Synthesis
International Scientific Conference of IT and Business-Related Research

RAZVOJ KRIPTO MODULA ZA BEZBEDNU RAZMENU KLJUČEVA NA ANDROID PLATFORMI

THE DEVELOPMENT OF CRYPTO MODULE FOR SECURE KEY EXCHANGE ON ANDROID PLATFORM

Petar Stepanović

Apstrakt:
Brz razvoj Interneta uslovio je potrebu za razvojem zaštite informacija preko Interneta. Korišćenjem Android, Java, Google Cloud Messaging tehnologija razvili smo kripto modul za razmenu ključeva preko Interneta sa svim funkcionalnostima koje su potrebne za realizaciju ove aplikacije.
U ovom radu će biti objašnjene osnove pomenutih tehnologija sa kratkim pregledom aplikacije koja omogućava sigurnu razmenu ključeva preko Interneta.

Ključne reči:
ključ, aplikacija, Internet, bezbednost, razmena.

Abstract:
The necessity of secure exchange of information over the Internet was triggered by sudden Internet advancements. We have used Android, Java and Google Cloud Messaging technologies to develop a crypto module for secure key exchange over the Internet with all the functions needed for the implementation of such application.
This paper shall elaborate on the fundamentals of the aforementioned technologies with the overview of the applications enabling secure key exchange over the Internet.

Key words:
key, application, Internet, security, exchange.

UVOD

Savremene informacione tehnologije drastično su promenile današnji svet računarskih mreža, s obzirom na svakodnevni rast broja novih korisnika Interneta i samih internet servisa, povećava se kako količina tako i vrednost informacija pa slobodno možemo reći da je informacija jedan od najbitnijih resursa današnjice. U današnje vreme Internet ulazi u razne aspekte svakodnevnog života. On se koristi za komunikaciju, trgovinu, informisanje, elektronsko bankarstvo, itd. Usled ovoga Internet nam olakšava dosta toga ali isto tako napadačima olakšava da nelegalnim napadima dođu do čuvanih infromacija. Obezbeđivanjem kanala kroz koji prolaze naše informacije umanjujemo ne samo privatnu već i sve češće ekonomsku štetu.

Za ciljnu grupu koja će koristiti ovaj modul izabrani su korisnici koji koriste uređaje na Android platformi i koji koriste aplikacije za ramenu poruka preko interneta. Opredelili smo se za ovu ciljnu grupu korisnika jer je sve veći broj korisnika koji koristi aplikacije za razmenu poruka preko Interneta, kako za privatno ili neobavezno takozvano ćaskanje, tako i onih koji koriste ovu mogućnost u poslovne svrhe.

U skladu sa prethodno navedenim prednostima i manama komunikacije preko Interneta za temu ovog rada opredelili smo se za izradu kripto modula za razmenu ključeva na Android platformi. Aplikacija pruža korisnicima mogućnost da se registruju kako bi mogli bezbedno da komuniciraju sa dugim registrovanim korisnicima tako što prvo izračunaju ključ koji će kasnije koristiti za šifrovanje željenih poruka.

Šifrovanjem infromacija onemogućujemo neželjenoj strani da ima uvid u sadržaj naše informacije. Šifrovanje je proces transformacije orignalne informacije (otvoreni tekst) u šifrovani podatak (šifrat). Obrnut proces šifrovanju je, dešifrovanje koje na osnovnu šifrata rekonstruiše prvobitnu nešifrovanu poruku. U oba procesa pored otvorenog teksta i šifrata koristi se vrednost koja se naziva ključ šifrovanja. Broj simbola koji predstavlja ključ tj. njegovu dužinu zavisi od šifrarskog sistema koji se koristi i od veličine otvorenog teksta. Postoje dva osnovna tipa kriptografije simetrična i asimetrična kriptografija.

Kao što je prethodno napomenuto za ciljnu grupu su izabrani korisnici koji koriste uređaje na Android platformi. Postoje razni načini i uređaji preko kojih možemo da razmenjujemo poruke ali smo se opredelili za pametne mobilne uređaje zbog njihove rasprostranjenosti i dostupnosti u skoro svakoj situaciji. Za Android platformu smo se opredelili jer po nekim istraživanjima Android korisnici čine čak 85% ukupnog broja korisnika svih operativnih sistema na globalnom nivou.

Za izradu same Android aplikacije danas postoji nekoliko integrisanih razvojnih okruženja od kojih su najpopularnija Eclipse, Android Studio, IntelliJ IDEA. Na sloju komunikacije mogu se koristiti razne tehnologije neke od njih su Java sa raznim bibliotekama koje omogućavaju laku komunikaciju između servera i korisnika, Python, PHP, itd. Za skladištenje poruka često se koriste MySQL, Oracle, ali u ovom radu za skladištenje poruka i komunikaciju sa uređajem koristili smo *Google Cloud Messaging* servis.

E-mail: petarstepanovic@gmail.com
DOI: 10.15308/Synthesis-2015-81-84

Struktura aplikacije se može podeliti na dve celine: korisnički deo tj. Android aplikacija i komunikacioni deo tj. lokalni server i *Google Cloud Messaging* servis. Android aplikacija sadrži informacije o korisniku koje omogućavaju komunikaciju između dva korisnika i pristup lokalnom serveru. Lokalni server sadrži ključ koji garantuje da se Android aplikacija poveže baš sa željenim serverom i kasnije povezuje Android aplikaciju sa *Google Cloud Messaging* servisom. *Google Cloud Messaging* servis preuzima podatke sa lokalnog servera, skladišti ih i kasnije prosleđuje između korisnika.

Koristeći navedene tehnologije, aplikacija je realizovana u skladu sa navedenim zahtevima a uputsvto za korišćenje aplikacije dato je zajedno sa detaljnim pregledom njene strukture i slikama svih važnijih delova ove aplikacije.

REZULTATI I DISKUSIJA

SOFTVERSKO REŠENJE

Cilj projekta

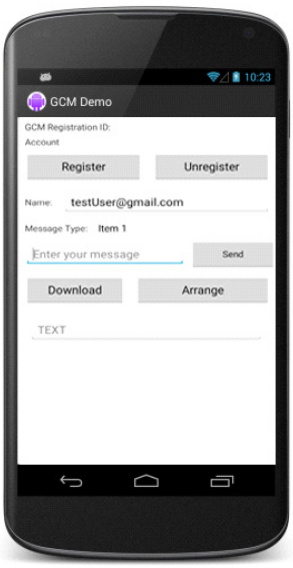
Na osnovu prethodnih primera došli smo na ideju da osmislimo protokol koji bi po ugledu na „Satelitski Scenario“ omogućio da iz generatora nasumičnih vrednosti, koji se nalazi na Internetu, nakon nekoliko faza usaglašavanja svojih lokalnih kopija originalnog niza, dva korisnika (Alisa i Bob) dobiju identični simetrični ključ bez slanja ključa kroz komunikacioni kanal i bez poznavanja lokalne kopije ključa drugog korisnika. Za formiranje lokalnih kopija inspiraciju smo našli u igri „Potapanje brodova“, koju igraju dva igrača sa po dve koordinatne mreže (matrice) čije su dimenzije unapred poznate. Na svojoj lokalnoj matrici igrač raspoređuje brodove i beleži pogotke protivnika, dok na drugoj matrici beleži svoje pogotke. Analogno ovoj igri i naši korisnici prave nasumičnu lokalu kopiju zajedničke matrice sa servera, nakon čega nasumično „gađaju“ jedan drugog kako bi ustanovili na kojim poljima se poklapaju i tako dobili zajednički ključ.

Arhitektura celokupnog softverskog rešenja

Arhitektura ovog softverskog rešenja se sastoji iz serverske i klijentske strane. Na serveru je postavljena dvodimenzionalna matrica koju klijent preuzima sa servera. Ova matrica je nasumičnim redosledom popunjena vrednostima koje mogu da iznose „0“ ili „1“. Nakon preuzimanja, klijent od zajedničke matrice pravi lokalnu kopiju. Nakon toga, po principu „Satelitskog Scenarija“, vrši se ispravljanje grešaka u korelisanom nizu. Prva faza je faza destilacije, druga faza je usaglašavanje informacija. Nakon toga, formira se novi niz u koji se upisuje koji parovi lokalne matrice sada transponovane u jednodimenzionalni niz se zadražavaju a koji odbacuju, taj niz šalju oba korisnika i na osnovu primljenog niza ključevi se usklađuju i oba klijenta formiraju identični ključ. O ovom delu protokola pričaćemo detaljnije malo kasnije.

Implementacija

Za implementaciju ovog protokola u aplikaciju korišćen je *Android Studio*. Ovo je integrisano razvojno okruženje(*IDE)* za Android platformu. Glavna logika aplikacije smeštena je u klasi *MainActivity*. Pritiskom na dugme *Download,* čije je pravo ime identiteta *downloadBT,* pozivamo funkciju *DownloadFile* koja pokreće pregledač *(browserIntent)* kome se prosleđuje adresa datoteke koja je smeštena na serveru i tako pokreće preuzimanje i skladištenje datoteke na memoriju telefona. Zatim pritiskom na dugme *Arrange,* čije je pravo ime identiteta *arrangeBT* pozivamo sledeće funkcije redom: *readBytes, getKeys, Coding, informationReconciliation, privacyAmplification.*



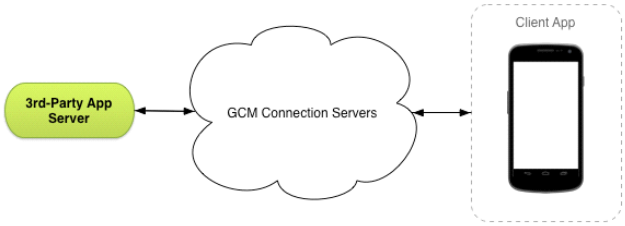
Slika 1. Grafički izgled aplikacije

Objašnjenje korišćenih funkcija:

- *readBytes*
  U ovoj funkciji vrši se čitanje matrice sa servera i upisivanje u dvodimenzionalni niz [i][j].
- *getKeys*
  U ovoj funkciji se prvo formira incijalna promenljiva *(seed) m* koja se uvodi radi veće verovatnoće slučajih vrednosti koordinata koje se nasumično biraju. One mogu da sadrže vrednost od 0 do 50. Nakon što je izabran nasumični par koordinata, na toj poziciji se u lokalnoj matrici prepisuje vrednost zajedničke matrice koja može da sadrži „0“ ili „1“. Tako se nasumično popunjava polovina lokalne matrice, dok se u preostalu polovinu upisuje vrednost „0“. Ovim putem se obezbeđuje da udeo zajedničkog znanja bude ½ tj. da količina informacije ne prelazi 50%.
- *Coding*
  Ovom funkcijom se prolazi kroz niz *for* petljom i određuje parnost paketa koji se sastoje od dva uzastopna bita. Ako je parnost izabranog para „1“ onda se zadržava prvi bit, i tom metodom se uređuje kompletan niz.
- *informationReconciliation*
  Pomoću ove funkcije prethodno uređeni nizovi se potuno izjednačavaju. Ovaj deo protokola počinje tako što seprethodno uređeni nizovi podele u blokove unapred određene dužine za koje se kasnije izračunava bit parnosti. Ako se na određenom bloku bit parnosti razlikuje, to znači da postoji razlika između nizova u tom bloku. Ova funkcija se izvršava u više rundi tj. dok se ne isprave sve greške. Nizovi koji su preostali nakon izvršavanja ove funkcije predstavljaju materijal za generisanje ključa.
- *privacyAmplification*
  Faza protokola o kojoj pričamo se sa razlogom poslednja izvršava. Ova funkcija služi za potupuno izbacivanje napadačeve uzajamne infromacije sa nizovima korisnika koji žele da formiraju zajednički ključ. Ovo postižemo tako što jedna strana bira kombinaciju funkcija kojeće se primeniti i nakon toga šalje kombinaciju brojeva drugom korisniku. Vrednost jednog broja predstavlja jednu kriptografsku funkciju, takođe, vrednost ovih brojeva je unapred dogovoren.

DOI: 10.15308/Synthesis-2015-81-84

Od 1250 bita koje prepišu oba klijenta, vreme koje je potrebno da se dobije materijal koji koristimo za formiranje ključa u proseku iznosi 2 do 4 sekunde, a prosečna dužina niza koji predstavlja materijal ključ iznosi 400 bita.



Slika 2. Arhitektura Google Cloud Messaging sistema

KOMUNIKACIJA

Google Cloud Messaging za Android je besplatan servis koji omogućava programerima da šalju podatke sa servera do Android aplikacija na Android uređajima i da šalju podatke sa aplikacije do servera. Preko ovog servisa moguće je obostrano poslati obaveštenje o promeni nekog stanja ili poslati podatak veličine do 4 kb. Ovaj servis je zadužen i za čuvanje poruka i njihovo kasnije prosleđivanje Android aplikaciji pokrenutoj na određenom uređaju.

Arhitektura Google Cloud Messaging servisa

- *GCM* je server koji se nalazi u ovom servisu preuzima poruke sa lokalnog servera i prosleđuje poruke Android aplikaciji (klijentu).
- Lokalni server/*Smack CCS Client* je komponenta koju implementiramo kako bi radila sa izabranim *GCM* serverom. Lokalni server prosleđuje poruke *GCM* server, on ih stavlja u red i tako uređene ih čuva dok klijent ne uspostavi vezu sa Internetom.
- Klijentska aplikacija je Android aplikacija za koju je implementirana mogućnost korišćenja *Google Cloud Messaging* servisa. Da bi aplikacija primala poruke od *GCM* servera prvo je potrebno da se aplikacija registruje za korišćenje usluga *Google Cloud Messaging* servisa.

Smack CCS (XMPP) Server

Google Cloud Messaging (GCM), Cloud Connection Server (CCS) je krajnja tačka *XMPP (Extensible Messaging and Presence Protocol)* koja omogućava konstantnu, asinhronu, dvosmernu vezu sa *Google* serverom. Ista konekcija se koristi za slanje i primanje poruka između severa i Android uređaja. Neke od prenosti korišćenja ovog protokola su:

- Asinhroni pristup XMPP protokola omoguće da se sa manje utrošenih resursa šalje više podataka.
- Komunikacija je dvosmerna što znači da sever može da šalje poruke Android uređaju, ali isto tako i Android uređaj može da šalje poruke nazad serveru.
- Uređaj šalje poruke istom vezom preko koje je i primio poruku što uveliko smanjuje utrošak baterije pri slanju poruka.

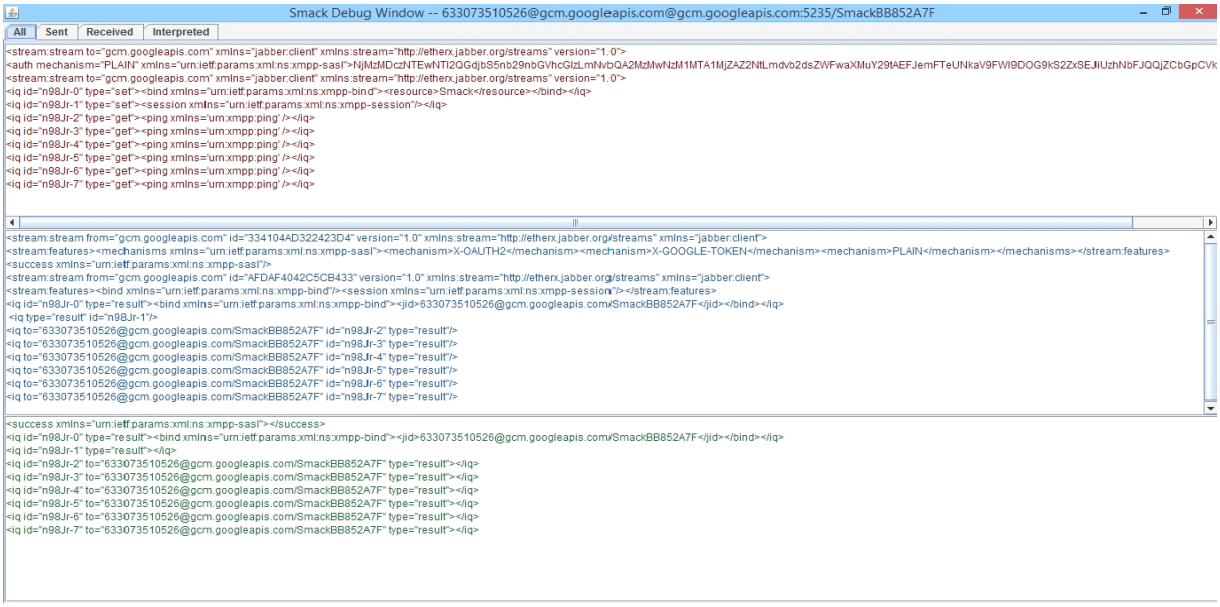
IMPLEMENTACIJA CCS/XMPP LOKALNOG SERVERA

Implementaciju ovog dela našeg modula odlučili smo da uradimo preko *NetBeans* okruženja. Za programski jezik izabrali smo Javu uz koju smo koristili *Smack* biblioteku. *Smack* je *XMPP (Jaber)* klijentska biblioteka otvorenog koda koja služi za razmenu podataka. Ova Java biblioteka se ugrađuje u aplikaciju i omogućava nam da napravimo kompletnu *XMPP* klijent aplikaciju ili da integrišemo neke osnovne funkcionalnosti ovog protokola, kao što su slanje obaveštenja, itd. Ovu bilbioteku povezujemo sa našom Java aplikacijom, tako što iz *NetBeans*-a dodamo (smack.jar) u folder pod nazivom *libraries* i tek tada možemo koristiti funkcionalnosti ove biblioteke. Nakon što smo podesili *Google Cloud Messaging* servis i dodali potrebne biblioteke, možemo krenuti sa predstavljanjem nekih od bitnijih delova koda našeg lokalnog servera.

Da bi smo povezali lokalni server sa servisima koje pruža *Google Cloud Messaging* potrebno je da obezbedimo ProjectID i SeverKey.

REZIME

Kao poboljšanje kripto modula planira se spajanje novonastale aplikacije sa već postojećim projektom šifrovanja sms poruka *One Time pad*-om na Android platformi. Spajanjem



Slika 3. Grafički izgled lokalnog servera

DOI: 10.15308/Synthesis-2015-81-84

ove dve aplikacije proširili bismo mogućnosti naše aplikacije i na sms komunikaciju. Još jedno planirano poboljšanje našeg modula je to da naše dvodimenzonalne matrice proširimo sa još jednom ili više dimenzija, zajednički ključ bi tada bio višedimenzionalni oblik do koga bi napadač još teže došao.

U ovom radu smo prošli kroz neka od pitanja vezanih za razmenu ključeva simetrične i asimetrične kriptografije i malo bolje se upoznali sa osnovnim načelima razmene ključeva. Razvojem kripto modula za razmenu poruka na Android platformi sa planiranim poboljšanjima dobijamo kompletno, samostalno rešenje razmene šifrovanih podataka koje ima primenu u skoro svim sferama našeg društva.

U veoma kratkom vremenskom periodu naš modul može generisati ključeve velikih dužina. U ovom radu razmenu ključeva smo primenili u aplikaciji za razmenu poruka ali ovaj modul se može primeniti pri razmeni bilo kog tipa informacijena raznim sistemima i platformama. Zbog velike dužine ključa mogu se primeniti razni kriptološki algoritmi koji bi odgovarali potrebama korisnika. Sa uspešno razmenjenim ključevima i šifrovanim sadržajem naše poruke, može se reći da je naš modul za razmenu poruka potpuno siguran.

Međutim, fleksibilnost i otvorenost Android sistema ostavlja veliku mogućnost da sami uređaji ili kod napisan od strane programera sadrži elemente koji bi narušili bezbednost našeg sistema. Pretpostavlja se da je kompanija *Samsung,* koja je jedna od vodećih proizvođača Android pametnih uređaja, na nekim modelima pametnih uređaja ostavljala takozvani bekdor (eng. *backdoor*) preko kojeg je moguće aktivrati mikrofon, kameru, imati uvid u geolokaciju ili pristupiti fajlovima na uređaju. U ovom slučaju treća strana bi lako mogla doći do izračunatog ključa ili imati uvid u karaktere poruke koje smo unosili dodirivanjem ekrana našeg uređaja.

Iz svega gore navedenog možemo zaključiti da samo ako smo u potupunosti samostalno razvili hardverski i softverski deo našeg modula možemo biti sigurni da je naš modul apsolutno bezbedan.

LITERATURA

Google Developers. (2015). Google Cloud Messaging for Android. Preuzeto 25. Avgusta 2014. sa https://developer.android.com/google/gcm/index.html

Maurer, U.M. (1993). Secret Key Agreement by Public Discussion. Information Theory, IEEE Transactions, 39(3), 733-742. DOI: 10.1109/18.256484

Milosavljević, M., & Adamović, S. (2013). *Osnovi teorije informacija i kodovanja*. Beograd: Univerzitet Singidunum.

Milosavljević, M., & Adamović, S. (2014). *Kriptologija 2.* Beograd: Univerzitet Singidunum.

Thawte. (2013). *History of Cryptograpy.* Preuzeto 10. Novembra 2014. sa http://book.itep.ru/depository/crypto/Cryptography_history.pdf

Veinović, M., & Adamović, S. (2013). *Kriptologija 1.* Beograd: Univerzitet Singidunum.

DOI: 10.15308/Synthesis-2015-81-84