgauti privatų raktą iš microSD kortelės yra neįmanoma, todėl svarbu yra apsaugoti jį nuo panaudojimo su kriptografinėmis funkcijomis.

Standartinis būdas pasinaudoti kriptoprocesoriaus funkcijomis yra per biblioteką, kuri yra integruota į *Android* programėlę. Naudojantis šia biblioteka, vartotojo PIN kodas yra perduodamas į microSD kortelės kriptoprocesorių autentifikavimui.

Išlaikant bendrą mobiliojo telefono operacinės sistemos saugumą yra geriausias būdas išlaikyti apsaugotą prieigą prie kriptografinio procesoriaus. Be privilegijuotos prieigos prie operacinės sistemos „root“ vartotojo, jokia kita programa ar vartotojas negalės klausytis operacijų atliekamų su microSD kortele. Šis būdas taip pats apsaugo nuo neautorizuoto mobiliojo telefono klaviatūros pasiklausymo.

ALK turi keletą apibrėžtų standartų, kurie padengia ALK kanalo saugą. ECMA išleistas standartas ECMA-385 apibrėžia esamas ALK saugumo paslaugas ir NFC-SEC protokolą. ECMA-385 standartas buvo praplėstas pridedant ECDH ir AES palaikymą. Šie standartai naudoja rakto sutarimo mechanizmą, kuris yra paremtas ISO/IEC 11770-3 standartu.

Kita standartų organizacija („NFC Forum“) išleido LLCP specifikacijos versiją 1.3, kuri prideda neautentifikuoto, bet apsaugoto duomenų perdavimo galimybę, užtikrinant žinučių, kurios yra apsikeičiamos tarp įrenginių, slaptumą ir konfidencialumą.

Nors šie standartai jau egzistuoja, kad užtikrintų ALK kanalo saugą tarp dviejų vienodo lygio įrenginių, bet jie nėra realizuoti mobiliuose įrenginiuose. Tai susiję su faktu, kad ALK ryšio atstumas yra labai mažas, todėl nėra tikėtina, kad kas nors sugebės pasinaudoti ALK duomenų perdavimu. Dėl šios priežasties, ALK saugumo standartai dar nėra plačiai naudojami. Pasiūlytame mobiliosios VPN identifikacijos sprendime jautri informacija nėra perduodama ALK ryšiu.

4.5. Eksperimento išvados

Eksperimento metu gautos sistemos charakteristikos yra pakankamos sistemos realizavimui ir panaudojimui realioje aplinkoje. ALK ryšio stebėta greitaveika, kuri yra apie 40 KB/s, yra pakankama tokio tipo sistemai, nes perduodamas informacijos kiekis yra nedidelis. Taip pat nereikia iš anksto suporuoti įrenginių norint pasinaudoti mobiliąja VPN identifikacija. Siūlomos sistemos identifikacijos greitaveika priklauso nuo autentifikavimo raktų ir sertifikatų dydžio, nes dėl nedidelės ALK greitaveikos vienos identifikacijos metu bendras informacijos perdavimas šiuo komunikacijos kanalu gali užtrukti iki 1 sekundės. Identifikacijos maišos reikšmės pasirašymo procesas kriptografiniame procesoriuje užtrunka apie 2 sekundes. Bendras vartotojo autentifikavimo laikas nuo identifikacijos pradžios iki pabaigos trunka iki 3 sekundžių. Skirtingas privačiųjų raktų ilgis PKCS#11 microSD kriptoprocesoriuje neįtakojo pasirašymo proceso trukmės.

Bendras mobiliosios VPN identifikacijos sistemos saugumas yra užtikrinamas apsaugant vartotojo identifikacijos raktus ir prieigą prie jų. Kadangi šie raktai yra laikomi FIPS140-2 3 saugumo lygį užtikrinančioje laikmenoje, todėl užtenka užtikrinti, kad būtų apsaugota prieiga prie privačiųjų raktų panaudojimo su kriptografinėmis funkcijomis.

Pasiūlytas VPN identifikacijos sprendimas suteikia saugumo ir perkeliamumo naudą su minimalia kaina. Kriptoprocesoriumi paremtos microSD kortelės pritaikymas apsaugo privačius raktus ir visas susijusias kriptografines operacijas.

Jeigu mobilusis telefonas, kriptografinė microSD kortelė būtų pamesti ar pavogti, visi privačių raktų objektai ir susijusios funkcijos būtų nepasiekiamos nežinant PIN kodo. Žmogus, norintis gauti prieigą, galėtų bandyti spėlioti PIN kodą, bet jis yra pakankamai ilgas, kad suteiktų pakankamai laiko raktų savininkui anuliuoti visas su kriptografiniais objektais susijusias prieigas ir atšaukti sertifikatus, susijusius su prarastais raktais. Kortelės su kriptografinėmis funkcijomis nesiskiria nuo standartinių kortelių vizualiai ar iš operacinės sistemos perspektyvos. Kriptoprocesorių turinčios microSD kortelės, su vienoda atmintimi, kaina yra tik nežymiai didesnė.

Dauguma sąveikos per VPN identifikacijos procesą su pasiūlytu sistemos prototipu yra automatizuota, todėl reikalauja mažai vartotojo įsiterpimo, tuo pačiu suteikdamas patobulintą saugumą ir perkeliamumą.

5. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

1. Atlikus esamų VPN identifikacijos sprendimų analizę buvo pastebėta, kad nei vienas iš esamų sprendimų neužtikrina VPN identifikacijos mobilumo ir aukščiausio lygio identifikacijos raktų apsaugos. Didžiausią saugumą, šiuo metu, gali užtikrinti fiziniai saugumo įrenginiai. Tačiau kai kurios fizinės įrangos priežiūra, palaikymas reikalauja papildomų žmogiškųjų resursų ir finansinių išlaidų.
2. Kriptografiškai stiprios, mobiliosios VPN identifikacijos sistemos prototipo sukūrimui buvo pasirinkta fizinis saugumo įrenginys, PKCS#11 standarto microSD kortelė, nereikalaujantis priežiūros ir papildomų žmogiškųjų resursų palaikymo. Šioje kriptografinėje microSD kortelėje yra saugomi vartotojo identifikacijos raktai. Mobiliajai VPN identifikacijos sistemai perkeliamumą suteikia tai, kad PKCS#11 microSD yra įdedama į *Android* mobilųjį telefoną ir naudojama kartu su telefone įrašyta *NFCSec* programėle. Raktų objektai niekada nepalieka PKCS#11 kortelės atminties ir visos operacijos su jais yra atliekamos microSD kortelėje, jei vartotojas žino PIN kodą. *Android* mobilusis telefonas užtikrina sistemos perkeliamumą, o PKCS#11 microSD – FIPS140-2 3 lygio saugumą.
3. Mobiliąja VPN identifikacijos sistema palyginus su esamomis matome, kad siūlomos sistemos pranašumai yra aukščiausio lygio identifikacijos raktų apsauga ir mobilumas. Siūlomos sistemos veikimui taip pat nėra būtina prieiga prie interneto, jei VPN susijungimas vyksta per vietinius tinklus. Norint pasinaudoti šiuo sprendimu reikia į mobilųjį telefoną ir vietinį kompiuterį įdiegti programinę įranga, kuri pasirūpina VPN identifikacijos procesu.
4. Atlikus projektuojamos sistemos saugumo buvo nustatyta, kad pagrindinis pasiūlytos sistemos prototipo saugumo tikslas yra apsaugoti privačiojo rakto prieigą ir galimybę naudoti kriptografines funkcijas su privačiojo rakto objektu. Kadangi išgauti privatų raktą iš microSD kortelės yra neįmanoma, todėl svarbu yra apsaugoti jį nuo panaudojimo su kriptografinėmis funkcijomis. Išlaikant bendrą mobiliojo telefono operacinės sistemos saugumą yra geriausias būdas išlaikyti apsaugotą prieigą prie kriptografinio procesoriaus.
5. Atlikus sistemos prototipo charakteristikų tyrimą buvo nustatyta, kad siūlomos sistemos bendra identifikavimo trukmė trunka iki 3 sekundžių. Daugiausiai laiko užtrunka maišos reikšmės pasirašymas su identifikacijos raktu.
6. Atlikus RSA kriptosistemos funkcijų analizę buvo nustatyta, kad siūlomos sistemos realizavimui galima panaudoti tik RSA parašo funkciją. RSA raktų generavimas vyksta už siūlomos sistemos ribų.
7. Atlikus PKCS#11 standarto panaudojimo analizę buvo nustatyta, kad pagrindinė sistemoje panaudojimo funkcija, kuri vykdo reikiamą RSA pasirašymą, pagal PKCS#11 standartą yra apibrėžiama *CKM_RSA_PKCS*.

6. LITERATŪRA

1. Taneski, V., Hericko, M., Brumen, B. „Password security — No change in 35 years?“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6859779
2. Wood, D., Stoss, V., Chan-Lizardo, L., Papacostas, G.S., Stinson, M.E. „Virtual private networks“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=10290
3. RSA Laboratories. „PKCS #11 Base Functionality v2.30: Cryptoki“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-30/pkcs-11v2-30b-d6.pdf
4. Muzzi, F.A.G., Chiaramonte, R.B., Moreno, E.D. „The Hardware-based PKCS#11 Standard using the RSA Algorithm“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5256823
5. The Internet Society. Internet X.509 Public Key Infrastructure Certificate and CRL Profile [interaktyvus]. [Žiūrėta 2015-04-10]. Prieiga per: https://tools.ietf.org/rfc/rfc2459
6. Adi, W., Ouertani, N., Hanoun, A., Soudan, B. „Deploying FPGA self-configurable cell structure for micro crypto-functions“ [interaktyvus]. [Žiūrėta 2014-12-16]. Prieiga per: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5202368
7. NFC Forum. „NFC in Action“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: http://nfc-forum.org/what-is-nfc/nfc-in-action/
8. Choon Hoong Ding, Sarana Nutanong, and Rajkumar Buyya. Peer-to-Peer Networks for Content Sharing [interaktyvus]. [Žiūrėta 2016-03-17]. Prieiga per: http://www.cloudbus.org/papers/P2PbasedContentSharing.pdf
9. C. Patauner, H. Witschnig, D. Rinner, A. Maier, E. Merlin, E. Leitgeb. „High Speed RFID/NFC at the Frequency of 13.56 MHz“ [interaktyvus]. [Žiūrėta 2015-06-16]. Prieiga per: http://www.eurasip.org/Proceedings/Ext/RFID2007/pdf/s1p4.pdf
10. Hsu-Chen Cheng, Wen-Wei Liao, Tian-Yow Chi, Siao-Yun Wei. „A secure and practical key management mechanism for NFC read-write mode“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5745999
11. Devharsh Trivedi. „Near Field Communication“ [interaktyvus]. [Žiūrėta 2015-08-22]. Prieiga per: https://www.researchgate.net/figure/277131825_fig3_Figure-13-Comparison-of-NFC-RFID-Infrared-and-Bluetooth
12. Du Meng. „Implementation of a host-to-host VPN based on UDP tunnel and OpenVPN Tap interface in Java and its performance analysis“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6554047
13. The Internet Society. „Securing L2TP using IPsec“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: https://tools.ietf.org/html/rfc3193

14. IVPN. „PPTP vs L2TP/IPSec vs OpenVPN“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: https://www.ivpn.net/pptp-vs-l2tp-vs-openvpn
15. The Internet Society. „PKCS #1: RSA Encryption“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: https://tools.ietf.org/html/rfc2313
16. EMC Corporation. „PKCS #1 v2.2: RSA Cryptography Standard“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: http://www.emc.com/collateral/white-papers/h11300-pkcs-1v2-2-rsa-cryptography-standard-wp.pdf
17. Perlman, R. „An overview of PKI trust models“ [interaktyvus]. [Žiūrėta 2014-12-15]. Prieiga per: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=806987
18. Statista. „Distribution of Android operating systems used by Android phone owners in May 2016, by platform version“ [interaktyvus]. [Žiūrėta 2015-06-16]. Prieiga per: http://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/
19. OASIS Open. „CKM_RSA_PKCS_FIPS“ [interaktyvus]. [Žiūrėta 2015-06-16]. Prieiga per: https://www.oasis-open.org/committees/download.php/50390/CKM_RSA_PKCS_FIPS_186_4_v03.pdf
20. The Internet Society. „The Base16, Base32, and Base64 Data Encodings“ [interaktyvus]. [Žiūrėta 2015-06-16]. Prieiga per: https://tools.ietf.org/html/rfc4648
21. OpenVPN Technologies Inc.. „Security Overview“ [interaktyvus]. [Žiūrėta 2015-06-16]. Prieiga per: https://openvpn.net/index.php/open-source/documentation/security-overview.html
22. Microsoft. „What is TLS/SSL?“ [interaktyvus]. [Žiūrėta 2015-06-16]. Prieiga per: https://technet.microsoft.com/en-us/library/cc784450%28v=ws.10%29.aspx
23. GO-Trust. „GO-Trust Launches the First Portable, FIPS 140-2 Level 3 and FIPS 201 Security for iPhones and iPads“ [interaktyvus]. [Žiūrėta 2015-06-16]. Prieiga per: http://www.go-trust.com/go-trust-launches-the-first-portable-fips-140-2-level-3-and-fips-201-security-for-iphones-and-ipads/
24. Ala A. Abdulrazeg., Norita Md Norwawi., Nurlida Basir. „Security Measurement Based On GQM To Improve Application Security During Requirements Stage“ [interaktyvus]. [Žiūrėta 2016-05-10]. Prieiga per: http://sdiwc.us/digitlib/journal_paper.php?paper=00000310.pdf
25. The Internet Society. „PKCS #1: RSA Cryptography Specifications Version 2.0“ [interaktyvus]. [Žiūrėta 2015-06-16]. Prieiga per: https://tools.ietf.org/html/rfc2437
26. National Institute of Standards and Technology. SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES [interaktyvus]. [Žiūrėta 2016-03-17]. Prieiga per: http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

7. PRIEDAI

7.1. priedas. Straipsnis

Straipsnis A. Nauckūnas, E. Sakalauskas, „Mobile Identification to VPN System Using PKCS11 Crypto Processor and NFC“ buvo pateiktas ir pristatytas „Information Society and University studies 2016” (IVUS 2016) konferencijoje, kuri vyko 2016 m. balandžio 28 d., Kaune. Straipsnis buvo atspausdintas konferencijos leidinyje 72-76 psl.

Mobile identification to VPN system using crypto processor and NFC

A. Nauckūnas, E. Sakalauskas
Kaunas University of Technology
Faculty of Mathematics and natural sciences
Kaunas, Lithuania
arvydas.nauckunas@gmail.com

***Abstract* — Mobile VPN identification system is presented that uses microSD based Crypto processor and NFC. Whole system includes three main communicating devices: remote VPN server, local computer and mobile phone with microSD based crypto processor. The overall identification system's security is estimated.**

***Keywords* — VPN, NFC, OpenVPN, Cryptoki, PKI, Crypto processor.**

I. Introduction

Mobile phone has become part of every person's life. Now mobile phones can replace many devices while in the same time providing more features to the users. Phone now acts as a photo camera, calculator, video camera and other devices.

In today's age of information there are a lot of different computerized systems where users have to remember their user names and passwords in order to work with and maintain those systems. With increasing amount of information and its sensitivity and a need to comply with security requirements, mobile identification systems are becoming more relevant to every person. Passwords become even longer to comply with growing security standards and they need to be frequently changed.

Public-private key authentication method is being used increasingly. This provides improved security compared to password authentication due to the fact that a key length is usually much longer than password and does not represent meaningful words or patterns therefore it is impossible to guess or brute-force with currently known hardware. Unfortunately using PKI brings its own challenges such as safe key storage and portability combined with ease of use.

Computers that are communicating with the private networks are usually connected to these networks via an unsecure channel. It could be either the Internet or a special Internet Service provider link. In order to reach these networks in a secure manner VPN connections are established. This ensures data encryption and integrity. To establish this secure channel password authentication or more secure private-public key authentication is used. This brings forth challenges how to use mobile phone as a private key storage and how to establish VPN for local computers.

II. Proposed Mobile VPN identification Solution

One of the most popular ways to handle a private key is to store it on a local computer. But this approach restricts usage with only one computer. Portability problem is usually addressed by carrying a private key in USB thumb drive. As an additional security measure password can be set on a private key file to prevent its usage without knowing the password. This protection usually can be cracked by using dictionary or brute force attacks.

Introducing private key to the computer systems where it can be directly read adds risk of it being stolen.

The proposed solution is a way to solve private key portability and usage for VPN identification security issues.

A. Solution components

Components for the solution were selected based on how widely spread the technology is and security benefit it brings.

- NFC enabled smartphone with a microSD card slot and Android 4.0.3 or higher. It was selected because of a wide spread and Android version 4.0.1 and lower has already faced NFC related security concerns. Also currently Android takes most of the market share of all smartphone operating systems. Where version 4.0.3 and higher make up more than 90% of Android OS;
- microSD based Crypto processor which complies with PKCS#11 standard. Hardware solution for private key storage was selected since it protects private key without an ability to extract it and only provides one entry point which is PIN protected for crypto function operations. Selected microSD cards also provides storage as standard cards without crypto processor;
- NFC reader built-in or attached to a local computer. NFC short range communication minimizes risk of various attacks such man in the middle. Does not require special setup before usage;

- Android application that handles NFC communication and interaction with microSD card functions. Through the application data is passed to crypto functions on microSD card. It also acts as VPN profile storage that contain VPN parameters and handles NFC communication;
- Agent application on a local Windows computer that handles communication between OpenVPN client and NFC;
- Remote OpenVPN server [1] as part of complete solution provides access to secured network resources over VPN tunnel based on user identification.

This solution can also be used with other smartphones as well if there is vendor provided library for crypto microSD card for particular OS.

Mobile devices with iOS do not have a slot for external microSD card. Therefore they are unable to make best use of this solution. Tested crypto microSD card vendor enables to use this proposed solution by connecting microSD card reader via iOS device lightning port.

B. VPN identification

Mobile identification in this solution taps into existing TLS authentication protocol and OpenVPN "management" and "management-external-key" functions [2].

Main VPN identification steps in Fig. 1 of the proposed solution:

1. Non-sensitive data transfer from mobile phone to a local computer;
2. Initial TLS handshake until TLS authentication;
3. SHA1 (160 bits) and MD5 (128 bits) hash calculation according to TLS authentication protocol. Transferring a combined hash with length of 288 bits to Android mobile phone application for hash signing with a private key within crypto processor based microSD card;
4. Finalizing TLS handshake.

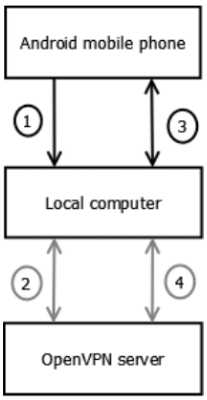


Fig. 1. VPN identification steps of the proposed solution

Identification steps 2 and 4 in Fig. 1 are handled by OpenVPN therefore they will not be discussed further.

User initiates identification by selecting profile in Android application which relates OpenVPN configuration parameters, CA and client certificates and identifier for a private key that is stored on PKCS#11 microSD. After that user PIN code has to be entered for interaction with objects in microSD. If code is correct then user touches mobile phone to NFC reader. From this point Android application and agent service installed on a local computer handles identification process.

Android application sends VPN parameters, CA and client certificates to a local computer via NFC protocol. Agent service on a local computer employs received information to initiate OpenVPN TLS connection. This covers identification step 1.

During Step 3 according to TLS protocol SHA1 [3] (160 bits) and MD5 [4] (128 bits) hash values of all previous communication are calculated. So the total value of 288 bits has to be signed within 60 seconds with a private key corresponding to previously provided client certificate. Agent program sends this value to Android application which passes it to microSD along with a private key identifier for signing operation. Signature is then sent back to agent program via NFC and passed on to OpenVPN server for authentication.

If hash value was signed with correct private key and signature was submitted to OpenVPN server within 60 seconds of TLS initialization then authentication is successful and VPN connection gets established.

Once VPN connection is established mobile phone can be removed from NFC reader since it is used only to provide configuration and user authentication at initial stages. There is no further interaction with mobile or crypto microSD card after hash signature is sent to a local computer.

In case network connection is lost and VPN tunnel disconnects then mobile identification steps have to be repeated in order to establish new VPN connection.

III. SYSTEM SECURITY

Whole system security is based on how each component is secured and where sensitive data is located and used.

System is built from three main communicating devices:

- Android mobile phone – used as OpenVPN client CA and client certificate holder. Private key is stored as an object on microSD card that supports PKCS#11;
- Local computer – intended OpenVPN client that will establish VPN connection and gain access to remote resources;
- Remote OpenVPN server – used to provide access to remote networks or resources. It can be located on internal network without any access from Internet to secure private network resources or connection can be established over Internet.

A. PKCS#11 microSD card

PKCS#11 microSD card cryptographic functions in the system are only reachable through provided Android library which uses microSD read and write commands to invoke special crypto processor functions.

Objects contained in the card can only be manipulated with when API authenticates against the card with initially set PIN code. Without it any PKCS#11 operations with the card objects (private keys, certificates) are prohibited.

Main function used for this solution is RSA PKCS #1 v1.5 signature RSASP1 (K, m), where input parameters are as follows [5]:

K - RSA private key, where K has one of the following forms:

- pair (n, d)
- quintuple (p, q, dP, dQ, $qInv$)

m - message representative, an integer between 0 and n-1

Output is a signature s represented by an integer between 0 and n-1. Assumption of the calculation is that private key K is valid.

1. If the message representative m is not between 0 and n-1 then message is too long and cannot be signed.

2. If the first form (n, d) of K is used:

2.1 Let $s = m^d \bmod n$.

Else, if the second form (p, q, dP, dQ, $qInv$) of K is used:

2.2 Let $s\_1 = m^{dP} \bmod p$.

2.3 Let $s\_2 = m^{dQ} \bmod q$.

2.4 Let $h = qInv\ (\ s\_1 - s\_2\ ) \bmod p$.

2.5 Let $s = s\_2 + hq$.

3. Output s.

RSA signature length depends on a private key size. For private key which size is 1024 bits the signature size is 128 bytes and for 2048 bit – 256 bytes.

Timings were measured when signing with different key sizes in crypto processor. Signing, including user authentication against microSD card and other related operations, took around 2 seconds in total. In this particular crypto processor realization there was no difference in timings when signing with different key sizes.

B. Android mobile phone

System security on mobile phone relies on operating systems security. Since private key is not retrievable from microSD card and without knowledge of the PIN code it is not possible to use any PKCS#11 functions with private key object.

The only way PIN of microSD card can be obtained is if user gives this information to someone or mobile phone keyboard button presses are being monitored. More difficult approach to acquiring PIN code would be to monitor all system calls sent to microSD card.

Prevention from technological aspect of stealing PIN code can be mitigated by carefully selecting applications that mobile phone user is installing and what permissions they require. As well as keep mobile phone running without phone user acquiring elevated user privileges. Getting root permissions on mobile phone widens attack surface and introduces more possible security threats with bigger impact due to full access to operating system and its processes.

C. NFC channel

NFC communication is possible only between two devices. In this case it is between mobile phone and Windows computer with NFC reader. Reader can be built-in or connected as peripheral. Although theoretical working distance of NFC is less than 10cm after number of tests with mobile phones and several NFC readers the actual measured distance is no more than 1.5cm. And that can only be achieved if mobile phone is held at certain angle against NFC reader. This distance limits possible information theft or man-in-the-middle attacks unless reader itself has been physically tampered with.

Even if NFC communication would be eavesdropped in this proposed solution NFC channel handles only data that are not sensitive and cannot be used for anything except for currently ongoing authentication.

Data transferred via NFC is:

- CA and Client certificates – can be publicly viewed;
- OpenVPN parameters – file containing remote OpenVPN server IP address and port number. This information can be obtained by doing port scan;
- Hash values – related only to current TLS connection and cannot be used elsewhere. Even with current connection they become invalid after 60 seconds;
- Signature – related only to current TLS connection and cannot be used elsewhere. Even with current connection they become invalid after 60 seconds.

Data transfer rates were tested with different phones and NFC readers. When phones were in close enough proximity to establish NFC communication the data rates were always according to standard specifying data rate of 424 kbit/s. Actual transfer speed measured was around 40KB/s. Amount of data that needs to be transferred during VPN identification mainly depends on certificate and signature sizes. OpenVPN configuration parameters used are up to size of 1KB. With tested key length of 2048 bits whole NFC data transfer took less than 0.2 seconds.

D. Local computer

A local computer has to be secure enough for particular VPN connections that are to be established. No sensitive data related to VPN connection is stored on a local computer. During VPN tunnel setup phase local computer handles only

information that is transferred via NFC. In general if more than one user is using local computer some files can be inspected by another user. After successful establishment of VPN tunnel all configuration data are removed by running agent program. Sensitive data such as a private key is never present on local computer. If unauthorized user gets access to computer he will not be able to reach remote network resources over VPN.

Agent program only allows initiate VPN connection when user is logged in with his user profile. It is not possible to establish connection if user is in lock screen. To avoid someone else establishing VPN connection without knowledge of current user working with the computer agent service shows notification when VPN connection is being established. Only single OpenVPN connection instance is allowed at one time.

IV. Overall security analysis

Main focus of proposed system security is a way how a private key is accessed and ability to use it with cryptographic functions. Retrieval of a private key object from microSD card is not possible therefore it is important to prevent its usage with cryptographic functions.

Standard way to use cryptographic processor functions is via library built into Android application. Through that library user presented PIN code is passed on to microSD card crypto processor for authentication.

Maintaining overall operating systems security is best way to keep access to crypto processor secured. Without privileged access to operating systems privileged "root" user no other application or person will be able to listen system calls to microSD card. This also prevents from unauthorized inspection of keyboard input.

NFC has several defined standards that cover NFC channel security. ECMA [6] has released standard ECMA-385 [7] that defines available NFC security services and protocol NFC-SEC. While extending protocol with ECMA-386 [8] adding ECDH and AES support. These standards use key agreement mechanism based on ISO/IEC 11770-3 [9].

Another standardization body NFC Forum has released version 1.3 of the LLCP specification [10] that adds an unauthenticated secure data transport option to ensure privacy and confidentiality of messages exchanged between peer devices.

While these standards exist to ensure NFC channel security between two peer devices they are not yet implemented on mobile phones. This relates to the fact that due to very short NFC communication range it is unlikely that someone will be able to abuse NFC data transfers therefore NFC security standards are not yet widely implemented. In proposed solution no sensitive data are being transferred.

V. Conclusion

The proposed VPN identification solution brings security and portability benefits with minimal cost. Adoption of crypto processor based microSD card secures private keys and all related cryptographic operations.

Even if mobile phone or cryptographic card would be lost or stolen a private key object and related functions would not be accessible without knowledge of the PIN code. One could try to brute force PIN code but it is long enough to give enough time for the owner to invalidate all access and revoke certificates related to the stolen keys. Cryptographic cards do not differ from standard cards in visual appearance or from operating system perspective. Cost of same size microSD cards with crypto processor are only marginally higher.

Most of the interaction during VPN identification process with the proposed solution has been automated therefore requiring less user interaction while providing improved security and portability benefits.

The proposed solution is used to authenticate local computer against a remote OpenVPN server and create VPN connection with the help of smartphone. Similar solutions that use crypto microSD card authenticate not a local computer but smartphone itself in order to create VPN tunnel between phone and remote VPN server.

VI. Abbreviations and Acronyms

NFC – near field communication;

API – application programming interface;

PIN – private identification number;

PKCS#11 - standard defines a platform-independent API to cryptographic tokens;

TLS – transport layer security;

SHA1 – secure hashing algorithm version 1;

USB – universal serial bus;

VPN – virtual private network;

IP address – numerical label assigned to each device participating in a computer network that uses the Internet Protocol for communication;

port – endpoint of communication in an operating system;

AES – Advanced Encryption Standard, symmetric encryption protocol;

ECDH – Elliptic curve Diffie–Hellman key agreement protocol;

CA – certificate authority.

References

[1] OpenVPN [Online]. Available: https://openvpn.net/

[2] OpenVPN. Management interface [Online]. Available: https://openvpn.net/index.php/open-source/documentation/miscellaneous/79-management-interface.html

[3] US Secure Hash Algorithm 1 (SHA1) [Online]. Available: https://tools.ietf.org/html/rfc3174

[4] The MD5 Message-Digest Algorithm [Online]. Available: http://tools.ietf.org/rfc/rfc1321

[5] PKCS #1: RSA Cryptography Specifications Version 2.0. RSASP1 [Online]. Available: https://tools.ietf.org/html/rfc2437#section-5.2.1

[6] ECMA International [Online]. Available: http://www.ecma-international.org/

[7] NFC-SEC: NFCIP-1 Security Services and Protocol [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-385.pdf

[8] NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-386.pdf

[9] ISO/IEC 11770-3:2015. Information technology -- Security techniques -- Key management -- Part 3: Mechanisms using asymmetric techniques [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=60237

[10] NFC Forum Technical Specifications [Online]. Available: http://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/