



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације

ДИПЛОМСКИ – МАСТЕР РАД

Кандидат: Роман Павловић
Број индекса: E10821

Тема рада: Један приступ развоју заштићеног разговора на мобилним телефонима са оперативним системом Симбиан

Ментор рада: проф. Др Никола Теслић

Нови Сад, Јун 2009.



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР :	
Идентификациони број, ИБР :	
Тип документације, ТД :	Монографска документација
Тип записа, ТЗ :	Текстуални штампани материјал
Врста рада, ВР :	Дипломски – мастер рад
Аутор, АУ :	Роман Павловић
Ментор, МН :	проф. Др Никола Теслић
Наслов рада, НР :	Један приступ развоју заштићеног разговора на мобилним телефонима са оперативним системом Симбиан
Језик публикације, ЈП :	Српски / латиница
Језик извода, ЈИ :	Српски
Земља публикаовања, ЗП :	Република Србија
Уже географско подручје, УГП :	Војводина
Година, ГО :	2009
Издавач, ИЗ :	Ауторски репринт
Место и адреса, МА :	Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, ФО : (поглавља/страна/ цитата/табела/слика/графика/прилога)	
Научна област, НО :	Електротехника и рачунарство
Научна дисциплина, НД :	Рачунарска техника
Предметна одредница/Кључне речи, ПО :	Заштићени разговор, Симбиан, мобилна платформа, криптија
УДК	
Чува се, ЧУ :	У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, ВН :	
Извод, ИЗ :	
Датум прихватања теме, ДП :	
Датум одбране, ДО :	
Чланови комисије, КО :	Председник: др Михајло Катона
	Члан: др Небојша Пјевалица
	Члан, ментор: проф. Др Никола Теслић
	Потпис ментора



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, ANO :			
Identification number, INO :			
Document type, DT :	Monographic publication		
Type of record, TR :	Textual printed material		
Contents code, CC :	Master Thesis		
Author, AU :	Roman Pavlović		
Mentor, MN :	PhD Nikola Teslić		
Title, TI :			
Language of text, LT :	Serbian		
Language of abstract, LA :	Serbian		
Country of publication, CP :	Republic of Serbia		
Locality of publication, LP :	Vojvodina		
Publication year, PY :	2009		
Publisher, PB :	Author's reprint		
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)			
Scientific field, SF :	Electrical Engineering		
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems		
Subject/Key words, S/KW :	Secure call, Symbian, cryptography		
UC			
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, N :			
Abstract, AB :			
Accepted by the Scientific Board on, ASB :			
Defended on, DE :			
Defended Board, DB :	President:	PhD Mihajlo Katona	
	Member:	PhD Nebojša Pjevalica	Menthor's sign
	Member, Mentor:	PhD Nikola Teslić	

SADRŽAJ

1. Uvod.....	1
1.1 Osnovni zadaci rada	1
2. Opis platforme i razvojnog okruženja.....	4
2.1 Symbian mobilna platforma	4
2.2 Carbide.C++ razvojno okruženje	9
2.3 Opis Nokia E61 mobilnog uređaja	11
3. Realizacija.....	12
3.1 Opis principa rada zaštićenog razgovora	12
3.2 Primena principa rada na Symbian OS-u	14
3.3 Kontrolni modul SCAAppUi	19
3.4 Moduli korisničke sprege	22
3.4.1 SCAMainView	22
3.4.2 SCAMain	25
3.4.3 SCACallLogView	25
3.4.4 SCACallLog	27
3.4.5 CallLog	28
3.4.6 SCASettingsView	29
3.4.7 SCASettings.....	31
3.4.8 SCASettingsSettings.....	31
3.4.9 SavedSettings.....	32
3.5 Moduli glavnog mehanizma (Engine).....	32
3.5.1 SharedData.....	33
3.5.2 CRCChecksum.....	35

3.5.3	Crypt	36
3.5.4	DataCall	37
3.5.5	CommHandle	39
3.5.6	StreamerEngine.....	40
3.5.7	QueHandler.....	42
4.	Testiranje i verifikacija	43
5.	Zaključak	47
6.	Literatura.....	49
7.	Dodatak.....	50
7.1	Testni scenariji za ručno testiranje	50
7.2	Testovi za automatsko testiranje	64

SPISAK SLIKA

Slika 2.2.1 Modularni prikaz Symbian OS-a	5
Slika 2.2.2 Organizacija elemenata fizičke arhitekture.....	7
Slika 2.2.3 Arhitektura programskog procesora AP-a (SoC).....	9
Slika 2.2.4 Carbide C/C++ razvojno okruženje	10
Slika 2.2.5 Izgled mobilnog uređaja Nokia E61	11
Slika 3.3.1 Šematski prikaz principa zaštićenog razgovora.....	14
Slika 3.3.2 Podela programa po modulima i po celinama.....	17
Slika 3.3.3 Izgledi korisničkih sprega (<i>levo glavna korisnička sprega, u sredini sprega za podešavanje rada programa, desno sprega istorijata poziva</i>)	19
Slika 3.3.4 Dijagram stanja prelaza modula za kontrolu poziva.....	39
Slika 4.1 Pristupi testiranju (<i>levo ručno testiranje, desno automatsko</i>).....	43
Slika 4.2 Grafički prikaz performansi sistema za osnovni algoritam za šifrovanje.....	45

SKRAĆENICE

API	- <i>Application Programming Interface</i>
CPU	- <i>Central Processor Unit</i>
GSM	- <i>Global System for Mobile communications</i>
CSD	- <i>Circuit Switched Data</i>
DLL	- <i>Dynamic linking library</i>
OS	- <i>Operating system</i>
MMU	- <i>Memory management unit</i>
RTOS	- <i>Real-Time Operating System</i>
SDK	- <i>Software Development Kit</i>
PCM	- <i>Pulse Code Modulation</i>
AMR	- <i>Adaptive Multi-Rate</i>
APS	- <i>Audio Proxy Server</i>
API	- <i>Application Programming Interface</i>
UI	- <i>User Interface</i>
CRC	- <i>Cyclic Redundancy Check</i>
DES	- <i>Data Encryption Standard</i>
AES	- <i>Advanced Encryption Standard</i>
RAM	- <i>Random access memory</i>
FR	- <i>Full rate</i>
AMR-NB	- <i>Adaptive Multi-Rate - Narrow Band</i>
EDGE	- <i>Enhanced Data Rates for GSM Evolution</i>
WiFi	- <i>Wireless</i>
MSC	- <i>Message Sequence Charts</i>

1. Uvod

Razvojem tehnologije tokom poslednjih godina došlo je do sve veće upotrebe mobilnih uređaja. Mobilni uređaji su postali standard za govornu komunikaciju. Komunikacija po definiciji podrazumeva proces prenosa informacija od jednog izvora do drugog posredstvom komunikacionog kanala ili medijuma. U kontekstu zaštićenog razgovora, iz definicije komunikacije se izdvajaju dva bitna pojma, informacija i komunikacioni kanal.

Informacije poseduju veliku moć kada se koriste na pravi način. Svi imaju potrebu da čuvaju informacije tajnim, od drugih ljudi, organizacija, kompanija pa sve do država. Postoje razni načini da se dođe do poverljivih informacija, a jedan od najprisutnijih načina je prisluškivanje telefonskih razgovora. Prisluškivanje se vrši presretanjem informacija na komunikacionom kanalu.

Cilj ovog rada je da opiše jedan način zaštite razgovora na mobilnim GSM (Global System for Mobile communications) mrežama koristeći CSD (Circuit Switched Data) kanal za prenos podataka (data call).

1.1 Osnovni zadaci rada

Osnovni zadatak rada je da se na bazi postojeće podrške na mobilnim telefonima sa operativnim sistemom Symbian S60 treća edicija (inicijalna verzija) realizuje zaštićen razgovor koristeći poziv za prenos podataka. Podrška treba da se realizuje u vidu zasebnog programa. Korisnička sprega programa je potrebno da obezbedi:

Izgled sličan matičnom programu za uspostavljanje poziva

- Pristup bazi kontakata
- Evidenciju poziva (istorijat poziva)
- Podešavanja rada programa

U okviru same podrške, potrebno je obezbediti:

- Uspostavljanje poziva
- Sam tok poziva
- Prekid poziva za prenos podataka

Preko zaštićenog poziva, treba da se prenose tokovi podataka koji se preuzimaju sa audio ulaza na telefonu, dok podaci koji se primaju u toku razgovora, treba da se prenose na audio izlaz na telefonu. Potrebno je primeniti i jednostavan algoritam za šifriranje.

Izgled glavnog ekrana programa treba da daje uvid u stanje programa, to jest stanje poziva. Traka sa menijima treba da sadrži dva dugmeta, jedno za slanje programa u pozadinu i jedno za prikaz menija sa opcijama. Sa glavnog prozora potrebno je obezbediti opcije:

- Unos novog broja koji se želi pozvati
- Izabir telefonskog kontakta iz baze kontakata
- Prikaz ekrana sa podešavanjima
- Prikaz informacija o verziji programa

Prilikom uspostave poziva i samog toka poziva opcije u meniju treba da se menjaju u skladu sa stanjem programa. Prilikom dolazećeg poziva je potrebno da se obezbedi opcija za prihvatanje poziva, dok je prilikom samog razgovora potrebno da se obezbedi opcija za prenos izlaznog audio toka na zvučnik i prekid veze.

Programom treba da se obezbedi praćenje istorije poziva koja je nezavisna od podrazumevanog istorijata poziva samog telefona. Ova usluga programa obezbeđuje dodatnu zaštitu informacija o izvršenim pozivima. Korisnička sprega treba da prikaže listu poziva sortiranu po vremenu poziva i da da informaciju o tipu poziva. Treba da se razlikuju tri tipa poziva: primljeni, zvani i propušteni pozivi. Opcije treba da omoguće korisniku da pozove telefonski broj iz liste, da obriše stavku iz istorijata ili da obriše celu listu istorijata poziva.

Podešavanja rada programa treba da obezbede korisniku mogućnost da podesi parametre korišćenog šifrovanja. Treba da obezbedi izbor algoritma, ručno unošenje ključa ili autogenerisanje ključa. Dodatne opcije podešavanja su nivo jačine zvuka i izbor da li se program pokreće sa pokretanjem samog sistema. Podešavanja sistema se čuvaju u datoteci. Datoteka treba da je zapisana u binarnom obliku kako ne bi bila čitljiva.

Poseban zahtev programa je da bude aktivan svo vreme rada sistema. Kako bi se obezbedilo nesmetano korišćenje mobilnog uređaja, potrebno je da se obezbedi mogući rad

programa u pozadini i po potrebi mogućnost da se prikazuje u prvom planu (prilikom dolazećeg poziva za šifrovani razgovor).

Sama programska podrška, zaštićenog razgovora, treba da se realizuje modularno. Modul za kontrolu rada modema, modul za rad sa Audio tokovima podataka i modul za kriptovanje podataka. Algoritmi za šifrovanje treba da se realizuju u vidu posebnih Symbian DLL biblioteka koje se dimački učitavaju u program. Princip rada glavnog mehanizma za kontrolu zaštićenog razgovora je opisan u poglavlju Realizacija.

Program je potrebno da se realizuje za mobilne uređaje sa operativnim sistemom Symbian treća edicija (verzije 9.1, 9.2, 9.3).

Rad je organizovan u šest poglavlja.

1. Poglavlje “Uvod” sadrži opis zadatka rada.
2. U poglavlju “Opis platforme i razvojnog okruženja” su izloženi osnovni koncepti korišćene platforme i razvojnog okruženja.
3. U poglavlju “Realizacija” je opisana arhitektura programa. Dat je pregled i objašnjenje osnovnih klasa i njihovih funkcija.
4. Poglavlje “Ispitivanje i verifikacija” se bavi ispitivanjem očekivanog rada programa.
5. Poglavlje “Zaključak” ukratko daje osvrt na realizovani sistem, koncept zaštićenog razgovora i dobijene rezultate.
6. Poglavlje “Literatura” sadrži literaturu koja je korišćena prilikom izrade rada.
7. Poglavlje “Dodatak” sadrži tabele testnih slučajeva i MSC dijagrame za automatsko testiranje.

2. Opis platforme i razvojnog okruženja

Ciljna platforma korišćena za razvoj programske podrške je Symbian 9.1 treća edicija (inicijalna verzija). Symbian mobilna platforma je izabrana iz dva razloga. Prvi je široka rasprostranjenost na tržištu pametnih telefona. Postoji veliki broj uređaja sa ovim operativnim sistemom, a konkretno tokom razvoja je korišćen uređaj Nokia E61. Drugi razlog je stabilnost sistema. Činjenica da je Symbian prisutan na tržištu već 9 godina i da pokriva polovinu tržišta pametnih tefona dovoljno govori o njegovoj stabilnosti [11]. U nastavku poglavlja je dat pregled Symbian platforme, skupa alata za razvoj i uvid u karakteristike korišćenog mobilnog uređaja.

2.1 Symbian mobilna platforma

Symbian operativni sistem je nastao 1998 godine. Napravili su ga Nokia, Motorola, Psion i Erricsson a kasnije im se priključio i Siemens. Nastao je iz EPOC operativnog sistema koji je razvio Psion. EPOC je bio prvi OS napravljen namenski za mobilne uređaje. EPOC je 32bitni operativni sistem koji podržava višenitni rad. Nastao je iz želje proizvođača mobilnih uređaja da se napravi univerzalni operativni sistem kojeg bi svi mogli koristiti na različitim uređajima i platformama [4] .

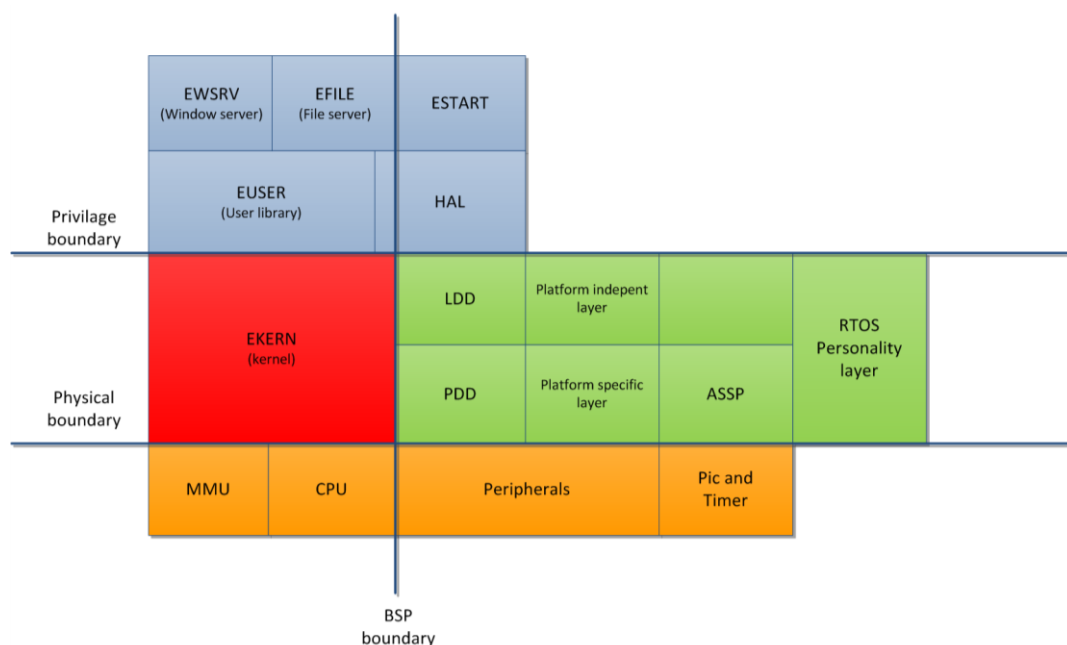
Symbian OS je modularno koncipiran operativni sistem. Svi nivoi sistema su modularni od jezgra do programskih sprega najvišeg nivoa. Tako na primer upravljenje diskom obavlja poslužitelj datoteka (file server), dok recimo upravljanje grafičkom spregom je zadatak poslužitelju prozora (window server). Međutim, postoji jedan modul koji je glavni i koji rukuje memorijom, kontroliše i preraspoređuje zadatke a to je jezgro (kernel), EKA2.

EKA2 jezgro [3] je jednokorisničko što znači da ne postoji koncept prijavljivanja na sistem više korisnika kao što je to slučaj kod sistema koji su namenjeni PC računarima Windows, Linux... Jezgro podržava višeprocesni rad. Prebacuje procesorsko vreme između zadataka niti koje rade istovremeno. EKA2 jezgro je zasnovano na nivoima prioriteta tako da zadaci sa višim

nivoima prioriteta imaju obezbeđenu naklonost procesora, i imaju prednosti u odnosu na zadatke nižeg prioriteta. Jezgro obezbeđuje rad u realnom vremenu što garantuje zadacima da se izvršavaju u konačnom vremenu.

Arhitektura Symbian operativnog sistema se sastoji iz više glavnih modula. Prikaz organizacije jezgra sistema se može videti na Slici 2.2.1.

1. **Nanojezgro (nanokernel)** – Glavna uloga je da obezbedi jednostavnu podršku za niti sa operacijama za njihovu sinhronizaciju i zakazivanje. Ime, nanojezgro, je dato jer su operacije koje radi jednostavnije od većine operativnih sistema u realnom vremenu. Nanojezgro obezbeđuje početno rukovanje izuzecima. Izuzetke dalje prosleđuje različitim nivoima na dalju obadu. Takođe pruža jednostavne funkcije za rad sa vremenskim kontrolama. Obezbeđuje objekte za sinhronizaciju, mutekse i semafore na nivou samog nanojezgra.
2. **Symbian OS jezgro** – Pruža funkcije potrebne za rad ostatka sistema Symbian OS-a. Zasniva se na nanojezgru kako bi obezbedio složenije objekte kao što su korisničke niti, procesi, indentifikatori (handle), dinamički učitane biblioteke, objekti za međunitnu komunikaciju i još mnoge druge. Obezbeđuje objekte za sinhronizaciju: Symbian OS semafori i muteksi. Semafori koje obezbeđuje jezgro su standardni brojački semafori koji podržavaju zadržavanje više niti i koje oslobađaju niti u redosledu u zavisnosti od njihovih prioriteta. Za razliku od nanojezgra, Symbian OS jezgro dozvoljava dinamičko zauzimanje memorije. Pruža komponentu jezgra za zauzimanje memorije – dinamičke memorije jezgra, koja koristi usluge modula poznatog kao model memorije.

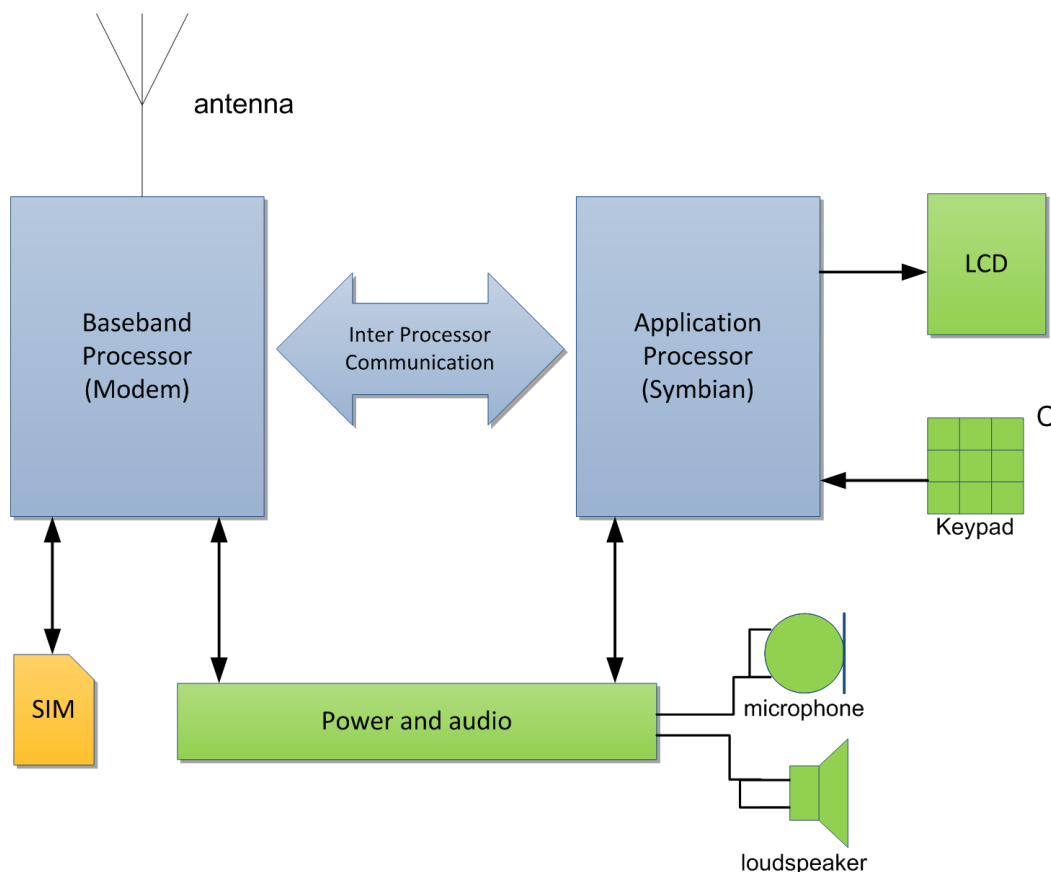


Slika 2.2.1 Modularni prikaz Symbian OS-a

3. **Model memorije** – Jezgro EKA2 sistema se za rukovanje memorijom oslanja na ovaj modul. Modul sakriva memorijsku arhitekturu ASIC-a od jezgra. Tako da memorijski model obuhvata bitne MMU razlike, kao što je razlika između tipa virtualno ili fizički označene keš memorije, ukoliko sistem uopšte poseduje MMU jedinicu. Symbian trenutno podržava tri memorijska modela: direktni (nema MMU), kretajući (slično starijoj verziji jezgra EKA1), višememorijski model (korišćeni kod ASICa sa fizički označenim kešom kao što su Intel x86 i noviji ARM procesori)
4. **Nivo osobenosti** (personality layer) – Nanojezgro je dizajnirano da pruži samo dovoljno funkcionalnosti da se pokrene GSM stek. Ideja koja stoji iza toga je da se proizvođačima dozvoli pokretanje i GSM steka i njihovog rukovaoca ličnim podacima (PIM – personal information management) na istom procesoru. Većina današnjih proizvođača je napisala signalne stekove za postojeće RTOS sisteme kao što su Nucleus ili uITRON. Nivo osobenosti je emulacioni nivo koji stoji iznad nanojezgra kako bi pružio RTOS API-je klijentskim programima.
5. **ASSP/promenljivi dodatak** – Tipično, CPU i razni elementi fizičke arhitekture na mobilnim uređajima su napravljeni u jednom integrisanom čipu. Ovakvi čipovi se oslovljavaju sa ASSP (Application-Specific Standard Product). U starijoj verziji jezgra podrška za ASSP je bila integrisana u jezgro što je zahtevalo ponovno prevođenje jezgra za svaki novi procesor. Ovakav pristup omogućava lako prilagođavanje sistema novom procesoru bez izmena na jezgru sistema.
6. **Upravljački programi** – Služe za povezivanje perifernih elemenata fizičke arhitekture sa ostatkom Symbian operativnog sistema.
7. **Dodaci (Extensions)** – Dodaci su samo upravljački programi koje jezgro automatski pokreće tokom perioda podizanja sistema, tako da se na njih može gledati samo kao na proširenja jezgra. Na primer, modul za otklanjanje grešaka (debugger) je proširenje jezgra koje se može isključiti iz sistema bez ponovnog prevođenja jezgra. Nakon podizanja dodataka, koje se dešava rano u procesu pokretanja sistema, sistem nastavlja normalno sa procesom podizanja sistema sve dok se ne aktivira nit za superviziju koja pokreće ostale dodatke sistema. Dodaci koje učitava nit za superviziju nisu kritični za rad jezgra.
8. **EUSER** – Omogućava spregu između korisničkih biblioteka i samog jezgra. Razlikuju se tri tipa funkcija koje nudi:
 - Metode klase iz biblioteka koje se izvršavaju u celosti na strani klijenta, kao što su metode nizova i deskriptora.

- Funkcije za pristup programskim spregama jezgra u privilegovanom režimu kao što je provera vremena ili lokalnih podešavanja.
 - Funkcije kojima se od jezgra zahteva da manipuliše sopstvenom memorijom kako što je stvaranje procesa i učitavanje dinamičkih biblioteka.
9. **Poslužitelj datoteka** (file server) – Poslužitelj koji radi u klijentskom modu, kako bi omogućio korisničkim nitima da kontrolišu diskove, direktorijume i datoteke.
10. **Poslužitelj za kontrolu prozora** (window server) – Poslužitelj koji omogućava pristup korisničkoj grafičkoj sprezi i tastaturi.

Telefoni bazirani na Symbian operativnom sistemu su pre svega napravljeni da ostvaruju visok kvalitet glasovne komunikacije i dugo vreme trajanja baterije, a tek posle toga da podržavaju nove usluge i programe. Sagledavajući arhitekturu uređaja koji podržavaju Symbian operativni sistem dolazi se do dve celine koje zavise jedna od druge. To su mobilna radio sprega procesora osnovnog polja (BP – baseband processor) to jest GSM modema i programskog procesora (AP – application processor) koji pokreće korisničku spregu i programe višeg nivoa. Ove dve celine okružuje niz periferija kao što su: baterija, zvučnici, SIM kartica... Na Slici 2.2.2 se može videti uobičajna realizacija organizacije platforme gde se AP i BP nalaze u zasebnim integrisanim kolima koja su povezana međuprocessorskom vezom velike brzine (IPC).



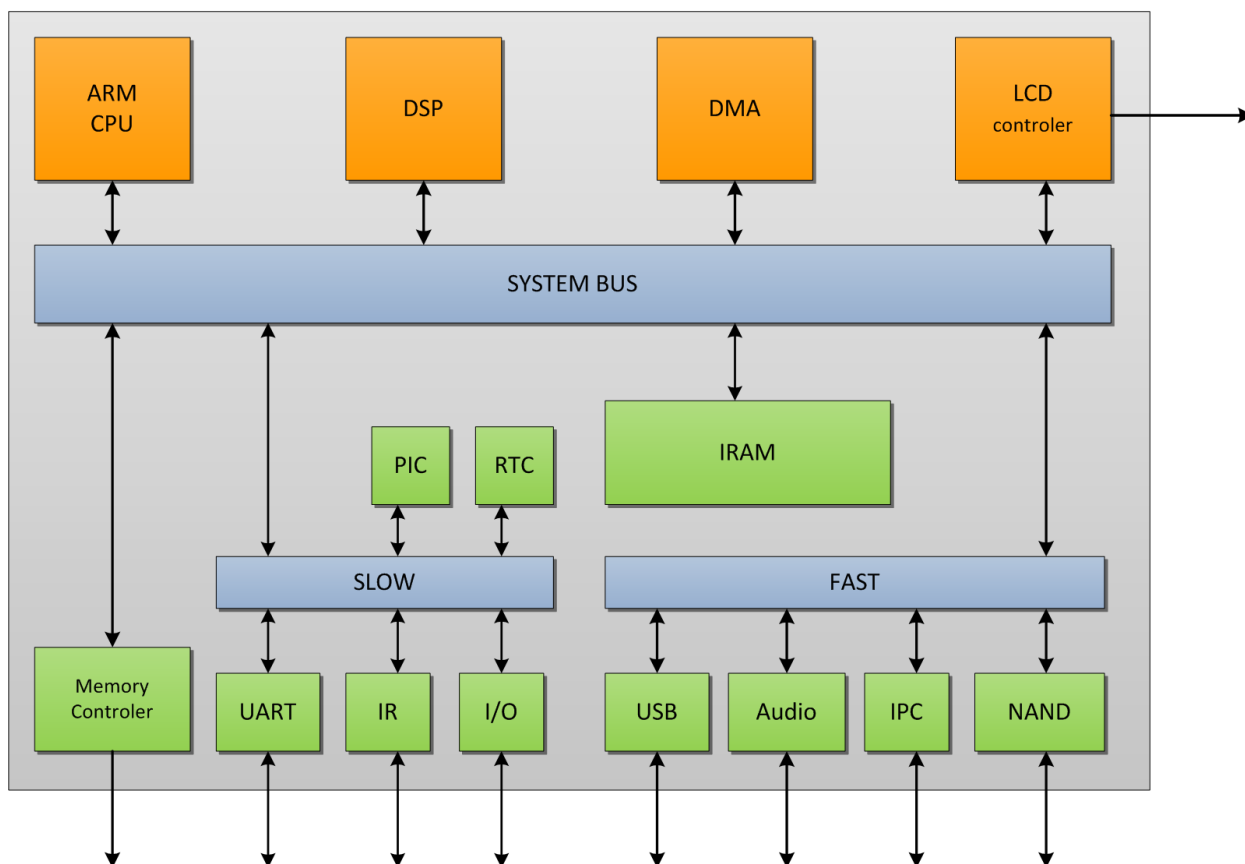
Slika 2.2.2 Organizacija elemenata fizičke arhitekture

Obe celine, AP i BP, isključuju korišćenje usluga druge u svom radu. BP zahteva programsku podršku za rad u realnom vremenu, periodično upravljanje napajanjem i obezbeđuje sigurnost na mreži. AP očekuje da radi u jednom od dva moda – pod punom snagom kada korisnik vrši interakciju sa uređajem ili u modu dubokog sna (deep sleep) kada se telefon nalazi u stanju mirovanja. AP sadrži okruženje i biblioteke koje služe za podršku ugrađenim programima i programima nezavisnih proizvođača.

1. **Procesor osnovnog pojasa** (Baseband processor) – je modem za prenos podataka i govora na telefonu. Sadrži elektroniku potrebnu za rad na 2.5G i 3G mobilnim mrežama, DSP procesor sa algoritmima za kodovanje i dekodovanje informacija i procesor na kojem radi stek za kontrolu poziva. Procesor sa jedne strane komunicira sa baznim stanicama na mreži, a sa druge sa AP procesorom. Programska podrška koja radi na BP se naziva telefonski stek, ili samo skraćeno Stek. Telefonski stek je složen sistem koji se razvijao sa razvojem telefonskih standarda. Tipičan telefonski stek sadrži između 2 i 8 MB koda, i zahteva do 2MB memorije RAM-a za izvršavanje. BP rukovalac napajanjem je visoko optimizovan za što duže vreme trajanja poziva uz minimalnu potrošnju baterije. Ovo podrazumeva da u stanju mirovanja BP prelazi u mod u kome se svake 2 sekunde proveravaju obaveštenja sa mreže, za razliku od moda kad je aktivan kada je to vreme svedeno na 4.6 ms. IPC sprega je znatno unapređena u odnosu na ranije verzije Symbiana, od jednostavnog serijskog prolaza do veze sa protokolom od 10Mb/s. Ova veza može da sadrži pet kanala. Kanal za kontrolu telefona, kontrolu sistema, podataka, tokova podataka i BP debagera. Audio podaci se usmeravaju, od BP-a posećenom digitalnom audio magistralom, direktno do audio elemenata fizičke arhitekture mikrofona i slušalice odnosno zvučnika. Ova magistrala svodi kašnjenja na minimum sa zagaranovanim radom u realnom vremenu i niskom potrošnjom, tako što zaobilazi AP. Ako bi govorni podaci prolazili kroz AP to bi uvelo dodatna kašnjenja zbog prenosa preko IPC magistrale i baferovanja podataka a povećala bi se i potrošnja baterije. BP kontrološe SIM karticu koja sadrži tajne informacije (kao što je IMSI broj) i algoritme potrebne za prijavljivanje na mrežu.

2. **Programski procesor** (AP – application processor) – je srce telefona sa Symbian operativnim sistemom. AP je primer organizacije „Sistem na čipu“ (SoC – System-on-Chip). Sadrži ARM CPU, memoriju, kontroler rada ekrana, audio spregu, multimedijalne ubrzivače i mnoge druge periferije. SoC je poznat pod još dva imena, ASCI (Application-specific Integrated Circuits) za namenski prilagođene čipove i ASSP (Application-specific Semiconductor Parts) za komercijalne čipove. SoC su dizajnirani i proizvedeni od strane više glavnih kompanija za proizvodnju čipova: Texas Instruments ima svoju familiju nazvanu OMAP, dok Intel ima njihovu familiju proizvoda XScale. Svaka potkomponenta čipa je posebna intelektualna svojina (IP – intellectual property). Svi IP blokovi su povezani vezama sa magistralama preko

standardnih industrijskih sprega. Najpoznatija kompanija za licenciranje IP-a je ARM koja licencira ARM CPU jezgra. Na Slici 2.2.3 se može videti blok šema ARM 926 procesora. Detaljan opis rada Symbian OS se može naći u knjizi [3] iz poglavlja Literatura.



Slika 2.2.3 Arhitektura programskog procesora AP-a (SoC)

2.2 Carbide.C++ razvojno okruženje

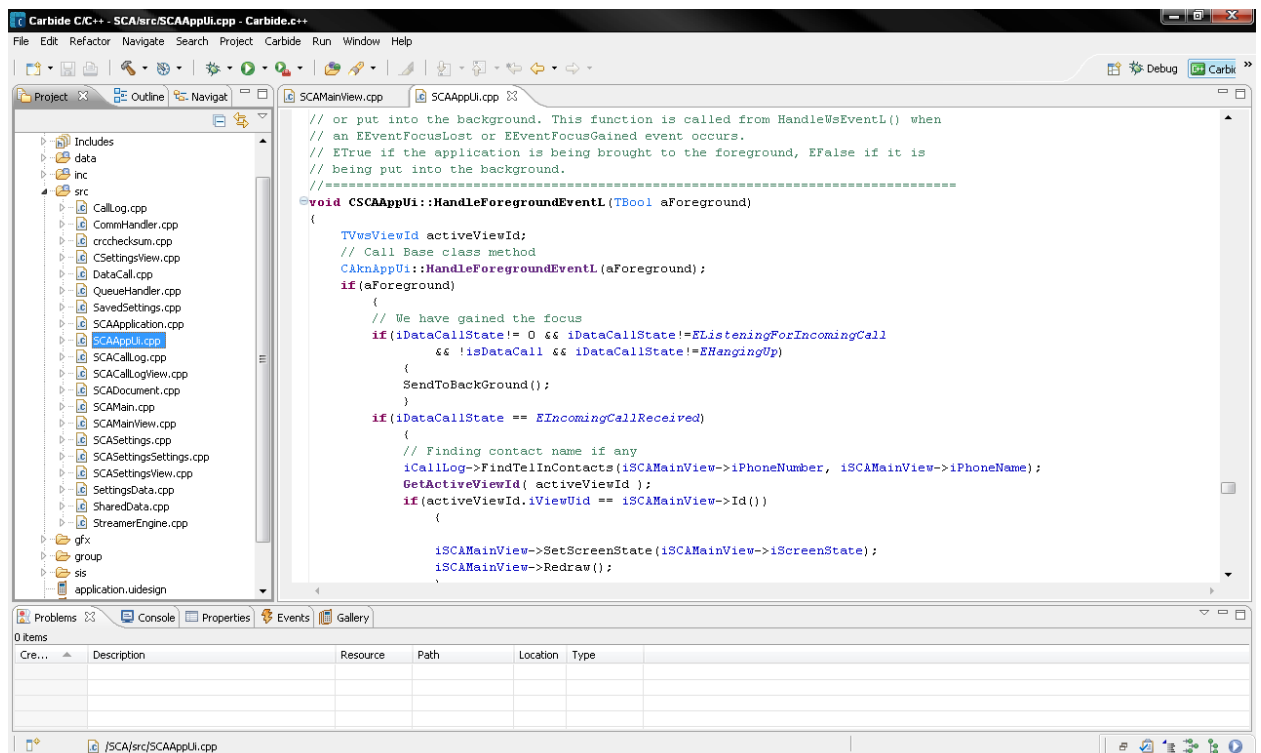
Carbide.C++ se zasniva na Eclipse platformi, koja je pokrenuta u Novembru 2001. godine od strane IBM-a i još sedam drugih kompanija, kao projekat otvorenog koda sa ciljem da se napravi besplatno i višepatformsko integrisano razvojno okruženje (IDE). Carbide.C++ IDE se zasniva na Eclipse IDE verziji 3.3 sa dodatnim modulima koji omogućavaju Eclipse-u da rukuje Symbian projektima. Pošto se Carbide.C++ zasniva na Eclipse-u, on se može dodatno prilagoditi dodavanjem dodatnih modula, kao što je sistem za kontrolu verzija projekata, UML sistem za modelovanje...

Instalacija razvojnog okruženja je jednostavna jer dolazi u jednoj zasebnoj Windows instalacionoj datoteci. Izgled okruženja se velikim delom podudara sa standardnim Eclipse IDE-om što se može videti na slici 2.2.4.

Pošto je Carbide.C++ razvijen da rukuje Symbian C++ projektima, standardno se sa leve strane nalazi C/C++ pogled na projekat (Project view), koji je dizajniran da na pregledan način prikaže izvorne, projektne i resursne datoteke. U sredini IDE-a se nalazi editor koda. Sa donje strane razvojnog okruženja se nalazi više prozora od kojih su najbitniji Problems, Console, Properties i Events. Prethodno opisani izgled se odnosi na osnovnu perspektivu. Pored ove, Carbide.C++ pruža i niz drugih, kao što su Debug (za ispravljanje grešaka), Qt, C++ i SVN Synchronizing. Svaka od ovih perspektiva se može dodatno prilagoditi.

Skup alata za razvoj programske podrške (SDK) je potrebno instalirati nezavisno od Carbide.C++ razvojnog okruženja. Korišćena je verzija 9.1 Maintenance Release jer programi razvijani sa ovim skupom alata rade na svim verzijama Symbiana treće edicije. Uz skup razvojnih alata dolaze Emulator, GCC i ARMv5 kompajleri i alati za rad sa grafičkim komponentama.

Razvoj programa za Symbian je znatno olakšan uslugom ispravljanja grešaka na fizičkoj arhitekturi. To podrazumeva da se kroz kod programa može prolaziti koračno dok se program izvršava na mobilnom uređaju. Da bi telefon mogao da se poveže sa Carbide.C++ radnim okruženjem potrebno je na mobilni uređaj instalirati program TRK, a u samom okruženju podesiti parametre veze sa mobilnim uređajem.



Slika 2.2.4 Carbide C/C++ razvojno okruženje

2.3 Opis Nokia E61 mobilnog uređaja

Nokia E61 je telefon namenjen poslovnim korisnicima, što znači da je posebna pažnja obraćena na komunikaciju (načine povezivanja), unos teksta, brzinu rada i odziv, trajanje baterije i kvalitet izrade. Na tržištu se pojavio oktobra 2005, kao prvi Nokijin telefon iz ovog segmenta mobilnih uređaja.

Telefon se zasniva na operativnom sistemu Symbian treće edicije (9.1 3rd edition). Sa stanovišta fizičke arhitekture telefon je zapakovan u aluminijumsko kućište dimenzija 117x40x14mm. Telefon je znatno širi od većine uređaja na tržištu čime je stvoren prostor da se ugradi cela QWERTY tastatura i ekran dimenzije 2.9 inča. Ekran je rezolucije 320x240 sa paletom od 16 miliona boja. Uređaj poseduje zvučnik koji se nalazi na gornjoj strani telefona čime je izbegnuta pojava mikrofonijske usled blizine mikrofona. Jezgro fizičke arhitekture čini Texas Instruments OMAP ARM 9 procesor sa dva jezgra i radnim taktom na 220MHz. Sistemska memorija je veličine 128MB od kojih je 64MB RAM. Nakon podizanja Symbian operativnog sistema ostaje maksimalno slobodno 22MB memorije. Telefon poseduje dodatne elemente fizičke arhitekture za obradu zvuka i grafike. Podržani kodeci na nivou fizičke arhitekture su AMR, FR, NB-AMR čime je smanjeno opterećenje procesora prilikom obrade govora. Telefon omogućava proširivanje memorije dodavanjem miniSD memorijske kartice. Snaga uređaja se ogleda u raznovrsnoj podršci za povezivanje sa drugim uređajima. Podržava tehnologije GPRS, CSD, EDGE, WiFi, 3G, Bluetooth i Infrared. Veliki ekran i mogućnost višestrukog povezivanja zahtevaju jak izvor napajanja koji se dobija iz litijum-jonske baterije kapaciteta 1500mAh.



Slika 2.2.5 Izgled mobilnog uređaja Nokia E61

3. Realizacija

Program zaštićenog razgovora je razvijen iz realne potrebe korisnika da ima obezbeđenu privatnost prilikom obavljanja razgovora. Analizom mogućnosti za ostvarivanje ove ideje razmotrena su dva aspekta. Prvi aspekt je izbor platforme. Izabrana je platforma Symbian OS. Razlozi su široka rasprostranjenost na tržištu pametnih mobilnih telefona sa jedne strane i kvalitetna i dobro dokumentovana programska podrška sa druge strane. Drugi aspekt se odnosio na nosioca glasa preko mreže. Izabrana je linija za prenos podataka (CSD data call) koja ostavlja mogućnost korisniku da prenosi digitalizovani glas brzinom od 9600bit/sec [5] [6]. Za razvojno okruženje je izabran Carbide C/C++. Okruženje je zasnovano na Eclipse IDE-u, što pruža udoban rad sa raznovrsnim pogodnostima Eclipse dodatka za rad, kao što su alati za ispravljanje grešaka i praćenje memorije.

U sledećem poglavlju je dat prikaz koncepta funkcionisanja ovog sistema.

3.1 Opis principa rada zaštićenog razgovora

Programska podrška se realizuje kao korisnički program koji ima privilegije pristupa zaštićenim resursima mobilnog uređaja. Potrebne su privilegije za pristup ulaznim i izlaznim tokovima podataka, privilegije za pristup GSM modemu i prava pristupa elementu fizičke arhitekture za obradu audio signala sa podrškom za AMR (Adaptive Multi Rate) kodek.

Govor korisnika sa mikrofona se dobija direktnim pristupanjem ulaznom audio toku podataka. Potrebno je obezbediti da program zaštićenog razgovora ima privilegovano pravo pristupa ovom toku podataka. Neprivilegovan pristup ulaznom audio toku podataka može dovesti do toga da drugi program na mobilnom uređaju u toku zaštićenog razgovora snima govor dobijen sa mikrofona. Preporučljivo je čitati ulazni audio tok podataka u okvirima dužine 20ms što preračunato u količini bajtova iznosi 320 16-tobitnih PCM odbiraka [2].

$$\mathcal{S} = \mathcal{C} \cup \mathcal{O} \cup \mathcal{D} \cup \mathcal{B} \cup \mathcal{E} \cup \mathcal{F} \cup \mathcal{G} \cup \mathcal{H} \cup \mathcal{I} \cup \mathcal{J} \cup \mathcal{K} \cup \mathcal{L} \cup \mathcal{M} \cup \mathcal{N} \cup \mathcal{O} \cup \mathcal{P} \cup \mathcal{Q} \cup \mathcal{R} \cup \mathcal{S} \cup \mathcal{T} \cup \mathcal{U} \cup \mathcal{V} \cup \mathcal{W} \cup \mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z} \quad (1)$$

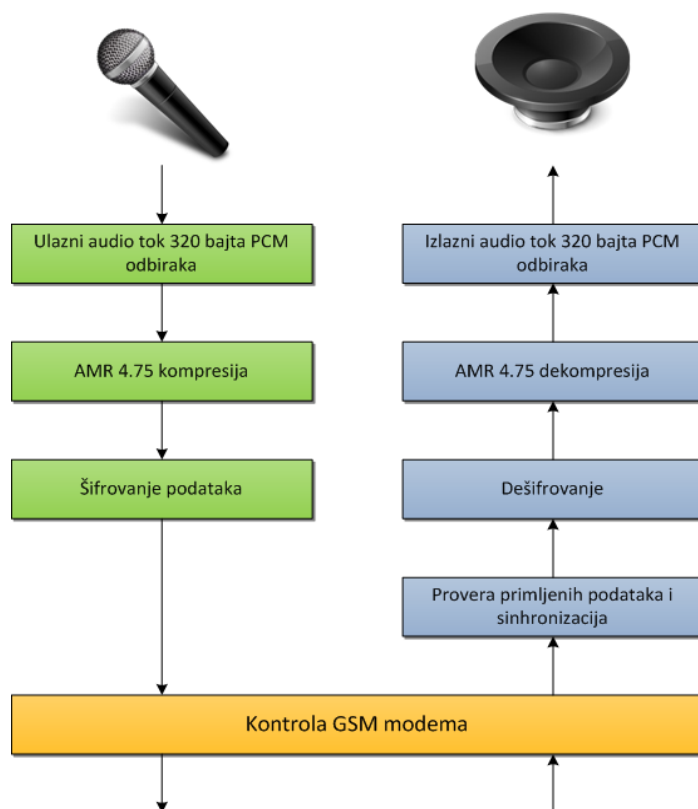
Prikupljeni odbirci govora se kompresuju koristeći AMR algoritam. Algoritam je namenjen mobilnoj telefoniji i prilagođen je kompresiji govora. Izabran je AMR 4.75 model AMR algoritma za kompresiju. AMR 4.75 mod AMR algoritma je odabran zato što ima nominalni protok od 4.75 kb/s. Izabrani mod AMR-a, sa lošijim kvalitetom i manjim protokom, je nametnut korišćenjem CSD poziva za prenos podataka koja ima malu propusnu moć od 9.6 kb/s. AMR 4.75 ima stepen kompresije reda veličine 25. Jedan okvir od 320 bajta PCM odbiraka (20ms govora) kompresuje u okvir od 13 bajta.

$$13bajta * \frac{1}{0.02sec} = 5200bita/sec \quad (2)$$

Nakon kompresije podaci se šifruju. Algoritam za šifrovanje nije striktno određen ovim radom. Mora se voditi računa da algoritmi koji bolje šifruju zahtevaju više procesorskog vremena i uvode složeniju logiku sinhronizacije, što može da uvede znatna kašnjenja u ceo sistem. Treba obratiti pažnju na to da algoritmi za šifrovanje koji rade nad blokovima zahtevaju blokove fiksne dužine, pri čemu je neophodno prilagoditi kompresovane okvire govora toj veličini.

Nakon šifrovanja podataka, podaci se šalju sagovorniku preko CSD poziva za prenos podataka. Poziv za prenos podataka je podržan od strane svih mobilnih operatera. Ranije je korišćen za povezivanje uređaja na internet, pri čemu se ostvaruje stalna veza između dva uređaja, zauzimajući jedan vremenski kanal. Vezu ostvarenu na ovaj način uređaji vide kao serijsku vezu, pri čemu je maksimalni mogući protok podataka 9.6 kbit/sec u punom dupleks režimu.

Paralelno sa slanjem podataka izvršava se prijem podataka šifrovanog govora sagovornika. Podaci se primaju i proverava se validnost primljenih podataka. U zavisnosti od tipa CSD veze, transparentne ili netransparentne, potrebno je uvesti proveru validnosti primljenih podataka. Ukoliko se koristi transparentni mod moguće je da paketi ne budu primljeni u celosti usled gubitaka u mreži u toku prenosa, do i od bazne stanice. Okviri unutar paketa koji nisu validni se odbacuju. Dozvoljen je gubitak od 100ms (ljudsko uho ne može da primeti razliku u kvalitetu govora) što predstavlja 5 okvira razgovora. Primljeni paketi se dešifruju. Dešifrovani paketi govora se dekompresuju iz AMR formata u PCM format, koji se dalje pušta na izlazni audio tok podataka mobilnog uređaja, to jest zvučnik. Na slici Sl.3.1.1 se može videti šematski prikaz principa rada.



Slika 3.3.1 Šematski prikaz principa zaštićenog razgovora

Potrebno je da oba učesnika u razgovoru imaju korisničke programe realizovane na istom principu. Ovo omogućava da uređaji koji koriste različite mobilne sisteme ostvaruju zaštićeni razgovor međusobno na isti način.

Kritični deo ovog principa je algoritam za šifrovanje. Algoritam za kriptovanje se realizuje kao zasebni modul koji se dinamički uključuje u sistem. Svi moduli treba da imaju istovetne programske sprege kako bi program za šifrovanje mogao da ih koristi. Dve stvari obezbeđuju sigurnost poziva. Prva je zaštita prilikom inicijalne uspostave poziva, prilikom koje treba obezbediti sigurnu razmenu ključeva. Drugi je algoritam za šifrovanje. Algoritmi za šifrovanje su blokovskog tipa, što znači da su paketi određene dužine i da se šifruju unapred dogovorenim ključem iste dužine.

3.2 Primena principa rada na Symbian OS-u

Program sigurnog poziva SCA je realizovan kao korisnički program koji ima pristup privilegovanim resursima Symbian operativnog sistema. Privilegije koje su potrebne za rad programa omogućavaju pristup korisničkim podacima i izmenu korisničkih informacija, pristup GSM modemu telefona, pristup informacijama o pozivu, pristup elementu fizičke arhitekture za rad sa izlaznim i ulaznim audio tokovima, pristup sistemskim funkcijama za šifrovanje i

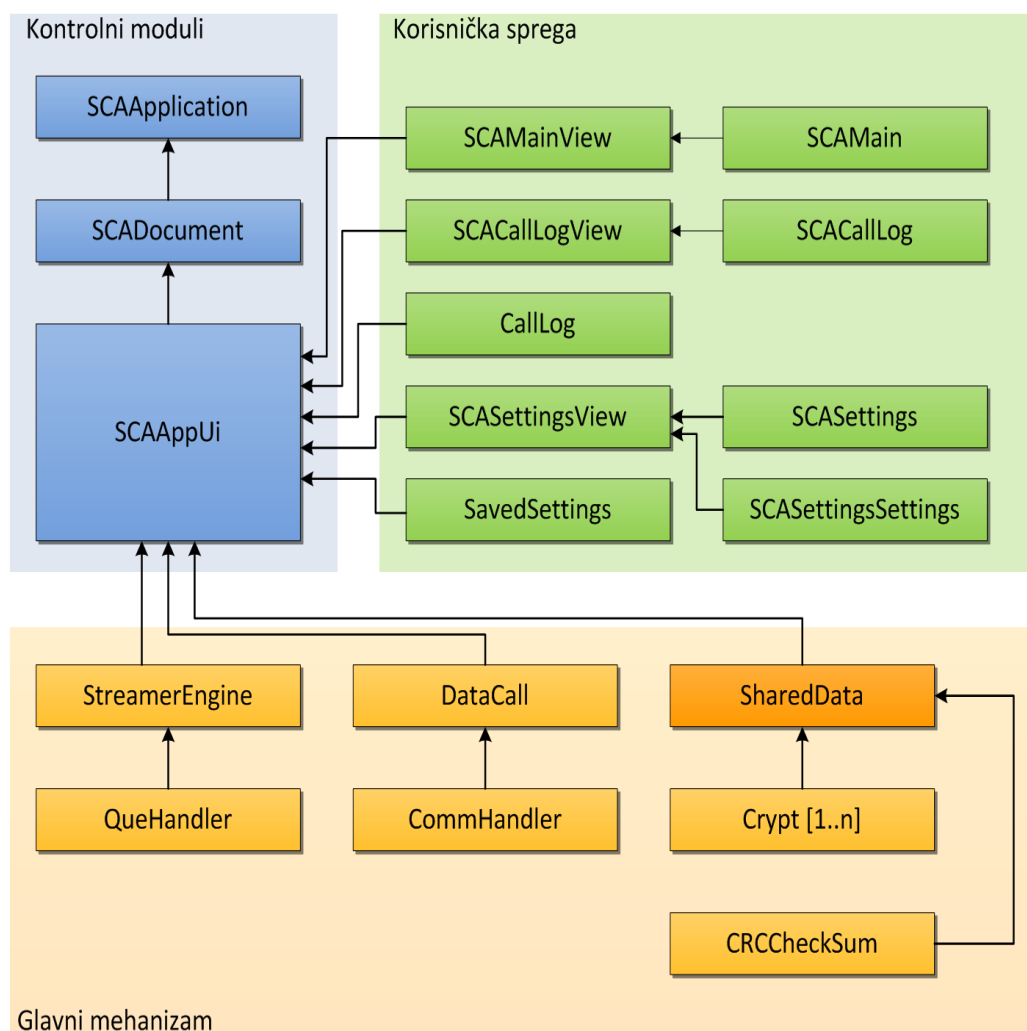
rukovanje sistemskim obaveštenjima. U nastavku je dat pregled primenjenih privilegija sa njihovim kratkim opisima:

- CommDD (Communications Device Driver) – Direktan pristup svim upravljačkim programima elemenata fizičke arhitekture za komunikaciju. Potrebno radi korišćenja GSM modema.
- LocalServices - Pristup lokalnim uslugama kao što su Bluetooth, Infrared ili USB. Može se koristiti i u programima koji se ne prijavljuju sistemu. Ne omogućava funkcije kontrole poziva.
- MultimediaDD (Multimedia Device Driver) - Pristup kritičnim funkcijama za kontrolu multimedije, kao što je direktan pristup upravljačkim programima elemenata fizičke arhitekture kamere, mikrofona, zvučnika, elementa za kodovanje i dekodovanje multimedije. Trenutno se najčešće koristi od strane APS (Audio Proxy Server) paketa koji omogućava pristup audio tokovima u punom dupleks režimu rada.
- NetworkControl – Mogućnost pristupanja i izmena kontrole mrežnog protoka. Primeri su kontrola linija poziva i menjanje prioriteta poziva.
- NetworkServices – Dozvoljava pristup udaljenim uslugama koje mogu da se naplaćuju korisniku od strane mreže. Primeri usluga sistema sa ovom mogućnošću su uspostavljanje poziva, slanje SMS poruka, WLAN i GPRS.
- ReadDeviceData – Omogućava čitanje poverljivih podataka o mreži i mobilnom uređaju. Omogućava čitanje informacija kao što su identifikacija ćelija operatera ili EMAI broja uređaja.
- ReadUserData – Programi sa ovom mogućnošću mogu da čitaju poverljive korisničke informacije kao što su kontakti. Zaštićeni korisnički podaci su zavisni od platforme i ne moraju uvek biti isti na različitim platformama.
- SwEvent – Omogućava obradu sistemskih događaja kao što je pritisak na dugme tastature, promena fokusa programa i odlazak u pozadinu. Omogućava i razmenu poruka između različitih programa.
- UserEnvironment – Pruža pristup uslugama koje obezbeđuju podatke iz korisnikovog fizičkog okruženja. Ova mogućnost daje uslugu programu za snimanje audio zvuka ili korišćenje kamere. Sa ovom mogućnošću je nemoguće dobiti lokaciju korisnika. Lokaciju korisnika obezbeđuje mogućnost Local.
- WriteDeviceData – Omogućava pristup pisanja poverljivih podataka mobilnog uređaja. Omogućava podešavanje opcija koje određuju ponašanje uređaja.
- WriteUserData – Omogućava pristup pisanja poverljivim korisničkim informacijama kao što je dodavanje kontakata. Ova usluga je zavisna od platforme.

Na Slici 3.3.2 se može videti organizacija programa po modulima i celinama. U sledećem poglavlju će svaki modul biti detaljnije opisan.

Ulazni audio tok sa mikrofona i izlazni audio tok na slušalicu se upravljaju koristeći Audio Proxy Server – APS [4]. APS je programski omotač audio elemenata fizičke arhitekture. Glavna uloga APS je da zaobiđe probleme i restrikcije koje se tiču proizvođača audio komponenti mobilnih uređaja kada se upravlja tokovima audio podataka u programima nezavisnih proizvođača. Komponenta omogućava lako korišćenje i podešavanje audio tokova u punom dupleks režimu rada. Podržava različite kodeke: AMR, G711, G729, iLBC i PCM. Program koristi AMR-NB 5.15 kodek. Ovaj tip AMR kodne šeme zahteva nominalni protok od 5,15 Kbit/sec. Ispitivanjem realnog protoka CSD-a došlo se do zaključka da mreža u realnim uslovima svojom propusnom moći CSD linije za prenos podataka omogućava korišćenje ovog moda AMR-a. Kvalitet zvuka je subjektivno bolji u odnosu na 4.75 mod AMR-a. Audio tokovi imaju frekvenciju odabiranja od 8000Hz i koriste samo jedan kanal (Mono). Prioritet audio tokova je podešen na najvišu vrednost 100, koja garantuje da klijent ne može biti prekinut od strane drugih korisničkih programa kada preuzme kontrolu nad audio tokovima. Parametri APSa su podešeni tako da pružaju rad u punom dupleks režimu i da obezbeđuju najbolji kvalitet zvuka koji audio elementi fizičke arhitekture pružaju.

Nakon dobijenih kompresovanih podataka govora, podaci se šifruju. Moduli za šifrovanje se realizuju nezavisno od samog programa. Realizuju se kao Symbian DLL biblioteke koje se dinamički učitavaju u program. Algoritmi za šifrovanje su blokovskog tipa. Realizovani su moduli DES i AES [7] [8]. Ovi algoritmi zahtevaju ključeve po kojima se podaci šifruju. Ključevi su dužine 16 karaktera i unose se preko korisničke sprege. Realizovana je i mogućnost automatskog generisanja ključeva pre svakog poziva. Kao što je i navedeno u opisu principa rada, pored samog kriptovanja podataka govora kritična stvar je i početna razmena ključeva. Ključevi se razmenjuju pre početka razgovora i ostaju nepromenjeni do kraja razgovora. Ova razmena ključeva se ostvaruje po principu Difi-Helman [7]. Difi-Helman je kriptografski protokol koji omogućuje korisnicima koji se ne poznaju da razmene ključeve na siguran način preko prenosnog medijuma koji nije siguran. Tako dobijeni ključ se zatim koristi za šifrovanje. Za realizaciju ovog protokola iskorišćena je ugrađena programska podrška Symbian OS-a.



Slika 3.3.2 Podela programa po modulima i po celinama

Modul za kontrolu GSM modema ima zadatak slanja i primanja podataka. Modul za kontrolu poziva uspostavlja poziv, prati stanje poziva u toku razgovora i prekida poziv. Moduli su realizovani kao aktivni objekti. Aktivni objekti su jedna od karakteristika Symbian operativnog sistema koja omogućuje da se mnogo racionalnije troše resursi telefona [2]. Modul postavlja zahtev raspoređivaču (scheduler) da prati određeni događaj u sistemu i prekida sa radom dok se događaj ne desi ili ga sistem ne obavesti da se događaj nije desio kako je očekivano. Praktično je reč o automatu stanja koji menja stanja u zavisnosti od sistemskih obaveštenja. Modul za rukovanje GSM modemom preuzima podatke od modula za rad sa audio tokovima preko posebnog modula koji kontroliše deljene podatke.

Modul za kontrolu deljenih podataka igra bitnu ulogu u sistemu. Ovim modulom se sinhronizuju podaci pristigli sa modema i podaci koji dolaze sa ulaznog audio toka. Modul uključuje u sebe jedan ili više modula za šifrovanje. Podaci koji dolaze sa modema se automatski dešifruju, odnosno podaci za slanje koji idu na modem se automatski šifruju unutar modula.

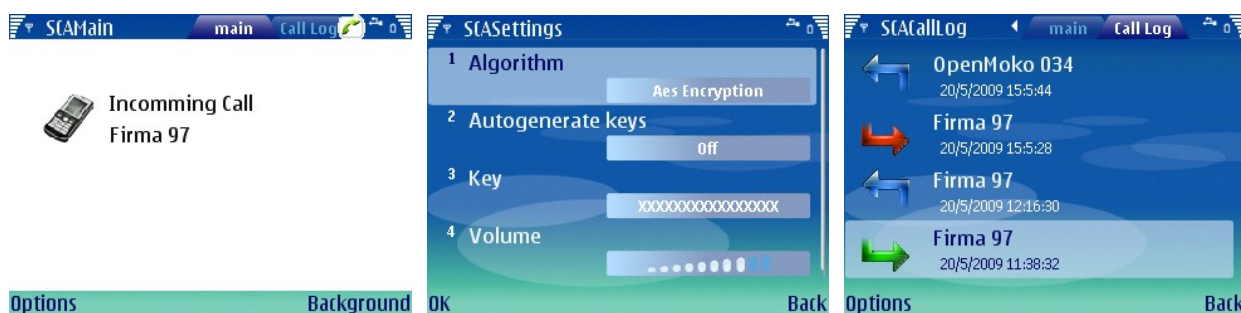
Grafička korisnička sprema se sastoji iz tri dela. Izgled grafičke korisničke sprege se može videti na Slici 3.3.3.

Prvi deo je glavni ekran na kojem se ispisuje stanje u kojem se nalazi program odnosno poziv. Izgled korisničke sprege je jednostavan, sadrži ikonicu telefona koja se menja u zavisnosti od stanja poziva, glavni tekst na kome se ispisuje tekst stanja programa i pomoćni tekst na kojem se ispisuju dodatne informacije kao što je broj zvanog ili dolazećeg poziva. Traka menija (menubar) sadrži dva dugmeta Options i Background. Sa Background se program šalje u pozadinu, dok se na dugme Options dobijaju dodatne informacije kao što su podešavanja, izabir broja iz korisničke liste kontakata, informacije o tekućoj verziji programa i algoritama.

Drugi deo grafičke korisničke sprege daje uvid korisniku u istorijat poziva. Sprema sadrži listu primljenih, propuštenih i pozivanih poziva. Svaki poziv je određen sa tri parametra na korisničkoj sprezi, ikonicom koja pokazuje tip poziva (primljen, propušten ili pozivan), informacijom o broju telefona sa kojim je obavljen razgovor i vremenom kada je obavljen razgovor. Traka menija sadrži dva dugmeta Options i Back. Pritiskom na dugme Back se vraća na glavnu korisničku sprema. Dugme Options nudi tri usluge: pozivanje broja iz istorijata poziva, brisanje tekuće stavke u listi istorijata i brisanje svih stavki iz liste istorijata. Prelazak sa glavne sprege na sprema koja prikazuje istorijat poziva se obavlja pritiskom na desni taster navigacionih dugmadi.

Treći deo korisničke sprege je prozor za podešavanje rada programa. Sastoji se iz liste opcija za podešavanje i trake menija sa dva dugmeta. Sledi pregled opcija za podešavanje:

- Autoboot – Postavlja parametar automatskog podizanja programa sa sistemom. Ima dva stanja uključeno i isključeno (on/off)
- Encryption – Daje korisniku mogućnost da izabere jedan od podržanih algoritama za šifrovanje razgovora. Stavke algoritama se automatski učitavaju u zavisnosti od učitanih modula za šifrovanje.
- Autogenerate Key – Opcija uključuje, odnosno isključuje automatsko generisanje ključeva za šifrovanje. Ima dva stanja uključi i isključi (on/off)
- Key – Daje korisniku mogućnost da unese novi ključ za šifrovanje. Ključ koji se unese se šifrjuje jednim od postojećih algoritama i kao takav se čuva u programu. Ključ mora biti dužine 16 karaktera. U slučaju da neki od algoritama koristi 8 bajtne ključeve na algoritmu je da odabere koji deo ključa će koristiti. Ukoliko korisnik unese ključ koji je manji od 16 karaktera dobija se poruka obaveštenja da ključ mora biti dužine 16 karaktera i vraća se na polje za unos.
- Volume – Omogućuje korisniku da podesi nivo jačine zvuka. Vrednost nivoa jačine je određena vrednošću od 1 do 10.



Slika 3.3.3 Izgledi korisničkih sprega (*levo glavna korisnička sprega, u sredini sprega za podešavanje rada programa, desno sprega istorijata poziva*)

Modul koji spaja glavni mehanizam sa korisničkom grafičkom spregom i ujedno ih i kontroliše je modul za kontrolu. Svaki program sa korisničkom spregom poseduje ovakav modul. Modul na početku pokreće i inicijalizuje glavni mehanizam i module korisničke sprege. Parametri za inicijalizaciju rada programa se nalaze u datoteci settings.dat i mogu se menjati u toku rada programa preko korisničke sprege za izmenu podešavanja. Na početku rada programa se učitava i istorijat poziva iz datoteke calllog.log. Nakon inicijalizacije programa modul za kontrolu postavlja program u stanje mirovanja, u kojem se sluša na dolazeći poziv. Modul prelazi u stanje mirovanja u jednom od dva slučaja, ukoliko korisnik inicira poziv ili ukoliko se detektuje dolazeći poziv. Slika promene stanja se može videti na slici 3.3.4 promene stanja modula za kontrolu poziva. Modul za kontrolu obrađuje i događaje vezane za slanje programa u pozadinu odnosno podizanje programa u prvi plan. Jedna od stvari koja se tom kontrolom postiže jeste neprimetan rad programa, pri čemu se program podiže u prvi plan samo u slučaju da je detektovan dolazni poziv zaštićenog razgovora. Modul će detaljnije biti opisan u poglavlju 3.5.4.

Sprega između glavnog mehanizma i modula za kontrolu je realizovna preko funkcija povratnog poziva (callback functions). Na taj način se korisnička sprega uz posredstvo modula za kontrolu uvek nalazi u konzistentnom stanju sa glavnim mehanizmom.

3.3 Kontrolni modul SCAAppUi

Modul pokreće, inicijalizuje i kontroliše glavni mehanizam i korisničku spregu. Svi zahtevi korisničke sprege za uslugama glavnog mehanizma i obrnuto se odvijaju posredstvom ovog modula. Modul se sastoji iz jedne klase SCAAppUi koja nasleđuje klase CaknViewAppUi i MdataCallCallBack. Klasu SCAAppUi instancira klasa SCADocument. SCADocument se instancira od strane SCAApplication koja je ulazna tačka programa i ne ostvaruje nikakvu dodatnu funkcionalnost. U natavku je dat pregled bitnih promenljivih i funkcija sa kratkim opisom njihove funkcionalnosti:

Promenljive:

- ***CSCAMainView* iSCAMainView*** – Glavna korisnička sprega
- ***CSCACallLogView* iSCACallLogView*** – Korisnička sprega istorijata poziva
- ***CSCASettingsView* iSCASettingsView*** – Korisnička sprega podešavanja
- ***TBool iCanDial, iCanHangup, iCanAnswer, iCanCancel, iIncCall*** – Promenljive određuju koje su od osnovnih operacija moguće u određenom trenutku
- ***TBool isCallLogDialTimerSeted*** – Vrednost određuje da li je aktivan vremenski brojač koji se aktivira nakon zahteva za pozivom iz korisničke sprege istorijata poziva
- ***TInt iDataCallState*** – Kazuje u kom stanju se nalazi glavni mehanizam
- ***TInt iTimerCounter*** – Broj ulazaka u Periodic() funkciju vremenskog brojača
- ***CDataCall* iMyCallDialer*** – Daje mogućnost pristupa funkcionalnosti modula za kontrolu poziva
- ***CStreamerEngine* iAudioEngine*** – Daje mogućnost pristupa funkcionalnosti modulu za kontrolu audio tokova
- ***RLibrary iCipherDesEnc, iCipherAesEnc*** – Dinamički učitani moduli za šifrovanje
- ***CSharedData* iShared*** – Omogućava pristup promenljivima i funkcionalnosti modulu za rad sa deljenim podacima programa
- ***CSavedSettings* iSavedSettings*** – Omogućava pristup tekućim informacijama podešavanja programa
- ***CCallLog* iCallLog*** – Modul u kome se čuva lista istorijata poziva
- ***CPeriodic* iUpdateTimer*** – Vremenski brojač koji se koristi prilikom ograničenja trajanja dužine razgovora

Funkcije:

- ***TKeyResponse HandleKeyEventL(const TKeyEvent& aKeyEvent, TEventCode aType)***
Funkcija obrađuje systemske događaje pritiska na dugmadi tastature. U njoj se reaguje na navigaciju između glavne korisničke sprege i sprege istorijata poziva. Obrđuje se i pritisak na numerčku tastaturu prilikom čega se aktivira prozor za unos broja.
- ***void HandleCommandL(TInt aCommand)***
Standardna funkcija u kojoj se vrši obrada komandi programa. Komande koje se ovde obrađaju pozivane su iz grafičkih korisničkih sprege.
- ***void ShowTabs(TBool aState)***
Funkcija koja iscertava tabulatore prilikom prelaska sa jedne na drugu korisničku spregu.

- ***void HandleWsEventL(const TWsEvent &aEvent, CCoeControl *aDestination)***
 Redefinisana sistemska funkcija kojom se obrađuju događaji promene grafičke sprege. Funkcija je potrebna jer učestvuje prilikom rukovanja slanja u pozadinu ili njenog podizanja u prvi plan, kao i zbog rukovanja crvenim i zelenim dugmadima na tastaturi.
- ***void BringToFront()***
 Funkcija diže program iz pozadine u prvi plan.
- ***void SendToBackGround()***
 Funkcija šalje program u pozadinu.
- ***void HandleForegroundEventL(TBool aForeground)***
 Redefinisana sistemska funkcija koja se direktno poziva prilikom slanja i podizanja programa iz pozadine.
- ***void LoadSettingsValues(CSavedSettings& aSettings)***
 Učitavanja podešavanja iz settings.dat datoteke i punjenje u objekat tipa CSavedSettings čija se referenca prosleđuje kao parametar.
- ***void SaveSettingsValues(CSavedSettings& aSettings)***
 Upisuje podešavanja rada programa u datoteku settings.dat radi trajnog čuvanja podataka u memoriji telefona. Poziva se neposredno pre izlaska iz programa.
- ***void AddCallLogEntry(const TDesC& aName, TInt aType)***
 Funkcija koja dodaje novi elemenat u listu istorijata poziva. Za parametre prima deskriptor u kome se nalazi broj telefona i tip poziva koji može biti primljeni, propušteni ili zvani (vrednosti 0, 1 i 2 respektivno).
- ***static TInt Period(TAny * aPtr)***
 Funkcija se poziva nakon isteka iUpdateTimer-a i služi kako bi bilo moguće ograničiti dužinu trajanja poziva na vreme određeno vremenskom kontrolom.
- ***void StartTimer (TTimeIntervalMicroSeconds aDelay, TimeIntervalMicroSeconds Period)***
 Pokreće se vremenska kontrola. Prvi parametar označava posle koliko vremena će vremenska kontrola prvi put isteći. Drugi parametar označava vremenski period svakog sledećeg isteka kontrole.
- ***void DataCallNotification(TInt aEvent)***
 Funkcija koja je nasleđena iz MdataCallCallBack klase. Ovo je funkcija za povratno pozivanje kojim modul za rukovanje pozivom (DataCall) komunicira sa kontrolnim modulom.

3.4 Moduli korisničke sprege

Moduli korisničke sprege su automatski generisani od strane Carbide C/C++ okruženja za rad, izuzev modula CallLog i SavedSettings. U ovom poglavlju će biti opisani moduli, način njihovog generisanja i potrebnih izmena u skladu sa potrebama rada programa. Moduli se mogu podeliti u tri gupe: moduli glavne korisničke sprege, moduli korisničke sprege istorijata poziva i moduli korisničke sprege podešavanja rada programa. Automatski generisani moduli korisničke sprege sadrže osnovni skup funkcija korisničke sprege.

3.4.1 SCAMainView

SCAMainView je osnovni modul glavne korisničke sprege. Instancira se u konstruktoru kontrolnog modula SCAAppUi. Aktivan je svo vreme rada programa, što znači da se konstruktor izvršava jednom u toku rada programa. Svaki prelazak sa modula glavne korisničke sprege na neki drugi inicira pozivanje funkcije DoDeactivate() i obrnuto svaki prelazak na glavnu korisničku spregu inicira se pozivanjem funkcije DoActivate(). Prilikom prikazivanja glavne korisničke sprege instancira se modul SCAMain, a prilikom zatvaranja korisničke sprege ovaj modul se uništava. SCAMainView modul obrađuje komande glavne korisničke sprege. Komande glavne korisničke sprege se nalaze deklarisanе u datoteci SCAMain.hrh. Za komande koje zahtevaju pokretanje glavnog mehanizma (prilikom uspostavljanja poziva) komande se prepuštaju kontrolnom modulu SCAAppUi na dalju obradu. U nastavku je dat pregled bitnih funkcija i promenljivih:

Komande koje podržava modul (nalaze su SCAMain.hrh):

- ***ESCAMainViewDialMenuItemCommand*** - Prozor za unos novog broja
- ***ESCAMainViewContactsMenuItemCommand*** – Prikaz liste kontakata
- ***ESCAMainViewSettingsMenuItemCommand*** – Prelazak na spregu podešavanja
- ***ESCAMainViewDialPhoneNumber*** – Komandom se pokreće pozivanje
- ***ESCAMainViewEnd_CallMenuItemCommand*** - Prekid poziva
- ***ESCAMainViewLoudMenuItemCommand*** – Uključivanje i isključivanje zvučnika
- ***ESCAMainViewAnswerMenuItemCommand*** – Odgovor na primljeni poziv
- ***ESCAMainViewAboutMenuItemCommand*** – Prikaz informacija o programu
- ***ESCAMainViewExitMenuItemCommand*** – Izlazak iz programa

Promenljive:

- ***CSCAMain* iSCAMain*** – Modul za rad sa grafičkim komponentama korisničke sprege
- ***TMenuState iMenuState*** – Sadrži informaciju u kom stanju se nalazi meni.
- ***TScreenState iScreenState*** – Govori u kom stanju je trenutno korisnička sprega. Promenljiva je bitna jer se ispisi na ekranu moraju uskladiti sa stanjem glavnog mehanizma, kada su u pitanju stanja razgovora.
- ***TBool isMainViewActive*** – Sadrži informaciju da li je glavna korisnička sprega trenutno aktivna.
- ***RBuf iPhoneName*** – Sadrži ime iz liste kontakata sa kojim je povezan broj telefona poziva koji se trenutno obavlja.
- ***RBuf iPhoneNumber*** – Sadrži broj telefona poziva koji se trenutno obavlja
- ***TBool iAudioEngineInitialized*** – Sadrži informaciju da li je audio mehanizam valjano inicijalizovan.

Funkcije:

- ***TUId Id()***
Vraća identifikacioni broj glavne korisničke sprege.
- ***void HandleCommandL(TInt aCommand)***
Rukuje komandama glavne korisničke sprege.
- ***static TInt RunPhoneDialQueryL(TDes& aData, TDes& aTitle, TBool aUseDefaults = ETrue, const TDesC* aOverridePrompt = NULL)***
Prikazuje dialog za unos broja telefona. Prvi parametar koji prima označava deskriptor u koji će se smestiti unešeni broj, drugi parametar označava tekst naslova dijaloga, treći parametar označava da li će se koristiti podrazumevani naslov ili tekst iz drugog parametra.
- ***TBool HandleKeyEventPhoneNumberQuery(TInt aKeyCode)***
Funkcija se poziva od strane kontrolnog modula kada se detektuje da je korisnik pritiskom na neki od tastera brojeva inicirao postupak unosa broja za pozivanje. Parametar aKeyCode se prenosi kako bi se prvi pritisnuti broj tastera mogao uneti u dijalog za unos broja.
- ***TBool HandleCallLogDialQuery()***
Funkcija se poziva kada se iz korisničke sprege istorijata poziva pokreće zvanje željenog broja odabirom jedne od stavki istorijata.
- ***TInt ShowContactList()***
Prikazuje listu kontakata i nudi mogućnost da se izabere jedan kontakt.
- ***void DynInitMenuPaneL(TInt aResourceId, CEikMenuPane* aMenuPane)***
Funkcija u zavisnosti od stanja u kojem se nalazi glavna korisnička sprega dinamički menja sadržaj opcionog menija. Funkcija je redefinisana standardna sistemska funkcija koja se poziva svaki put pre iscrtavanja menija na korisničku spregu.

- ***void SetMenuState(TMenuState aState)***
Postavlja stanje menija u odnosu na stanje glavne korisničke sprege.
- ***void Redraw()***
Funkcija osvežava ekran. Potrebno ju je pozvati svaki put kad se želi promeniti nešto na korisničkoj sprezi.
- ***void SetScreenState(TScreenState aScreenState)***
Omogućava postavljanje promenljive iScreenState na određenu vrednost stanja kako bi sadržaj korisničke sprege uvek mogao biti konzistentan sa stanjem glavnog mehanizma.
- ***void DoActivateL(const TVwsViewId& aPrevViewId,TUId aCustomMessageId,const TDesC8& aCustomMessage)***
Funkcija je automatski generisana. Poziva se svaki put kada se aktivira glavna korisnička sprega i postavlja spregu u stanje definisano u promenljivoj iScreenState.
- ***void DoDeactivate()***
Funkcija je automatski generisana. Poziva se svaki put kada se prelazi na drugu korisničku spregu (gasi se glavna korisnička sprega).
- ***TBool HandleDialMenuItemSelectedL(TInt aCommand)***
Rukuje komandom za unos broja iz opcionog menija.
- ***TBool HandleContactsMenuItemSelectedL(TInt aCommand)***
Rukuje komandom za prikaz korisničke liste kontakata.
- ***TBool HandleSettingsMenuItemSelectedL(TInt aCommand)***
Rukuje komandom za prikaz korisničke sprege podešavanja rada programa.
- ***TBool HandleEnd_CallMenuItemSelectedL(TInt aCommand)***
Rukuje komandom za prekid poziva.
- ***TBool HandleLoudMenuItemSelectedL(TInt aCommand)***
Rukuje komandom za uključivanje i isključivanje zvučnika u toku poziva.
- ***TBool HandleAnswerMenuItemSelectedL(TInt aCommand)***
Rukuje komandom za prihvatanje poziva.
- ***TBool HandleExitMenuItemSelectedL(TInt aCommand)***
Rukuje komandom za izlazak iz programa.
- ***void SetStatusLabel(const TDesC& aStatusText)***
Postavlja tekst labele statusnih informacija glavne korisničke sprege.
- ***void SetPhoneLable()***
Postavlja tekst labele za dodatne informacije na tekst koji se nalazi u promenljivoj iPhoneName.

3.4.2 SCAMain

Modul je povezan sa prethodnim modulom. Instancira se svaki put kada se aktivira glavna korisnička sprega. Uloga modula je kontrola grafičkih komponenti glavne korisničke sprege. Modul je automatski generisan. Dodate su funkcije za izmenu elemenata korisničke sprege. U nastavku je pregled elemenata korisničke sprege i funkcija za njihovu izmenu:

TScreenState nabrojiva lista:

- ***EScreenIdle***
- ***EScreenIncommingCall***
- ***EScreenDialing***
- ***EScreenConnecting***
- ***EScreenConnected***
- ***EScreenDisconnecting***

Elementi korisničke sprege:

- ***CEikLabel* iStatusLabel*** – Labela za ispis statusne poruke
- ***CEikLabel* iPhoneNumberLabel*** – Labela za ispis dodatnih informacija
- ***CEikImage* iStatusImage*** – Slika koja prikazuje trenutno stanje poziva

Funkcije:

- ***void SetStatusLabel(const TDesC& aStatusText)***
Postavlja novi tekst statusne labele koji se dobija kao parametar funkcije.
- ***void SetPhoneLabel(const TDesC& aPhoneText)***
Postavlja novi tekst labele za dodatne informacije koji se dobija kao parametar funkcije
- ***void SetStatusImage(TScreenState aImgNo)***
Postavlja sliku trenutnog stanja poziva. Ulazni parametar funkcije je promenljiva tipa nabrojive liste TScreenState.

3.4.3 SCACallLogView

Ovaj modul realizuje grafičku korisničku spregu istorijata poziva. Modul se oslanja na SCACallLog modul za kontrolu grafičkih komponenti i modul CallLog za dobavljanje i rukovanje informacijama o istorijatu poziva. Modul je automatski generisan od strane Carbide C/C++ razvojnog okruženja. Izvršene su potrebne izmene kako bi se prilagodio radu programa. Modul se instancira od strane SCAAppUI modula za kontrolu. Obraduje događaje interakcije korisnika sa grafičkom spregom istorijata poziva. Komande koje modul obrađuje se nalaze u datoteci SCACallLog.hrh.

Grafička korisnička sprega se sastoji od liste koja sadrži informacije o istorijatu poziva. Svaki element liste označava jednu stavku istorijata. Kroz listu se kreće navigacionim

dugmadima gore i dole. Sa navigacionim dugmetom levo se prelazi na glavnu korisničku spregu. Pored liste istorijata poziva sadrži i traku menija. Traka sadrži dva dugmeta. Levo dugme Options nudi opcioni meni sa tri stavke: Dial (pozovi broj), Delete (obriši trenutno odabranu stavku) i Delete All (obriši ceo istorijat poziva). Desno dugme Back vraća korisnika na glavnu korisničku spregu. Sve operacije vezane za istorijat se obavljaju u ovom modulu.

U nastavku slede bitne funkcije i promeljive vezane za rad ovog modula:

Komande sadržane u *TSCACallLogViewCommands*:

- *ESCACallLogViewDialMenuItemCommand*
- *ESCACallLogViewDeleteMenuItemCommand*
- *ESCACallLogViewDelete_AllMenuItemCommand*

Promenljive:

- *TInt iSelectedCallLogEntry* – Trenutno odabrana stavka istorijata poziva
- *CSCACallLog* iSCACallLog* – Modul za rukovanje grafičkim komponentama
- *CCallLog* iCallLogSource* – Modul sa informacijama istorijata poziva

Funkcije:

- *TUId Id()*
Vraća identifikacioni broj grafičke sprege istorijata poziva.
- *void HandleCommandL(TInt aCommand)*
Rukuje komandama grafičke sprege. Parametar *aCommand* je komanda na koju treba odreagovati. Spisak komandi je naveden prethodno.
- *void SetCallLogSource(CCallLog* aCallLogSource)*
Funkcija služi da se modulu dodeli pokazivač na modul *CallLog* koji sadrži informacije o istorijatu poziva.
- *void DoActivateL(const TVwsViewId& aPrevViewId, TUId aCustomMessageId, const TDesC8& aCustomMessage)*
Automatski generisana funkcija koja je važna jer se poziva svaki put kada se korisnička sprega istorijata poziva otvara.
- *void DoDeactivate()*
Automatski generisana funkcija koja se poziva svaki put kada se zatvara korisnička sprega istorijata poziva (prelazi se na drugu korisničku spregu).
- *TBool HandleDialMenuItemSelectedL(TInt aCommand)*
Rukovanje Dial (pozovi odabranu stavku) stavkom iz opcionog menija.

- ***TBool HandleDeleteMenuItemSelectedL(TInt aCommand)***

Rukovanje Delete (obriši odabranu stavku) stavkom iz opcionog menija.

- ***TBool HandleDelete_AllMenuItemSelectedL(TInt aCommand)***

Rukovanje Delete All (obriši sve stavke iz istorijata poziva) stavkom iz opcionog menija.

3.4.4 SCACallLog

Modul kontroliše grafičke komponente korisničke sprege istorijata poziva. Modul je automatski generisan od strane Carbide C/C++ razvojnog okruženja. Zaduženja modula su da kontroliše grafički prikaz liste istorijata poziva. Načinjene su izmene u odnosu na generisani kod u skladu sa specifičnim zahtevima programa. Informacije o istorijatu poziva se dobavljaju iz CallLog modula, sa kojim je povezan preko pokazivača na ovaj modul. Pokazivač na modul CallLog se dobija prilikom samog stvaranja modula.

Lista je realizovana kao DoubleLargeStyle lista, što podrazumeva da su elementi liste veliki i da je podeljena na dva dela, deo koji sadrži ikonicu i deo koji sadrži tekstualne informacije. Tekstualne informacije se sastoje iz teksta koji opisuje ime ili broj telefona poziva i vremena kada se poziv desio. Postoje tri ikone, od koji svaka definiše jedan tip poziva primljeni, propušeni ili pozivani.

U nastavku je dat pregled bitnih funkcija za rad ovog modula:

Nabrojiva lista slika tipova poziva TListBoxImages:

- ***EListBoxSca_aifIncoming_callIndex*** - Vrednost ikonice dolaznog poziva
- ***EListBoxSca_aifMissed_callIndex*** – Vrednost ikonice propuštenog poziva
- ***EListBoxSca_aifOutgoing_callIndex*** - Vrednost ikonice odlaznog poziva
- ***EListBoxFirstUserImageIndex***

Promenljive:

- ***CCallLog* iCallLog*** – Modul koji sadži informacije istorijata poziva
- ***CAknDoubleLargeStyleListBox* iListBox*** – grafička komponenta liste istorijata poziva

Funkcije:

- ***void InitializeControlsL()***

Automatski generisana funkcija. Služi za inicijalizaciju liste istorijata poziva. Dobavljaju se stavke iz liste CallLog modula jedna po jedna i ubacuju se u grafičku komponentu liste poziva.

- ***void AddListBoxItemL(CEikTextListBox* aListBox, const TDesC& aString)***

Automatski generisana funkcija koja služi za dodavanje novog elementa u listu na nivou rada sa grafičkim elementima. Kao ulazne parametre prima pokazivač na listu i tekst u kome je sadržan opis elementa.

- **CAknDoubleLargeStyleListBox* ListBox()**

Automatski generisana funkcija koja vraća pokazivač na listu istorijata poziva.

- **void DeleteCurrentItem()**

Funkcija briše iz liste trenutno odabrani elemenat liste.

- **void DeleteAllItems()**

Funkcija briše sve elemente iz liste istorijata poziva.

3.4.5 CallLog

Modul CallLog sadrži informacije o istorijatu poziva. Modul se pokreće sa pokretanjem programa od strane modula za kontrolu SCAAppUi. Prilikom pokretanja modula učitavaju se sačuvane informacije istorijata poziva iz calllog.log datoteke. Tokom rada programa modul omogućava da se informacije istorijata poziva održavaju ažuriranim. Prilikom završetka rada programa informacije istorijata poziva se automatski čuvaju u datoteci calllog.log. Realizuje funkcije za učitavanje sačuvanih informacija istorijata poziva, čuvanje informacija, dodavanje, brisanje i dobavljanje stavki istorijata poziva. CallLog modul se sastoji iz dve klase TCallLogEntry i CCallLog. TCallLogEntry sadrži informacije jedne stavke istorijata. Svaka stavka je definisana sa tri parametra: telefonski broj poziva, vreme poziva i tip poziva. Tip poziva može biti primljeni, propušteni ili pozivani. U nastavku je dat pregled bitnih funkcija klase CcallLog:

Promenljive:

- ***RArray<TCallLogEntry> iList*** – Sadrži elemente liste istorijata poziva

Funkcije:

- ***void GetAllEntries()***

Popunjava listu elemenata istorijata poziva iz datoteke calllog.log.

- ***TCallLogEntry GetEntry(TInt id)***

Dobavlja elemenat liste istorijata poziva po rednom broju iz liste. Redni broj elementa iz liste se prosleđuje preko parametra id.

- ***void DeleteAllEntries()***

Briše sve elemente iz liste istorijata poziva

- ***void DeleteEntry(TInt id)***

Briše određeni elemenat iz liste istorijata poziva. Redni broj elementa za brisanje se prosleđuje preko parametra id.

- ***void AddCallEntry(const TDes& aTel, const TDes& aTime, TInt aCallType)***

Funkcija dodaje novi elemenat na prvo mesto u listi. Na taj način lista je sortirana po vremenu popunjavanja. Funkcija prima tri parametra. Prvi parametar očekuje deskriptor sa brojem telefona poziva. Drugi parametar očekuje deskriptor sa vremenom poziva. Tip poziva se prosleđuje preko trećeg parametra.

- ***TInt CountEntries()***

Funkcija vraća broj elemenata iz liste istorijata poziva.

- ***void Close()***

Funkcija upisuje elemente liste istorijata poziva u datoteku calllog.log.

- ***TInt FindTelInContacts(const TDesC& aTelNumber, TDes& aContactName)***

Funkcija pretražuje listu korisničkih kontakata po kriterijumu telefonskog broja. Cilj pretrage je ime kontakta povezanog sa telefonskim brojem. Prima dva parametra, prvi je deskriptor u kojem se nalazi telefonski (kriterijum pretrage) i drugi u koji se smešta ime kontakta, ukoliko je pronađen.

- ***void ChangeEntryType(TInt aId, TInt aCallType)***

Funkcija omogućava izmenu informacije o tipu poziva određenog elementa liste istorijata poziva. Očekuju se dva parametra: redni broj elementa u listi kao prvi i novi tip poziva kao drugi parametar.

3.4.6 SCASettingsView

Korisnička grafička sprega podešavanja programa je realizovana ovim modulom. Modul se instancira samo jednom, na početku rada kontrolnog modula SCAAppUi. Kontrolise interakciju sa korisnikom sa jedne strane i grafičke elemente sprege sa druge strane. Grafička korisnička sprega podešavanja rada programa se sastoji od liste elemenata za podešavanje. Modul omogućava podešavanje modula koji se koristi za šifrovanje, podešavanje da li će se ključ automatski generisati, omogućava unos ključa dužine 16 karaktera i podešavanje nivoa jačine zvuka. Drugi deo korisničke sprege se sastoji od trake menija koja sadrži dva dugmeta Ok kojim se prihvataju podešavanja i Back kojim se izmene u podešavanjima poništavaju.

Da bi modul mogao da izvrši zahtevane promene u radu programa, modulu je omogućen pristup modulima za rad sa audio tokovima (AudioStreamer), za rad sa deljenim podacima (SharedData). U nastavku je dat kratak pregled funkcija sa njihovim opisima:

Komande:

- *ESCASettingsViewEnumeratedTextPopup1*
- *ESCASettingsViewVolume*
- *ESCASettingsViewComboAlgorithm*
- *ESCASettingsViewGenerateKeysAutomaticly*
- *ESCASettingsViewKey*

Promenljive:

- *CSharedData* iShared* – Modul koji kontroliše deljene podatke
- *CStreamerEngine* iAudioEngine* – Modul za rad sa audio tokovima
- *CSavedSettings* iSavedSettings* – Modul za skladištenje podešavanja rada programa

Funkcije:

- *TUId Id() const*
Funkcija vraća indentifikacioni broj korisničke sprege podešavanja.
- *void HandleCommandL(TInt aCommand)*
Automatski generisana funkcija koja rukuje komandama korisničke sprege. Komande koje se obrađuju su navedene prethodno.
- *void SetSharedData(CSharedData* aSharedData)*
Podešava pokazivač na modul za rad sa deljenim podacima programa.
- *void SetAudioEngine(CStreamerEngine* aAudioEngine)*
Podešava pokazivač na modul za rad sa audio tokovima podataka.
- *void SetSavedSettings(CSavedSettings* aSavedSettings)*
Podešava pokazivač na modul za čuvanje podešavanja rada programa.
- *void DoActivateL(const TVwsViewId& aPrevViewId, TUId aCustomMessageId, const TDesC8& aCustomMessage)*
Automatski generisana funkcija koja se poziva svaki put kada se pokreće korisnička sprega podešavanja rada programa. Funkcija inicijalizuje komponente podešavanja i njihove vrednosti.
- *void DoDeactivate()*
Automatski generisana funkcija koja se poziva svaki put kada se prelazi sa korisničke sprege podešavanja rada programa na neku drugu korisničku spregu.

3.4.7 SCASettings

Zaduženje modula je kontrola grafičkih komponenti korisničke sprege za podešavanje rada programa. Modul je automatski generisan od strane Carbide C/C++ razvojnog okruženja. Modulu je dodata određena funkcionalnost kako bi se prilagodio radu programa. Implementirana je funkcija za učitavanje vrednosti podešavanja iz modula SavedSettings. Modul je povezan sa SharedData i AudioStreamer modulima kako bi se izmene podešavanja mogle primeniti na glavni mehanizam. Dodata je nova funkcionalnost čuvanja podataka prilikom deaktiviranja korisničke sprege za podešavanja rada programa. U nastavku je dat kratak pregled funkcija i promenljivih sa kratkim opisima:

Promeljive:

- ***TSCASettingsSettings& iSettings*** – Modul za čuvanje trenutnih vrednosti podešavanja korisničke sprege za podešavanje
- ***CSharedData* iShared*** – Modul za rad sa deljenim podacima programa
- ***CStreamerEngine* iAudioEngine*** – Modul za rad sa audio tokovima podataka
- ***CSavedSettings* iSavedSettings*** – Modul za čuvanje vrednosti podešavanja rada programa

Funkcije:

- ***void LoadSettingValuesL()***
Funkcija postavlja vrednosti sprege na vrednosti trenutnih podešavanja rada programa.
- ***void SaveSettingValuesL()***
Funkcija je automatski generisana i poziva se svaki put kada se neka od komponenti izmeni.
- ***void SaveSettingsOnOk()***
Funkcija čuva trenutne vrednosti podešavanja korisničke sprege za podešavanje rada programa u modulu za čuvanje trenutnih podešavanja (SavedSettings)
- ***void HandleUserGeneratedKeyChange()***
Funkcija relizuje zaštitu unosa ključa za šifrovanje razgovora. Proverava se dužina ključa, koji mora biti 16 karaktera.

3.4.8 SCASettingsSettings

Automatski generisan modul koji sadrži trenutne vrednosti grafičkih komponenti korisničke sprege za podešavanje rada programa. Modul poseduje funkcije za čitanje i izmenu ovih vrednosti. U nastavku je dat pregled grafičkih komponenti:

- ***TInt iComboAlgorithm*** – Vrednost polja za izabir algoritma za šifrovanje
- ***TBool iGenerateKeysAutomaticly*** – Vrednost koja definiše da li se ključ automatski generiše
- ***TBuf<KKeyMaxLength> iKey*** – Deskriptor sa vrednošću ključa za šifrovanje
- ***TBool iAutoBoot*** – Vrednost automatskog pokretanja programa sa sistemom
- ***TInt iVolume*** - Sadrži trenutnu vrednost nivoa jačine zvuka

3.4.9 SavedSettings

Vrednosti trenutnih podešavanja rada programa se čuvaju u ovom modulu. Modul realizuje odgovarajuće funkcije za podešavanje i čitanje vrednosti ovih promenljivih. Iz potrebe da se vrednosti trajno čuvaju u memoriji telefona i nakon rada programa realizovana je serijalizacija objekta. Serijalizacija omogućava lak upis i čitanje celog objekta u tokove podataka za rad sa datotekama. U nastavku je dat pregled bitnih funkcija i promenljivih:

Promenljive:

- ***TBool iIsAutoStartOn*** – Vrednost podešavanja za automatsko pokretanje programa sa sistemom.
- ***TInt iEncryption*** – Vrednost koja označava korišćeni algoritam za šifrovanje.
- ***TBool iIsAutoGenerateKeysOn*** – Vrednost koja označava da li se ključ za šifrovanje podataka automatski stvara.
- ***TBuf<MAX_KEY_LENGTH> iUserGeneratedKey*** – Bafer koji sadrži vrednost trenutno validnog korisničkog ključa za šifrovanje (ključ je u šifrovanom obliku).
- ***TInt iVolume*** – Vrednost podešavanja nivoa jačine zvuka

Funkcije:

- ***void SetAllToDefault()***
Postavlja vrednosti na unapred definisane vrednosti.
- ***void InternalizeL(RReadStream& aStream)***
Serijalizacija objekta prilikom čitanja iz datoteke. Kao parametar prima referencu na objekat toka podataka za čitanje iz datoteke.
- ***void ExternalizeL(RWriteStream& aStream) const***
Serijalizacija objekta prilikom pisanja u datoteku. Kao parametar prima referencu na objekat toka podataka za pisanje u datoteku.

3.5 Moduli glavnog mehanizma (Engine)

Glavni mehanizam realizuje uspostavljanje, sam tok i prekid poziva. Moduli glavnog mehanizma rukuju ulaznim i izlaznim tokovima podataka, radom GSM modema, tokom poziva i bibliotekama za šifrovanje. Moduli su podeljeni u tri grupe:

- Moduli za rad sa ulaznim i izlaznim audio tokovima
- Moduli za kontrolu GSM modema i toka poziva
- Moduli za rad sa deljenim podacima

3.5.1 SharedData

Modul rukuje deljenim podacima i stvara spregu između modula za rad sa modemom i modula za rad sa audio tokovima podataka. Ovi moduli su spregnuti preko ulaznog i izlaznog bafera. Za svaki od ovih bafera postoje dve funkcije, jedna za dobavljanje i druga za postavljanje podataka. Modul za rad sa GSM modemom (CommHandler), preuzima podatke dobijene od modula za rad sa audio tokovima pomoću funkcije MicrophoneDataGet, a primljene podatke postavlja u izlazni bafer pomoću funkcije PlayBufferAppend. Sa druge strane modul za rad sa audio tokovima (AudioStreamer) ulazne podatke sa mikrofona postavlja u ulazni bafer sa funkcijom MicrophoneDataAppend a podatke koji se puštaju na zvučnik dobavlja funkcijom MicrophoneDataGet. Oba bafera su realizovana kao cirkularni baferi. Modul rukuje ovim baferima tako da se isključuje situacija prepunjavanja bafera, a samim tim i prepisivanja podataka. Šifrovanje i dešifrovanje podataka se obavlja prilikom čitanja bafera.

Sastavni deo modula je klasa MDataCallCallBack koja realizuje povratne pozive funkcija za komunikaciju sa ostali modulima.

U nastavku je dat pregled važnih funkcija sa kratkim opisima:

Nabrojiva lista EDataCallStates:

- ***EListeningForIncomingCall*** – Stanje čekanja na dolazni poziv.
- ***EIncomingCallReceived*** - Stanje u koje se prelazi kada je poziv primljen.
- ***EDiallingNumber*** – Stanje se aktivira kada se pokrene poziv.
- ***EAnsweringCall*** – U stanje se prelazi nakon odgovora na poziv.
- ***EGettingPort*** – U stanje se prelazi nakon stanja EansweringCall ili EdiallingNumber.
- ***EOpeningPort*** – U stanju se otvara prolaz a prelazi se iz EOpeningPort stanja.
- ***ENotifyStatusChange*** – Program se nalazi u ovom stanju u toku razgovora. U ovom stanju se prati stanje poziva (proveravaju se događaji sa mreže).
- ***EHangingUp*** – Program prelazi u ovo stanje kada primi signal za raskid poziva sa mreže
- ***EReadWrite2Comm*** - Stanje u kome se nalazi prilikom čitanja i pisanja podataka na prolazu (portu).
- ***EConnect*** – Program prelazi u ovo stanje nakon uspešnog uspostavljanja veze.
- ***EDisconnect*** – Program prelazi u ovo stanje kada je poziv prekinut od strane korisnika sa kojim je uspostavljena veza.
- ***EFailedDialling*** – Stanje u kojem se program nalazi nakon neuspelog uspostavljanja veze. Uzrok neuspelog uspostavljanja veze može biti zauzetost mreže.
- ***ETimerExpired*** – Stanje u kojem se nalazi nakon isteka vremenske kontrole za ograničenje trajanja poziva. Nakon ovog stanja se prelazi u stanje EdemoDisconnect.
- ***EDemoDisconnect*** – Program u ovom stanju raskida vezu zbog isteka vremenske kontrole koja ograničava vreme trajanja poziva.

Nabrojiva lista EdataCallConnection:

- ***EIncomingCall***
- ***EOutgoingCall***

Promenljive:

- ***HBufC8* iInputBuffer*** – Kružni bafer za smeštanje ulaznih podataka koji se preuzimaju sa mikrofona.
- ***HBufC8* iOutputBuffer*** – Kružni bafer za smeštanje izlaznih podataka koji se puštaju na zvučnik.
- ***TUint8 iActiveEncryption*** – Sadrži informaciju o rednom broju algoritma za šifrovanje iz liste učitanih biblioteka za šifrovanje.
- ***CCrypt *iCrypt[EncryptionNo]*** – Niz pokazivača na učitane biblioteke za šifrovanje. Biblioteke se učitavaju u kontrolnom modulu.
- ***TBuf8<KMaxKeySize> iKey*** – Sadrži ključ za šifrovanje podataka.
- ***TUint8 iInputPacketNo*** – Sadrži broj paketa koji se nalaze u ulaznom baferu.
- ***TUint8 iOutputPacketNo*** – Sadrži broj paketa koji se nalaze u izlaznom baferu.
- ***EdataCallConnection iConnection*** – Sadrži informaciju koja kazuje koji je tip poziva. iConnection može da označava ili dolazni ili odlazni poziv.
- ***CRCChecksum iCrc*** – modul za rad sa cikličnom proverom redundanse (CRC).

Funkcije:

- ***void Initialize()***
Postavlja vrednosti ulaznog i izlaznog bafera na nule (resetuje bafer).
- ***void InitializeCrypt()***
Funkcija dovodi module za šifrovanje u početno stanje.
- ***void GenerateKey(TUint8 aSize)***
Stvara ključ određene dužine na slučajan način. Dužina ključa se zadaje parametrom aSize, a označava dužinu u broju bitova.
- ***void SetKey(TDes8& aKey, TUint8 aSize)***
Postavlja ključ na novu vrednost. Deskriptor u kojem se nalazi ključ se zadaje prvim parametrom aKey, dužina ključa se zadaje parametrom aSize. Vrednost veličine ključa treba da je manja ili jednaka veličini deskriptora iz kojeg se ključ kopira.
- ***void CryptKey(TDes8& aKey, TUint8 aSize)***
Funkcija služi za šifrovanje ključa. Ključ se šifrjuje unapred određenim algoritmom koji je definisan u promenljivoj KKeyCrypt. Ključ se postavlja na novu šifrovanu vrednost. Vrednost ključa koji se šifrjuje se prosleđuje kroz aKey parametar, a veličina ključa koji se šifrjuje se zadaje kroz drugi parametar aSize.
- ***void DeCryptKey(TDes8& aKey, TUint8 aSize)***

Funkcija dešifruje ključ koji joj se prosledi kao vrednost prvog parametra. Veličina ključa koji se dešifruje se zadaje drugim parametrom *aSize*. Dešifrovani ključ se smešta u isti deskriptor iz kojeg je pročitao.

- ***void MicrofoneDataAppend(TDes8& aData)***

Upisuju se novi podaci u ulazni cirkularni bafer. Funkcija očekuje deskriptor sa osmobarbitnim podacima. Podaci će biti upisani samo u slučaju da u baferu ima dovoljno mesta da se smeste.

- ***TUint8 MicrofoneDataGet(TDes8& aData, TUint8 packetNo)***

Funkcija dobavlja podatke iz ulaznog bafera. U isto vreme funkcija i šifrira podatke. Podaci koji se dobavljaju se smeštaju u deskriptor koji se zadaje kao prvi parametar funkcija. Broj paketa koji se preuzimaju se zadaje drugim parametrom *packetNo*.

- ***void PlayBufferAppend(TDes8& aData, TUint8 requestSize)***

Funkcija upisuje nove podatke u izlazni kružni bafer. Podaci koji se žele upisati, zadaju u deskriptoru koji se prosleđuje kao prvi parametar, a količina podataka koji se upisuju iz tog bafera se zadaje drugim parametrom *requestSize*.

- ***TUint8 PlayBufferFill(TDes8& aData, TUint8 requestSize)***

Dobavlja podatke iz izlaznog kružnog bafera. Broj paketa koji se žele dobiti se zadaje drugim parametrom *requestSize*, a mesto gde se paketi smeštaju se zadaje prvim parametrom. Paketi koji se dobavljaju iz izlaznog bafera se istovremeno i dešifruju. U slučaju da u baferu nema zatražena količina paketa neće biti dobavljen ni jedan paket.

3.5.2 CRCCheckSum

Paketi podataka šifrovanog AMR govora se šalju na mrežu u redosledu u kojem stižu sa ulaznog audio toka podataka sa mikrofona. Podaci se u toku transporta na mreži mogu izgubiti usled čega je potrebno izvršiti proveru ispravnosti sadržaja paketa. Zadatak modula je rad sa cikličnom proverom redundanse CRC (cyclic redundancy check) kojom se program štiti od paketa koji se oštećuju u toku transporta na mreži. CRC bajti se proračunavaju i dodaju na pakete pre samog slanja paketa na mrežu. U nastavku je dat pregled funkcija modula sa kratkim opisom:

Funkcije:

- ***void appendCRC(unsigned char *buffer, unsigned char payloadLength)***

Funkcija izračunava CRC podataka paketa sadržanih u baferu koji se prosleđuje kao prvi parametar funkcije i dodaje se u paket. Drugi parametar funkcije određuje veličinu paketa koji se nalazi u baferu.

- **unsigned checkCRC(unsigned char *buffer, unsigned char packetLength)**

Funkcija proverava validnost paketa izračunavajući njegov CRC. Paket koji se proverava, prosleđuje se u baferu kao prvi parametar funkcije. Drugi parametar funkcije označava dužinu paketa koji se proverava. Funkcija vraća 1 ukoliko je paket ispravan odnosno 0 ukoliko paket nije ispravan.

- **unsigned CRCChecksum::calcCRC(unsigned char *buffer, unsigned char payloadLength)**

Funkcija je slična appendCRC funkciji izuzev što ne dodaje izračunati CRC kod u paket. Paket koji se proverava, prosleđuje se u baferu kao prvi parametar funkcije, a dužina paketa se zadaje kao drugi parametar. Funkcija vraća CRC kod kao povratnu vrednost.

3.5.3 Crypt

Program zaštićenog razgovora je realizovan na način da moduli za šifrovanje podataka budu razvijani nezavisno od samog programa. Moduli za šifrovanje se realizuju kao Symbian OS DLL (Dinamicly linked library). Prilikom početka rad programa, tačnije prilikom pravljenja modula za kontrolu, biblioteke se dinamički učitavaju u program. Modul koji omogućava korišćenje učitanih modula je Crypt. U njemu su definisane virtualne metode koje svaki modul za šifrovanje mora da sadrži kako bi mogao valjano da se koristi u programu za zaštićeni razgovor. U nastavku su date funkcije koje svaka dinamička biblioteka za kriptovanje mora da sadrži:

Funkcije:

- ***void InitializeL(TMyType *aInternalKey, TMyType *aExternalKey, TMyType aBlockByteNumber)***

Funkcija služi za podešavanje parametara šifrovanja. Prvi parametar je privatni ključ koji se prosleđuje kao pokazivač na niz neoznačenih brojeva. Drugi parametar je javni ključ. koji se prenosi u funkciju na isti način kao i prvi parametar. Treći parametar označava broj bajta nad kojim radi algoritam za šifrovanje.

- ***void BlockEncryption(TMyType *aDataBlock, TMyType *aEncryptedDataBlock)***

Funkcija služi za šifrovanje podataka. Prima dva parametra. Prvi parametar treba da sadrži bafer sa podacima za šifrovanje. Podaci se nakon šifrovanja smeštaju u bafer koji se prosleđuje kao drugi parametar funkcije.

- ***void BlockDecryption(TMyType *aDataBlock, TMyType *aDecryptedDataBlock)***

Funkcija radi suprotno od BlockEncryption, dešifruje podatke. Podaci za dešifrovanje se zadaju u baferu koji se prima kao prvi parametar funkcije. Dešifrovani podaci se smeštaju u bafer koji se prosleđuje kao drugi parametar.

- ***void ConstructL()***

Dvokoračni konstruktor u kome se stvaraju baferi i inicijalizuju promenljive.

- ***TVersion Version()***

Funkcija vraća strukturu sa podacima o tekućoj verziji biblioteke za šifrovanje.

- ***void GetCipherName(TDes& aPhoneNumber)***

Funkcija za dobavljanje imena biblioteke za šifrovanje. Ime biblioteke se smešta u deskriptor koji se prosleđuje kao parametar funkcije.

3.5.4 DataCall

Modul ima zadatak da prati stanje poziva, uspostave, raskida i samog toka. Realizovan je kao aktivni objekat (klasa koja nasleđuje klasu CActive). Aktivni objekti su specifičnost Symbian operativnog sistema kojom se omogućava nekoj klasi da obrađuje asinhrono događaje. Modul je realizovan kao automat stanja. Automat stanja se nalazi u redefinisanoj funkciji RunL() iz ActiveObject klase koja se poziva svaki put kada se desi asinhroni događaj.

Stanja u kojima se modul može naći nalaze se u nabrojivoj listi EdataCallStates iz modula za rad sa deljenim podacima (SharedData). Modul se pokreće od strane modula za kontrolu (SCAAppUi). Prilikom pokretanja modula, prvo se sakupljaju informacije o telefonu, pobrojavaju se linije, izabira se linija za prenos podataka (data call) i postavlja se u stanje čekanja na dolazni poziv. Iz stanja čekanja može da se pređe u jedno od dva stanja: primljenog poziva (IncomingCallReceived) ili pozivanja (DialingNumber). U slučaju primljenog poziva modul prelazi u stanje prekida veze (HangingUp), ukoliko je poziv odbijen ili je prekinut pre nego što je odgovoreno. U slučaju da se poziv odbije prilikom pozivanja broja, prelazi se u stanje prekida veze kao i u prethodnom slučaju. U slučaju da je na poziv odgovoreno prelazi se u stanje odgovaranja na poziv (AnsweringCall). U isto stanje može i da se pređe iz stanja pozivanja (DialingNumber) ukoliko pozivani broj odgovori na poziv. Iz stanja odgovaranja na poziv uvek se prelazi u stanje dobavljanje prolaza (GettingPort). Nakon dobavljanja prolaza prelazi se u stanje otvaranja prolaza (OpeningPort). U stanju otvaranja prolaza pokreće se modul za rukovanje GSM modemom (CommHandler). Jednom kada je otvoren prolaz veza je uspostavljena pri čemu se prelazi u stanje praćenja stanja poziva (NotifyStatusChange). U ovom stanju se osluškiju signali koji se dobijaju sa mreže. Nakon primljenog signala HANG UP,

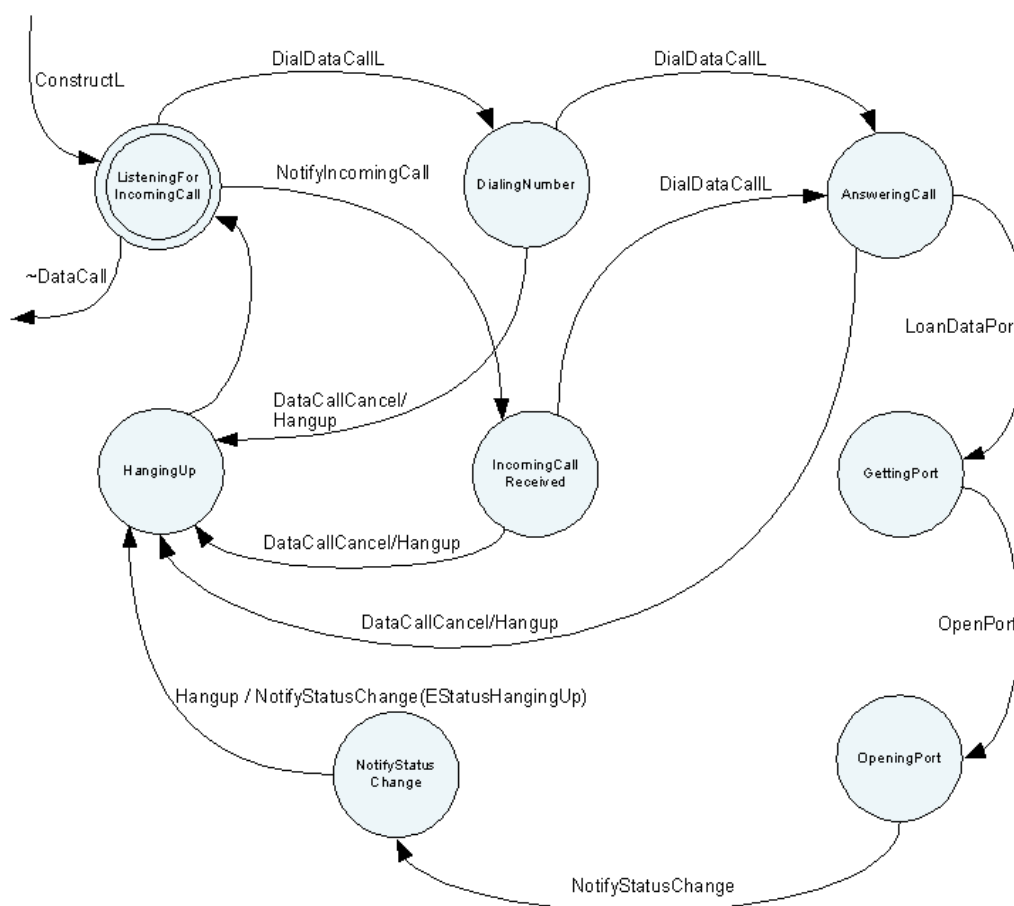
prelazi se u stanje čekanja na poziv. U slučaju da se pozivani broj prekine, modul će primiti signal `EstatusHangingUp`. Drugi način da se raskine poziv je pozivanjem funkcije `Hangup()`, prilikom koje se automat stanja prebacuje u stanje prekida veze (`HangingUp`). U stanju prekida veze postavlja se asinhroni događaj čekanja na dolazni poziv, a automat stanja se postavlja na sledeće stanje primljenog poziva.

Modul je povezan sa korisničkom spregom preko modula za kontrolu. Veza je realizovana funkcijom ponovnog poziva `DataCallNotification` iz klase `MDataCallCallBack` koja pripada modulu za rad sa deljenim podacima. Modul za kontrolu na taj način održava informacije sa glavne korisničke sprege konzistentnim sa modulom za kontrolu poziva.

Dijagram stanja preleza se može videti na slici Sl. 3.3.4. U nastavku su navedene funkcije sa kratkim opisima:

Funkcije:

- ***void DialDataCallL(const TDesC& aPhoneNumber)***
Funkcijom se započinje novi poziv. Broj telefona koji treba da se pozove se prosleđuje kao parametar funkcije.
- ***void Hangup()***
Funkcija služi za prekid veze. Funkcijom se automat stanja podešava na stanje `HangingUp`.
- ***void DataCallRead(void)***
Funkcija dobavlja telefonski broj prilikom primljenog poziva.
- ***void AnswerDataCallL()***
Funkcijom se odgovara na primljeni poziv. Prelazi se iz stanja primljeni poziv u stanje odgovaranja na poziv `AnsweringCall`.
- ***void DataCallCancel()***
Otkazuje se poziv. Poziva se u slučaju kada se želi prekinuti poziv pre nego što je veza uspostavljena.
- ***void WaitDataCall()***
Postavlja automat u stanje čekanja na dolazni poziv.
- ***void RunL()***
Redefinisana sistemaska funkcija iz klase `CActive` koja se poziva svaki put kada se desi asinhroni događaj čije je praćenje zatraženo iz ovog modula. Unutar ove funkcije se nalazi telo automata stanja.



Slika 3.3.4 Dijagram stanja prelaza modula za kontrolu poziva

3.5.5 CommHandle

Modul za kontrolu rada GSM modema. Modul realizuje funkcije koje služe za slanje i primanje podataka preko GSM modema. Modul je realizovan kao automat stanja, to jest klasa koja je realizovana u ovom modulu nasleđuje CActive sistemsku klasu. Sam automat se nalazi u redefinisanoj sistemskoj funkciji RunL() koja se poziva svaki put kada se desi asinhroni događaj. Modul za rad sa GSM modemom se pokreće od strane modula za kontrolu poziva kad se nađe u stanju OpeningPort. Modul se zaustavlja kada modul za kontrolu poziva pređe u stanje raskida veze (HangingUp).

Početno stanje automata stanja je EWaitingForTransmit. U ovom stanju se očekuje slanje paketa za početnu uspostavu poziva, razmena ključeva. Nakon ovog stanja se prelazi u stanje slanje šifrovanih podataka EWritingData. Sledeće stanje u koje se prelazi nakon slanja podataka je EWaitingForData, stanje u kome se čitaju podaci sa linije. Dobavljanje podataka za slanje i smeštanje primljenih podataka se obavlja posredstvom modula za rad sa deljenim podacima (SharedData) koji je opisan u poglavlju 3.5.1.

U nastavku je dat spisak bitnih funkcija sa kratkim opisom:

Nabrojiva lista EWritingStates:

- ***EWritingData*** - Stanje slanja podataka sa ulaznog audio toka
- ***EWaitingForTransmit*** – Stanje početne razmene kontrolnih paketa i ključeva
- ***EWaitingForData*** - Stanje primanja podataka sa mreže

Funkcije:

- ***void RunL()***
Sistemska redefinisana funkcija koja se poziva svaki put kada se desi asinhroni događaj. Funkcija realizuje automat stanja.
- ***void ConstructL()***
Dvokoračni konstruktor. Automat stanja se postavlja u početno stanje i pokreće se razmena ključeva i kontrolnih paketa.
- ***void PrepareControlPacket()***
Funkcija stvara kontrolni paket za početnu razmenu ključeva.
- ***void HandleControlPacket()***
Funkcija izvršava obradu kontrolnih paketa.
- ***void HandleDataPacket()***
Funkcija rukuje šifrovanim paketima. Radi nad jednim paketom.

3.5.6 StreamerEngine

Zaduženje modula je rukovanje ulaznim i izlaznim audio tokovima podataka. Ulazni audio tok podataka sadrži digitalizovane podatke govora koji se dobijaju sa mikrofona mobilnog uređaja u AMR 5.15 formatu. Frekvencija odabiranja podatka govora je podešena na 8000Hz, a broj kanala je podešen na 1 (Mono). Izlazni audio tok podataka sadrži dešifrovane podatke govora u AMR 5.15 formatu koji se puštaju na zvučnik telefona. Ulazni podaci se smeštaju u ulazni bafer modula za rukovanje deljenim podacima. Izlazni podaci se dobavljaju iz istog modula i već su dešifrovani i spremni za puštanje na zvučnik. Modul nudi funkcije za podešavanje parametara zvuka, kao što su podešavanje nivoa pojačanja i kvaliteta zvuka. Realizovane su funkcije za inicijalizaciju, početak i završetak kontrole audio tokova. Modul za kontrolu (SCAAppUi) kontroliše rad ovog modula. Inicijalizacija modula se poziva prilikom stvaranja modula u konstruktoru modula za kontrolu. Pokretanje i zaustavljanje kontrole audio tokova se poziva iz funkcije za ponovni poziv kojom modul za rad sa pozivom komunicira sa modulom za kontrolu. Modul ne pristupa direktno elementima fizičke arhitekture mikrofona i zvučniku, već za to se oslanja na modul QueHandler.

U nastavku je dat pregled funkcija i promenljivih modula sa kratkim opisima :

Nabrojiva lista TAppState označava u kom stanju se nalazi modul:

- **ENotReady** – Nije spreman za preuzimanje kontrole nad tokovima
- **EReady** – Spreman za preuzimanje kontrole nad tokovima
- **EStreaming** – Stanje tokova označava da je kontrola audio tokova u toku

Nabrojiva lista Tcodecs označava koji koder je podešen za kompresiju audio podataka:

- **ECodecNone** - Nije podešen ni jedan koder
- **ECodecPCM** – Podešen je koder za rad sa PCM (Pulse Code Modulation)
- **ECodecAMR** - Podešen je AMR (Adaptive Multi-Rate) koder
- **ECodecG711** - Podešen je G.711 koder
- **ECodecG729** – Podešen je G.729 koder
- **ECodeciLBC** – Podešen je iLBC (internet Low Bitrate Codec) koder

Promenljive:

- **RMsgQueue<TAPSCommBuffer> iReadQ** – Bafer u koji se smeštaju pročitani podaci sa mikrofona.
- **RMsgQueue<TAPSCommBuffer> iReadCommQ** – Bafer preko koga se izdaju komande modulu za rad sa ulaznim audio tokom.
- **RMsgQueue<TAPSCommBuffer> iWriteQ** – Bafer u koga se smeštaju podaci koji se puštaju na zvučnik.
- **RMsgQueue<TAPSCommBuffer> iWriteCommQ** – Bafer preko kojeg se izdaju komande modulu za rad sa izlaznim audio tokom.
- **CQueueHandler* iPlayCommHandler** – Modul za rad sa izlaznim audio tokom.
- **CQueueHandler* iRecCommHandler** – Modul za rad sa ulaznim audio tokom.
- **CQueueHandler* iLoopbackHandler** – Modul koji objedinjuje prethodna dva modula.
- **TAppState iStatus** – Sadrži vrednost stanja u kojem se nalazi modul. Označava da li je modul spreman za preuzimanje kontrole nad tokovima ili već ima kontrolu.
- **TCodecs iCurrentCodec** – Trenutno podešeni audio kodek.
- **CSharedData *iShared** – Modul za rukovanje deljenim podacima.
- **TBool iSpkr** – Da li je uključen zvučnik.
- **TInt iMaxGain** – Maksimalna vrednost pojačanja zvuka.
- **TInt iMaxVolume** – Maksimalna vrednost jačine zvuka.

Funkcije:

- **void SetAMR()**
Podešava vrednosti APSa (Audio Proxy Server) za rad sa AMR koderom.
- **void SetCodecMode(TAPSCodecMode aMode)**
Funkcija podešava vrednosti koda za rad sa željenim kodekom.
- **void ToggleSpeaker()**
Funkcija uključuje odnosno isključuje zvučnik za glasnu reprodukciju.

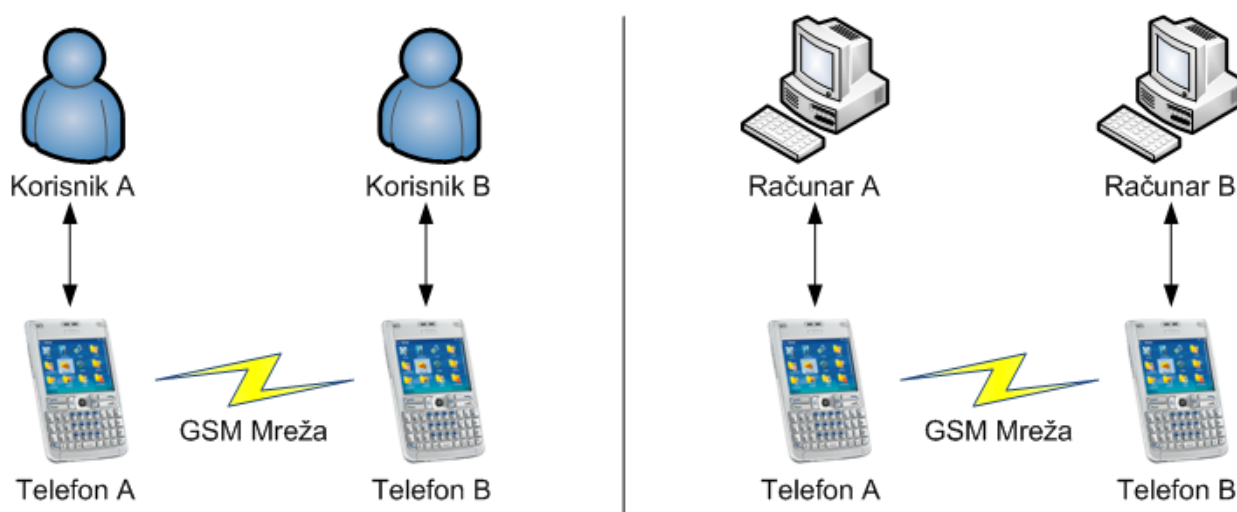
- ***TInt MaxGain()***
Funkcija vraća vrednost maksimalnog pojačanja zvučnika.
- ***TInt MaxVolume()***
Funkcija vraća vrednost maksimalne jačine zvuka slušalice.
- ***TInt GetGain()***
Funkcija vraća vrednost trenutno podešenog pojačanja zvučnika.
- ***TInt GetVolume()***
Funkcija vraća trenutno podešenu vrednost jačine zvuka slušalice.
- ***void SetGain(const TInt aGain)***
Funkcija podešava vrednost pojačanja zvučnika.
- ***void SetVolume(const TInt aVolume)***
Funkcija podešava vrednost jačine zvuka slušalice.
- ***void Initialize()***
Funkcijom se pokreće rad modula za kontrolu audio tokova.
- ***void StartL()***
Funkcija pokreće preuzimanje kontrole nad audio tokovima podataka.
- ***void Stop()***
Funkcija prekida kontrolu modula nad audio tokovima podataka.
- ***void Settings()***
Podešava vrednosti audio tokova i rada samog APSa.
- ***virtual void StartOutputStream()***
Funkcija se poziva automatski svaki put kad se završi pokretanje ulaznog audio toka.
- ***virtual void NotifyError(const TInt aError)***
Funkcija se poziva ukoliko se desi greška prilikom kontrole modula audio tokovima. Greška se prikazuje u sistemskim prozorima za ispis greške.

3.5.7 QueHandler

Modul realizuje direktan pristup elementima fizičke arhitekture zvučnika i slušalice. Da li će modul raditi sa mikrofonom ili slušalicom zavisi od početnog stanja u koji se postavi prilikom pokretanja modula. Modul je realizovan kao aktivni objekat. Posедуje funkcije za zaustavljanje i pokretanje samog modula `Start()` i `DoCancel()`, kao i funkciju za dobavljanje trenutnog stanja u kojem se nalazi modul.

4. Testiranje i verifikacija

Testiranje i verifikacija su izvršeni primenom dva različita pristupa. Prvi pristup predstavlja testiranje programa ručno, koristeći testne scenarije kojima se ispituje funkcionalnost korisničke sprege, subjektivni kvalitet glasa i kašnjenja govora. Drugi pristup je automatsko testiranje programa kako bi se ispitala stabilnost rada programa u uslovima velikog opterećenja. Na slici Sl. 4.1 su grafički prikazani oba pristupa testiranju rada programa.



Slika 4.1 Pristupi testiranju (*levo ručno testiranje, desno automatsko*)

Testiranje funkcionalnosti programa se vršilo na osnovu niza testnih slučajeva koji pokrivaju većinu situacija u kojima program može da se nađe tokom korišćenja. Za svaki testni slučaj definisan je poseban testni scenario. Testni scenariji se sastoje iz 6 polja: redni broj, kratak opis testnog scenarija, početni uslovi, postupak izvršavanja, očekivani rezultat i rezultat izvršavanja.

Početni uslovi testnog scenarija označavaju stanje u kojem program treba da se nalazi pre početka izvršavanja testnog scenarija. Postupak izvršavanja prikazuje akcije koje korisnik treba da izvrši. Očekivani rezultat prikazuje stanje programa u kojem on treba da se nađe nakon izvršenja testnog scenarija. Ukoliko se rezultat testa poklapa sa očekivanim rezultatom test je prošao, u suprotnom nije prošao. U dodatku se nalaze tabele sa testnim slučajevima i rezultatima izvršavanja testnih slučajeva.

Automatsko testiranje ispituje stabilnost rada programa prilikom velikog opterećenja, što znači da se program podvrgava malom broju testova sa velikim brojem ponavljanja. Ovim testovima se ispituje osnovna funkcionalnost programa kao što je pozivanje broja, prihvatanje ili odbijanje poziva.

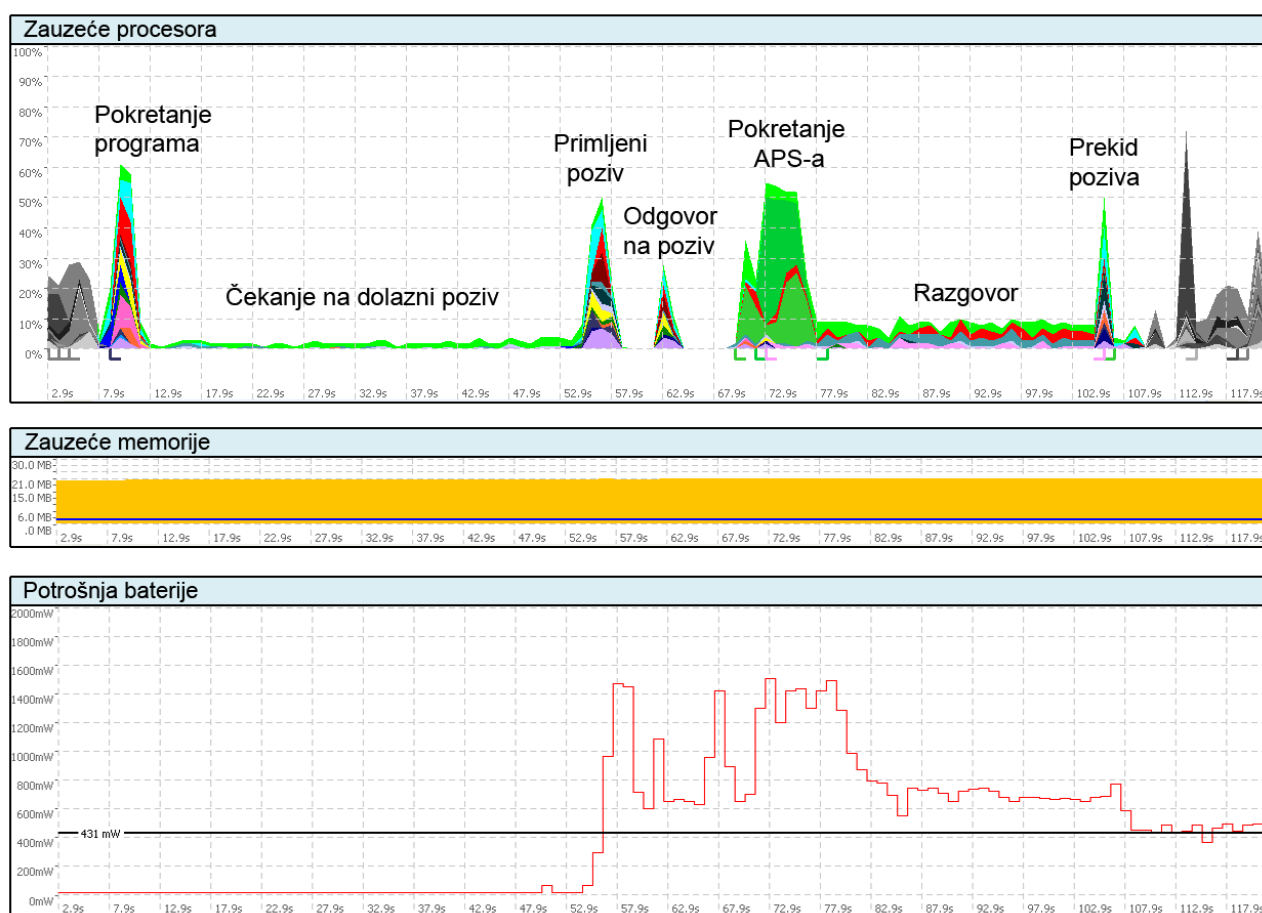
Primenjeni sistem za automatsko testiranje rada programa na Symbian OS-u se sastoji iz dva dela. Prvi deo je program na računaru koji izvršava testove zadavajući mobilnom uređaju niz komandi koje simuliraju pritiske na dugmad tastature. Svaki test se sastoji iz niza komandi koje se izvršavaju redom, jedna za drugom. Komande mogu biti pritisci na dugmad tastature, čekanje na određeno stanje poziva ili komande za dobavljanje rezultata rada (slike korisničke sprege i spisak pokrenutih programa). Svaki test se nalazi u zasebnoj datoteci sa ekstenzijom .tst. Testovi se izvršavaju jedan za drugim, onim redosledom u kojem su navedeni u datoteci Test_Suite.ts. Rezultat rada se smešta u direktorijum sa istim imenom kao što je i ime testa. Rezultat se sastoji iz slike korisničke sprege u trenutku završetka rada testa u .bmp formatu i spiska pokrenutih programa u tekstualnom formatu. Rezultat rada programa, da li je test prošao ili ne, se dobija poređenjem slike izgleda korisničke sprege nakon izvršenog testa sa referentnom slikom. Drugi deo sistema za testiranje je program koji se pokreće na mobilnom uređaju. Program prima komande sa računara i izvršava ih. Komunikacija između računara i mobilnog uređaja se odvija preko tehnologija za bežičnu komunikaciju uređaja u određenom radijusu (Bluetooth). Na koji će se računar i preko kojeg prolaza (porta) povezati, program na mobilnom telefonu čita iz datoteke Testing.cfg koja se nalazi u direktorijumu c:/Data/Shared na samom uređaju. U dodatku su prikazani MSC dijagrami realizovanih testova automatskog testiranja.

Nakon izvršenog testiranja i verifikacije funkcionalnosti pristupilo se analizi performansi rada sistema. U ovu svrhu se koristio Performance Investigator, proširenje samog razvojnog okruženja Carbide.C++. Proširenje analizira podatke koji se prikupljaju na telefonu posebnim programom s60_3_0_Prof. Svi prikupljeni podaci se snimaju u memoriji telefona, kako bi se naknadno prebacili na PC i analizirali. Parametri koji su se ovom analizom pratili su:

1. Zauzeće i opterećenje procesora
2. Potrošnja memorije
3. Potrošnja baterije

Analizirana su tri slučaja, kada se razgovor šifruje sa osnovnim algoritmom, DES algoritmom i AES algoritmom za šifrovanje razgovora. U nastavku su prikazani dobijeni rezultati analize.

Prvi analizirani slučaj, kada se razgovor obavlja šifrovanjem osnovnim algoritmom, prikazan je na Slici 4.2. Na grafiku zauzeća procesora se može videti da program ni u jednom slučaju ne troši više od 60% procesora duže od 3 sekunde. Najveća potrošnja procesora se dostiže prilikom pokretanja programa, primanja poziva i pokretanja modula za rad sa audio tokovima (pokretanje APS-a). U toku samog razgovora zauzeće procesora ne prelazi 10%. U stanju čekanja na dolazni poziv program troši 1-2% procesora. Analizom zauzeća memorije utvrđeno je da program zauzima 152KB memorije, od kojih 24KB pripada programskom steku, 4KB dinamički učitanim bibliotekama i 124KB dinamički zauzetoj memoriji. U toku rada programa nema bitnih oscilacija u količini zauzete memorije, što se može videti na grafiku zauzeća memorije. U stanju čekanja na poziv, program troši izuzetno malo snage, oko 10mW. Snaga koju program troši se drastično povećava u toku razgovora, dostignuta je vrednost od 1500mW, dok je prosek oko 1000mW.



Slika 4.2 Grafički prikaz performansi sistema za osnovni algoritam za šifrovanje

Drugi analizirani slučaj je uspostava poziva kada se koristi DES algoritam za šifrovanje. Dobijeni rezultati ne odstupaju bitno od prethodnog slučaja. Izmereno je smanjenje zauzeća procesora na nivo od 40% u svom maksimumu. U toku samog razgovora, nivo zauzeća ne prelazi 6%, što se može objasniti implementiranjem DES algoritma koristeći sistemske biblioteke za šifrovanje podataka. Potrošnja memorije je veća za 52KB i iznosi 204KB. Povećanje se odnosi isključivo na zauzeće dinamičke memorije, koje iznosi 176KB. Potrošnja snage je ostala na nivou iz prethodnog slučaja.

AES algoritam za šifrovanje je znatno napredniji od prethodna dva, pa su i izmereni rezultati na nivou gornjih granica prethodna dva slučaja. Zauzeće procesora je na nivou prvog slučaja, 60% u maksimumu i 10% u toku razgovora. AES je implemetiran koristeći iste sistemske biblioteke kao i u slučaju DESa, pa je izmereno zauzeće memorije od 204KB. Potrošnja snage je na nivou prethodna dva slučaja.

5. Zaključak

Ideja o zaštićenom razgovoru je nastala iz sve veće potrebe za očuvanjem privatnosti informacija razgovora. Mali broj komercijalnih rešenja na ovom polju čini ga vrlo interesantnim za razvoj. U radu je razrađen i realizovan princip funkcionisanja jednog rešenja. Razgovor se obavlja preko CSD poziva za prenos podataka govora, što omogućava korišćenje složenih algoritama za šifrovanje. CSD tip poziva je podržan od strane svih GSM mreža. Ograničenje u količini prenešenih podataka od 9.6 kbita/s koje nameće korišćeni poziv za prenos podataka govora se zadovolja kompresovanjem glasa AMR 5.15 algoritmom. Nad tako kompresovanim podacima primenjuju se blokovski algoritmi za šifrovanje kao što je DES. Pored jakog algoritma za šifrovanje bitna je i početna razmena ključeva, koja je realizovana primenom Difi-Helman-Merkle protokola za razmenu ključeva po nesigurnom kanalu. Pristupi audio tokovima podataka se vrše u privilegovanom režimu rada, pri čemu ni jedan drugi program nema prava pristupa ovim tokovima. Ovim se korisnik štiti od mogućih programa koji snimaju razgovore, a nalaze se na samom mobilnom uređaju.

Korisnička sprega je jednostavna i pruža sve potrebne informacije o pozivima na jednom mestu. Posедуje glavnu korisničku spregu koja daje uvid u trenutno stanje poziva. Realizovano je i praćenje istorije poziva. Lista istorijata poziva je realizovana u zasebnoj korisničkoj sprezi i pruža uvid u osnovne informacije o pozivu. Takođe omogućava pozivanje brojeva iz liste istorijata. Program omogućava korisniku da podesi parametre algoritma za šifrovanje i jačine zvuka sa korisničke sprege za podešavanje rada programa. Sam program je napravljen tako da ne ometa korisnika u radu sa ostalim uslugama telefona, što znači da program može da radi i u pozadini i da se pokreće samo kada se primi poziv za sigurni razgovor. Govorni poziv se i dalje obavlja sa matičnim programom za razgovor.

Algoritmi za šifrovanje se realizuju u posebnim Symbian DLL bibliotekama koje se dinamički iščitavaju u program što omogućava lako unapređivanje sigurnosti rada programa dodavanjem jačih algoritama za šifrovanje.

Za realizaciju programa za sigurni razgovor izabran je Symbian operativni sistem zbog svojih prednosti koje se odnose pre svega na zastupljenost sistema na tržištu pametnih telefona. Po poslednjim ispitivanjima, američke firme Canalys, Symbian je zauzimao 47.1% tržišta [11]. Ostale prednosti sistema su bogato razvojno okruženje Carbide C/C++, mobilni uređaji visokih performansi koji podržavaju sistem i opširna dokumentacija programskih sprega sistema.

Funkcionalnost programa je testirana primenom testnih scenarija, dok je stabilnost rada programa pod uslovima velikog opterećenja testirana primenom automatskih testova. Program je prošao sve testove. Subjektivnom procenom kvaliteta razgovora je ustanovljeno da je razgovor na nivou glasovnog poziva. Izmereno je konstantno kašnjenje od dve sekunde koje je prouzrokovano korišćenom linijom za prenos podataka to jest topologijom same mreže. Izmeren je i kvalitet razgovora u pokretu (vožnja automobilom), pri čemu razgovor ne gubi na kvalitetu sve do brzine 60km/h. Kvalitet razgovora se znatno narušava pri brzinama većim od 80km/h.

Kroz analizu i realizaciju programa istakle su se dve bitne mane sistema. Prva mana je to što je Symbian zatvorena platforma i ne daje uvid u funkcionisanje jezgra sistema. Drugim rečima, ne može se tvrditi da sam sistem nema ozbiljne sigurnosne propuste. Druga mana je izmereno kašnjenje od 2 sekunde, koje je nešto manje od satelitskog poziva, a znatno veće od običnog GSM glasovnog poziva.

6. Literatura

- [1] *"Quick Recipes on Symbian OS"*, Michael Aubert, Jun 2008
- [2] *"Symbian OS C++ for Mobile Phones"*, Richard Harrison i Mark Shackman, Jun 2007
- [3] *"Symbian OS Explained"*, Jo Stichbury, Oktobar 2004
- [4] *"S60 Programming - A Tutorial Guide"*, Paul Coulton, Februar 2007
- [5] *"Common problems when attempting secure calls"*, Global Teck, Avgust 2008
- [6] *"Analysis of packet loss in transport layer over ATM networks"*, Kazumi Kumazoe, Oktobar 1997
- [7] *"An overview of public key cryptography"*, Martin E. Hellman, Novembar 2006
- [8] *"Differential cryptanalysis of DES-like cryptosystems"*, Journal , Januar 1991
- [9] <http://wiki.forum.nokia.com/>
- [10] <http://developer.symbian.com/>
- [11] <http://www.gartner.com/>

7. Dodatak

7.1 Testni scenariji za ručno testiranje

Tabele testnih slučajeva korišćene prilikom ručnog testiranja funkcionalnosti i subjektivnog kvaliteta poziva:

TESTIRANJE GRAFIČKE KORISNIČKE SPREGE			
Rb	Opis testa	Očekivani rezultat	Uspešnost
1	Opis: Pokretanje programa.	Na ekranu telefona treba da se pojavi glavna korisnička sprega SCA programa. Stanje treba da sadrži tekst Secure Call.	PROŠAO
	Početni uslovi: SCA program nije upaljen. Sistem treba da se nalazi na glavnom ekranu sistema (Home Screen).		
	Postupak: Tel A: Pritisniti desno programsko dugme (SoftKey) SCA Tel A: Sačekati da program izvrši pokretanje modula za rad sa audio tokovima		
2	Opis: Izlazak iz programa.	Program treba da se ugasi i da se na ekranu telefona prikaže glavni ekran sistema (HomeScreen).	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavna korisnička sprega.		
	Postupak: Tel A: Pritisnuti Options->Exit		

3	Opis: Slanje programa u pozadinu Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavna korisnička sprega. Postupak: Tel A: Pritisnuti desno programsko dugme “background”	Program treba da ode u pozadinu. Na ekranu treba da se vidi glavna korisnička sprega sistema (HomeScreen).	PROŠAO
4	Opis: Slanje programa u pozadinu pritiskom na Crveno dugme (RedKey).	Program treba da ode u pozadinu. Na ekranu treba da se vidi glavna korisnička sprega sistema (HomeScreen).	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavna korisnička sprega.		
	Postupak: Tel A: Pritisnuti crveno dugme RedKey		
5	Opis: Reagovanje programa na pritisak zelenog dugmeta (GreenKey) .	Ne treba ništa da se desi. Korisnička sprega i stanje programa ostaju ne promenjeni.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavna korisnička sprega.		
	Postupak: Tel A: Pritisnuti zeleno dugme GreenKey		
6	Opis: Reagovanje programa na pritisak dugmadi brojeva	Na glavnoj korisničkoj sprezi programa treba da se pojavi prozor za biranje sa upisan tekstom “123” u polje za unos broja.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavna korisnička sprega.		
	Postupak: Tel A: Pritisnuti dugme “1” na tastaturi Tel A: Pritisnuti dugme “2” na tastaturi Tel A: Pritisnuti dugme “3” na tastaturi		

7	Opis: Prikazivanje liste kontakata telefona. Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.	Na korisničkoj sprezi treba da se pojavi lista kontakata iz telefonskog imenika.	PROŠAO
	Postupak: Tel A: Pritisnuti Options->Contacts		
8	Opis: Slanje programa u pozadinu i ponovno vraćanje u prvi plan kada se nalazi u listi kontakata.	Program nakon pritiska crvenog dugmeta treba da ode u pozadinu. Nakon pritiska desnog programskog dugmeta treba da se vrati u prvi plan sa prikazanom listom kontakata.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: Tel A: Pritisnuti Options->Contacts Tel A: Pritisnuti crveno dugme (RedKey) Tel A: Sačekati da se pojavi glavna korisnička sprega sistema i zatim pritisnuti desno programsko dugme SCA		
9	Opis: Izabiranje broja iz liste kontakata. Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Postupak: Tel A: Pritisnuti Options->Contacts Tel A: Pritisnuti navigaciono dugme dole Tel A: Pritisnuti OK dugme	Na ekranu treba da se pojavi prozor sa odabranim brojem iz liste kontakata i imenom iz liste kontakata.	PROŠAO
10	Opis: Slanje u pozadinu programa nakon izabranog broja iz liste kontakata i zatim ponovno podizanje u prvi plan.	Program nakon odabira tel. kontakta treba da prikaže prozor sa brojem. Zatim se sklanja u pozadinu i prikazuje se glavna korisnička sprega sistema. Ponovno se pojavljuje glavna korisnička sprega sa prozorom za pozivanje odabranog broja.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: Tel A: Pritisnuti Options->Contacts Tel A: Izabrati jedan kontakt Tel A: Pritisnuti crveno dugme (RedKey) Tel A: Pritisnuti desno programsko dugme SCA		

11	Opis: Slanje programa u pozadinu kada se nalazi u listi istorijata poziva.	Nakon prvog koraka se prikazuje lista istorijata poziva. Nakon drugog koraka se program šalje u pozadinu.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: Tel A: Pritisnuti desno navigaciono dugme Tel A: Pritisnuti crveno dugme (RedKey)		
12	Opis: Slanje programa u pozadinu kada se nalazi u podešavanjima programa.	Nakon prvog korake se prikazuje korisnička sprega za podešavanje rada programa. Nakon drugog se program šalje u pozadinu	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: Tel A: Pritisnuti Options->Settings Tel A: Pritisnuti crveno dugme (RedKey)		
13	Opis: Dodavanje nove stavke u istorijat poziva nakon pozivanja broja.	Program nakon izabranog broja i potvrde početka poziva prelazi u stanje pozivanja. Nakon pritiska crvenog dugmeta poziv se prekida, a nakon koraka 6 na ekranu u listi istorijata poziva broj koji je pozivan treba da se vidi na 1. mestu.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: Tel A: Pritisnuti Options->Contacts Tel A: Izabrati jedan broj iz liste Tel A: Pritisnuti levo programsko dugme OK Tel A: Nakon 3sec. pritisnuti crveno dugme Tel A: Pritisnuti desno navigaciono dugme		
14	Opis: Pozivanje broja. Program na pozivanom telefonu je u prednjem planu.	Program na telefonu A nakon koraka 3 ispisuje „Dialing number“ i ime telefona B iz liste kontakata. Telefon B nakon koraka 5 ispisuje „Incoming Call“ i tel. Broj ili ime tel. A iz lise kontakata.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: tel A: Pritisnuti Options->Contacts tel A: Izabrati broj telefona B tel A: Potvrditi pozivanje pritiskom na OK tel B: Sačekati na poziv od telefona A tel B: Pritisnuti crveno dugme (RedKey)		

15	Opis: Pozivanje broja. Program na pozivanom telefonu je u pozadini.	Program na telefonu B kad primi poziv zaštićenog razgovora treba da se pojavi u prvom planu ida se na korisničkoj sprezi ispiše „Incoming Call“ i ime telefona B iz liste kontakata	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: tel B: Pritisnuti crveno dugme (RedKey) tel A: Pritisnuti Options->Contacts tel A: Izabрати broj telefona B tel A: Potvrditi pozivanje pritiskom na OK tel B: Sačekati na poziv od telefona A tel B: Pritisnuti crveno dugme (RedKey)		
16	Opis: Pozivanje broja. Program na pozivanom telefonu se nalazi podešavanjima programa.	Nakon primljenog poziva na telefonu B, program treba da se prebaci na glavnu korisničku spregu i da se prikaže tekst „Incoming Call“ i ime telefona A iz liste kontakata	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: tel B: Pritisnuti Options->Settings tel A: Pritisnuti Options->Contacts tel A: Izabрати broj telefona B tel A: Potvrditi pozivanje pritiskom na OK tel B: Sačekati na poziv od telefona A tel B: Pritisnuti Options->Answer		
17	Opis: Pozivanje broja. Program na pozivanom tel. se nalazi u podešavanjima programa u pozadini.	Nakon primljenog poziva na telefonu B, program treba da se prikaže sa glavnom korisničkom spregom u prvom planu sa ispisanim tekstom „Incoming Call“ i ime telefona A iz liste kontakata	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: tel B: Pritisnuti Options->Settings tel B: Pritisnuti crveno dugme (RedKey) tel A: Pritisnuti Options->Contacts tel A: Izabрати i pozvati broj telefona B tel B: Sačekati na poziv i odgovoriti		

18	Opis: Pozivanje broja. Program na pozivanom tel. se nalazi u listi kontakata.	Nakon primljenog poziva poziva na telefonu B, dobija se signalizacija da je primljeni zaštićeni poziv, u prvom planu nalazi se lista kontakata, na Back se prikazuje glavna sprega sa teksto „Incoming Call“ i imenom tel. A iz liste kontakata	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: tel B: Pritisnuti Options->Contacts tel A: Pritisnuti Options->Contacts tel A: Izabrati broj telefona B tel A: Potvrditi pozivanje pritiskom na OK tel B: Sačekati na poziv od telefona A tel B: Pritisnuti desno programsko dugme (Back) tel B: Pritisnuti crveno dugme (RedKey)		
19	Opis: Pozivanje broja. Program na pozivanom tel. se nalazi u listi kontakata u pozadini.	Nakon primljenog poziva ne telefonu B program se prikazuje u prvom planu sa listom kontakata na korisničkoj sprezi. Nakon koraka 7 poziv se odbija.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: tel B: Pritisnuti Options->Contacts tel B: Pritisnuti crveno dugme (RedKey) tel A: Pritisnuti Options->Contacts tel A: Izabrati broj telefona B tel A: Potvrditi pozivanje pritiskom na OK tel B: Sačekati na poziv od telefona A tel B: Pritisnuti crveno dugme (RedKey)		
20	Opis: Pozivanje broja. Program se nalazi u prvom planu sa aktivnim prozorom za pozivanje. Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.	Nakon primljenog poziva na telefonu B, na korisničkoj sprezi se ispisuje tekst „Incoming Call“ sa dodatnim tekstom imena telefona A	PROŠAO

	Postupak: tel A: Pritisnuti Options->Contacts tel A: Izabrati broj telefona B tel A: Potvrditi pozivanje pritiskom na OK tel B: Sačekati na poziv od telefona A tel B: Pritisnuti „Cancel“ programsko dugme	iz liste kontakata	
21	Opis: Pozivanje broja. Program se nalazi u pozadini sa aktivnim prozorom za pozivanje. Početni uslovi: SCA program je pokrenut i nalazi se u pozadini sa aktivnim prozorom za pozivanje. Postupak: tel A: Pritisnuti Options->Contacts tel A: Izabrati broj telefona B tel A: Potvrditi pozivanje pritiskom na OK tel B: Sačekati na poziv od telefona A tel B: Pritisnuti „Cancel“ programsko dugme tel B: Pritisnuti zeleno dugme (GreenKey)	Nakon primljenog poziva na telefonu B, program se podiže u prvi plan. Na korisničkoj sprezi se ispisuje tekst „Incoming Call“ sa dodatnim tekstom imena telefona A iz liste kontakata	PROŠAO

TESTIRANJE ZAŠTIĆENOG POZIVA			
Rb	Opis testa	Očekivani rezultat	Uspešnost
22	Opis: Pozivanje govornog poziva. Program se nalazi u prvom planu. Aktivna glavna korisnička sprega. Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega. Postupak: tel A: Pozovi tel. B preko govornog poziva tel B: Sačekati na poziv od telefona A tel B: Prihvatiti poziv tel B: Prekinuti poziv na crveno dugme	Nakon primljenog govornog poziva program odlazi u pozadinu. Nakon koraka 3 program se vraća u prvi plan (Normalno ponašanje sistema)	PROŠAO

23	Opis: Pozivanje govornog poziva. Program se nalazi u pozadini. Aktivna glavna korisnička sprega.	Nakon primljenog govornog poziva, program se ne podiže u prvi plan.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u pozadini. Aktivna je glavna korisnička sprega.		
	Postupak: tel A: Pozvati tel. B preko govornog poziva tel B: Sačekati na poziv od telefona A tel B: Odbiti poziv		
24	Opis: Pozivanje govornog poziva. Program se nalazi u prvom planu. Aktivna je korisnička sprega podešavanja.	Nakon primljenog govornog poziva program odlazi u pozadinu. Nakon koraka 3 program se vraća u prvi plan (Normalno ponašanje sistema)	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je korisnička sprega za podešavanje rada programa.		
	Postupak: tel A: Pozovi tel. B preko govornog poziva tel B: Sačekati na poziv od telefona A tel B: Prihvatiti poziv		
25	Opis: Pozivanje govornog poziva. Program se nalazi u pozadini. Aktivna je korisnička sprega podešavanja.	Nakon primljenog govornog poziva, program se ne podiže u prvi plan.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u pozadini. Aktivna je korisnička sprega za podešavanje rada programa.		
	Postupak: tel A: Pozvati tel. B preko govornog poziva tel B: Sačekati na poziv od telefona A tel B: Odbiti poziv		

27	Opis: Pozivanje govornog poziva. Program se nalazi u prvom planu. Aktivna glavna korisnička sprega sa listom kontakata.	Nakon primljenog govornog poziva program odlazi u pozadinu. Nakon koraka 3 program se vraća u prvi plan (Normalno ponašanje sistema)	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega sa listom kontakata.		
	Postupak: tel A: Pozovi tel. B preko govornog poziva tel B: Sačekati na poziv od telefona A tel B: Prihvatiti poziv		
28	Opis: Pozivanje govornog poziva. Program se nalazi u pozadini. Aktivna glavna korisnička sprega sa listom kontakata.	Nakon primljenog govornog poziva, program se ne podiže u prvi plan.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u pozadini. Aktivna je glavana korisnička sprega sa listom kontakata.		
	Postupak: tel A: Pozvati tel. B preko govornog poziva tel B: Sačekati na poziv od telefona A tel B: Odbiti poziv		
29	Opis: Pozivanje govornog poziva. Program se nalazi u prvom planu. Aktivna glavna korisnička sprega sa aktivnim prozorom za pozivanje.	Nakon primljenog govornog poziva program odlazi u pozadinu. Nakon koraka 3 program se vraća u prvi plan (Normalno ponašanje sistema)	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega aktivnim prozorom za pozivanje.		
	Postupak: tel A: Pozovi tel. B preko govornog poziva tel B: Sačekati na poziv od telefona A tel B: Prihvatiti poziv		

30	Opis: Pozivanje govornog poziva. Program se nalazi u pozadini. Aktivna glavna korisnička sprega sa aktivnim prozorom za pozivanje.	Nakon primljenog govornog poziva, program se ne podiže u prvi plan.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u pozadini. Aktivna je glavana korisnička sprega sa laktivnim prozorom za pozivanje.		
	Postupak: tel A: Pozvati tel. B preko govornog poziva tel B: Sačekati na poziv od telefona A tel B: Odbiti poziv		
31	Opis: Poziv zaštićenog razgovora. Razgovor prekida pozivana strana.	Nakon koraka 3 poziv zaštićenog razgovora je uspostavljen i moguće je pričati. Nakon koraka 4 program treba da se vrati u početno stanje čekanja na poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: tel A: Pozivati telefon B iz liste kontakata tel B: Sačekati poziv tel B: Prihvatiti poziv pritiskom na zeleno dugme (GreenKey) tel A,B: Sačekati stanje Connected tel B: Pritisnuti crveno dugme (RedKey)		
32	Opis: Poziv zaštićenog razgovora. Razgovor prekida strana koja poziva.	Nakon koraka 3 poziv zaštićenog razgovora je uspostavljen i moguće je pričati. Nakon koraka 4 program treba da se vrati u početno stanje čekanja na poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: tel A: Pozivati telefon B iz liste kontakata tel B: Sačekati poziv tel B: Prihvatiti poziv pritiskom na zeleno dugme (GreenKey) tel A,B: Sačekati stanje Connected tel A: Pritisnuti crveno dugme (RedKey)		

33	Opis: Poziv zaštićenog razgovora. Razgovor prekida strana koja poziva.	Nakon koraka 3 poziv zaštićenog razgovora je uspostavljen i moguće je pričati. Nakon koraka 4 program treba da se vrati u početno stanje čekanja na poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: tel A: Pozvati telefon B iz liste kontakata tel B: Sačekati poziv tel B: Prihvatiti poziv Options->Answer tel A,B: Sačekati stanje Connected tel A: Prekinuti vezu Options->End call		
34	Opis: Poziv zaštićenog razgovora. Prihvatanje veze i pritiskanje zelenog dugmeta više puta.	Nakon koraka 3 veza treba da počne da se uspostavlja. Nakon koraka 4 program treba da nastavi sa uspostavljanjem veze. Nakon koraka 6 program se vraća u početno stanje čekanja na poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana sprega.		
	Postupak: tel A: Pozvati telefon B ručnim unošenjem broja tel B: Sačekati na poziv tel B: Prihvatiti poziv na zeleno dugme tel B: Pritisnuti zeleno dugme 3 puta tel A,B: Sačekati stanje Connected tel A: Prekinuti vezu na crveno dugme		
35	Opis: Poziv zaštićenog razgovora. Prekid razgovora pritiskom više puta na crveno dugme.	Prvim pritiskom na crveno dugme se prekida veza, sledećim pritiskom se program šalje u pozadinu. Program treba da se nalazi u pozadini.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: tel A: Pozvati telefon B iz liste kontakata tel B: Sačekati i prihvatiti poziv tel A,B: Sačekati stanje Connected tel B: Prekinuti vezu sa više pritisaka na crveno dugme		

36	Opis: Poziv zaštićenog razgovora. Telefon B ne odgovara na poziv, nakon čega telefon A prekida vezu.	Nakon koraka 2 telefon prima poziv. Nakon koraka 3 program se vraća u početno stanje čekanja na poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: tel A: Pozvati telefon B iz liste kontakata tel B: Ne odgovoriti na poziv tel A: Prekinuti zvanje sa crvenim dugmetom		
37	Opis: Poziv zaštićenog razgovora. Telefon B odbija dolazeći poziv.	Nakon odbijenog poziva program treba da se vrati u početno stanje čekanja na poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: tel A: Pozvati telefon B iz liste kontakata tel B: Sačekati na poziv tel B: Odbiti poziv na crveno dugme		
38	Opis: Poziv zaštićenog razgovora. Telefon A poziva samog sebe.	Nakon pozivanja dobija sistemski prozor sa informacijom „Number busy“, nakon čega se program vraća u početno stanje.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: tel A: Pozvati sopstveni broj		
39	Opis: Poziv zaštićenog razgovora. Telefon B poziva nepoznati broj.	Nakon pozivanja nepoznatog broja prikazuje se poruka „Network Error“, nakon čega se program vraća u početno stanje	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: Tel B: Izabrati opciju Options->Dial Tel B: Uneti broj „11222333123“ i pozvati		

40	Opis: Poziv zaštićenog razgovora. Telefon B poziva broj koji nije dostupan.	Nakon pozivanja nedostupnog broja prikazuje se poruka „Network Error“, nakon čega se program vraća u početno stanje	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: Tel B: Isključiti mobilni uređaj Tel A: Pozvati telefon B iz liste kontakata		
41	Opis: Poziv zaštićenog razgovora. Telefon B ne odgovara na poziv (propušteni poziv).	Nakon što mreža prekine poziv usled neodgovaranja na poziv program se vraća u početno stanje. Nakon koraka 3 u listi istorijata poziva, prikazan je poziv kao propušteni.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: tel A: Pozvati telefon B iz liste kontakata tel B: Ne odgovoriti na poziv. Pustiti da odzvoni poziv do kraja tel B: Pritisnuti navigacioni taster desno		
42	Opis: Poziv zaštićenog razgovora. Telefon A prekida poziv pre nego što se telefon B javio.	Nakon koraka 3 program treba da se vrati u početno stanje čekanja na poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: Tel A: Pozvati telefon B iz liste kontakata Tel B: Sačekati na i ne odgovarati na poziv Tel A: Prekinuti poziv na crveno dugme		
43	Opis: Poziv zaštićenog razgovora. Telefon A prekida vezu u stanju Connecting.	Nakon koraka 2 program prelazi u stanje Connecting. Nakon korak 3 program se vraća u početno stanje slušanja na dolazni poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu.		
	Postupak: Tel A: Pozvati telefon B iz liste kontakata Tel B: Sačekati na poziv i odgovoriti Tel A: Prkinuti poziv na crveno dugme		

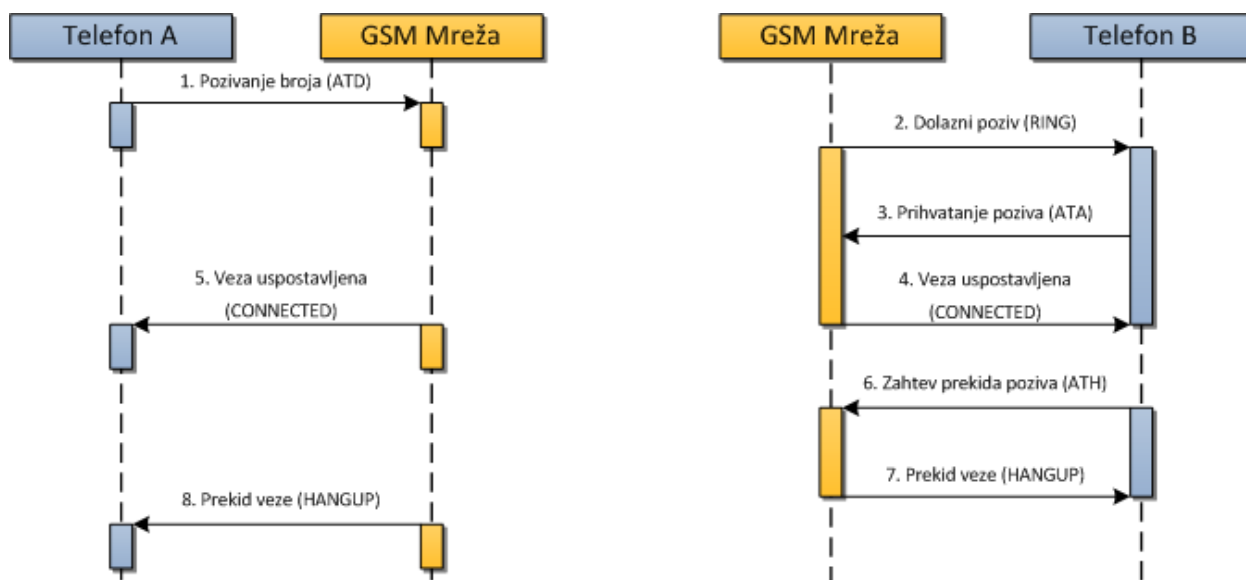
44	Opis: Poziv zaštićenog razgovora. Pozivani telefon prekida vezu u stanju Connecting.	Nakon koraka 2 program prelazi u stanje Connecting. Nakon korak 3 program se vraća u početno stanje slušanja na dolazni poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: Tel A: Pozvati telefon B iz liste kontakata Tel B: Sačekati i primiti poziva Tel B: Prekinuti vezu na crveno dugme		
45	Opis: Poziv zaštićenog razgovora. Reagovanje programa na pritiske dugmadi brojeva u toku poziva.	Nakon korak 3 poziv je uspostavljen i razgovor je moguć. Nakon korak 4 ne treba ništa da se desi. Nakon korak 5 program se vraća u početno stanje.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u prvom planu. Aktivna je glavana korisnička sprega.		
	Postupak: Tel A: Pozvati telefon B iz liste kontakata Tel B: Sačekati i primiti poziv Tel A,B: Sačekati na stanje Connected Tel A: Pritisnuti dugmad „1“, „6“ Tel B: Prekinuti vezu na Options->End Call		
46	Opis: Poziv zaštićenog razgovora. Rad programa nakon što sistem ode u Sleep režim rada.	Nakon 1 minuta od početka stanja Connected sistem odlazi u Sleep mod rada. Program ne sme da ode pozadinu. Nakon koraka 6 program treba da se vrati početno stanje čekanja na dolazni poziv.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u pozadini. Aktivna je glavana korisnička sprega.		
	Postupak: Tel A: Podesiti Tools->Settings->Phone->Display->Power saver time out na 1 min Tel A: Pokrenuti program SCA Tel A: Pozvati telefon B iz liste kontakata Tel B: Sačekati i odgovoriti na poziv Tel A,B: Sačekati stanje Connected Tel A: Prekinuti poziv posle 2 min na crveno		

47	Opis: Poziv zaštićenog razgovora. Uključivanje zvučnika u toku razgovora.	Nakon izvršenog testa zvuk treba da se prebaci sa slušalice na zvučnik na telefonu B.	PROŠAO
	Početni uslovi: SCA program je pokrenut i nalazi se u pozadini. Aktivna je glavana korisnička sprega.		
	Postupak: Tel A: Pozvati telefon B iz liste kontakata Tel B: Sačekati i odgovoriti na poziv Tel A,B: Sačekati na stanje Connected Tel B: Pritisnuti programsko dugme „Loud“		

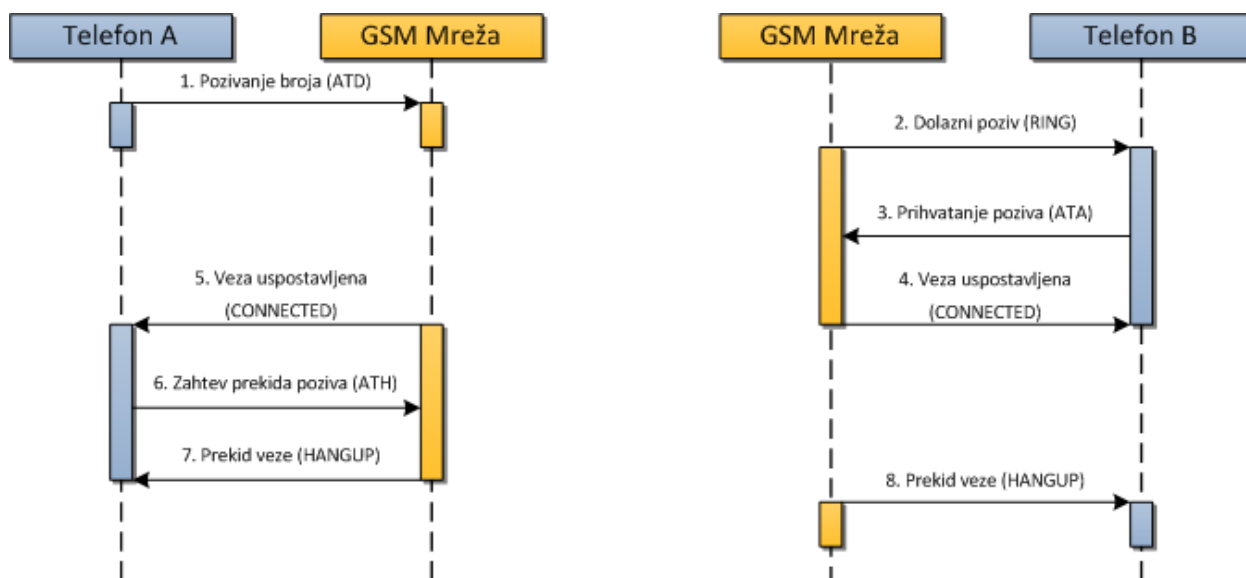
7.2 Testovi za automatsko testiranje

Testovi koji se koriste prilikom automatskog testiranja. Svaki test se sastoji iz dva dela, strane koje poziva (na slici levo) i strane koja prima poziva (na slici desno).

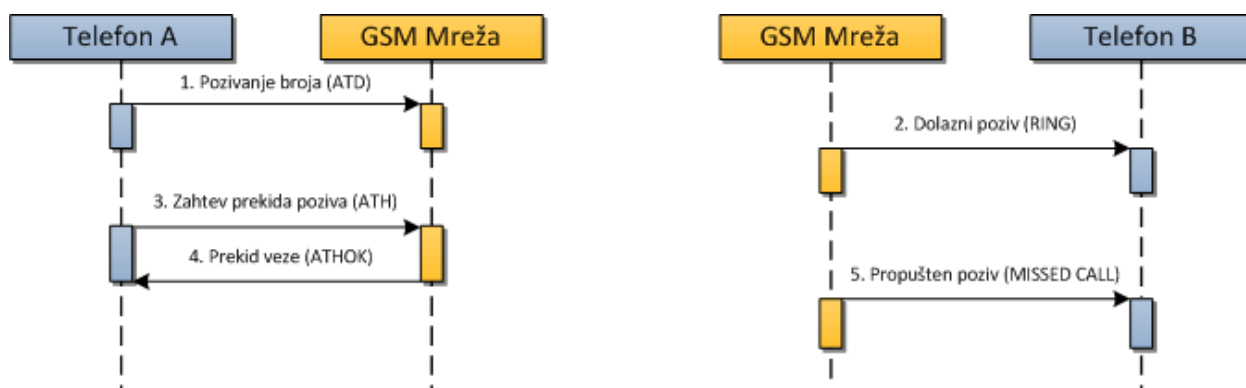
1. Telefon A poziva telefon B koji odgovara na poziv. Oba telefona čekaju na stanje Connected nakon čega telefon B prekida vezu.



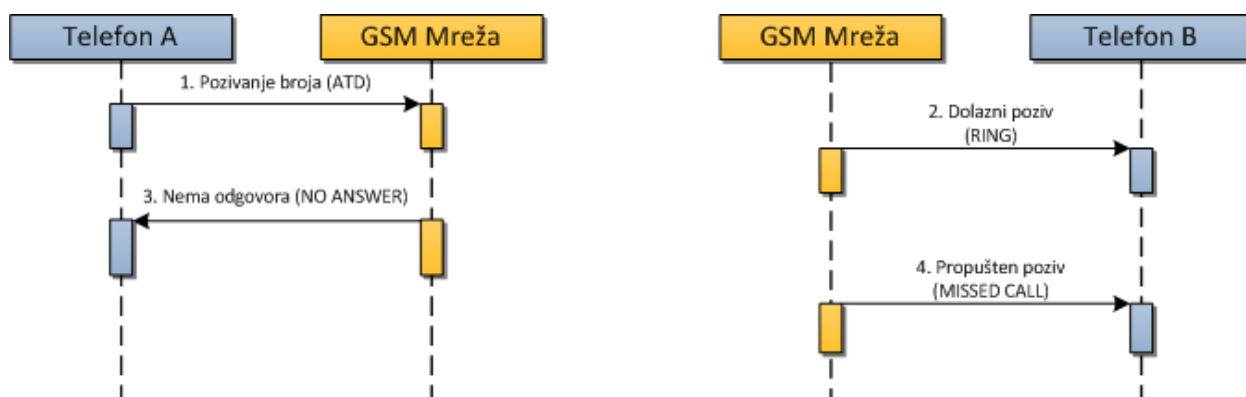
2. Telefon A poziva telefon B koji odgovara na poziv. Oba telefona čekaju na stanje Connected nakon čega telefon A prekida vezu.



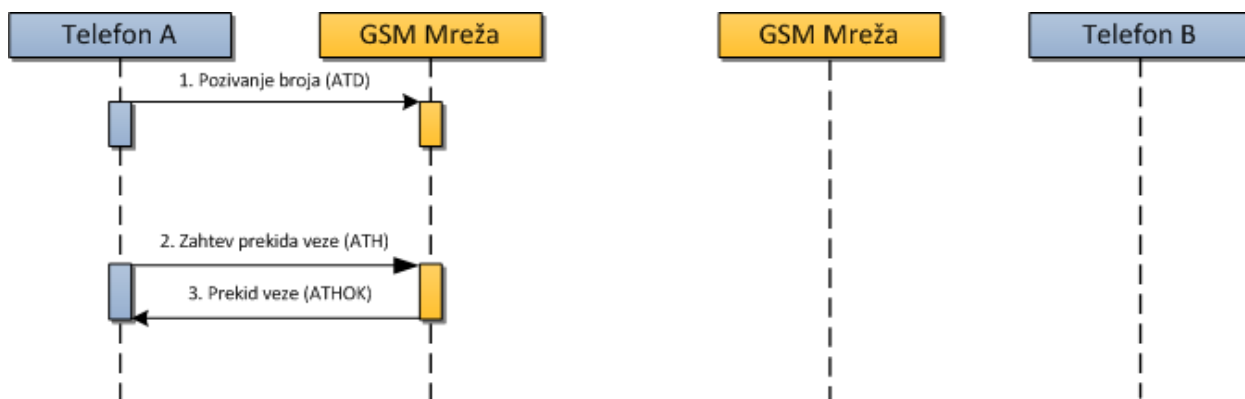
3. Telefon A poziva telefon B koji ne odgovara na poziv. Pre isteka poziva telefon A prekida vezu, što uzrokuje propušteni poziv na telefonu B.



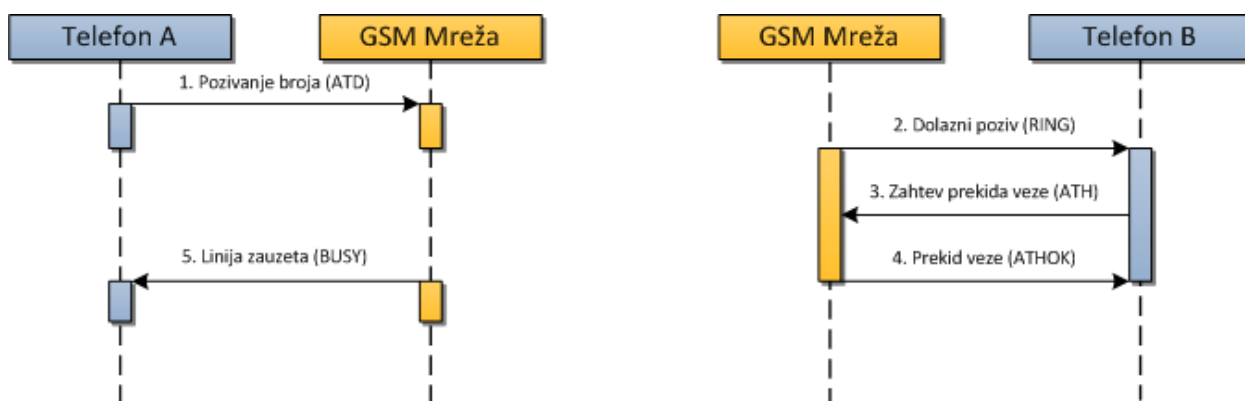
4. Telefon A poziva telefon B koji ne odgovara na poziv. Nakon isteka vremena trajanja pozivanja mreža prekida poziv. Telefon B beleži propušteni poziv u istoriju poziva.



5. Telefon A poziva telefon B. Pre nego što telefon B primi poziv telefon A prekida poziv.



6. Telefon A poziva telefon B koji odbija poziv.



7. Telefon A poziva telefon B koji prihvata poziv. Oba telefona čekaju na stanje Connected, nakon čega telefoni prekidaju vezu istovremeno.

