Kriptiranje datoteka zasnovano na lokaciji

Dmitrović, Matej

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni preddiplomski studij računarstva

ENKRIPCIJA DATOTEKA ZASNOVANA NA LOKACIJI

Završni rad

Matej Dmitrović

Osijek, 2019.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 19.09.2019.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Matej Dmitrović
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3907, 25.09.2018.
OIB studenta:	67929836880
Mentor:	Doc.dr.sc. Zdravko Krpić
Sumentor:	Dr.sc. Bruno Zorić
Sumentor iz tvrtke:	
Naslov završnog rada:	Kriptiranje datoteka zasnovano na lokaciji
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene mentora:	19.09.2019.
Datum potvrde ocjene Odbora:	25.09.2019.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT

IZJAVA O ORIGINALNOSTI RADA

Osijek, 25.09.2019.

Ime i prezime studenta:	Matej Dmitrović
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3907, 25.09.2018.
Ephorus podudaranje [%]:	4

Ovom izjavom izjavljujem da je rad pod nazivom**: Kriptiranje datoteka zasnovano na lokaciji**

izrađen pod vodstvom mentora Doc.dr.sc. Zdravko Krpić

i sumentora Dr.sc. Bruno Zorić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD

Danas većina ljudi modernog svijeta uza sebe ima mobilni uređaj. Bio taj uređaj za povezivanje s rodbinom, pravljenje podsjetnika ili fotografiranje scena, on igra važnu ulogu u svakodnevnom ljudskom životu. Međutim, uvijek postoji mogućnost od napada i krađe uređaja ili njegovih podataka. Zbog toga, osiguravanje tih podataka – slika, dokumenata, video-zapisa i slično - je od velike važnosti. Do danas su osmišljeni mnogobrojni načini kako bi se očuvali podaci: od obične šifre do prepoznavanja ljudskog lica. Lokacija mobilnog uređaja također može biti korištena u takve svrhe. Parametri lokacije – geografska širina, dužina i visina – mogu služiti prilikom stvaranja ključa za zaštitu podataka. Danas već postoje implementacije koje koriste takvu vrstu zaštite. "Geo-Encryption Lite" je aplikacija za prijenos i zaštitu podataka opisana u [1], koja pomoću lokacije uređaja i korisničke lozinke dozvoljava otvaranje poslanog podatka samo na određenoj lokaciji. U [2] je detaljan opis geo-lokacijskog sustava zaštite koji uz lokaciju koristi i briznu uređaja. Također je razrađen postupak sekvencijalno zaštite podataka preko nekoliko lokacije. Kako bi korisnik bio u mogućnosti otvoriti zaštićenu datoteku, mora u pravilnom redoslijedu proći kroz određen broj lokacija.

U ovom radu razmatrat će se upotreba lokacije mobilnog uređaja pri zaštiti podataka. U drugom poglavlju se opisuje teorijska osnova kriptografije i korištenje lokacije u zaštiti podataka. Implementacija rješenja jednog takvog sustava je opisana u trećem poglavlju, dok se u četvrtom poglavlju daju zaključci rada i smjernice za buduće radove.

1.1. Zadatak rada

U teorijskom dijelu rada potrebno je istražiti mogućnost ugrađivanja i primjene zemljopisne lokacije korisnika kod kriptiranja i dekriptiranja datoteka. U praktičnom dijelu rada potrebno je ostvariti programsko rješenje za Android sustav koje omogućuje kriptiranje i dekriptiranje datoteka uz korištenje koncepata opisanih u teorijskom dijelu rada.

2. KRIPTOGRAFIJA

Prema [3], kriptografija je znanstvena disciplina koja se bavi prijenosom poruke od pošiljatelja do primatelja na takav način da ne postoji treća osoba koja može protumačiti tu poruku. Svaki oblik kriptografije ima svoje prednosti i nedostatke – i kod svakog oblika je moguće protumačiti izvornu poruku, ali kriptografija se temelji na tome da je tu poruku vremenski i resursno zahtjevno za dešifrirati.

Potrebno je razmotriti nekolicinu osnovnih pojmova korištenih u radu. Kriptiranje (ili enkripcija, engl. *encryption*) podatka označava transformiranje u šifrirani podatak koji se ne može lako odgonetnuti, dok dekriptiranje (engl. *decryption)* podataka označava pretvorbu šifriranog podatka u izvorni podatak kojeg je moguće protumačiti. Šifrirana poruka ili podatak (još se naziva šifra, engl. *cipher, ciphertext*) je krajnji proizvod kriptiranja.

2.1. Razvoj kriptografije

Najraniji kriptografski postupci počinju već od davnih Grka kada je korištena naprava za šifriranje zvana *skital*. Šifriranje je tokom povijesti korišteno u vojne i diplomatske svrhe – primjer jednog takvog šifriranje je Cezarova šifra, koja je korištena za vrijeme Rimskog cara Gaja Julija Cezara u diplomatskim svrhama. U nešto modernijem vremenu, tijekom Drugog svjetskog rata korištena je kriptografska naprava ENIGMA čija je svrha bila da sakrije Nacističke poruke od Savezničkih postrojbi. Ostali bitniji postupci i naprave za kriptiranje su detaljnije opisane u knjizi [4].

Najrašireniji kriptografski postupci se pojavljuju tek 70-ih godina dvadesetog stoljeća. Direktan uzrok tome je razvoj financijskih transakcija. Od tada, kriptografija postaje bitnija širem broju ljudi. Kako bi novi kriptosustavi bili korišteni na internacionalnim razinama, američka institucija za nacionalnu sigurnost (engl. *National Security Agency,* NSA) odobrila standard enkripcije pod nazivom standard za enkripciju podataka (engl. *Data Encryption Standard,* DES) po čijem standardu je razvijen prvi moderniji kriptografski algoritam s istim nazivom DES, opisan u [3]. Utemeljenjem DES standarda označava se početak digitalne kriptografije. Algoritmi su se počeli primjenjivati, ne direktno na tekstu, nego na bitovima koji tvore tekst. Bitovi su najčešće kriptiranu u grupama, odnosno blokovima – iako postoje metode koje kriptiraju bit po bit. Standard koji nasljeđuje DES i danas se najviše primjenjuje u digitalnoj kriptografiji je napredni standard za enkripciju (engl. *Advanced Encryption Standard,* AES). AES je definirala američka

tvrtka za nacionalnu sigurnost NSA (engl. *National Security Agency*) za sakrivanje ključnih informacija.

2.2. Kriptografski postupci

Kriptografski postupci, prema [3], podijeljeni su po tipu operacije, načinu obrade podatka te po tajnosti i javnosti ključeva. Po tipu operacije razlikuju se supstitucijske i transpozicijske šifre. Kod supstitucijskih šifri, svaki element podatka se zamjenjuje s nekim drugim elementom, dok kod transpozicijskih šifri se elementi podatka permutiraju (poput anagrama - riječi ili fraze koja permutiranjem redoslijeda slova postaje nova riječ ili fraza). Supstitucijske šifre ne moraju sadržavati elemente iz izvornog podatka, dok transpozicijske sadrže samo te elemente, ali u drugačijem redoslijedu.

Postupci po obradi podatka se dijele na blokovne i protočne šifre. Blokovne šifre su šifre kod kojih se sekvencijalno obrađuju blokovi elemenata koristeći unikatni ključ. Za razliku od blokovnih šifri, kod protočnih šifri obrađuje se element po element uz usporedno stvoreni niz ključeva. Načelno kod takvih postupaka, šifriranje slijedećeg elementa ovisi o nizu prijašnjih elemenata. Kod blokova, šifriranje se vrši na većem broju elementa uz korištenje jednog ključa. Složenije protočne šifre, koje pamte elemente unazad nekoliko iteracija šifriranja, zahtijevaju veću memoriju. Blokovne šifre ne zahtijevaju memoriju budući da se šifriranje događa u jednom trenutku, neovisno o prijašnjim elementima. Zbog potrebom za manjom količinom resursa i jednostavnosti blokovnih šifra, suvremeni kriptosustavi najčešće koriste blokovne šifre.

Kriptografski postupci po tajnosti i javnosti ključeva se dijele na simetrične kriptosustave i kriptosustave s javnim ključem. Simetrični sustavi se temelje na ključu koji služi za šifriranje i dešifriranje podatka. Budući da je taj ključ samo jedan, tajnost ključa je bitna. Zbog toga se ovi postupci još nazivaju kriptosustavi s tajnim ključem. Zbog korištenja jednog ključa i općenite jednostavnosti, sustavi s tajnim ključem se smatraju jako efikasnim. Nedostatak dolazi upravo kod potrebe za tajnosti ključa – kako bi se ključ mogao podijeliti između dvije stranke koje ga koriste (pošiljatelja i primatelja), on također treba biti kriptiran.

Kriptosustavi s javnim ključem su asimetrični jer se ključ za dešifriranje ne može saznati iz ključa za šifriranje. Svatko može kriptirati podatak sa takozvanim javnim ključem, ali samo osobe koje imaju odgovarajući (privatni ili tajni) ključ mogu je dekriptirati.

Svaka od navedenih podjela može se kombinirati kako bi nastao kriptografski postupak. Tako, na primjer, Vigenerova šifra je supstitucijski, protočni kriptosustav s tajnim ključem. Viegnerova

šifra svakom elementu podatka mijenja element paralelno sa elementom ključa. Ako se riječ FAKULTET kriptira ključem BROJ, prvi element podatka (F) kriptirat će se prvim elementom ključa (B). Analogno, drugi element podatka (A) kriptirat će se drugim elementom ključa (R). Ovim primjerom pokazano je kako je Vigenerova šifra: supstitucijska jer svaki element zamjenjuje drugim elementom; protočna jer paralelno kriptira u dani ključ; i sustav s tajnim ključem, zbog toga što zadani ključ mora biti tajna kako se šifra ne bi razbila.

2.2.1. Kriptosustavi s javnim ključem

Kriptosustavi s javnim ključem su noviji sustavi. Prema [3], sve do 1976. korišteni su isključivo kriptosustavi sa tajnim ključem. Tek navedene godine, Whitfield Diffie i Martin Hellman su predstavili rješenje kod kojih se podatak kriptira javnih ključem, kojeg svi mogu koristiti, a tajni ključ koji ga dešifrira je poznat samo jednoj osobi. Na taj način bilo tko može šifrirati poruku, ali samo osoba sa tajnim ključem je može dešifrirati. Razvojem ovog postupka otvorile su se mogućnosti za razvoj modernijih vrsta enkripcije.

2.2.2. Kriptosustavi s hibridnim ključem

Postupci s javnim ključem su pogodni za korištenje budući da nije potrebno sakrivati tajni ključ. Međutim, oslanjaju se na složene matematičke operacije koju su naposljetku neefikasne. Kriptosustavi s tajnim ključem su s druge strane efikasni, ali se oslanjaju na tajnosti ključa. Hibridni kriptosustavi kombiniraju obje metode kako bi se dobio efikasan postupak koji se ne oslanja na tajnosti ključa. Hibridni postupci koriste tajni ključ za kriptiranje podataka, te potom javni ključ za kriptiranje tajnog ključa. Budući da tajni ključevi ne sadrže velike količine podataka, neefikasnost javnog ključa nije očigledna. Kriptirani se šalje sa kriptiranim tajnim ključem. Kako bi se podatak mogao dekriptirati, prvo se dekriptira tajni ključ korištenjem primateljevog javnog ključa. Korištenje hibridnog ključa u konkretnoj programskoj biblioteci može se pročitati u [5] gdje je dodatno opisan način djelovanja hibridnog ključa. Podatak je potom moguće dekriptirati korištenjem sada dekriptiranog tajnog ključa. Strogo gledano, kriptostustavi s javnim ključem više nisu korišteni, zato što se u svim implementacijama takvog sustava koriste sustavi s hibridnim ključem.

2.2.3. Hash funkcija

Kao dodatak šifriranju pomoću javnog ili tajnog ključa, uvodi se postupak pod nazivom *hashiranje* (engl. *hashing*). *Hash* je prema [3] ne-reverzibilna deterministička matematička funkcija koja podatke promjenjive veličine pretvara u podatke fiksne veličine. To znači da je

podatke koje prođu kroz *hash* funkciju gotovo nemoguće iščitati iz vraćene vrijednosti u ograničenom vremenskom razdoblju. Zato što su podatci ne-reverzibilni, najčešće se koriste za očuvanje integriteta i autentifikaciju poslanog podatka.

Digitalni potpis je autentifikacijski postupak za ispitivanje integriteta poslanog podatka. Digitalni potpis radi na principu sličnome javnog ključu – razlika je u tome što se podatak uopće ne kriptira. Umjesto kriptiranja, podatak se provuče kroz *hash* funkciju te se dobivena vrijednost doda na kraj podatka. Na primateljevoj strani poslani podatak se također provlači kroz istu *hash* funkciju. Ukoliko je dobivena vrijednosti jednaka vrijednosti nadodanoj na poslani podatak, tada je dokazano da je integritet podatka postojan. Postojanje integriteta označava da podatak nije promijenjen, ali ne osigurava da podatak nije neovlašteno čitan. Zbog tog nedostatka, digitalni potpis se može upotrebljavati uz razne kriptografske postupke koje osiguravaju da se podatak ne može iščitati tokom prijenosa.

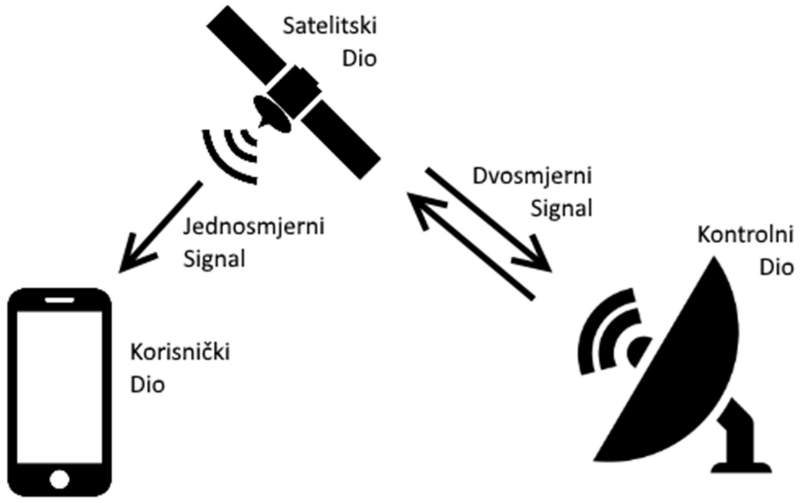
Spremanje lozinki u bazu podataka se oslanja na korištenje *hasha.* Spremanje čiste lozinke (engl. *plaintext*) u bazu podataka označava prodor sigurnosti, jer u tom slučaju osoba koja ima pristup bazi podataka ima pristup lozinkama svih korisnika (te tako i njihovim računima). Lozinke se u bazi spremaju korištenjem *hash* funkcije. Korisnička lozinka u sustavima za prijavu se tada prvo provlači kroz *hash* funkciju, te se potom uspoređuje s lozinkom u bazi podataka.

2.3. Kriptografija zasnovana na lokaciji

Kriptosustavi javnog i tajnog ključa koriste ključeve prilikom kriptiranja podataka. Fleksibilnost takvih kriptografskih sustava je zasnovana na pretpostavci da ključ može biti podatak sam po sebi. Odabir tipa i sadržaja ključa je bitan. Najčešće je korištena lozinka – tekstualni ključ koju korisnik sam odabire. Odabir lozinke je bitan jer o njoj ovisi sigurnost čuvanih podataka. Odabiranjem kratke lozinke povećava se mogućnost neovlaštenog preuzimanja podataka. Odabirom pojma koji označava korisnika se također smanjuje razina sigurnosti podataka.

Svjetski sustav za pozicioniranje (engl. *Global Positioning System*, GPS) je tehnologija osmišljena od strane vojske Sjedinjenih Američkih Država - prvobitno u vojne svrhe, no danas se može javno koristiti besplatno. Za korištenje je samo potreban uređaj koji može primiti podatke GPS-a. Prema podacima iz 2017. u [6], GPS radi pomoću 28 satelita koji kruže oko Zemlje i šalju podatke o lokaciji uređaja koji zahtjeva lokaciju. Na slici 2.1 prikazan je postupak dohvaćanja lokacije. Sateliti i kontrolni dijelovi međusobno sinkroniziraju informacije o lokaciji. Satelit potom pošalje usklađene podatke uređaju koji zahtjeva lokaciju. Lokacija je definirana

kao uređena trojka geografske širine, dužine i visine koja opisuje točno jednu poziciju za površini Zemlje. Zbog teoretske raspodjele Zemlje na meridijane i paralele, preslikavanje geografske mreže u jednu lokaciju je injektivno, što znači da ne postoje dvije točke na Zemlji koje imaju istu geografsku širinu, dužinu i visinu. Zbog svojstva injektivnosti, lokacija može služiti kao jedinstveni ključ prilikom kriptiranja. Kombiniranjem GPS lokacije i jednog od navedenih kriptografskih postupaka može se napraviti kriptografski postupak zasnivan na lokaciji, što znači da se kriptirani podatak može dekriptirati samo na lokaciji za koju je bio kriptiran. Korištenje lokacije kao ključ je osnova sustava u [1] i [2], ali je donesen zaključak da je korištenjem samo lokacije moguće u relativno malom broju iteracija pogoditi šifru.

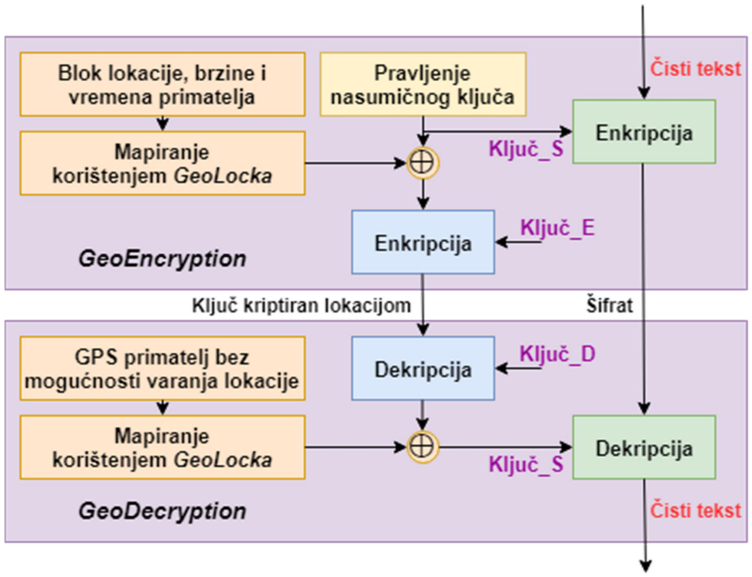


Sl.2.1. Pojednostavnjeni prikaz načina rada GPS sustava, prema uzoru na [6].

Predložena Android aplikacija za dijeljenje podataka među korisnicima prema [1] pod imenom „Geo-Encryption Lite“ koristi GPS sustav pri kriptiranju datoteka. Predloženi algoritam obuhvaća dohvaćanje statičkih vrijednosti lokacije (dužina, širina, visina) koja će s korisnički-unesenom šifrom koristiti XOR binarni operator kako bi se dobio javni ključ za šifriranje. Taj ključ se koristi pri kriptiranju uz uporabu nekog od algoritama za kriptiranje podataka. Postupak je definiran kao kriptosustav s javnim ključem, što znači da se šifra mora poslati uz kriptirani podatak. Samo kada je primajući korisnik na točnoj lokaciji, moguće je dekriptirati kriptiranu datoteku.

Algoritam predložen u [2] uz statičke varijable lokacije koristi i dinamičke podatke (brzina) pri pravljenju ključa za kriptiranje. Korišten je model parametara koji pomoću pozicije, brzine i vremena (engl. *position, velocity, time;* PVT) generira kod za kriptiranje. PVT sam po sebi nije

dovoljan za veću razinu sigurnosti - zato je, slično kao u [1], korišten u XOR operatoru zajedno sa nasumično stvorenim ključem. Rezultat operacije je šifriran ključ koji se dodatno kriptira asimetričnim postupkom. Prilikom dekriptiranja datoteke, PVT je korišten za dekriptiranje poslanog ključa koji naposljetku otključava datoteku. Cijeli postupak može se iščitati iz dijagrama (Sl.2.2).



Sl.2.2. Dijagram tijeka algoritma predložen i napravljen prema [2].

Lokacija kao ključ kriptiranja se može smatrati nepouzdanim zbog toga što je lokacija vezana uz korisnika. Korisnik koji želi kriptirati neki podatak će koristiti lokaciju koju on koristi, ili će prilikom slanja podatka biti korištena lokacija koju primatelj poznaje. Napadači koji poznaju rutinu korisnika mogu u relativno malom broju pokušaja dekriptirati datoteku - zbog toga je potrebno odabrati lokaciju nevezanu uz korisnika. Problem je to što je korisnik obvezan pristupiti odabranom mjestu kriptiranja. Ako je to mjesto udaljeno, potrebno je putovati do te lokacije kako bi se podatak mogao kriptirati, odnosno dekriptirati.

2.3.1. Načini prijevare lokacije

Prijevara lokacije (engl. *location spoofing*) ima usku svrhu. Najčešće se koristi kako bi se zaobišle lokacijske barijere koje blokiraju neki sadržaj. Za prijevaru lokacije postoje fizički alati, ali i programski alati. GPS sateliti komuniciraju preko signala, stoga je moguće promijeniti amplitudu jednog takvog signala kako bi se poslala kriva vrijednost lokacije. Međutim, alati namijenjeni za tu svrhu dostižu visoke cijene na tržištu. Danas je sve popularnije korištenje virtualnog privatnog umrežavanja (engl. *Virtual Private Network*, VPN), sigurnosne poveznice

između korisnika i interneta. VPN se ponajviše koristi kako bi se izbjeglo geo-ključanje sadržaja web-stranica. Geo-ključanje (engl. g*eo-locking*) je sakrivanje ili blokiranje sadržaja klijentima koji se ne nalaze u određenoj državi. Još jedna korist VPN-a je sakrivanje identiteta korisnika. Korisnici VPN-a su stoga zaštićeni od bilo kakvog neovlaštenog načina prikupljanja podataka.

Android mobilni uređaji imaju opciju uključivanja lažiranje lokacije (engl. *mock location*). Uključivanjem navedene opcije moguće je koristiti neke od već napravljenih aplikacija za prijevaru lokacije koje će, kada se zatraži GPS lokacija, jednostavno mijenjati dobivene statičke i dinamičke vrijednosti lokacije (bez mijenjanja signala). Stoga, predložena rješenja [1] i [2] je teoretski moguće prevariti ako napadač zna približnu vrijednost parametara. U tu svrhu, oba rješenja se ne oslanjaju na samu lokaciju, nego koriste parametre lokacije za kriptiranje ključeva koji se koriste u kriptiranju podataka.

Predloženo rješenje iz [2] je dodatno nadograđeno: blokovi za lokacijsko kriptiranje podataka mogu se međusobno ulančavati. Ulančavanjem blokova se postiže da se podatak može dekriptirati točnim redoslijedom lokacija. Tako na primjer, ako je podatak kriptiran na lokaciji L_1, potom L_2 i L_3, moguće ga je dekriptirati ako se prvo obavlja dekripcija na lokaciji L_3, pa L_2 i naposljetku L_1. Na taj način je osigurana veća razine zaštite podatak od prijevare lokacije, zato što za N korištenih lokacija, mora se točno prevariti N lokacija. Vjerojatnost za prijevare lokacije eksponencijalno raste: ako svaka lokacija L ima vjerojatnost $p(L)$ da bude prevarena - tada za n broj lokacija, vjerojatnost da dođe do proboja podataka iznosi $p(L_1) * p(L_2) * ... * p(L_n)$. U slučaju da svaka lokacija ima istu vjerojatnost da bude prevarena, izlazna vjerojatnost postaje $p(L) * p(L) * ... * p(L) = p(L)^n$.

2.3.2. Sigurnost i privatnost u mobilnim aplikacijama

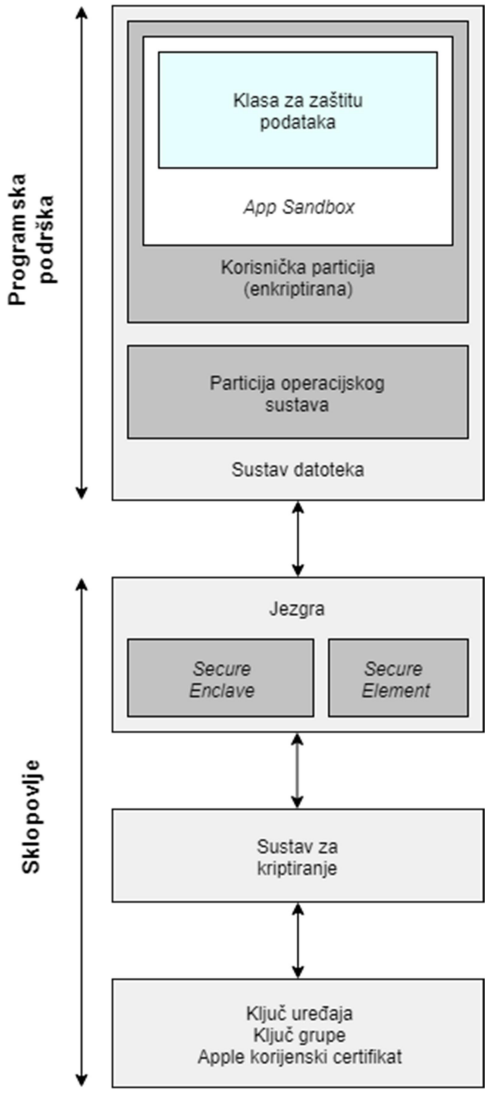
Informacija je danas najtraženija roba. Tvrtke diljem svijeta žele znati sve više o svojim korisnicima kako bi mogli stvarati proizvode koje odgovaraju njihovim kupcima. Uloga mobilnih uređaja u tome je znatna jer su korišteni kao najveći izvor informacija. Razne mobilne aplikacije kao skrivenu funkcionalnost imaju slanje korisničkih podataka poslužiteljima, ali toga je sve manje. Uvedeni su zakoni koji ograničavaju, i naposljetku kažnjavaju, neovlašteno preuzimanje podataka. Dohvaćanje informacija, iako ne šteti korisniku, prodor je sigurnosnog sustava i privatnosti korisnika, ali postoje agresivniji napadači koji traže više od korisnika. Korisnik je u tu svrhu prva i najslabija sigurnosna barijera – on treba odlučiti o legitimnosti neke aplikacije, jer korisnik je taj koji nesvjesno može dati dozvolu napadu. Prema Forbesovom članku [7] iz 2018., napadi pomoću invazivnih pretplata od korisnika uzimaju od 100 do 3600

američkih dolara godišnje. Australska komisija za tržišno natjecanje i potrošače (engl. *Australian Competition & Consumer Comission,* ACC) je 2018. na [8] objavila statistiku koja nalaže da su na taj način u 1,942 prijavljena slučaja ukradeno 84,584 američka dolara. Sigurnosni mehanizmi mobilnih uređaja predstavljaju drugu sigurnosnu barijeru. Kada se govori o mobilnim uređajima, najčešće se govori o mobitelima sa operacijskim sustavom Android ili iOS.

Android operacijski sustav je zasnovan na Linuksovoj jezgri koja se već godinama razvija i nadograđuje. Linuksova jezgra radi na korisnički-baziranom modelu dozvola (engl. *user-based permission model*) koji, u kratkim crtama, dopušta ili ne dopušta pristup jezgri ili datotekama ovisno o trenutnom korisniku. Taj model se proširuje i na grupe korisnika, tako da se mogu definirati i dopuštenja za specifične grupe korisnika. Još jedna važnija stavka je izoliranje procesa (engl. *process isolation*). To znači da je moguće izolirati bilo koji proces od jezgre ako se taj proces čini nametljivim. Nove sigurnosne stavke su nadodane na Linuksovu jezgru u Androidu. Jedna od stavki koja se nadodaje jest *Application Sandbox*, sigurnosni mehanizam koji odvaja i štiti aplikacije jedne od drugih. Android također ima mogućnost kriptiranja uređaja. Kada je uređaj kriptiran, svi korisnički podaci koji su u njemu ili se naknadno dodaju su automatski kriptirani. Prilikom kriptiranja se koristi *keystore* koji služi za generiranje i čuvanje ključeva za enkripciju podataka. *Keystore* se nalazi u posebnom dijelu sustava pod nazivom sigurnosno okruženje za izvođenje (engl. *Trusted Execution Enviroment,* TEE). TEE je zapravo dio *Trusty* operacijskog sustava koji se nalazi na istom procesoru kao i Android, ali je programski i sklopovski odvojen. *Trusty* i Android se paralelno izvode. *Keystore* nije namijenjen samo operacijskom sustavu, nego svim aplikacijama koje koriste kriptografske sustave s javnim ili tajnim ključem. Bilo kakav kriptografski ključ stvoren na Android uređaju automatski se sprema u *keystore.*

Sigurnosni sustav iOS operacijskog sustava zasnovan je na desetljećima iskustva stečenih razvijanjem stolnih računala. U svrhu boljeg korisničkog iskustva, sigurnosni mehanizmi su transparentni, a ključni mehanizmi su nedostupni za konfiguriranje kako ih korisnik ne bi slučajno isključio iz rada. Na slici (Sl.2.3) može se vidjeti da je sigurnosni sustav iOS-a podijeljen na dva dijela: sklopovski i programski dio. U sklopovskom dijelu se čuvaju razni certifikati i ključevi. *Crypto engine* je mehanizam za kriptiranje zaslužan za sigurni prijenos podataka između brze i glavne memorije. Između sklopovskog i prijenosnog dijela nalazi se jezgra operacijskog sustava koja dozvoljava da programski dio komunicira sa sklopovljem i obrnuto. U programskom dijelu nalazi se sustav datoteka koji je dosta sličan Androidu. Sustav datoteka je podijeljen na particije, a jedna od njih je i korisnička particija (engl. *user partition*) u

kojoj su spremljene i kriptirane korisničke datoteke. iOS, poput Androida, posjeduje *Application Sandbox* koji ograničava aplikacijama međusoban pristup.



Sl.2.3. Dijagram sigurnosnih mehanizama iOS sustava, napravljen po uzoru iz [9].

Privatnost u mobilnim aplikacijama, neovisno o operacijskom sustavu, ovisi isključivo o korisniku. Androidova trgovina aplikacija *Google Play Store* i iOS trgovina aplikacija *App Store* izlistavaju sadržaj i dozvole koje aplikacije koriste. Aplikacije bez ovlaštenih dozvola ne mogu pristupiti ključnim i korisničkim dijelovima mobilnih uređaja. Dozvole mogu biti ovlaštene jedino od strane korisnika. Stoga, korisnik je taj koji može omogućiti proboj privatnosti. Aplikacije koje predstavljaju rizik za proboj privatnosti su aplikacije koje traže višak ili pretjeran

broj dozvola. Tako na primjer, jednostavna aplikacija za uključivanje mobilne svjetiljke koja od korisnika traži dozvolu za čitanje kontakata predstavlja veliki rizik.

Ograničavanje korištenja lokacije je također nužno za očuvanje privatnosti. Neke aplikacije imaju pristup lokaciji uređaja, a tako i fizičkoj lokaciji osobe. Takve aplikacije predstavljaju stalan rizik od proboja privatnosti. Isključivanjem lokacijskih usluga kada se ne koriste takve aplikacije smanjuje se rizik i mogućnost da aplikacije dohvaćaju fizičku lokaciju čak i kada nisu korištene. Korištenje javnih bežičnih (engl. *Wi-Fi – wireless fidelity*) mreža predstavlja opasnost za privatnost i krađe podataka. Spajanjem na bežičnu mrežu uređaj se indirektno povezuje na napadački uređaj. Podatci kada se prenose preko takvih mreža prvo moraju proći kroz bežični usmjerivač (engl. r*outer*). Podatke je moguće uhvatiti prije nego što stupe u vezu za internetskim slojem. Nezaštićeni podaci su na taj način izloženi napadu i neovlaštenom mijenjaju sadržaja.

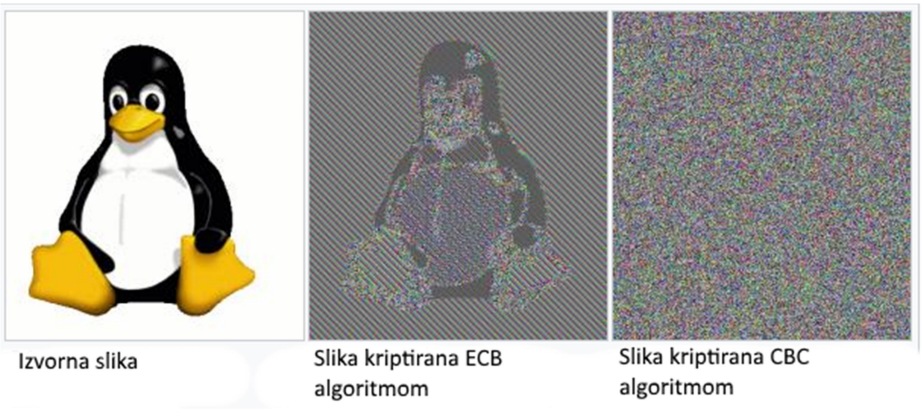
2.4. Uporaba kriptografije na Androidu

Sigurnosni mehanizmi, a tako i kriptografija, igraju ključnu ulogu u zaštiti podataka na računalima općenito, ali kod mobilnih uređaja potreba za zaštitom je znatno veća, zato što je mogućnost da napadači imaju fizički pristup znatno veća. Korištenjem kriptografskih postupaka i njihovih algoritama znatno se povećava razina sigurnosti podataka. Pravilan odabir standarda i algoritma za kriptiranje algoritam je bitan za čvrstu zaštitu. Prolaskom vremena, postupci koji su nekada bili smatrani neprobojnima zastarjeli su. Razvoj novijih standarda i algoritama osigurava da se održi sigurnost kriptiranih podataka, budući da za svaki algoritam postoji određena vrsta napada na koju je slab.

AES, detaljnije opisan u [10], je danas najkorišteniji standard za kriptiranje podataka. Zasniva se na nizu povezanih matematičkih operacija na podatkovnim blokovima koji uključuje i permutiranje i supstituiranje podataka (bitova). Veličina pojedinog bloka iznosi 128 bita, odnosno 16 bajta. Veličina ključa je bitan faktor prilikom kriptiranja u AES standardu, zbog toga što određuje broj rundi transformacija u pojedinom bloku. Ključ može biti 128 bitni, 192 bitni i 256 bitni. Za svaku veličinu po redu izvršit će se 10, 12, odnosno 14 rundi transformiranja. Veća veličina ključa osigurava veću razinu sigurnosti, budući da se blokovi dodatno transformiraju, te zbog toga što je sam ključ teži za otkriti. Veća veličina ključa također podrazumijeva veću količinu potrebne memorije kako bi se podatak mogao kriptirati.

Jedan od slabijih AES algoritama je elektronska kodna knjiga (*engl. Electronic Codebook*, ECB) koji svaki blok podataka šifrira istim ključem što znači da blokovi sa istim redoslijedom

bitovima bit će identični. Primjer zašto je ECB slab može se vidjeti na slici (Sl.2.4): na lijevoj strani je izvorna slika, dok je u sredini slika kriptirana putem ECB algoritma. Budući da su svi blokovi slike jednako kriptirani, mogu se očitati obrisi originalne slike. Na desnoj slici je prikaz enkripcije kod koje se ne može iščitati izvorni podatak.



Sl.2.4. Usporedba rezultata ECB i različitih algoritama pri kriptiranju slike.

Ulančavanje strojnih blokova (engl. *Cyber Block Chaining*, CBC) je algoritam AES standarda. Princip rada je sličan ECB-u. Razlika je u tome što prilikom kriptiranja bloka, taj blok se "provuče" kroz binarni operator XOR zajedno s prijašnjim (sada kriptiranim) blokom. Originalni podatak se ne može iščitati iz kriptiranog podatka. Problem nastaje kada se isti podaci kriptiraju istim ključem. Budući da ne postoji blok prije prvog bloka, prvi blok koristi XOR sa blokom popunjenim nulama. Nastali kriptirani podatak će uvijek imati isti šifrirani tekst. Kako bi svaki šifrirani tekst bio drugačiji, koristi se inicijalizacijski vektor (engl. *Initialization Vector,* IV). Koristeći IV, prvi blok koristi vrijednosti iz IV-a kako bi nastao šifrirani tekst. Kriptirani podatak je stoga drugačiji kada se koriste isti ključevi tako da se pojačava složenost, a time i sigurnost algoritma.

Na Android mobilnim uređajima verzijama 4.4 pa nadalje, moguće je kriptirati korisnikovu particiju na disku korisnikovom lozinkom. Enkripcija koristi AES-ov CBC algoritam sa veličinom podatkovnog bloka od 128 bita. Ključ za enkripciju se također kriptira radi dodatne sigurnosti. Kriptiranjem particije osigurava se dodatna zaštita od vanjskih i unutarnjih napada. Postupak nije prikladan kod fizičkih napada gdje napadač zna lozinku uređaja. Android verzija 7.0 pa nadalje podržava enkripciju baziranu na datotekama (engl. *file-based encryiption*) - dodatak koji dozvoljava da se različite datoteke kriptiraju različitim ključem. Kriptiranjem ključnih podataka nije potrebno kriptirati cijelu particiju diska.

Android prilikom generiranja ključa nudi nekoliko klasa za odabir algoritma. Za kriptiranje podataka koristi se klasa *Cipher* u kojoj je moguće odabrati algoritam za kriptiranje (npr. AES/CBC), dok za stvaranje potpisa, odnosno *hasha*, koriste se klase *MessageDigest, Mac* i *Signature*. Sve tri klase koriste *hash* funkciju, ali postoje razlike: *MessageDigest* je čista *hash* funkcija; *Mac* koristi šifru pri stvaranju potpisa; a *Signature* dodatno koristi niz bitova prvi provjeri valjanosti potpisa.

Objekt klase *Cipher,* nakon inicijalizacije algoritma i odabira kriptiranja, kao izlaz vraća šifrat u obliku niza bitova. Kako bi nastali šifrat bilo moguće dekriptirati, mora se stvoriti objekt klase *Cipher* s istim inicijaliziranim algoritma. Vraćena vrijednost tog objekta prilikom dekriptiranja jest niz bitova koji predstavlja binarni podatak – sliku, tekst, datoteku i sl. Neki postupci, poput CBC-a, zahtijevaju da bude postavljen IV prilikom inicijalizacije. Moguće je direktno unijeti niz bitova kao IV, ali Android pruža posebnu klasu *IvParameterSpec* za stvaranje IV niza. Prilikom kreiranja objekta te klase, potrebno je odrediti koja će tvornica specifikacija biti korištena - vrijednosti IV-a određene su tvornicom specifikacija. *SecureRandom* je jedna od tvornica koja postavlja nasumične vrijednosti IV-a. Zbog toga što nije ovisan ni o kojem podatku, nasumični odabir znatno povećava sigurnost. Ali zato što je IV nasumično generiran, mora biti spremljen u sigurnu pohranu jer se također koristi prilikom dekriptiranja podatka.

3. PROGRAMSKO RJEŠENJE ZA KRIPTIRANJE ZASNIVANO NA LOKACIJI

U prošlom poglavlju razmatrana je teorijska pozadina enkripcije zasnovane na lokaciji. U praktičnom dijelu demonstrirano je kako se lokacijska enkripcija može implementirati na principu tajnog (simetričnog) ključa koristeći GPS lokaciju mobilnog uređaja kao ključ za kriptiranje i dekriptiranje.

3.1. Specifikacija zahtjeva

Enkripcija zasnovana na lokaciji podrazumijeva kriptiranje i dekriptiranje podataka koristeći geografsku lokaciju uređaja. Mora postojati određena stopa tolerancije, budući da se točnost alata za dohvaćanje lokacije smanjuje sa većom preciznošću. Korisnik mora biti u mogućnosti sam odabrati željenu datoteku za kriptiranje. Kriptirane datoteke se spremaju i čuvaju u posebnom direktoriju sve do dok ih se ne dekriptira. Dekriptiranjem jedne takve datoteke će obrisati tu istu datoteku i u tom direktoriju stvoriti dekriptiranu verziju te datoteke. Ostale specifikacije opisane su u tablici 3.1.

Tab.3.1. Tablica specifikacija aplikacije za kriptiranje

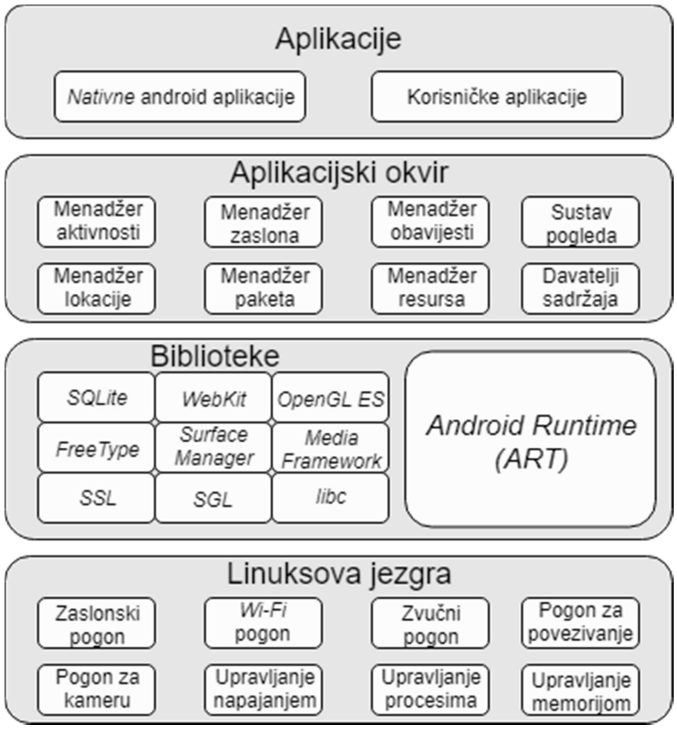
Korisnički slučaj 1.	Korisnik mora moći slobodno se kretati korisničkom particijom.
Korisnički slučaj 2.	Korisnik mora moći odabrati datoteku za kriptiranje.
Korisnički slučaj 3.	Korisnik mora moći odabrati lokaciju za koju će se ta datoteka kriptirati.
Korisnički slučaj 4.	Korisnik može otvoriti nekriptirane datoteke.
Korisnički slučaj 5.	Korisnik može slati kriptirane datoteke.
Korisnički slučaj 6.	Korisnik može pretraživati datoteku pomoći imena datoteke.
Korisnički slučaj 7.	Korisnik ima mogućnost odbijanja dozvole korištenja lokacijskih usluga.
Korisnički slučaj 8.	Korisnik ima mogućnost odbijanja dozvole čitanja i stvaranja datoteka u korisničkoj particji.

3.2. Opis platforme i tehnologija

Programsko rješenje će se napraviti na Android operacijskom sustavu pomoću alata Android Studio koji se služi programskim jezikom Java. Za dohvaćanje lokacije će se koristiti GPS sustav za navigaciju, a kao poveznica između GPS-a i Android Studija koristit će se klase *LocationManager* i *LocationListener*.

3.2.1. Android operacijski sustav

Android je operacijski sustav razvijen od strane Android inc., a od 2005. razvijan je od strane tvrtke Google. Već je spomenuto da je Android razvijen na Linuksovoj jezgri. Komunikacija između sklopovlja uređaja i programske podrške je sadržana u toj jezgri. Na Linuksovu jezgru nadodaju se razne biblioteke gdje je ART (engl. *Android Runtime*) jedna od važnijih. ART definira kako se određene naredbe izvršavaju i prevode u sklopovni jezik kako bi aplikacije bilo moguće brže pokretati. Java biblioteke su korištene kao adapter između aplikacije i standardnih C/C++ biblioteka. *Application Framework* je set usluga koje služe kao mjesto u kojem se aplikacije pokreću i održavaju. Na završnom sloju Android sustava nalaze se aplikacije. Aplikacije mogu biti *nativne* Android aplikacije – aplikacije koje su u paketu određene implementacije Androida - i aplikacije preuzete od strane korisnika. Cjelokupni prikaz Android operacijskog sustava prikazan je na dijagramu (Sl.3.1). Detaljnije o Android operacijskom sustavu se može saznati iz [11].



Sl.3.1. Dijagram organizacije Android operacijskog sustava po uzoru iz [11].

3.2.2. Android Studio

Android Studio je razvojno okruženje za razvijanje Android mobilnih aplikacija koji koristi Java programski jezik za definiranje i programiranje tijeka aplikacije, te proširivi opisni jezik (engl.

Extensible Markup Language, XML) za definiranje izgleda pojedinog prozora ili elementa. Prilikom stvaranja novog projekta potrebno je definirati osnovne podatke i za koju verziju Androida se pravi aplikacija. Načelno, novije verzije Androida imaju veći izbor paketa i elemenata za korištenje, dok ranije verzije podržavaju veći broj uređaja. Verzija 4.4. KitKat se vodi kao najsigurnija verzija za razvijanje zbog njezine stabilnosti i veliki broj korisnika koji obuhvaća.

3.2.3. Java

Java je objektno orijentirani programski jezik (engl. *Object-Oriented Programming Language*) koji je namijenjen za opću uporabu. To je postignuto tako što se Java aplikacije prevode u *bytecode.* Računalo koje u sebi posjeduje Java virtualni uređaj (engl. *Java Virtual Machine*) može pokrenuti aplikacije pisane u Javi, neovisno o arhitekturi računala. Sintaksa je slična programskim jezicima C i C++, ali posjeduje veću razinu apstrakcije spram oba jezika.

3.2.4. XML

XML je opisni jezik za definiranje pravila i formi koji su lako čitljivi čovjeku i uređaju. XML sam po sebi ne pokreće ništa, već služi za prijenos podataka. U Android Studiju XML se koristi kako bi se definirale konstante raznih tipova, te kako bi se dizajnirala struktura pojedinog prozora ili elementa.

3.3. Opis rješenja

Kriptografski postupak koji je korišten u ovom rješenju je kriptiranje tajnim ključem. Budući da je to simetrični postupak, kriptiranje i dekriptiranje je dosta slično - jedina je razlika u kojem smjeru postupak ide. Kao algoritam za kriptiranje korišten je AES-ov algoritam CBC.

Sl.3.2. Dijagram tijeka kriptiranja datoteke pomoću lokacije.

Zbog toga što je korišten CBC postupak, potrebno je definirati inicijalizacijski vektor - IV. IV je niz od 16 bajta popunjen nasumičnim brojevima. Enkripcija odabrane datoteke se vrši tako što se prvo odabere lokacija za koju se želi kriptirati datoteka. Odabrana lokacija, uz nasumično

generirani ključ - po uzoru na postupke iz [1] i [2] - služi za stvaranje ključa koji će se koristiti u AES/CBC kriptografskom algoritmu. Definiranjem ključa, pokreće se postupak kriptiranja pomoću AES/CBC algoritma. Stvara se datoteka u kojoj se prvo upisuje 16-bajtni IV, a potom se upisuje šifrat dobiven prolaskom podataka kroz AES/CBC. Dobivena datoteka sadrži tekst iz kojeg se ne može očitati izvorni podatak.

Sl.3.3. *Dijagram tijeka dekriptiranja lokacije pomoću lokacije.*

Postupak dekriptiranja pokreče se odabirom kriptirane datoteke. Najprije se iz kriptirane datoteka iščitava 16-bajtni IV. Potom se dohvaća trenutna GPS lokacija mobilnog uređaja iz koje se iščitava geografska dužina i širina kako bi se stvorio ključ za dekriptiranje. Generirani ključ koristi se uz AES/CBC algoritam i učitani IV za dekriptiranje datoteke. Kao izlaz nastaje izvorna datoteka.

3.4. Važni implementacijski detalji

Prije samog korištenja lokacije i datoteka na Android sustavu, moraju se dobiti dozvole od korisnika. Dozvole koje aplikacija treba navode se u datoteci *Manifest.xml*.

Dozvola ACCESS_LOCATION_FINE omogućuje dohvaćanje precizne lokacije uređaja, dok dozvole READ_EXTERNAL_STORAGE i WRITE_EXTERNAL_STORAGE omogućuju čitanje, odnosno pisanje (stvaranje) datoteka u korisničkom direktoriju.

```
5       <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
6       <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
7       <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Sl.3.4. Dozvole za pristup preciznoj lokaciji i rukovanjem datotekama.

3.4.1. Dohvaćanje datoteka

Kako bi bilo moguće uopće odabrati datoteku za kriptiranje, korisnik mora biti u mogućnosti pretražiti željenu datoteku u sustavu datoteka. U tu svrhu je kreirana klasa *FileListActivity* koja

nasljeđuje klasu *ListActivity*. Naslijeđena klasa ima dodatne funkcionalnosti poput dohvaćanja putanje direktorija kako bi se lakše mogli definirati adapteri.

Adapteri služe kao poveznica između podatka i *Activitya* u kojem ga se želi prikazati. U ovom projektu se definira adapter koji će povezati niz naziva datoteka i tipa datoteke sa *FileListActivityjem*. Na slici 3.5. prikazan je dio koda za postavljanje adaptera. Metoda kao parametar prima niz naziva datoteka. Stvara se novi niz koji sadržava uređeni par naziva datoteke i njenog tipa (datoteka ili direktorij). Kako bi *FileListActivity* mogao mijenjati izgled, napravljen je novi adapter *FileListAdapter* koji uz svaku datoteku stavlja odgovarajuću ikonu.

```
208     protected void setupAdapter(List<String> files)
209     {
210         List<ArrayList<String>> values = new ArrayList<>();
211
212         for (int i = 0; i < files.size(); i++) {
213             if (new File( pathname: path + File.separator + files.get(i)).isDirectory()) {
214                 values.add(i, new ArrayList<>(Arrays.asList(files.get(i), "Directory")));
215
216             } else {
217                 values.add(i, new ArrayList<>(Arrays.asList(files.get(i), "File")));
218             }
219         }
220
221         FileListAdapter adapter = new FileListAdapter(
222                 context: this,
223                 R.layout.list_view_item,
224                 values
225         );
226
227         setListAdapter(adapter);
228
229         Collections.sort(values, new FileComparator());
230     }
```

Sl.3.5. Metoda za postavljanje adaptera između zaslona korisnika i niza datoteka u trenutnom direktoriju.

3.4.2. Dohvaćanje trenutne lokacije uređaja

Za komunikaciju sa GPS sustavom, potrebna je poveznica između njega i Android sustava. Za tu svrhu koriste se klase *LocationManager* i *LocationListener* po uzoru na "Geo-Encryption Lite" iz [1]. *LocationManager* služi za postavljanje klase *LocationListener* koja periodički prima trenutačnu lokaciju uređaja od strane GPS sustava. Moguće je postaviti što će se izvršiti u slučaju da se priključi ili isključi pružatelj (engl. *provider*) lokacije i ako se status pružatelja promjeni.

U rješenju je napravljena klasa *CipherLocationListener* koja nasljeđuje *LocationListener* i dodaje joj određene funkcionalnosti za generiranje kriptografskog ključa. Može se dohvatiti trenutačno spremljena lokacija i pretvoriti je u tip podataka *string* koji se može koristiti za definiranje ključa. Za to se koristi posebna adapter klasa *LocationFormatter* koji može primiti objekte klase *Location* i *LatLng*. *LocationFormatter* dohvaća vrijednosti lokacije i zaokružuje ih na odabrani broj decimala. Zbog nepreciznosti GPS sustava, geografska širina i dužina zaokružuju se na tri decimale.

3.4.3. Odabiranje lokacije za enkripciju

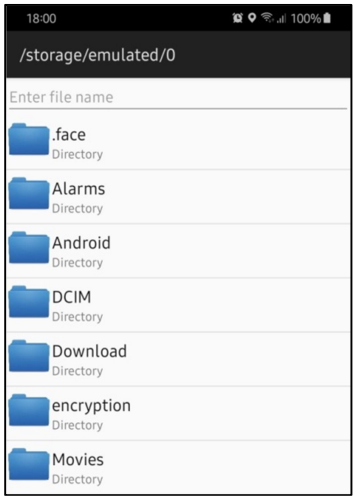
Odabir lokacije postiže se korištenjem već gotovog proizvoda Google karte. Prilikom kreiranja prozora koji sadrži Google kartu, stvara se pokazivač (engl. *marker*) pomoću kojeg korisnik može odabrati lokaciju. Odabir lokacije se vrši *drag and drop* metodom – korisnik pritisne pokazivač na jednu sekundu, te pomiče pokazivač po zaslonu. Svojstvo da se pokazivač može premještati već je definirano u samoj klasi pokazivača. Programski kod je potreban za dohvaćanje lokacije nakon što se pokazivač pomake. U tu svrhu koristi se *OnMarkerDragListener* sučelje koje definira tri metode. Jedna od tih metoda je *onMarkerDragEnd* koja se izvrši nakon što je pokazivač postavljen na kartu. Na slici 3.6. prikazan je isječak koda za odabiranje lokacije. Pritiskom na gumb „Odaberi“ poziva se metoda *onClick* koja šalje lokaciju odabranu na Google karti.

```
87      @Override
88      public void onMarkerDragEnd(Marker marker)
89      {
90          LocationFormatter formatter = new LocationFormatter(marker.getPosition());
91
92          tvLocation.setText(formatter.getAreaCoordinates());
93      }
94
95      @Override
96      public void onClick(View v) {
97          String coordinates = tvLocation.getText().toString();
98          Intent intent = new Intent();
99          intent.putExtra( name: "coordinates", coordinates);
100         setResult(RESULT_OK, intent);
101         finish();
102     }
```

Sl.3.6. Kod za dohvaćanje lokacije pokazivača i vraćanja dohvaćene lokacije pomoću koje će se odabrana datoteka kriptirati.

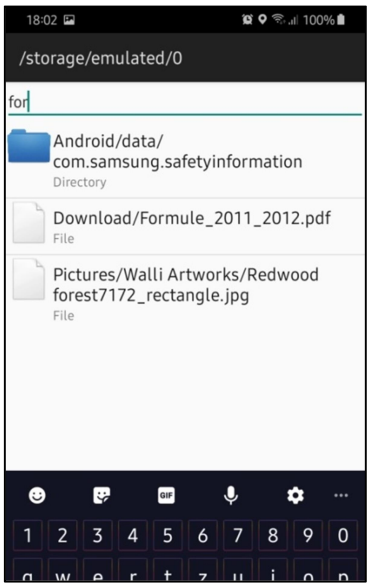
3.5. Prikaz rada rješenja

Otvaranjem aplikacije prikazuje se korisnički direktorij. Odabiranjem direktorija otvara se novi prozor u kojem je izlistan sadržaj odabranog direktorija. Pritiskom na Android tipku za vraćanje, zatvorit će se prozor s trenutno prikazivanim direktorijem.



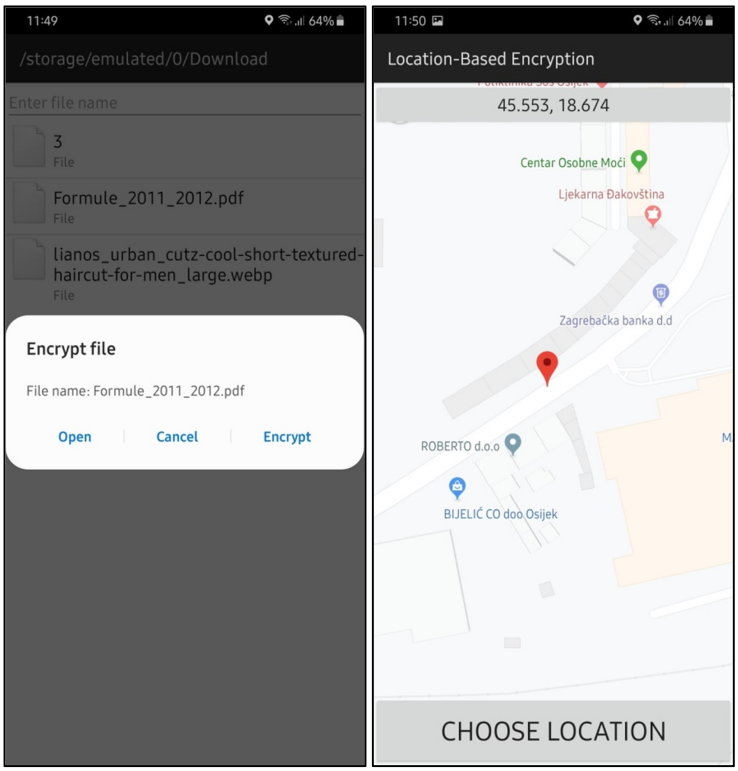
Sl.3.7. Početni prikazani direktorij

Pretraživanje direktorija i datoteka je moguće koristeći traku za pretraživanje na vrhu zaslona. Upisivanjem riječi izlistavaju se svi direktoriji i datoteke koje u svom imenu sadrže traženu riječ.



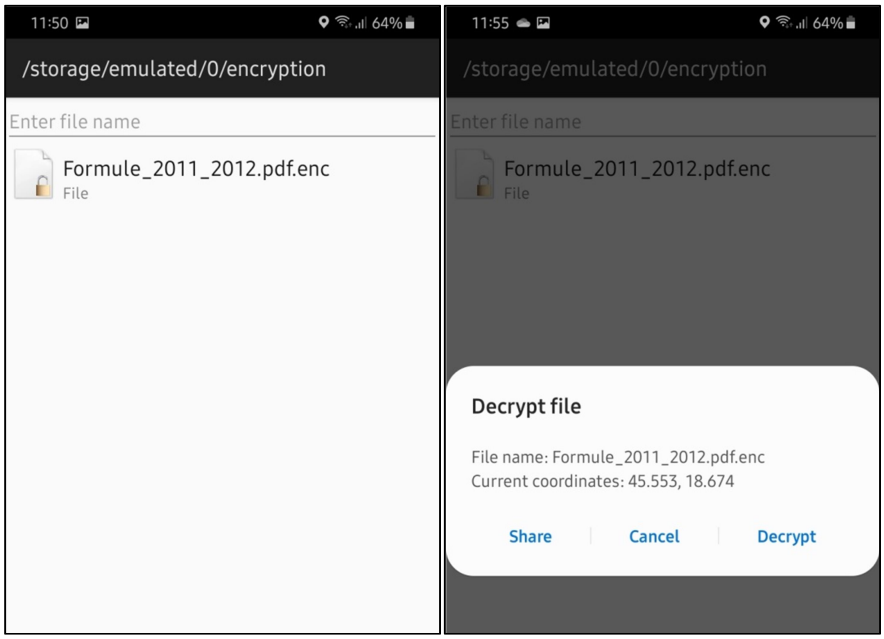
Sl.3.8. Pretraživanje svih datoteka i direktorija iz trenutačnog direktorija. Izlistavaju se svi direktoriji i datoteke koje u sebi sadrže „for“.

Odabiranjem datoteke otvara se dijaloški s tri moguće opcije. Odabiranjem *Cancel* gumba zatvara se dijaloški prozor. Odabiranjem *Open* moguće je otvoriti nekriptiranu datoteku u odgovarajućem pregledniku. Odabiranjem *Encrypt* gumba, otvara se novi prozor na kojem korisnik može odabrati lokaciju kriptiranja.



Sl.3.9. Dijaloški prozor (lijevo) za kriptiranje datoteke. Odabiranjem *Encrypt* otvara se Google karta (desno) za odabir lokacije.

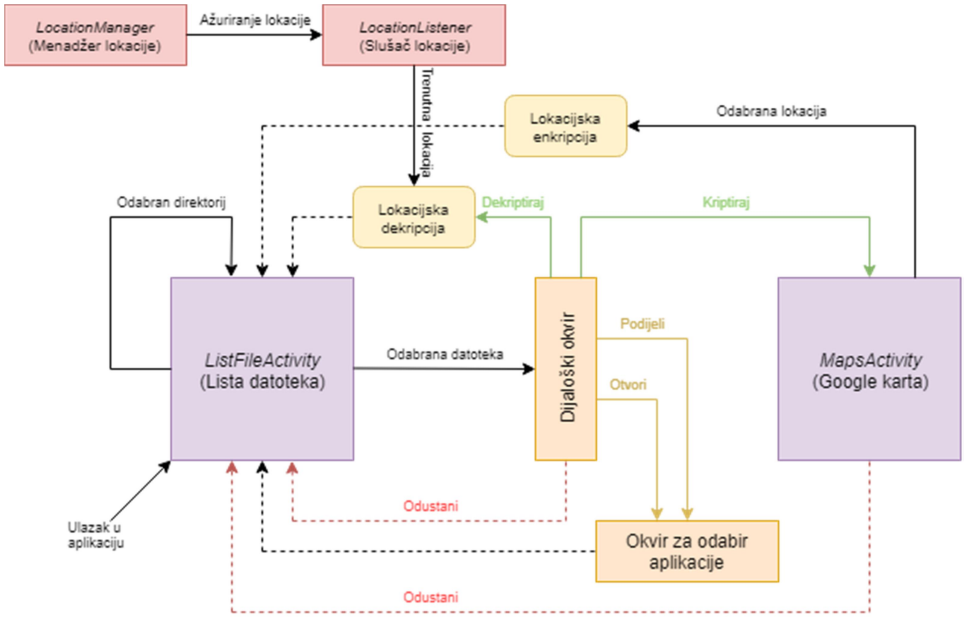
Odabirom lokacije i kriptiranjem datoteke u korisničkom direktoriju se stvara novi direktorij pod nazivom "*encryption*". U tom direktoriju sadržane su kriptirane datoteke. Odabirom kriptirane datoteke otvara se dijaloški prozor za nazivom datoteke i trenutačnom lokacijom. Prozor također sadrži tri opcije: *Cancel* za odustajanje od dekriptiranja, *Share* za slanje datoteke pomoću nekih od aplikacija za dijeljenje podataka, te *Decrypt* za pokušaj dekriptiranja datoteke. Za razliku od enkripcije, prilikom dekriptiranja nije moguće odabrati lokaciju, već je korištena trenutačna lokacija uređaja.



Sl.3.10. „*encryption*“ direktorij (lijevo) koji sadrži kriptirane datoteke. Odabiranjem datoteke otvara se dijaloški prozor (desno) za dekripciju datoteke.

Pritiskom *Decrypt* opcije, odabrana kriptirana datoteka će se dekriptirati i spremiti u direktoriju *“encryiption”*. Razlika između kriptirane i obične datoteke očituje se u njihovim nastavcima: kriptirana datoteka ima dodan nastavan “.enc”. U slučaju da je došlo do greške prilikom enkripcije, prikazat će se valjana poruka na prozoru.

Sl.3.11. Sadržaj direktorija „encryption“ nakon što je datoteka dekriptirana.



Sl.3.12. Cjelokupni prikaz tijeka aplikacije. Pune strelice označavaju normalan tijek izvođenja, dok iscrtane strelice označavaju vraćanje na pojedinu stavku.

3.6. Testiranje rješenja

Aplikacija je testirana od strane nekoliko osoba. Svaka osoba je popunila upitnik od osam pitanja postavljena pomoću Likertove skale koja se može vidjeti u tablici 3.2. Anketa je podijeljena na dvije grupe pitanja: četiri pitanja koja su vezana uz aplikaciju i četiri pitanja koja su vezana uz korištenje lokacije kao osnova kriptiranja. Svih osam pitanja predstavljeno je u tablici 3.3. zajedno sa svojim rednim brojem i grupom pitanja kojoj pripadaju. Na svako pitanje je moguće odabrati samo jedan odgovor iz Likertove skale.

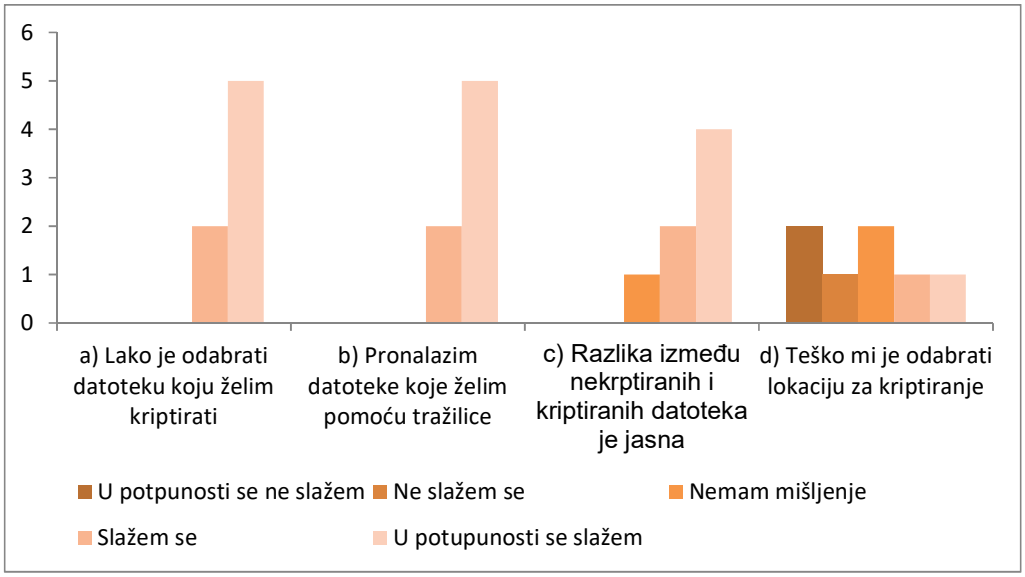
Tab.3.2. Likertova skala sa odgovarajućim vrijednostima.

Likertova skala	Vrijednost
U potpunosti se ne slažem.	1
Ne slažem se.	2
Nemam mišljenje.	3
Slažem se.	4
U potpunosti se slažem.	5

Tab.3.3. Osam pitanja postavljenih u anketi podijeljenih u dvije grupe.

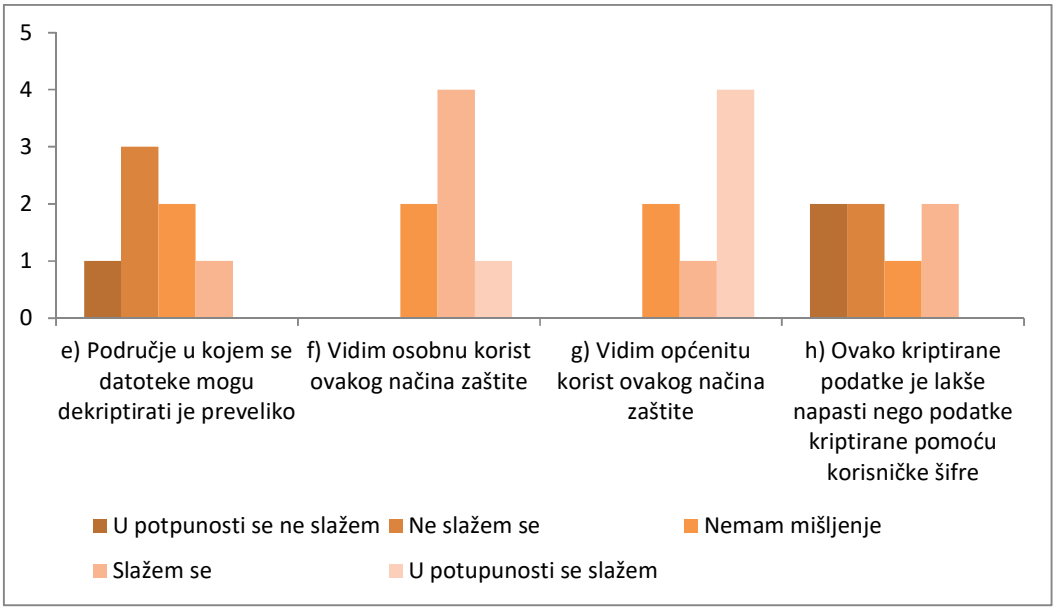
Grupa	Broj pitanja	Pitanje
Aplikacija	1	Lako je odabrati datoteku koju želim kriptirati.
	2	Pronalazim datoteke koje želim pomoću tražilice.
	3	Razlika između nekriptiranih i kriptiranih datoteka je jasna.
	4	Teško mi je odabrati lokaciju za kriptiranje.
Lokacija	5	Područje u kojem se datoteke mogu dekriptirati je preveliko.
	6	Vidim osobnu korist ovakvog načina zaštite.
	7	Vidim općenitu korist ovakvog načina zaštite.
	8	Ovako kriptirane podatke je lakše napasti nego podatke kriptirane pomoću korisničke šifre.

Na slici 3.13. nalaze se pitanja i rezultati iz prve skupine pitanja. Korisnici koji su testirali aplikaciju su zadovolji načinom kojim se mogu kretati kroz aplikaciju i tražiti željene datoteke, bilo to ručno ili pomoću tražilice. Smatrano je da razlika između enkriptiranih i običnih datoteka jest vidljiva, iako bi se u budućnosti ta stavka mogla poboljšati koristeći dodatna vizualna pomagala poput mijenjanja boje pozadine kod enkriptiranih datoteka. Najveća raznolikost mišljenja i iskustva dolazi kod odabiranja lokacije. Aplikacija trenutno koristi Google karte API kako bi se upravljalo pokazivačima, što znači da je pokazivač potrebno držani na jednu sekundu kako bi ga se mogao pomicati. Iskustvo je ukazalo da je takav način relativno težak za korištenje. Do problema ponajviše dolazi do pokazivača markera: ispitanici su teško mogli precizno odabrati pokazivač. Jedan od prijedloga rješenja je povećanje veličine markera, dok je drugi prijedlog dodavanje mogućnosti pomicanja pokazivača pritiskom na kartu.



Sl.3.13. Grafički prikazi rezultata ankete vezanih uz programsko rješenje.

Druga grupa pitanja i njihovi rezultati nalaze se na Sl. 3.14. Ova grupa pitanja bila je usmjerena na korištenje lokacije uz enkripciju i stav korisnika prema takvom pristupu. Zbog nepreciznosti GPS lokacije u većim decimalama, u aplikaciji mora postojati određena stopa tolerancije. Trenutačno ta stopa je u granicama od 15 metara – što znači da osoba koja kriptira podatak na jednoj lokaciji, može isti podatak dekriptirati u krugu od 15 metara. Većina ispitanika ipak smatra da ova stopa nije prevelika i da je 15 metara sasvim prihvatljivo područje, dok manjina ili smatra suprotno, ili pak nema naročito mišljenje. Većina korisnika smatra da bi ovakva vrsta zaštite imala potencijalnu korisnost u osobnom okruženju, ali da bi se ipak mogla malo poboljšati. Jedno od rješenja, još predloženo u [1], jest dodavanja korisničke lozinke. Tada bi lokacija služila kao dodatni izvor zaštite podataka. Mišljenje se pomalo mijenja kod korištenja ovakve enkripcije u opće svrhe. Smatrano je kako bi ovakva vrsta zaštite bila korisnija u općenitim okruženjima, poput čuvanja podataka banke, tvornica, i sl., nego u osobnom okruženju. Ovaj rezultat je neočekivan, budući da ustanove imaju javno poznatu adresu, stoga i lokaciju. Time bi lokacijska enkripcija trebala biti manje korisna u općim nego u osobnim svrhama, gdje adresa korisnika najčešće nije poznata. Polovica ispitanika ipak smatra da bi ovakva vrsta enkripcije bila teža za napasti nego korištenje lozinke, dok druga polovica nema naročito mišljenje ili smatra da je lozinka ipak bolji način zaštite. Razina sigurnosti lokacijske enkripcije može se svesti na osobnu razinu korisnika. Korisnik će najvjerojatnije koristiti lokaciju njemu poznatu ili onu na kojoj se najviše nalazi (npr. posao ili kuća). Korištenje nepoznate lokacije ili lokacije koja nije u njihovoj rutini nije pogodno jer to znači kako bi se podatak morao dekriptirati, korisnik mora biti fizički nazočan na odabranoj lokaciji. Kod lozinke korisnik ima mogućnost odabiranja šifre koja korisniku nije od osobne važnosti (ili pak nema nikakvo značenje) i odabiranje takve šifre ne utječe na korisnika. Zbog korištenja osobne lokacije, lokacijska enkripcija je u načelu slabija od tekstualne lozinke.



Sl.3.14. Grafički prikazi rezultata ankete vezanih uz korištenje lokacije prilikom kriptiranja.

4. ZAKLJUČAK

Sigurnost podatak postaje sve važnija – razni tipovi kriptografskih sustava omogućuju da se podaci zaštite od neovlaštenog čitanja sadržaja. Danas najkorišteniji kriptosustavi su sustavi s hibridnim ključem koji uvode efikasnost sustava s tajnim ključem i sigurnost sustava s javnim ključem. Kao dodatan sloj zaštite, moguće je koristiti lokaciju prilikom kriptiranja podataka. Na taj način se određuje područje u kojem se podatak može iščitati. Nedostatak toga je potrebna fizička nazočnost uređaja ako se želi koristiti udaljenija, sigurnija lokacija.

Kako bi se demonstrirao kriptografski sustav koji koristi lokaciju prilikom kriptiranja, napravljeno je programsko rješenje za Android mobilne uređaje. Koristeći najsuvremeniji standard za kriptiranje podataka, AES, i jedan od njegovih algoritama, CBC, aplikacija uspješno kriptira datoteke koristeći lokaciju kao ključ kriptiranja. Međutim, sama lokacija kao ključ kriptiranja nije dovoljno jaka – korisnik će najvjerojatnije kriptirati podatke u svom osobnom okruženju. Napadači koji znaju samo boravište korisnika lako mogu neovlašteno preuzeti podatke. Lokacija bi se stoga trebala koristiti kao dodatan sloj zaštite. Uz korisnički definiranu šifru moguće je iskoristiti lokaciju kako bi nastao ključ kojeg je teško probiti.

Sama aplikacija još nudi neke osnovne mogućnosti poput dijeljenja kriptiranih datoteka i otvaranja nekriptiranih datoteka. Aplikacija je testirana na nekoliko korisnika koji su zadovoljni funkcionalnostima koja pruža. Jedan od mogućih problema je nedovoljna razlika između nekriptiranih i kriptiranih datoteka. Uvođenjem dodatnih vizualnih pomagala, poput mijenjanja boje pozadine, može se postići bolji pregled između nekriptiranih i kriptiranih datoteka. Najveći problem dolazi kod odabira lokacije: pokazivač za odabir lokacije nije dovoljne veličine. Najočitije rješenje je povećanje pokazivača kako bi upravljanje postalo lakše. Dodatno rješenje je pomicanje pokazivača pritiskom na kartu. Korištenjem oba predložena rješenja omogućava se znatno lakše korištenje i vladanje odabirom lokacije. Postoji potencijalna upotreba ovakvog načina zaštite u osobne i općenite svrhe, ali donesen je zaključak da se lokacija ne bi trebala sama koristiti prilikom kriptiranja. Lokacija bi trebala biti dodatan sloj zaštite, a ne temelj na kojem se rade kriptografski sustavi.

LITERATURA

[1] A. Jha, S. Yangad, A. Surwase, S. Chaudhari, "Geo-Encryption Lite" - A location based Encryption Application for Android, International Journal of Computer Applications, sv. IV, br. 165, str. 13-17, 04.2017.

[2] D. E. Denning, L. Scott, A Location Based Encryption Technique and Some of Its Applications, ION National Technical Meeting, sv. 2003, str. 734-740, 01.2003.

[3] M. Maretić A. Dujella, Kriptografija, Element, Zagreb, 2007.

[4] J. Keegan, The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography, Knopf Doubleday Publishing Group, 2011.

[5] Zend, Encrypt and decrypt using hybrid cryptosystem, Zend Framework, https://docs.zendframework.com/zend-crypt/hybrid/, pristupljeno: 09.2019.

[6] The Library of Congress, What is GPS, Everyday Mysteries, https://www.loc.gov/rr/scitech/mysteries/global.html, pristupljeno: 08.2019.

[7] John Koetsier, App Scams: Sneaky 'Utility' Apps Are Stealing $260, $2500, or even $4700 Each Year ... Per User, Forbes, https://www.forbes.com/sites/johnkoetsier/2018/10/04/app-scams-cheap-utility-apps-are-stealing-260-2500-or-even-4700-each-year-per-user/, pristupljeno: 08.2019.

[8] Australian Competition & Consumer Comission, Scam statistics, Scamwatch https://www.scamwatch.gov.au/about-scamwatch/scam-statistics?scamid=23&date=2018, pristupljeno: 08.2019.

[9] Apple, iOS Security, https://www.apple.com/business/docs/site/iOS_Security_Guide.pdf, pristupljeno: 08.2019.

[10] Google, System and kernel security, Source, source.android.com/security/overview/kernel-security, pristupljeno: 08.2019.

[11] Techtopia, An Overview of the Android Architecture, Techtopia, https://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture, prisupljeno: 08.2019.

[12] Google, Encryption, Source, https://source.android.com/security/encryption, pristupljeno: 08.2019.

[13] Department of Homeland Security, Security Tip (ST19-003), CISA, https://www.us-cert.gov/ncas/tips/st19-003, pristupljeno: 09.2019.

SAŽETAK

U ovom radu istražuju se načini i metode korištenja lokacije prilikom kriptiranja podataka na mobilnim uređajima. Opisani su osnovni tipovi kriptiranja podataka, sigurnosni mehanizmi Android i iOS uređaja, te način na koji je implementirano kriptiranje u Android uređajima. Cilj rada je, koristeći neke od metoda opisanih u teorijskom dijelu, načiniti Android aplikaciju koja kriptira datoteke koristeći lokaciju. Aplikacija je načinjena u programskom alatu Android Studio koristeći programski jezik Java i opisni jezik XML. U aplikaciji, korisnik može odabrati lokaciju za koju će se datoteka kriptirati. Datoteka je kriptirana AES/CBC algoritmom s nasumičnim inicijalizacijskim vektorom i lokacijom kao ključem kriptiranja. Kriptiranu datoteku moguće je dekriptirati samo kada je Android uređaj na lokaciji odabranoj za kriptiranje. Aplikacija je laka za korištenje, iako postoji problem u korisničkom iskustvu kod upotrebe odabira lokacije. Smatra se kako bi lokacija trebala biti dodatak prilikom kriptiranja umjesto da se na lokaciji bazira cijeli sustav.

Ključne riječi: Android, enkripcija, kriptografija, lokacija, sigurnost

LOCATION-BASED ENCRYPTION, ABSTRACT

This thesis explores ways and methods of using location for encrypting data on mobile devices. Basic types of encryption, security mechanisms in Android and iOS devices, and ways of encryption implemented on the Android system are described in this paper. The goal is to create an Android application which uses some of the methods described in theoretical part. The application is created via the programming tool Android Studio which uses Java programming language and XML markup language. When a user chooses a file to encrypt, he can choose the location this file will be encrypted for. The file is encrypted using AES/CBC algorithm with randomized initialization vector and location as encryption key. The encrypted file is possible to decrypt only when the Android device is located at the chosen encryption location. The application is easy to use, although problems with user experience exist while picking encryption location. It is considered that the location should be used as an extra layer of protection while encrypting instead of being used as a basis of the whole system.

Key words: Android, encryption, cryptography, location, security,

ŽIVOTOPIS

Matej Dmitrović rođen je 18. prosinca 1997. godine u Osijeku. Pohađao je osnovu školu August Šenoa. Nakon osnovnoškolskog obrazovanja, uspije se u Prirodoslovnu-matematičku gimnaziju, Osijek. 2014. osvaja 23. mjesto na državnom natjecanju iz logike. 2016. godine upisuje se na Fakultet elektrotehnike, računarstva i informacijskih tehnologija, Osijek smjer računarstvo.

Matej Dmitrović

PRILOZI

1. Enkripcija zasnovana na lokaciju u .docx formatu
2. Enkripcija zasnovana na lokaciju u .docx formatu
3. Izvorni kod