پروژه باریم

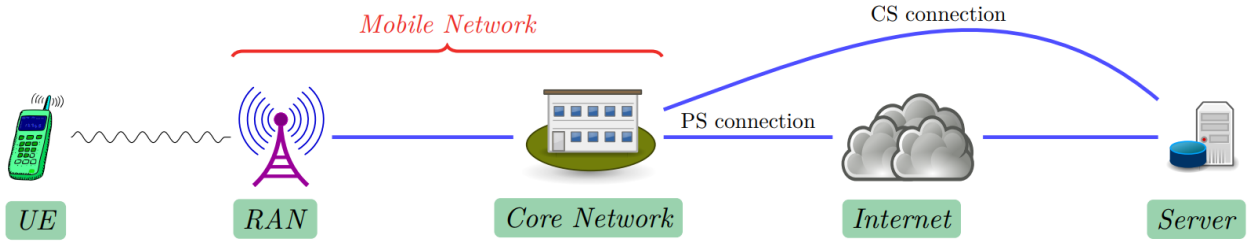
درس آشنایی با شبکه های تلفن همراه

غزل عربعلی - ۹۷۵۲۱۳۹۶، بهاره کاوسی نژاد - ۹۹۴۳۱۲۱۷

آخرین ویرایش: ۱۵ تیر ۱۴۰۳ در ساعت ۱۹ و ۱ دقیقه

۱ شرح پروژه

گسترش روزافزون شبکه های تلفن همراه به ویژه شبکه های نسل چهار و پنج، موجب شده است که این شبکه ها به عنوان بزرگترین شبکه دسترسی [۱] ، برای دستیابی به خدمات اینترنت بشمار آید. پرواضح است که در این بین، مساله امنیت [۲] برنامه های کاربردی [۳] و ساخت یک برنامه کاربردی با یک ارتباط امن، یکی از مهم ترین مسایل این حوزه خواهد بود. گرچه باید به این نکته توجه داشت که امنیت در یک ارتباط از طریق شبکه های تلفن همراه را، نباید تنها به مساله امنیت در دو سوی مشتری [۴] و خدمت گزار [۵] تقلیل داد؛ بلکه در جای جای این ارتباط، ما می توانیم با حملات متعددی مواجه شویم، که می تواند محرمانگی [۶] ، یکپارچگی [۷] و حریم خصوصی [۸] ما را هدف قرار دهد. شکل ۱.۱ نمایی از ارتباط یک مشتری با خدمت گزار را در بستر های مختلف از طریق شبکه های تلفن همراه به زیبایی نشان می دهد.



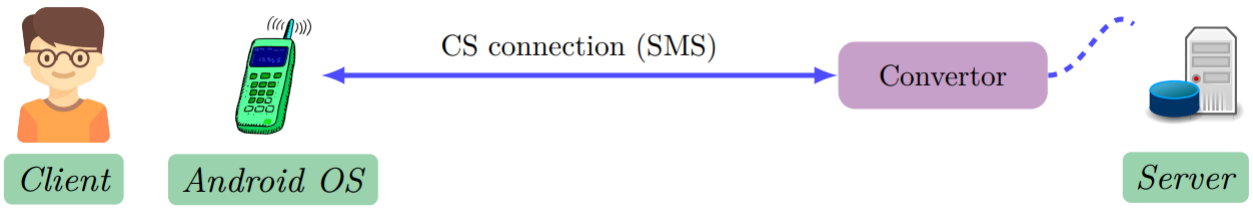
شکل ۱.۱: ارتباط بین مشتری با خدمت گزار از طریق شبکه های تلفن همراه بر روی بسترهای مختلف

در مساله پیش رو، فرض می کنیم که یک برنامه کاربردی داریم، که توسط برنامه UE می شود. UE از دیدگاه ما هر ابزاری است که توسط آن بتوان به شبکه های تلفن همراه متصل شد. UE می تواند گوشی تلفن همراه، تبلت و یا حتی هر شی در IoT [۹] باشد. گرچه در این پروژه، ما تنها بر روی گوشی های تلفن همراه و تبلت ها تمرکز خواهیم کرد.

برنامه کاربردی UE قرار است تا از طریق بسترهای موجود در شبکه های تلفن همراه به یک خدمت گزار مشخص متصل شوند و با آن تبادل اطلاعات داشته باشند. در این جا ما دو راه کار برای اتصال به خدمت گزار داریم. در راه کار نخست و بدیهی ترین شیوه، ما از طریق بستر اینترنت با خدمت گزار به تبادل داده مبادرت می ورزیم. ما اصطلاحا به این شیوه اتصال از طریق PS [۱۰] می گوییم.

[۱] Access Network
[۲] Security
[۳] Application
[۴] Client
[۵] Server
[۶] Confidentiality
[۷] Integrity
[۸] Privacy
[۹] Internet of Things
[۱۰] Packet-switched

بالاخره باید پذیرفت که دنیای اینترنت، مخاطرات پیدا و پنهان فراوانی دارد. اتصال از طریق خدمات CS [11] [12] نظیر تماس [13] و SMS [14] ، می تواند راه فراری از مخاطرات دنیای اینترنت باشد. در این پروژه، ما فرض می کنیم که اتصال مشتری به خدمت گزار را از طریق SMS ، برقرار خواهد شد.



شکل ۲.۱: معماری سطح بالای سامانه

در این جا برای سادگی فرض کنید که دو گوشی داریم. گوشی سمت مشتری و گوشی که ما به عنوان خدمت گزار از آن استفاده می کنیم. در سمت خدمت گزار (که در حقیقت یک گوشی معمولی است)، یک برنامه Android ای با کارکرد Backend نصب می شود. مشتری از طریق SMS فرمان ها را به سمت مقابل (خدمت گزار) ارسال می کند. مشتری می بایست به صورت مداوم اطلاعات مربوط به توان دریافتی و تکنولوژی سلول خدمتگزار [15] و مکان دریافت این اطلاعات را در صورتی که توان از یک سطح آستانه معین پایین بیاید در قالب یک پیام برای خدمت گزار ارسال کند. در این سامانه می بایست به نکات زیر دقت کنید:

- برنامه سمت خدمت گزار می بایست به صورت یک سرویس در Android باشد، البته برای مدیریت و پیکربندی آن می توان یک برنامه UI دار نیز داشته باشیم.
- فرض کنید که همگان پروتکل ارتباطی شما را که مبتنی بر SMS است می دانند. اگر اجازه دهیم SMS از هر شماره ای به سمت خدمت گزار ارسال شود، رویه ای در نظر بگیرید که جلوی دسترسی های غیرمجاز را بگیرد. شاید یک رویه ساده، ارسال یک رمز عبور [16] در ابتدای SMS است. تلاش کنید تا رویه های بهتری برای حل این چالش در نظر بگیرید.
- در هنگامی که مشتری درخواست خود را برای خدمت گزار ارسال می کند، خدمت گزار درخواست را می بایست اجرا کند و پاسخ را در یک SMS جداگانه برای مشتری ارسال کند. دقت کنید اگر بتوانید باید تشخیص بدهید که Delivery بر می گردد یا خیر. اگر برنگشت باید پیام را دوباره ارسال کنیم.
- در پیام ارسالی از سوی مشتری، می بایست مکان اندازه گیری، مقداری اندازه گیری و اطلاعات سلولی که به آن متصل است را ارسال کند.
- پروتکل ارتباطی را باید به صورت کامل مستند بکنید، و باید مبتنی بر پروتکل SMPP [17] باشد.

[11] Service
[12] Circuit-switched
[13] Call
[14] Short Message Service
[15] Serving Cell
[16] Password
[17] Short Message Peer-to-Peer

۲ توضیحات کدها

۱.۲ فایل SMPP.kt

کلاس SMPP که از DefaultSmppSessionHandler به ارث برده شده، برای مدیریت جلسات SMPP در یک برنامه Android طراحی شده است. این کلاس شامل متغیرهایی برای نگهداری تنظیمات SMPP (مانند میزبان، پورت، کاربر، و رمز عبور) و همچنین روشی برای پیکربندی این تنظیمات می‌باشد. متد configure برای تنظیم جزئیات اتصال SMPP استفاده می‌شود.

۱.۱.۲ متد sendSMS

متد sendSMS تلاش می‌کند تا یک پیام SMS به شماره داده شده ارسال کند. این متد ابتدا یک SMPP client و session ایجاد می‌کندو پیام را به قسمت‌هایی تقسیم می‌کند. با دو آستانه مقدار قدرت سیگنال را تست می کند. اگر سیگنال قدرت سلول سرویس‌دهنده کمتر از 50- و یا 110- باشد، قسمت‌های پیام را ارسال می‌کند. سپس به مدت ۱۰ ثانیه منتظر می‌ماند و در نهایت session و client را می‌بندد و از بین می‌برد. اگر خطایی رخ دهد، آن را در log ثبت می‌کند و مقدار false را بازمی‌گرداند.

۲.۱.۲ متد getSessionConfig

متد getSessionConfig پیکربندی جلسه SMPP را با استفاده از تنظیمات ذخیره شده در shared preferences بازمی‌گرداند. این متد یک شیء SmppSessionConfiguration را ایجاد می‌کند و نوع جلسه، میزبان، پورت، شناسه سیستم، و رمز عبور را تنظیم می‌کند.

۳.۱.۲ متد createSubmitSm

متد createSubmitSm یک شیء SubmitSm ایجاد می‌کند که برای ارسال پیام SMS استفاده می‌شود. این متد آدرس‌های مبدا و مقصد، کدینگ داده، و پیام کوتاه را تنظیم می‌کند و متن پیام را با استفاده از کاراکترست داده شده کدگذاری می‌کند.

۴.۱.۲ متد splitMessage

متد splitMessage پیام را به قسمت‌هایی با حداکثر طول بایت مشخص تقسیم می‌کند. این متد از یک کدکننده کاراکتر برای تبدیل پیام به یک buffer بایت استفاده می‌کند و پیام را به قسمت‌هایی تقسیم می‌کند که هر کدام طولی کمتر از maxByteLength دارند.

۵.۱.۲ متد extractServingCellSignalStrength

متد extractServingCellSignalStrength قدرت سیگنال سلول سرویس‌دهنده را از یک پیام استخراج می‌کند. این متد پیام را به بخش‌هایی بر اساس جداکننده ؛ تقسیم می‌کند و سپس اولین بخش را به عناصر جداگانه بر اساس ، تقسیم می‌کند و قدرت سیگنال را به عنوان یک عدد صحیح بازمی‌گرداند.

۶.۱.۲ متد firePduRequestReceived

در نهایت، متد firePduRequestReceived که از DefaultSmppSessionHandler به ارث برده شده، درخواست‌های PDU دریافت شده را پردازش می‌کند. اگر کدینگ داده پیام ۰ باشد، آدرس مبدا، مقصد، و محتوای پیام را چاپ می‌کند و یک پاسخ ایجاد می‌کند و بازمی‌گرداند.

۲.۲ فایل SMSActivation

در کلاس SMSActivation که از AppCompatActivity به ارث برده شده، متغیرهای لازم برای مدیریت مجوزهای دسترسی به پیامک‌ها، آداپتور چت و آیتم‌های چت تعریف می‌شود. در متد onCreate ، ابتدا چک می‌شود که آیا مجوز خواندن پیامک‌ها به برنامه داده شده است یا خیر. اگر مجوز داده نشده باشد، درخواست مجوز ارسال می‌شود و سپس پیامک‌ها خوانده می‌شوند. اگر مجوز قبلاً داده شده باشد، پیامک‌ها مستقیماً خوانده می‌شوند. سپس RecyclerView با استفاده از LinearLayoutManager مقداردهی اولیه می‌شود و آداپتور چت به آن متصل می‌شود. همچنین یک دکمه برگشت تنظیم می‌شود که با کلیک بر روی آن، فعالیت فعلی خاتمه می‌یابد.

۱.۲.۲ متد updateChatItems

در متد updateChatItems ، آیتم‌های چت جدید جایگزین آیتم‌های فعلی می‌شوند و به آداپتور اطلاع داده می‌شود که داده‌ها تغییر کرده‌اند. این متد برای به‌روزرسانی آیتم‌های چت استفاده می‌شود. متد loadChatItems نمونه‌ای از نحوه استفاده از این متد را نشان می‌دهد. در این متد، پیامک‌های جدید خوانده می‌شوند و سپس با استفاده از updateChatItems آیتم‌های جدید به‌روزرسانی می‌شوند.

۲.۲.۲ متد readSms

متد readSms پیامک‌های دستگاه را خوانده و آن‌ها را به لیستی از آیتم‌های چت تبدیل می‌کند. این متد ابتدا ستون‌های مورد نیاز برای خواندن پیامک‌ها (آدرس، متن و نوع پیامک) را مشخص می‌کند و سپس یک کوئری برای دریافت این داده‌ها از contentResolver ارسال می‌کند. در حالی که cursor به پیامک‌ها اشاره می‌کند، پیامک‌ها یکی یکی خوانده می‌شوند و اگر متن پیامک شامل عبارت " SMPP " باشد، متن پیامک پاک‌سازی می‌شود و سپس به لیست آیتم‌های چت اضافه می‌شود. در نهایت cursor بسته می‌شود و لیست آیتم‌های چت بازگردانده می‌شود.

۳.۲.۲ متد SMSActivation

در کلاس SMSActivation متغیرهای permission و requestCode برای مدیریت مجوزهای دسترسی به پیامک‌ها استفاده می‌شوند. همچنین recyclerView برای نمایش لیست پیامک‌ها، chatAdapter برای مدیریت آیتم‌های چت و chatItems برای نگهداری لیست پیامک‌ها تعریف شده‌اند. این متغیرها در متد onCreate مقداردهی اولیه می‌شوند و تنظیمات اولیه برای نمایش پیامک‌ها و مدیریت تعاملات کاربر با دکمه برگشت انجام می‌شود.

۳.۲ فایل SMSReceiver

کلاس SMSReceiver که از BroadcastReceiver به ارث برده شده است، برای مدیریت پیام‌های ورودی SMS طراحی شده است. در این کلاس، متد onReceive بازنویسی شده تا پیام‌های SMS ورودی را پردازش کند. وقتی یک پیام SMS دریافت می‌شود، این متد فراخوانی می‌شود و پیام‌ها را از طریق intent دریافت می‌کند.

ابتدا، اگر bundle که حاوی داده‌های پیام است، غیر null باشد، مجموعه‌ای از PDUs (Protocol Data Units) از bundle استخراج می‌شوند. هر PDU به یک پیام SMS تبدیل می‌شود و از آن شماره فرستنده و متن پیام استخراج می‌گردد. سپس این اطلاعات در log برای اهداف اشکال‌زدایی ثبت می‌شود.

در مرحله بعد، چک می‌شود که آیا متن پیام حاوی عبارت "پیام شما دریافت شد" است یا خیر. اگر این عبارت در پیام یافت شود، یک intent جدید برای شروع MainActivity ایجاد می‌شود. این intent شامل flag هایی است که نحوه اجرای MainActivity را تعیین می‌کنند و اطلاعات مربوط به پیام را به صورت extras به آن اضافه می‌کنند.

در نهایت، MainActivity با استفاده از intent ایجاد شده و شامل اطلاعات پیام، آغاز می‌شود. این اقدام به MainActivity اجازه می‌دهد که پیام را دریافت کرده و به آن پاسخ دهد یا آن را نمایش دهد. این فرآیند به طور کامل، امکان پردازش و مدیریت پیام‌های SMS را در برنامه فراهم می‌آورد.

۴.۲ فایل SMSActivation

در این کد، ابتدا در onCreate ، تنظیمات اولیه انجام می‌شود. این فعالیت از AppCompatActivity ارث‌بری می‌کند و از XML فایل activity_sms را به عنوان طرح برای نمایش استفاده می‌کند. نوار ابزار این فعالیت مخفی شده است تا به نمایشگر متصل شود. سپس، دسترسی به خواندن پیام‌های SMS چک می‌شود و اگر اجازه دسترسی صادر نشده باشد، درخواست مجوز از کاربر گرفته می‌شود. پیام‌های SMS خوانده شده از readSms بازیابی و در RecyclerView نمایش داده می‌شوند.

در onCreate ، RecyclerView برای نمایش موارد چت مقداردهی اولیه می‌شود. از LinearLayoutManager برای مدیریت ترتیب عناصر استفاده می‌شود و یک ChatAdapter جهت اتصال به RecyclerView ساخته و تنظیم می‌شود. دکمه بازگشت (backButton) برای بستن فعالیت فعلی پیکربندی شده است.

در updateChatItems ، لیست موارد چت به روزرسانی می‌شود و تغییرات به ChatAdapter اطلاع داده می‌شود تا موارد نمایش داده شده در RecyclerView به روز شوند.

loadChatItems یک مثال از تابع updateChatItems است که هر زمان که موارد چت جدیدی در دسترس باشد، برای به روزرسانی اطلاعات استفاده می‌شود.

در readSms ، از Content Provider مربوط به SMS برای بازیابی پیام‌های متنی استفاده می‌شود. این متد اطلاعاتی از شماره گیرنده (ADDRESS)، متن پیام (BODY) و نوع پیام (TYPE) را دریافت می‌کند. سپس پیام‌هایی که عبارت " SMPP " را در متن خود دارند را فیلتر و به لیست ChatAdapter . ChatItem اضافه می‌کند و در نهایت مجموعه داده را به عنوان خروجی بازمی‌گرداند.

۳ مراجع

۱ •