VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

ŠIFROVÁNÍ NAD TEXTOVÝMI ZPRÁVAMI PRO ANDROID

MESSAGE ENCRYPTION OVER TEXT MESSENGERS FOR ANDROID

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **DAVID BALVÍN**
AUTHOR

VEDOUCÍ PRÁCE **Doc. Dr. Ing. DUŠAN KOLÁŘ,**
SUPERVISOR

BRNO 2017

Zadání bakalářské práce/19727/2016/xbalvi00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Balvín David**
Obor: Informační technologie
Téma: **Šifrování nad textovými zprávami pro Android**
Message Encryption over Text Messengers for Android
Kategorie: Informační systémy

Pokyny:

1. Prostudujte aplikační možnosti platformy Android, dále prostudujte standardní a populární aplikace pro zasílání zpráv a textových zpráv (SMS).
2. Na základě analýzy navrhněte aplikaci, která umožní přenos šifrovaných zpráv s využitím existujících aplikací/protokolů - aplikace a protokoly konzultujte s vedoucím.
3. Aplikaci z bodu 2 implementujte v prostředí Android - podporované verze konzultujte s vedoucím.
4. Aplikaci důkladně otestujte.
5. Zhodnoťte svůj přínos i celou práci, diskutujte možná rozšíření.

Literatura:

- Ryan Cohen, Tao Wang: GUI Design for Android Apps, Apress, pp. 147, August 2014
- Vybrané stati na https://developer.android.com
- Aplikace Messenger, Xabber, Skype, apod.
- Dále dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- První 2 body zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Kolář Dušan, doc. Dr. Ing.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Cílem této práce bylo navrhnout a implementovat aplikaci, která bude zajišťovat zašifrovanou komunikaci pomocí sítě GSM. Aplikace zvládá automatické šifrování pomocí symetrické šifry při odesílání a dešifrování na požádání při přijmutí zprávy. Aplikace též kontroluje konzistenci zpráv v případě zřetězených zpráv. K aplikaci bylo vytvořeno uživatelské rozhraní, inspirované defaultní SMS aplikací pro operační systém Android a populárním komunikátorem Messenger.

Abstract

The main task of this thesis was to design and implement an application that would handle encrypted communication using the GSM network. The app is able to handle automatic encryption using a symetric-key algorithm for sending messages and decryption on demand for receiving/reading messages. The app also checks the consistency of received messages for concatenated SMS messages. A simple user interface was also created for this app, taking ideas from default SMS app for Android and popular Messenger.

Klíčová slova

Android, aplikace, kryptografie, SMS, GSM

Keywords

Android, application, cryptography, SMS, GSM

Citace

BALVÍN, David. *Šifrování nad textovými zprávami pro Android.* Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Dr. Ing. Dušan Kolář,

Šifrování nad textovými zprávami pro Android

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. Dr. Ing. Dušana Koláře. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
David Balvín
17. května 2017

Poděkování

Poděkování patří především panu Doc. Dr. Ing. Dušanu Kolářovi za odbornou pomoc a cenné rady při vývoji této práce. Poděkování patří také uživatelům, kteří aplikaci během vývoje otestovali a poskytnuli cennou zpětnou vazbu.

Obsah

Kapitola 1

Úvod

V dnešní informační době není přetažené říct, že správná data mají cenu zlata. Vzrůstající efektivita umělých inteligencí a neuronových sítí umožňují automatizovanější zpracování informací na internetu, mezi něž patří i komunikace uživatelů. Na druhé straně však vzniká stále větší potřeba pro bezpečnou komunikaci mezi dvěma subjekty, bez možnosti přečtení třetí stranou.

Při poslechu každodenních zpráv o napadení zařízení, prolomení zabezpečení či vyzrazení obchodních tajemství si člověk snadno uvědomí, jak i každodenní činnosti mohou být předmětem útoku na danou osobu. Jak i tak obyčejná věc, jako SMS, vyžaduje správnost HW zařízení, SW na tomto zařízení, protokolu a samotné sítě pro přenos. Proto je dobré tato zabezpečení ještě zvyšit pomocí zašifrování samotné zprávy. Pokud tedy útočník obejde zabezpečení GSM sítě a získá SMS zprávu, nezíská z ní žádná data, pouze zašifrovaný text.

Z takovýchto a dalších, níže popsaných důvodů, byla vytvořena aplikace...

Tombstone Talk

Kapitola 2

Analýza a specifikace požadavků

Tato kapitola se věnuje rozboru požadavků a popisu využívaných technologií, ať již programových či síťových. Z této analýzy vznikne návrh v další kapitole, který položí základy k samotné aplikaci. Též zde prozkoumáme možnosti již existujících aplikací a využijeme je k návrhu naší aplikace.

2.1 SMS

SMS, zkratka pro ***S**hort **M**essage **S**ervice* (česky *Služba krátkých textových zpráv*), je celosvětově dostupná služba pro posílání krátkých zpráv, většinou pomocí mobilních telefonů. Označení SMS se také často používá pro samotnou zprávu poslanou skrz tuto službu.

Služba SMS využívá nejčastěji ke svému provozu síť **GSM** (viz níže). Samotná SMS zpráva se dle standardu kóduje do maximálně 1120 bitů na jednu SMS. Standard umožňuje na kompatibilním zařízení využívat „řetězení zpráv“, pomocí odebrání určité kapacity zprávy pro definování hlavičky, viz níže. Dle použitého kódování se může jednat o následující maximální velikost textu pro jednu SMS zprávu (*Druhé číslo udává hodnotu při zřetězené zprávě*):

- **160 / 153 znaků** – 7 bitové kódování ve znakové sadě **GSM 03.38**
- **70 / 67 znaků** – 16 bitové kódování ve znakové sadě **UCS-2**

7 bitové kódování je implicitní, 16 bitového se využije pouze v případě, že zpráva obsahuje byť i jeden znak mimo 7 bitové kódování (například znaky emoji, definované v Unicode, jsou psány pomocí UCS-2). Standard ještě popisuje velikost 140 znaků pro kódovaná 8 bitová data, nicméně tímto se (minimálně ve verzi pro tuto práci) zabývat není třeba.

Služba je populární především pro její dostupnost, téměř každý dnes vlastní telefon schopný přijímat SMS. Dále je to také absence jakékoliv registrace, vyjma potřeby vlastnit SIM kartu (viz níže) od operátora a paušál či kredit pro odeslání. V neposlední řadě je to též v nutné kvalitě signálu, kdy i při velmi špatném je možno odeslat SMS, naopak u komunikátorů postavených na technologii IP často dochází k nemožnosti odeslat při slabém signálu.

Maximální počet znaků na zprávu značně omezuje možnosti šifrování, při použití mnohých algoritmů dojde ke zvětšení velikosti oproti původnímu textu. Je tedy třeba zvolit vhodný šifrovací algoritmus (viz níže) a taktéž použít kompresní algoritmus, například v aplikaci použité Huffmanovo kódování.

2.2 GSM

GSM, zkratka pro ***G**lobal **S**ystem for **M**obile Communications* (česky *Globální Systém pro Mobilní komunikaci*), je označení pro nejpoužívanější standard pro mobilní komunikaci a zároveň označení pro síť umožňující takovouto komunikaci. Byl navrhnut nejen pro umožnění telefonování, ale i používání další služeb, jako jsou zprávy SMS, datové přenosy, hlasová schránka, přesměrování hovorů a jiné. Jednalo se také, na rozdíl od předchozích systémů, o digitální přenos, který umožňuje šifrování, lepší využití jednotlivých buněk, menší spotřebu při hovoru a snadný přenos čísla na jiné zařízení pomocí SIM karty (viz níže). GSM síť je buňková, což znamená, že zařízení se do sítě připojuje pomocí nejbližší buňky, skrz kterou pak dále komunikuje se zbytkem sítě. Označuje se jako *síť druhé generace* (2G).

2.2.1 Historie GSM

Vývoj začal v 80. letech minulého století, kdy pracovní skupina pod názvem ***G**roupe **S**pécial **M**obile* vyvíjela a navrhla první verze tohoto standardu. V té době se více používaly dnes již nepoužívané standardy **NMT** a **AMPS**, které ale umožňovaly pouze telefonní hovory, na rozdíl od systému GSM.

Technické základy sítě byly definovány v roce 1987, první specifika GSM v roce 1990. Od roku 1998 se o specifikaci GSM stará projekt **3GPP** (***3**rd **G**eneration **P**artnership **P**roject*), který se vyjma GSM stará i o specifikace *třetí generace sítí* (**3G**, například **HSPDA**), které se soustavně označují jako **UMTS** (**U**niversal **M**obile **T**elecommunications **S**ystem), přidávající nové možnosti komunikace a zabezpečení.

2.2.2 GSM 03.38

GSM 03.38 [10] je znaková sada využívaná v SMS zprávách, definovaná pracovní skupinou GSM. Od dob, kdy se o GSM stará projekt 3GPP, se znaková sada označuje jako **3GPP 23.038**. Tabulka znaků je vyzobrazena v příloze A.1. Tabulka je rozdělena do dvou částí. V první části se znaky píši jejich standartním zakódováním. Druhá část vyžaduje zapsání znaku ESC a následné zapsání znaku dle tabulky. Tato druhá část tabulky se využívá k zapsání některých národních znaků, například pro české znaky ř, š, ž apod. Specifikace nevynucuje podporu pro tuto druhou část tabulky, v tomto případě se znak ESC při přijetí interpretuje jako mezera a následuje znak, který by pocházel z první části tabulky, či zakóduje do USC-2 během odesílání zprávy. Při použití rozšířené tabulky je potřeba definovat použité kódování rozšířené tabulky v hlavičce SMS zprávy.

2.2.3 GSM 03.40

GSM 03.40 [11] je standard popisující formát hlavičky SMS zprávy, sloužící pro nastavení příznaků zprávy a kontrolu přenosu zprávy. Standard popisuje vynucené části standardu, ale i nevynucené části, kde není zaručena 100% funkcionalita a zachování při přenosu sítí, například výše uvedené určení kódování rozšířené tabulky GSM 03.38. Od převzetí správy GSM projektem 3GPP se standard označuje jako **3GPP TS 23.040**, nicméně často se stále používá toto starší označení.

2.2.4 SIM

SIM, zkratka pro ***S**ubscriber **I**dentity **M**odule* (česky volně *Účastnický identifikační modul/karta*), je speciální čip, známý jako *SIM karta*, která umožňuje autentifikaci uživatele

do sítě GSM. Jako standard jej popisuje **3GPP TS 51.011** (dříve **GSM 11.11**). Pomocí této karty lze snadno měnit zařízení za zachování svého telefonního čísla. Některé telefony jsou prodávány jako tzv. *„SIM locked“*, znamenající že je možné v nich využít pouze SIM od jednoho operátora nebo skupiny operátorů. Jejich odblokování je často operátorem neposkytováno, či zpoplatněno.

V současné době se začíná zavádět nová technologie, tzv. **eSIM**. Jedná se o vestavěný čip přímo do zařízení, který na rozdíl od klasické SIM karty nelze vyjmout, zato jej ale lze přepisovat, proces přenosu čísla na jiné zařízení by tedy znamenal smazání identity na zařízení *A* a zapsání této identity na zařízení *B*.

V červnu roku 2013 přišel bývalý zaměstnanec CIA *Edward Snowden* s tvrzením, že NSA byla schopna zajistit šifrovací klíče karet a mohou tak odposlouchávat hovory a číst SMS zprávy bez povšimnutí uživatele či spolupráce mobilních operátorů. Společnost se nicméně brání, že k útoku sice došlo, ale klíče ukradeny nebyly.[1]

2.2.5 Zabezpečení

Původní specifikace GSM počítá pouze s ověřením uživatele u buňky, nikoliv naopak. Zároveň komunikace mezi těmito subjekty může (ale nemusí) být šifrována, nejčastěji pomocí šifrovacích algoritmů **A5/1** nebo **A5/2**, oba tyto algoritmy jsou již ale dnes brány za prolomené a tedy nebezpečné.[2]

Vyjma možnosti prolomení šifrování při použití výše uvedených algoritmů, pokud jsou tedy použity, pakliže specifikace jejich použití nijak nevynucuje, se jedná i o možnost použití falešné GSM buňky. Technicky sice náročná možnost útoku, nicméně v případě dostatečné blízkosti k oběti může útočník využít MITM[3] útoku a odposlouchávat komunikaci. Vyřešení tohoto problému přináší rozšíření UMTS, specificky USIM, která vyjma jiného vynucuje autentifikaci nejenom směrem uživatel -> buňka, ale i buňka -> uživatel.

Nevynucené zabezpečení a bezpečnostní úniky jsou důležitou motivací pro vypracování tohoto projektu. Všímáme si, že přenos je jednoduché zachytit a nepozorovaně odposlechnout. Tomuto zabrání šifrování zpráv.

2.3 Kryptografie

Kryptografie je *nauka o metodách utajování smyslu zpráv převodem do podoby, která je čitelná jen se speciální znalostí.* Slovo pochází ž řeckého **kryptós** (skrytý) a **gráphein** (psát). Spolu s kryptoanalýzou, vědou luštění zašifrovaných zpráv, se kryptografie pojmenovává jako **kryptologie**.

Utajování zpráv probíhá pomocí **šifrování**, což je algoritmus, který převádí čitelnou zprávu, označenou jako **otevřený text** na její nečitelnou podobu (neboli **šifrovaný text**) za pomoci **klíče**. Dále se algoritmy dělí do dvou tříd podle způsobu dešifrování textu:

- **Symetrické šifrování** – k šifrování a dešifrování je využito stejného klíče.
- **Asymetrické šifrování** – k šifrování je využito jiného klíče než k dešifrování.

[1] http://www.independent.co.uk/life-style/gadgets-and-tech/news/sim-card-maker-gemalto-says-it-was-hacked-by-gchq-and-nsa-but-encryption-keys-are-safe-10068767.html

[2] https://eprint.iacr.org/2008/147.pdf

[3] MITM je zkratka pro typ útoku „Man In The Middle“. Zpočívá ve způsobu zachycení komunikace uživatele se serverem a přesměrování obou směrů přes zařízení útočníka. Oběť se o útoku nemusí nijak dozvědět.

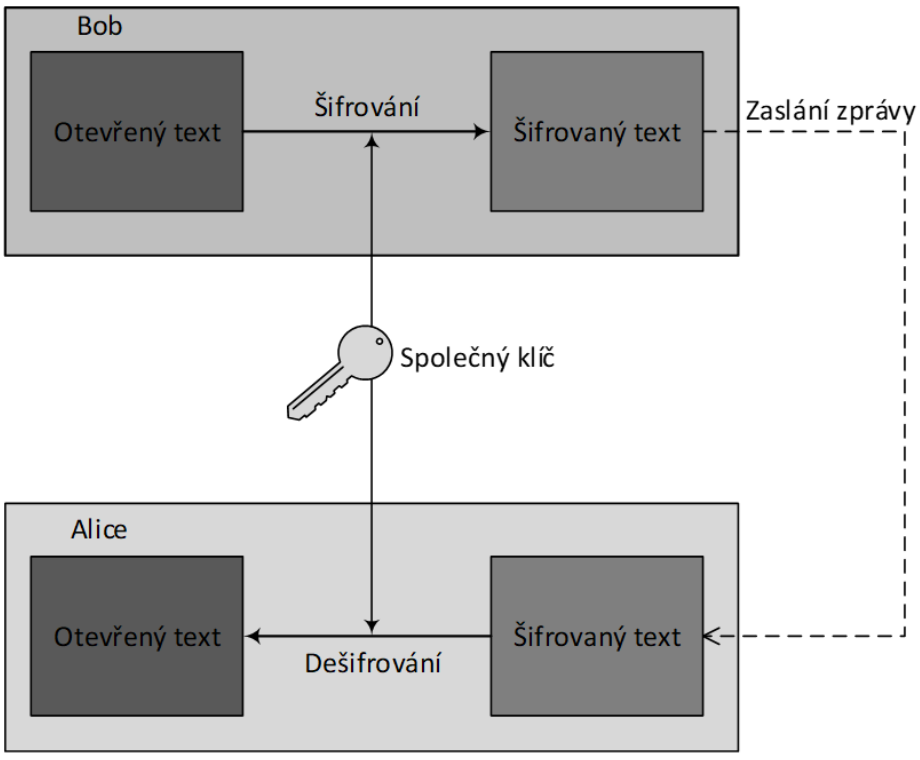
Taktéž lze kryptografické algoritmy rozdělit dle historického rozložení na:

- **Klasická kryptografie** – Od prvního použití kryptografie do přibližně poloviny 20. století.
- **Moderní kryptografie** – Od poloviny 20. století do současnosti.

Dále se také využívá tzv. **hashovací funkce**, která z celého textu vytvoří krátký řetězec s velkou pravděpodobností tento text identifikující. I drobná změna v původním textu drasticky změní výstup této funkce, může tedy sloužit k ověřování a podpisu (více níže). Tato metoda ale nemůže být brána se 100% jistotou, slouží pouze k další kontrole správnosti / autentičnosti přijaté zprávy.

2.3.1 Symetrické šifrování

Algoritmy na základě symetrické šifry využívají k šifrování i dešifrování **stejný klíč**. Pokud tedy chtějí dva modeloví účastníci komunikace, *Alice* (dále pouze *A*) a *Bob* (dále pouze *B*), zahájit šifrovanou komunikaci, musí si nejprve přes zabezpečený kanál či osobně předat/domluvit **společný klíč**, který bude známý pro obě strany. Následně *B* pomocí tohoto klíče zašifruje zprávu, kterou pošle *A*. Ta ji pomocí stejného klíče rozšifruje a může si přečíst původní text. Pokud bude chtít *A* odpovědět, zašifruje opět pomocí stejného klíče svoji odpověď, odešle ji *B* a ten za stále stejného klíče provede dešifrování a přečte si danou odpověď. Diagram této komunikace je vyzobrazen na obrázku 2.1.



Obrázek 2.1: Diagram komunikace s využitím symetrické šifry

Tyto algoritmy se dělí do dvou tříd, dle způsobu zpracování otevřeného textu:

- **Proudové** – Po jednotlivých bitech.

- **Blokové** – Po blocích pevné velikosti, s vhodným doplněním posledního bloku.

Kromě jednodušší implementace je hlavní výhodou symetrického šifrování především jeho rychlost.

Naopak hlavní nevýhodou je nutnost sdílení klíče. Vyjma problému s bezpečnou výměnou tohoto klíče pak nastává možnost snazšího odhalení klíče v případě slabší ochrany proti útočníkům na zařízení jednoho z účastníků komunikace (A či B), je možno komunikaci poslouchat obousměrně. Částečně se tomuto dá zabránit použitím dvou samostatných klíčů pro oba kanály (směr A -> B bude (de)šifrován pomocí jiného klíče, než B -> A), nicméně v případě možnosti zjištění jednoho klíče je vysoká pravděpodobnost zjištění i druhého.

Mezi algoritmy symetrické šifry se řadí známé **RC4**, ale také další jako **DES** či **AES**.

2.3.2 Asymetrické šifrování

Algoritmy na základě asymetrické šifry využívají k šifrování **odlišný klíč** než k dešifrování. Klíč použitý k šifrování zprávy se nazývá **veřejný klíč**, naopak klíč k dešifrování **soukromý klíč**. Pokud tedy chtějí dva modeloví účastníci komunikace zahájit šifrovanou komunikaci směrem B -> A, musí nejprve B získat od A její veřejný klíč. Tento klíč může být předán přes nezabezpečený kanál. Následně B pomocí tohoto klíče zašifruje zprávu, kterou pošle A. Ta ji pomocí svého soukromého klíče dešifruje a přečte si původní zprávu. Pokud naopak má probíhat komunikace směrem A -> B, musí A získat od B jeho veřejný klíč a dále již postupuje stejně. Diagram této komunikace je vyzobrazen na obrázku 2.2.

Algoritmus se tedy liší zásadně od symetrické šifry tím, že každá strana má dvojici veřejných a soukromých klíčů, kdy soukromý klíč je pro druhou stranu neznámý. V případě získání soukromého klíče B je možno odposlouchávat pouze směr A -> B, směr B -> A zůstává zabezpečen.
Hlavní výhodou algoritmu je absence nutnosti sdílet jeden stejný klíč pro obě osoby. Klíč pro zašifrování zprávy lze též přenášet po nezabezpečené lince, což u symetrické šifry nelze (pokud třetí strana zachytí klíče během předávání, zprávu může číst také). S tím souvisí i případná snazší výměna dvojice šifrovacích klíčů v případě podezření/odhalení odposlechu.

Nevýhodou je především vyšší režie oproti symetrickým šifrám, z důvodu složitějších počtů. Tato nevýhoda se často „obchází“ použitím asymetrické šifry pro výměnu klíče, který bude použit ke komunikaci se symetrickým šifrováním.

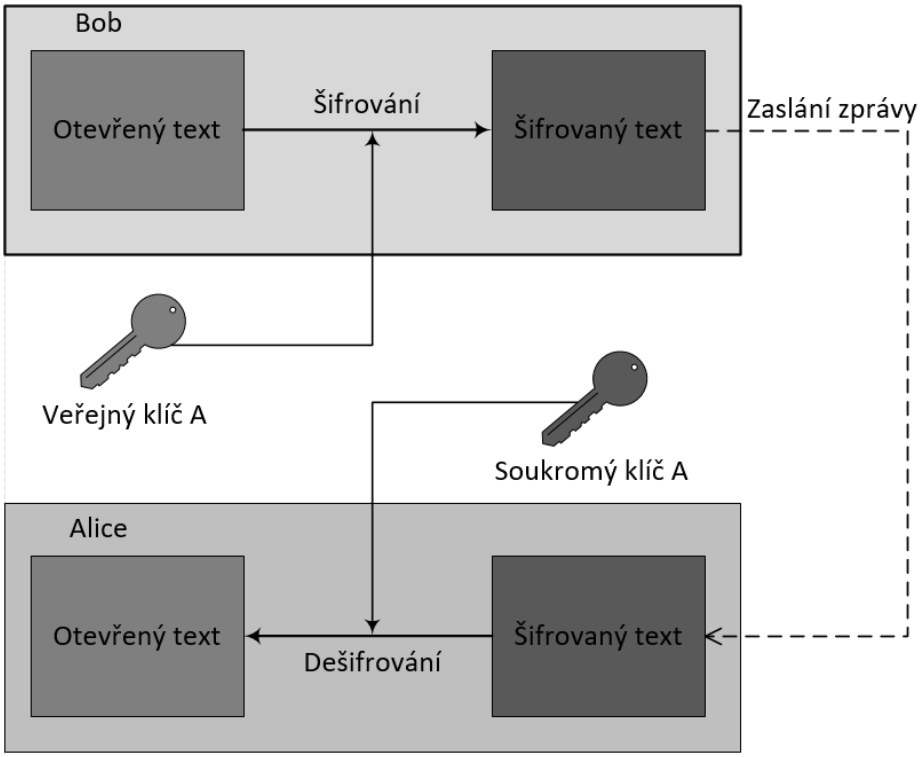
Nejznámějším a nejpoužívanějším algoritmem asymetrické šifry je **RSA**.

2.3.3 Steganografie

Mezi způsoby utajování zpráv patří také **steganografie**, což je *věda zabývající se ukrýváním zprávy jako takové*, například pomocí neviditelného inkoustu či v moderní době například do souborů s hudbou či obrázky. Jedná se též o snahu skrýt potenciálnímu odposlechu, že šifrovaná komunikace vůbec probíhá. Pro odposlouchávající stranu se může jednat o zdánlivě nevinné poslání fotky z dovolené, ale v tomto obrázku je ukryta tajná zpráva.

Například pokud k zobrazení obrázku využíváme pro každý pixel 24bitové barevné rozložení RGB, tedy pro každou ze tří základních barev 8 bitů, můžeme u každé složky odebrat 2 nejméně důležité bity a tyto využít k vložení naší zprávy či obrázku. U zdrojového obrázku sice dojde ke snížení kvality a počtu barev, nicméně pro lidské oko nepoznatelné. (Z 16,67 milionu barev na 262 144).

Využití steganografie pro účely našeho programu je plánováno jako budoucí rozšíření.



Obrázek 2.2: Diagram komunikace s využitím asymetrické šifry

2.4 Existující řešení

Při vypracování nové aplikace není určitě na škodu se podívat na již existující produkty a porovnat cílenou funkcionalitu s funkcionalitou těchto produktů. Všechny aplikace byly nalezeny na **Google Play**[4], při hledání pod tagy „sms“ a „encrypted“, uvedená hodnocení jsou hodnoty z Google Play. Následuje analýza vybraných aplikací a výběr možných prvků pro naší aplikaci.

2.4.1 Výchozí SMS aplikace

Výchozí SMS aplikace v Android 5.1 se skládá z rozhraní „tří oken“. V prvním okně „seznam konverzací“ aplikace standartně začíná. Při poklepnutí na libovolnou konverzaci se tato otevře v novém okně, kde se zobrazí historie konverzace s daným kontaktem, ve formě okna „detail konverzace“. Poslední možné „okno“ je „nová konverzace“, která umožňuje zasílání nové zprávy specifickému kontaktu a založení nové konverzace. Po odeslání zprávy je uživatel přesměrován právě do okna „detail konverzace“.

Při podržení ukazatele/prstu na určitém prvku (ať již konverzace či zpráva přímo v konverzaci) dojde k zobrazení nabídky dalších akcí. Jedná se například o smazání zprávy / konverzace či kopírování textu zprávy.

Aplikace též upozorní na chyby při odesílání, ať již nevyžadující zásah uživatele (například opožděné odeslání z důvodu ztráty signálu) či naopak takové, u kterých je potřeba SMS znovu odeslat (například došlo ke kritické chybě při odesílání).
Snímek aplikace je vyzobrazen na obrázku 2.3a.

[4] http://play.google.com/

2.4.2 Encrypted SMS

Hodnocení: 4,7 / 5
Druhá nejlepší aplikace z výběru. Pro každou konverzaci nabízí možnost výměny klíčů pro asymetrickou šifru, symetrickou nepodporuje. Po výměně si pamatuje, že k ní došlo a automaticky šifruje odchozí SMS. Aplikace nefunguje jako náhrada defaultní SMS aplikace, práce s ní je tedy problematická, viz nevýhody níže.

Mezi výhody jistě patří příjemné uživatelské rozhraní a přítomnost nápověd, například upozorní, že odchozí SMS nebude šifrovaná před výměnou klíčů. Aplikace také umožňuje nastavení „Master Password“, tedy hesla které je nutné zadat při každém spuštění aplikace, jinak není uživatel do aplikace puštěn. Další vrstvou bezpečnosti je i blokování vytváření screenshotu plochy a tedy dalším znesnadněním odcizení dat.

Nevýhodou je plýtvání místa v podobě pevně dané minimální velikosti textu na 6 řádků, krátké texty tedy zbytečně zabírají místo (na testovacím telefonu se pod sebe vešlo maximálně 5 zpráv). Taktéž je nepříjemné vynucování šifrování, při každém pokusu o nezašifrovanou SMS se objeví upozornění, není tedy možné mít pohodlně nezašifrované pouze některé konverzace.
Snímek aplikace je vyzobrazen na obrázku 2.3b.

2.4.3 SMS Encrypt

Hodnocení: 3,9 / 5
Při testování na referenčním telefonu (Lenovo A600, CyanogenMod OS) se aplikace sice spustí, ale nelze přijímat, odesílat ani číst zprávy. Dle screenshotu uživatelského prostředí se jednalo nejspíše o aplikaci pro Android 4.4, proto mohly nastat problémy z důvodu kompatibility.
Snímek aplikace je vyzobrazen na obrázku 2.3c.

2.4.4 Silence

Hodnocení: 4,6 / 5
Subjektivně nejlepší aplikace z výběru. Ihned při startu nabídne nastavení sebe samé jako výchozí aplikaci pro SMS. Následně nabídne přesun všech SMS do chráněné a zašifrované části uložiště aplikace, což zvyšuje bezpečnost, ale i omezí práci s SMS ostatními aplikacemi. Umožňuje stejně jako *Encrypted SMS* pouze asymetrickou šifru, nevyžaduje ale pro každou zprávu zašifrovaný stav.

Výhodou je moderně vypadající uživatelské rozhraní, jednoduchost ovládání a rychlost běhu. Též jako uživatelsky přívětivou vlastnost lze označit možnost změny barvy jednotlivých konverzací, je tedy možné pro různé kontakty mít různé barvy konverzací. Pro jednotlivé zprávy v konverzacích je i elegantně zobrazen stav přijetí zprávy, podobně jako je tomu například v aplikaci WhatsApp[5].

Nevýhodou je opět pouze asymetrická šifra, která taktéž potřebuje vlastnění aplikace na druhé straně. Není tedy možné poslat někomu zašifrovanou zprávu pomocí symetrické šifry pokud například nevlastní OS Android, či obecně chytrý telefon.
Snímek aplikace je vyzobrazen na obrázku 2.3d.

[5] https://play.google.com/store/apps/details?id=com.whatsapp&hl=cs

2.4.5 Text Encryption

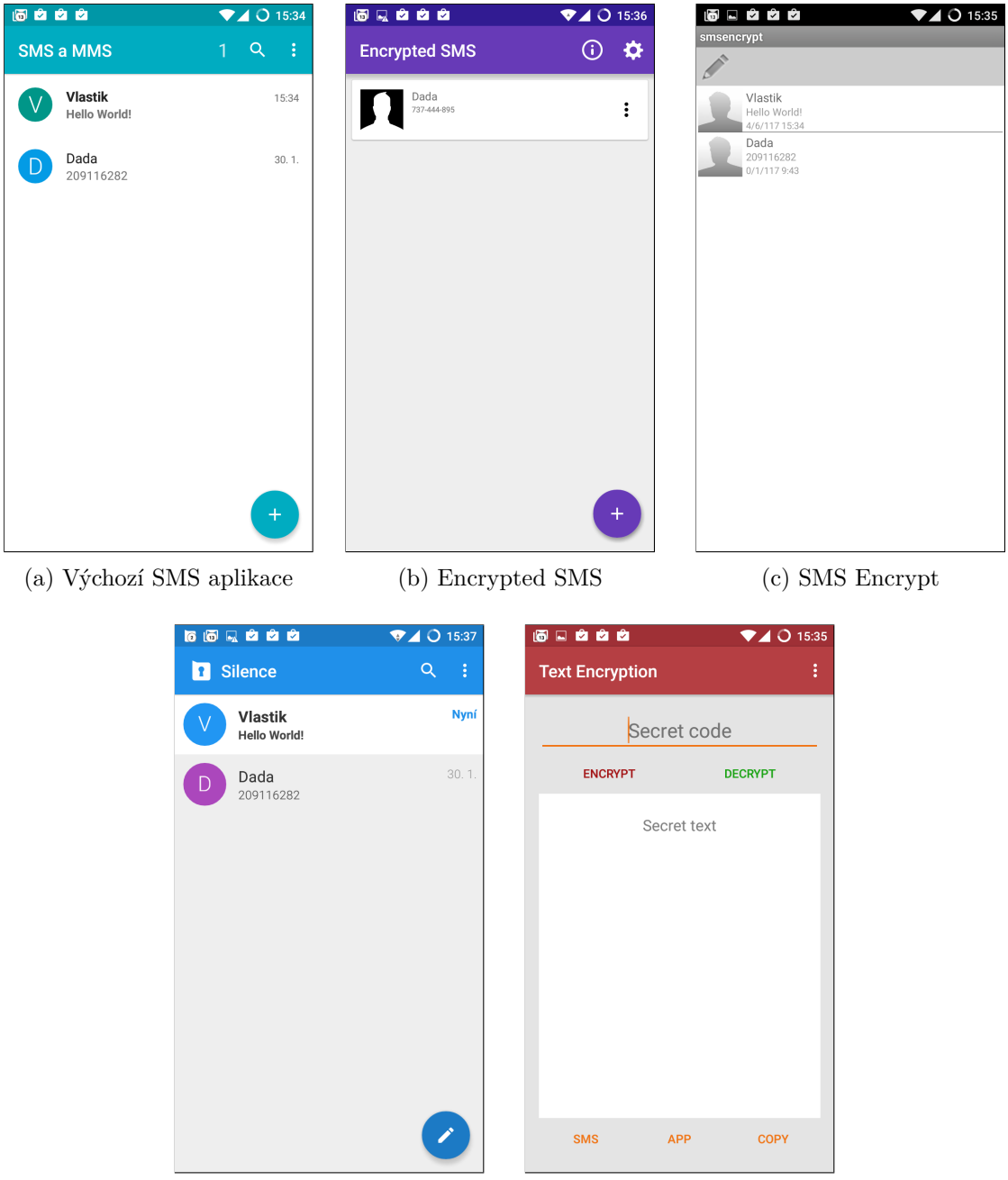
Hodnocení: 3,7 / 5
Aplikace, která sice neumožňuje přímo odesílat šifrované SMS, pouze umožňuje šifrování vybraného textu a následné sdílení, kde již lze vybrat SMS aplikace pro odeslání. Jedná se o jednoduchou, „jednooknovou“ aplikaci, která obsahuje pouze místo pro text, místo pro heslo a dvě tlačítka „encrypt“ a „decrypt“. Tato dvě tlačítka zašifrují, resp. dešifrují, zadaní text pomocí hesla, v případě zadání nesprávného hesla nedešifrují.

Výhoda aplikace zpočívá v její jednoduchosti, nejsou zde zbytečné rušivé prvky a aplikace vykonává činnost, ke které byla stvořena.

Nevýhodou je neoznámení šifrovacího algoritmu, opět je tedy nutné vlastnit tuto aplikaci oběma stranami pro šifrovanou komunikaci. Při testování na některých webových utilitách pro šifrování bylo při každém pokusu o dešifrování textu za použití stejného hesla oznámena chyba.
Snímek aplikace je vyzobrazen na obrázku 2.3e.

2.4.6 Využití pro aplikaci

Ze jmenovaných aplikací bude jistě hlavní inspirace a vzor v defaultní SMS aplikaci. Taktéž inspirací je uživatelská jednoduchost aplikace *Silence*, v neposlední řadě možnost samostatného šifrování textu bez odeslání SMS z aplikace *Text Encryption*.



(a) Výchozí SMS aplikace (b) Encrypted SMS (c) SMS Encrypt

(d) Silence (e) Text Encryption

Obrázek 2.3: Snímky hlavních oken jednotlivých testovaných aplikací

Kapitola 3

Programování na Androidu

3.1 Android OS

Operační systém **Android** je v aktivním vývoji již od roku 2003, důležitá událost ale nastala v roce 2005, kdy společnost **Google** odkoupila společnost **Android Inc.** a zakomponovala ji jako svoji dceřinou společnost [14]. V následujících letech si Google vydobyl vedoucí pozici na trhu, v současné době je 80 procent nových telefonů vybaveno OS Android [13].

Android je open-source OS, postavený na linuxovém jádru s drobnými úpravami pro potřeby mobilního nasazení (v prvních verzích Linux 2.3, v současné verzi Android 7.1 na Linux 4.4.1). Otevřenost Androidu umožnila vznik neoficiálních **ROM** pro telefony, například **CyanogenMod** (nyní již bez aktivního vývoje, pokračovatelem je **Lineage OS**) či **MIUI**.

Jednotlivé verze jsou značeny čísly, příslušnou verzí **API**[1] (například verze 19 pro Android 4.4) a také sladkostí v anglickém jazyce podle abecedy. Například Android 4.4 byl písmeno K – Kitkat, Android 5.0 písmeno L – Lollipop.

Z možností Linuxového jádra Android využívá například nastavování různých práv různým uživatelům. Každá aplikace spouštěna s unikátním UID (User ID), umožňující spouštět aplikace v tzv. „sandboxu“, kde bez implicitního nastavení a požadavku nemá aplikace přístup k prostředkům aplikace jiné.

Jedním z problémů, které musí vývojář často řešit, je rozdrobenost jednotlivých verzí Androidu na trhu. O vývoj nových aktualizací pro jádro Androidu se stará společnost Google, nicméně distribuci musí povolit a zahájit výrobce telefonu. Z důvodu nutnosti složitého testování kompatibility a s tím spojených nákladů je pro výrobce snadnější a ekonomičtější telefon vydat a pak na něj „zapomenout“. Tento fakt ukazuje následující tabulka. [3]

Android	2.3	4.0	4.1-4.3	4.4	5.0-5.1	6.0	7.0	7.1
API	10	15	16-18	19	21-22	23	24	25
Procento	1,0	0,8	9,1	18,8	32	31,2	6,6	0,5

Tabulka 3.1: Procentuální rozložení jednotlivých verzí OS Android dle verze a API, ke dni 17. května 2017

Z tabulky je patrné, že téměř 30% zařízení stále používá verzi 4.4 a nižší, pro kterou již dnes Google neposkytuje podporu. Tento fakt snižuje celkovou bezpečnost systému, jelikož

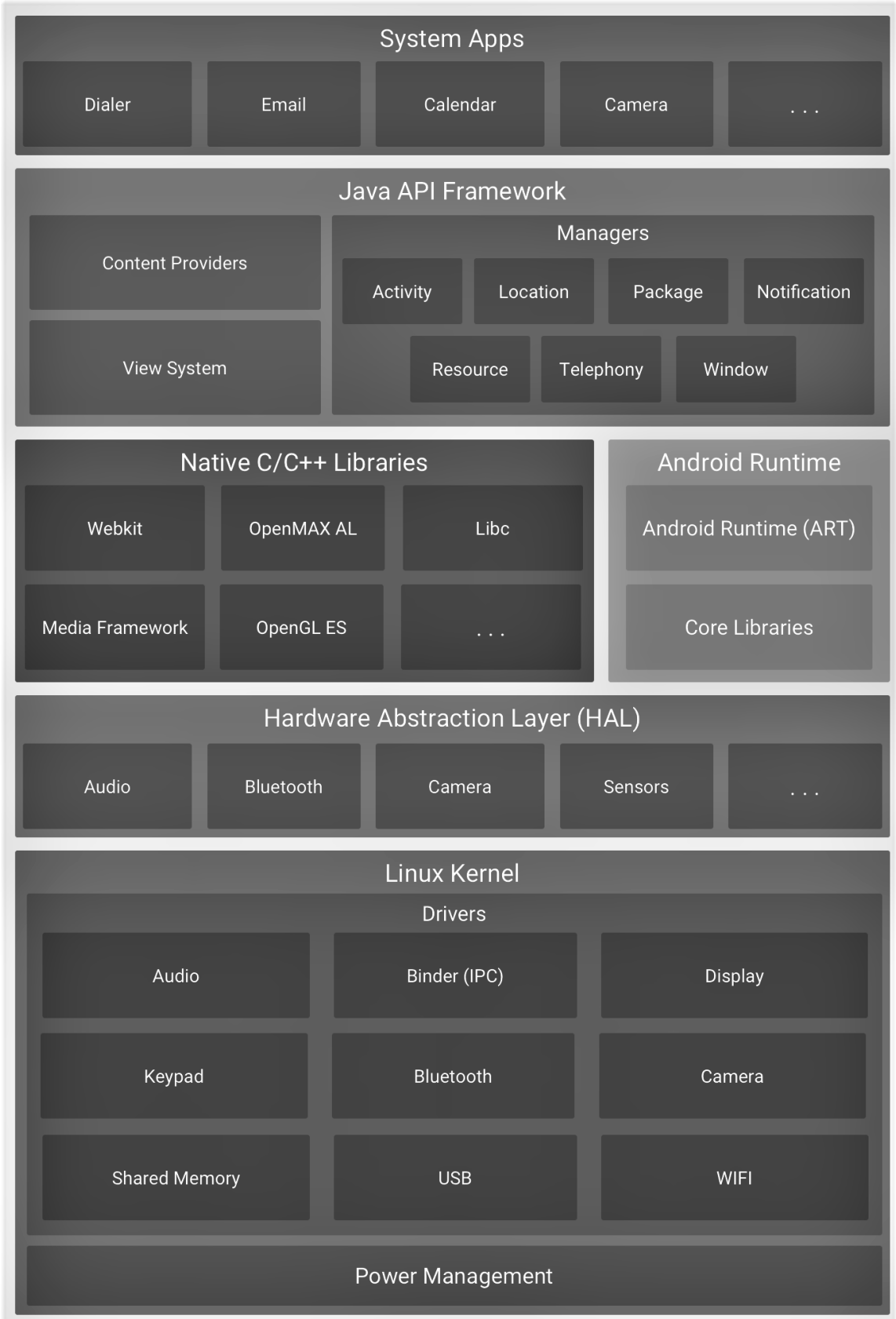
[1] API je zkratka pro ***A**pplication **P**rogramming **I**nterface*

se do těchto telefonů nedostanou nejnovější bezpečnostní aktualizace a zařízení jsou tedy nechráněná vůči různým druhům útoků.

3.2 Architektura Androidu

V popisu architektury se soustředíme spíše na model Android 5.0 (API 21 a vyšší), který výrazně urychlil běh aplikací oproti staršímu modelu. Tento model je rozložen do následujících částí a vyzobrazen na obrázku 3.1:

- **Linuxové jádro** – Upravené pro potřeby mobilních zařízení (vyšší výdrž, menší nároky na paměť RAM). Umožňuje například pouštět každou aplikaci v tzv. „Sandbox mode“, který aplikaci zamezí přístup k prostředkům jiné aplikace.
- **HAL** – Hardware Abstraction Layer, rozhraní pro abstrakci Hardware (dále jen HW), například pokud je přes API zavolána specifická HW komponenta, např. kamera, Android načte příslušnou knihovnu k obsluze této komponenty. Zároveň umožňuje výrobcům HW pomocí standardizovaného způsobu umožnit komunikaci Aplikace <-> HW.
- **ART** – Android Runtime, slouží pro dopředný (AOT – Ahead-Of-Time) překlad aplikace při instalaci na zařízení z DEX bytecode formátu do nativního jazyka procesoru. Tímto se model liší od předchozích verzí, které využívali překlad „za běhu“ (JIT – Just-In-Time) a využívaly interpret bytecode nazvaný Dalvik.
- **Nativní C/C++ knihovny** – nejsou dostupné přímo, ale pomocí Java API Framework (viz níže). Jsou zde obsaženy některé knihovny, například pro práci s grafikou (OpenGL, Vulkan), umožňují nízko úrovňový přístup k některým částem telefonu, často HW.
- **Java API Framework** – Souhrn všech API, která jsou dostupná pro aplikací běžící v OS Android. Umožňují například práci s User Interface, Notifikacemi a dalšími. Ulehčují práci vývojáře, abstrahují HW a standardizují komunikaci aplikací. Systémové aplikace (viz níže) využívají stejné API jako každá jiná aplikace.
- **Systémové aplikace** – Předinstalované aplikace, často bez možnosti je odinstalovat. Zajišťují základní funkcionalitu OS a HW. Patří mezi ně například SMS komunikátor, fotoaparát, webový prohlížeč a další. Od Android verze 4.4 lze většina systémových aplikací nahradit vlastní zvolenou aplikací, například již zmíněný SMS komunikátor naší aplikací a zastoupit tedy její funkcionalitu.



Obrázek 3.1: Model architektury OS Android [8]

3.3 Použité komponenty

V této sekci jsou vysvětleny jednotlivé části Android Framework, které jsou v aplikaci využity a tvoří tedy její aplikační jádro.

3.3.1 Soubor Manifest

Soubor Manifest je důležitým prvkem každé aplikace pro Android. Definuje potřebná práva aplikace pro práci, jednotlivá „okna“ v aplikaci a třídy jím odpovídající a taktéž služby, které běží na pozadí systému i mimo běh aplikace.

3.3.2 Aktivity a Fragmenty

Aktivity jsou hlavní možností interakce uživatele s aplikací. Každá aplikace má jednu hlavní aktivitu, která se spouští při startu aplikace (vyjma vyjímek, například reakce na notifikaci), a dále neomezeně mnoho dalších aktivit, které vykonávají další činnosti. Každá aktivita prochází svým vlastním „životním cyklem“, zobrazeným na obrázku v příloze B.1. Je vhodné metody aktivity přetížit a dosáhnout tím své vlastní funkcionality (například předávání parametrů jiné aktivitě).

Součástí aktivity také mohou být části zvané **fragmenty**. Těchto částí se využívá například ke sdílení některých částí uživatelského rozhraní, snižují tedy redundanci kódu. Každý fragment má vlastní životní cyklus, je ale spjatý s cyklem rodičovské aktivity [4].

3.3.3 Broadcast Receiver

Broadcast Receiver je třída sloužící k zachycení celosystémových hlášení „Broadcast“. Tato hlášení jsou různých typů a k zachycení mnohých je potřeba speciálních pravomocí. K aplikaci využívaného hlášení **SMS_DELIVER** je potřeba deklarovat v souboru Manifest pravomoc **BROADCAST_SMS**.

Práce s takovým hlášením probíhá pomocí třídy **Intent**, která v sobě ukrývá i detaily, v tomto případě například jednotlivé příchozí SMS.

3.3.4 Content Resolver / Content Provider

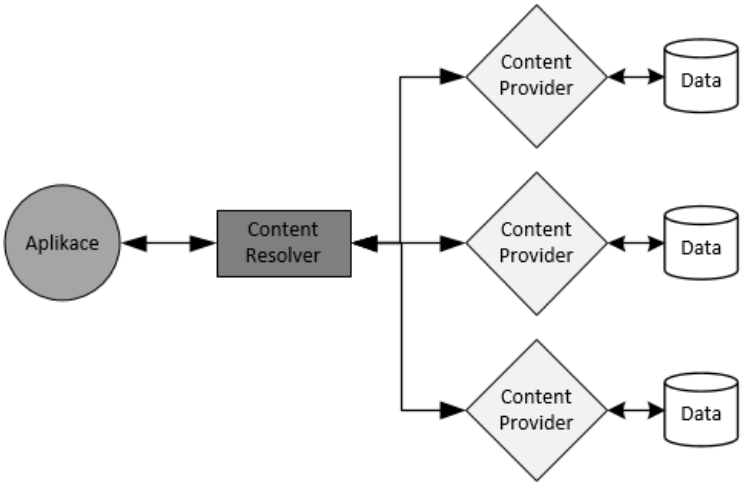
V každé aplikaci je dostupná jedna globální instance třídy **Content Resolver**. Tato třída se stará přijímání požadavků od uživatele a přesměrování je do patřičné instance třídy **Content Provider**. Slouží tedy na propojení jednotlivých aplikací a sdílení jejich dat ve standartizované podobě.

Content Resolver tedy abstrahuje komunikaci mezi databázemi, Content Provider abstrahuje práci s daty v těchto databázích, viz obrázek 3.2.

3.3.5 AsyncTask

Třída **AsyncTask** slouží pro zjednodušení výpočtu jednoduchých operací na pozadí hlavního vlákna pro uživatelské rozhraní, bez přímého použití vláken. Po skončení výpočtu vrátí instance třídy hodnotu zpět hlavnímu vláknu pro změnu rozhraní. Využívá se například pro načtení dat, které může trvat i několik vteřin, což by mohlo zamrznout hlavní vlákno a aplikace by se jevila jako neresponzivní.

Výhodná je především pro krátké operace, pro delší operace (například v řádu minut) je lepší využít jiných tříd, mj. *Executor*, *ThreadPoolExecutor* či *FutureTask* [2].



Obrázek 3.2: Diagram komunikace Content Provider a Content Resolver [15]

3.3.6 Android API

Android API je kritická část programování pro OS Android. Umožní standartizovanou práci s prostředky OS a kompatibilitu napříč zařízeními. Hlavní využívanou částí naší aplikace je API části *android.telephony* pro práci s SMS a také *android.support*, obsahující většinu tříd a metod pro práci s uživatelským rozhraním.

Pro práci s SMS jsou využívány především třídy **SmsManager** a **SmsMessage**, umožňující práci s příchozími a odchozími SMS.

3.4 Použité knihovny

3.4.1 SmsLib[2]

Tato knihovna je použita pro pokročilejší práci s PDU hlavičkou SMS zprávy. V této hlavičce se může ukrývat spousta užitečných hodnot, například informace o řetězení, které funkce API nedokážou jednoduše detekovat, hlavička je vývojáři dostupná pouze v binární podobě, je potřeba z ní hodnoty vyextrahovat ručně. Tato knihovna pomáhá tento krok zjednodušit pomocí parsování.

Knihovna je implementována v jazyce Java a dodávána jako zdrojový kód.

3.4.2 JCharset[3]

Pomocná knihovna pro konverzi pole bytů na znaky sady GSM 03.38, které Java knihovny defaultně neumožňují. Slouží především pro zjednodušení práce. Pomocí této knihovny především kontrolujeme aktuální znakovou sadu v nové SMS zprávě.

Knihovna je implementována v jazyce Java a dodávána jako .jar knihovna.

[2] https://github.com/tdelenikas/smslib-v3
[3] https://www.freeutils.net/source/jcharset/

3.4.3 Facebook Conceal[4]

K samotnému šifrování bylo z důvodu jednoduchosti a pravidla „Špatné šifrování je horší, než žádné šifrování“ plánováno využit knihovny Facebook Conceal. Tato knihovna si dává jako hlavní prioritu jednoduchost používání a rychlost oproti standartním možnostem. Byla vytvořena především pro potřeby aplikace Facebook, pro šifrování dat na externí SD kartě v telefonu, nicméně je možné ji použít i na šifrování obecných dat, tedy i textu. Jedná se o symetrickou šifru AES-GCM, neumožňuje ale implicitní výběr z několika různých šifer.

Bohužel možnosti této knihovny byly přeceněny a strávil jsem zbytečně mnoho člověkohodin[5] snahou o zprovoznění šifrované komunikace. Opakoval se tedy známý případ „Stagefright“, rozebírán například v předmětu ITS[6], kdy knihovna byla využita k funkcionalitě, pro kterou nebyla připravena a chovala se tedy nespecifikovaně. Osobně jsem si tedy ověřil teorii předmětu v praxi.

Knihovna je implementována v jazyce C++ a dodávána jako nativní NDK knihovna.

[4] http://facebook.github.io/conceal/

[5] Člověkohodina je termín používaný při vývoji software pro žízení, plánování a dělbu práce.

[6] Přednáška ze dne 11. 2. 2016, záznam dostupný online z https://video1.fit.vutbr.cz/av/records-categ.php?id=1283, časová značka 0:55:10.

Kapitola 4

Návrh aplikace

4.1 Inspirace

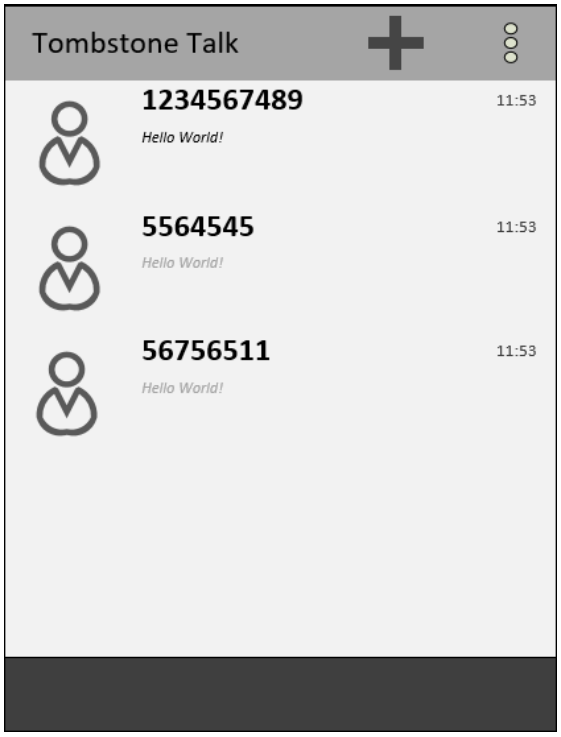
Hlavní inspirací byla výchozí SMS aplikace v Android 5.1 a to především po stránce uživatelského rozhraní. Z této inspirace vychází návrh.

4.2 Návrh

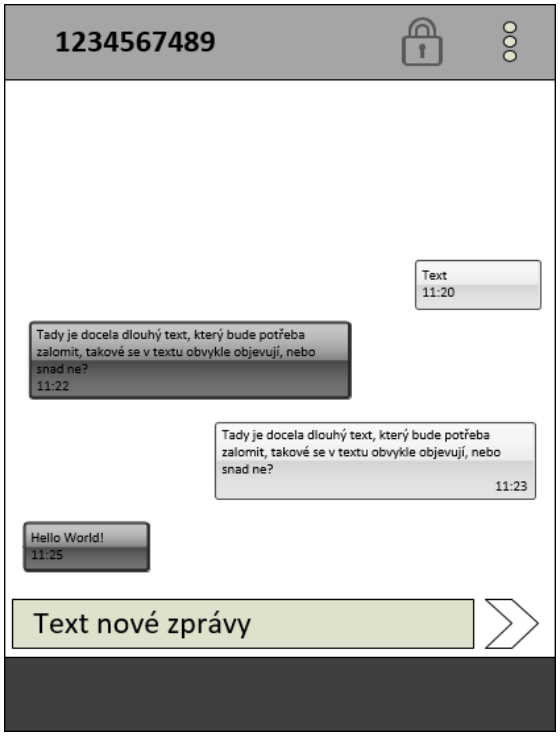
Návrh počítá s rozšířením schopnopností defaultní SMS aplikace za ponechání většiny známého prostředí. Tento přístup usnadní zažití uživatele a jeho orientaci v aplikaci, bude se tedy moci soustředit na naučení práce s novými elementy, jako je například níže popsaný přepínač aktivního šifrování [12] [6].

4.3 Uživatelské prostředí

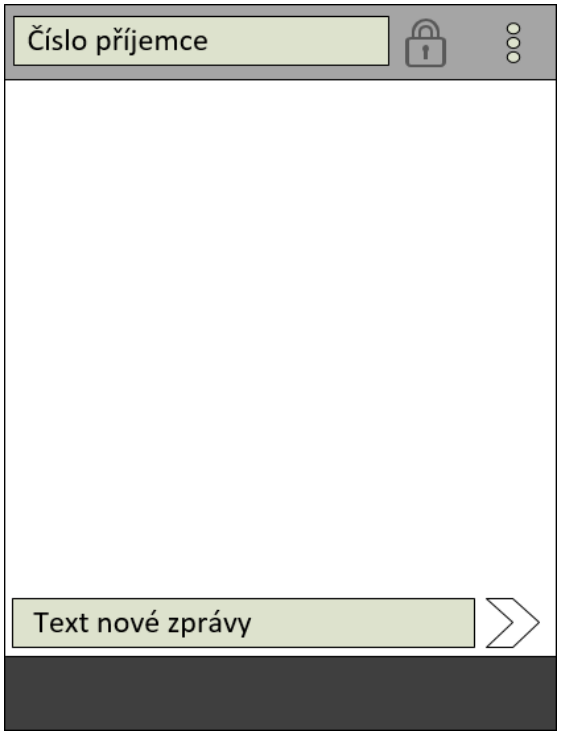
Jak již bylo zmíněno, uživatelské prostředí je ispirováno prostředím defaultní SMS aplikace Androidu 5.1, viz obrázek 2.3a. Jedná se tedy o „čtyřoknové“ prostředí, jednotlivá okna mohou být definována jako „seznam konverzací“, „detail konverzace“, „nová konverzace“ a „šifrovací utilita“. Obrázky 4.1 až 4.4 ukazují původní návrh jednotlivých oken.



Obrázek 4.1: Seznam konverzací



Obrázek 4.2: Detail konverzace



Obrázek 4.3: Nová konverzace

Obrázek 4.4: Šifrovací utilita

4.3.1 Seznam konverzací (Obr. 4.1)

Návrh seznamu konverzací počítá s jednoduchým vyzobrazením všech konverzací, které při klepnutí prstem zavedou uživatele do okna detailu konkrétní konverzace. Okno též bude obsahovat ikonu pro přechod na vytvoření nové konverzace, stejně jako panel s dalšími akcemi (jako je Nastavení apod.).

4.3.2 Detail konverzace (Obr. 4.2)

Toto okno by mělo obsahovat zobrazení všech SMS zpráv, pole pro text nové zprávy, tlačítko pro odeslání zprávy a také tlačítko pro nastavení aktivního šifrování zprávy. Implicitně bude šifrování vypnuto z důvodu zátěže na omezený počet znaků SMS.

4.3.3 Nová konverzace (Obr. 4.3)

Okno nové konverzace je velice podobné detailu konverzace, pouze neobsahuje seznam předchozích zpráv s kontaktem a místo označení okna číslem kontaktu se nachází na jeho místě pole pro zadání tohoto čísla.

4.3.4 Šifrovací utilita (Obr. 4.4)

Jednoduché okno s polem pro zadání hesla, otevřeného textu pro zašifrování a dvěma tlačítky pro zašifrování resp. dešifrování textu.

4.4 Šifrování

Šifrování by mělo probíhat za co nejmenší interakce s uživatelem. Uživatel by měl dodat pouze kriticky potřebné informace (například klíč k zašifrování), dále by již neměl poznat rozdíl mezi šifrovanou a nešifrovanou SMS, co se funkcionality odesílání týče. Rozhodně by měl být ale viditelně v uživatelském rozhraní informován, jaký režim je aktivní, tedy jestli následující SMS bude odeslána šifrovaně či nikoli.

(a) Šifrování neaktivní (b) Šifrování aktivní (c) Nastavení hesla

Obrázek 4.5: Návrhy rozhraní pro přepínání šifrování

Kapitola 5

Implementace

5.1 Implementační prostředí

Aplikace je vyvíjena v jazyce Java, za použití vývojového prostředí **Android Studio** verze 2.3 Aplikace byla vyvíjena a testována za použití emulátoru Android 4.4 (KitKat), 5.0 (Lollipop) a 6.0 (Marshmallow), z důvodu notných změn v těchto prostředích (především u verze 6.0 došlo k zásadní změně v pravomocích aplikace). Testování na „živém“ zařízení probíhalo na telefonu *Lenovo A6000* s OS CyanogenMod, odpovídající Android 5.1.

Minimální verze OS je stanovena na Android 4.4 (API verze 19) z důvodu kompatibility API, neb nahrazení defaultní SMS aplikace je podporováno až od této verze [5].

5.2 Start aplikace

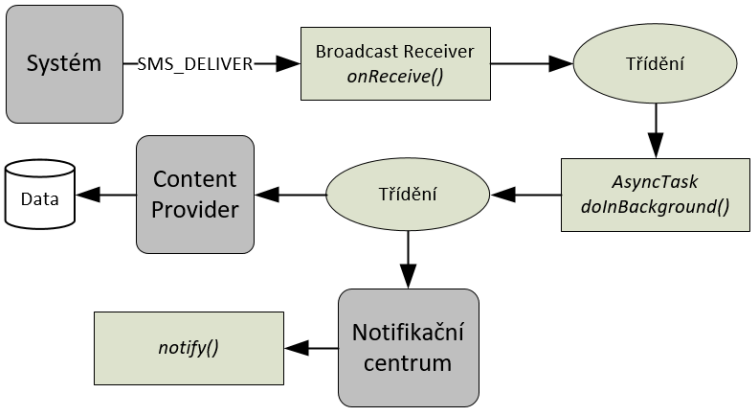
Aplikace zahajuje svoji činnost ve dvou možných stavech. Jednou z nich je otevření aplikace z nabídky všech aplikací, v tomto případě se tedy otevře aplikace v okně „seznam konverzací“. Na pozadí se kontaktuje **Content Provider** pro SMS a MMS zprávy a načtou se z každé konverzace poslední zprávy spolu s informací, zda-li byla poslední zpráva této konverzace již přečtená. V případě že nebyla je zpráva označena modrou barvou pro lepší přehlednost.

Druhou možností startu aplikace je reakce uživatele na notifikaci oznamující novou zprávu. V takovémto případě aplikace zahajuje svoji činnost přímo v okně „detail konverzace“ s konverzací, která obsahuje zprávu z notifikace. Zde se opět na pozadí kontaktuje Content Provider pro SMS a MMS zprávy, nicméně se ale načítají všechny zprávy pouze pro danou konverzaci a zobrazí se na obrazovce.

V obou případech je kontaktován Content Provider pomocí třídy ListFragment, rozšířenou o CursorAdapter, která zajišťuje aktuálnost hodnot, automatické scrollování a „recyklaci“ pohledů, viz výše.

5.3 Příchozí SMS

Aplikace pro příchozí SMS využívá od Broadcast Receiver odvozenou třídu **ReceiveSMS**, která poslouchá Broadcasty typu **SMS_DELIVER** a následně ze zprávy Intent pomocí standartní funkce *getMessagesFromIntent()*, která rozdělí veškeré zprávy do samostatných objektů **SMSMessage**. Tato funkce rozdělí i zřetězené SMS do samostatných, využíváme tedy funkce z knihovny **PduUtils** pro pokročilou práci s hlavičkou SMS, v tomto případě



Obrázek 5.1: Diagram zpracování příchozí SMS

pro získání referenčního čísla, které je pro všechny členské zřetězené SMS stejné, a identifikátoru SMS, které jednoznačně určuje danou SMS v řetezi.

Následně proběhne základní kontrola a řazení příchozích SMS, které putují jako parametr do AsyncTask pro další kontrolu, především konzistence, a následně k uložení do databáze SMS a MMS pomocí Content Resolveru. Ukládání pomocí API zajistí kompabilitu mezi různými zařízeními (Databáze může být na telefonu různých výrobců fyzicky uložena na různých místech). V AsyncTask probíhá také spojování zřetězených SMS do jedné, včetně náhrady nedostupných částí.

Případné dešifrování je inicializováno uživatelem, při podržení prstu nad libovolnou zprávou v okně „detail konverzace“. V tomto případě se uživateli zobrazí dialog žádající o zadání klíče, který bude použit pro dešifrování.

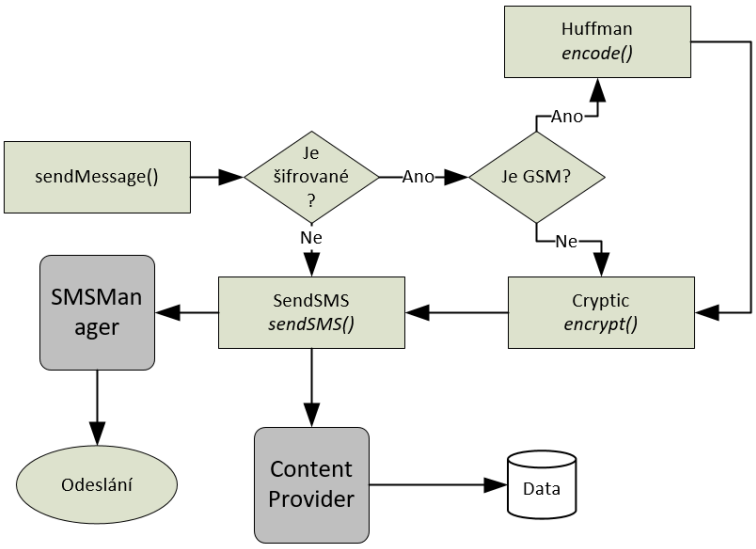
Při potvrzení se zadaný klíč použije k zavolání statické metody *decrypt()* třídy *Cryptic*. Tato metoda inicializuje novou instanci třídy *Cipher* pomocí dodaného šifrovacího klíče. Před šifrováním je ještě text převeden z formátu Base64 na pole bytů.

Následně proběhne samotné dešifrování. Pokud je zpráva v kódování GSM 03.38, je ještě nutné zprávu dekódovat z Huffmanova kódování, voláním metody *decode()* třídy *HuffmanCoding*. Ta zajistí i odstranění přebytečných bytů, vysvětlené v části 5.5: Huffmanovo kódování.

5.4 Odchozí SMS

Pro odchozí SMS je využito standartních funkcí třídy **SmsManager**, opět především pro zajištění maximální kompatibility na všech zařízeních. SmsManager obsahuje funkce *sendTextMessage()* a *sendMultipartTextMessage()*, které zajistí odeslání regulérní, resp. zřetězené SMS, včetně vytvoření správné hlavičky. Z těchto funkcí je v obou případech využita pouze *sendMultipartTextMessage()*, pro kterou dodáváme data rozdělená funkcí *divideMessage()*, která text rozdělí do patřičných částí včetně kontroly kódování. V případě UCS-2 tedy SMS zprávy správně rozdělí po 70, resp. 67 znacích.

Dále je též možno SMS označit speciálním Intent-em pro oznámení o odeslání a přijetí na druhém zařízení, které v naší aplikaci využíváme pro označení zprávy v chatu (například odeslaná ale ještě nepřijatá SMS je označena žlutě).



Obrázek 5.2: Diagram zpracování odchozí SMS

Jelikož tato funkce ale neukládá odeslané SMS do databáze, je potřeba je stejně jako u příchozích SMS manuálně vložit skrz Content Resolver, opět pro zajištění maximální kompatibility.

> *Beginning with Android 4.4 (API level 19), if and only if an app is not selected as the default SMS app, the system automatically writes messages sent using this method to the SMS Provider (the default SMS app is always responsible for writing its sent messages to the SMS Provider).* [9]

Pokud je zapnuté šifrování, tak před samotným voláním odesílacích funkcí a metod je ještě na základě aktuálního kódování provedena komprimace Huffmanovým kódováním, metodou *encode()* třídy *HuffmanCoding*. Toto kódování je použito pouze u znakové sady GSM 03.38, neb u UCS-2 by bylo naprosto nereálné a neefektivní jej použít. Kódování zajistí i zarovnání, vysvětlené v části 5.5: Huffmanovo kódování.

Šifrování, pokud je nastavené, probíhá pomocí statické metody *encrypt()* pomocné třídy *Cryptic*. Tato metoda inicializuje novou instanci třídy *Cipher* pomocí dodaného šifrovacího klíče. Následně proběhne samotné šifrování. Výstup šifrování je převeden do formátu Base64, pro kompatibilitu přenosu.

V případě neúspěchu zobrazí *Toast Notifikaci*[1] s chybovou hláškou.

5.5 Huffmanovo kódování

Huffmanovo kódování je princip, který na základě textu vytvoří tabulku kódování znaků, které nemusí mít stejnou velikost jako znaky původní, dokonce ani stejnou délku v rámci tabulky (např. jeden znak se může zakódovat do 3 bitů, zatímco jiný do 15) Komprimuje data na základě četnosti znaků, kdy častější znaky v textu mají po zakódovaní kratší délku než méně časté znaky.

[1]Toast Notifikace je drobný panel zobrazující se ve spodní části obrazovky. Používá se pro oznámení informací, které nevyžadují žádnou další interakci s uživatelem.

V aplikaci je při spuštění vytvořena stejná tabulka znaků, četnost znaků byla vytvořena na základě analýzy reálných textových zpráv v anglickém jazyce.

Pro potřeby znakové sady GSM 03.38 jsou znaky zarovnány do násobků sedmi. Zarování probíhá na bitové úrovni, kdy za poslední znak výstupu kódování je vložen bit 1 a N bitů 0, kde N udává počet bitů potřebný na správně zarovnání. Tento způsob též řeší stav, kdy poslední znak výstupu je správně zarovnán, v tomto případě je přidán jeden znak sady GSM 03.38 navíc (celkem 7 bitů). Tento způsob zarovnání byl inspirován algoritmem *PKCS#7*, jež je využit při šifrování. Při dešifrování je toto zarovnání odstraněno procházením od konce, kdy je odstraněn každý bit 0 do prvního bitu 1 včetně.

5.6 Encoding Watcher

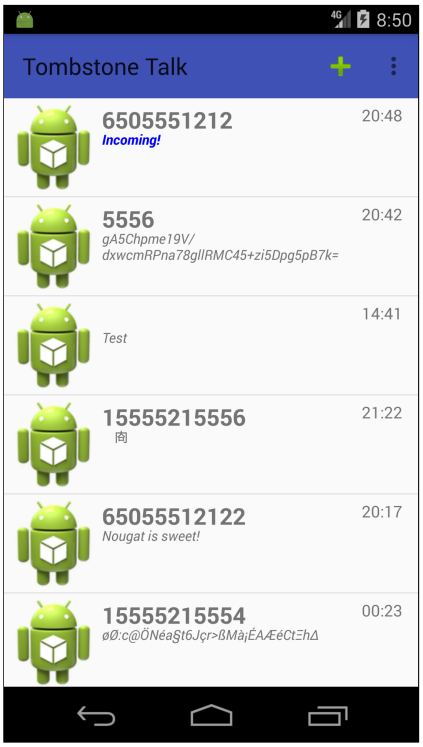
Encoding Watcher je pomocná třída, která zpracovává kontrolu změny kódování při psaní nové zprávy. V případě změny na toto upozorní příslušný prvek. Následně je možno upravit uživatelské rozhraní a případné kódování dle aktuální znakové sady.

5.7 Omezení aplikace

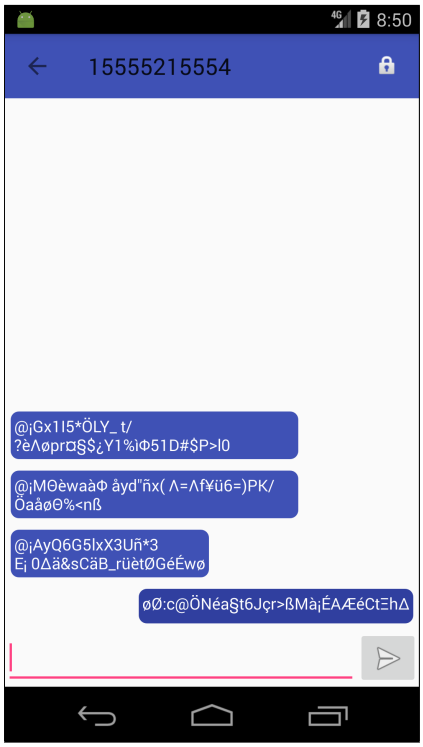
Každá aplikace, která má nahrazovat defaultní SMS aplikaci, musí také ve svém Manifestu deklarovat podporu a třídy pro zpracování MMS, „respond-via-message“ (sloužící pro odeslání odpovědi na hovor pomocí SMS) a „sendto“ (sloužící pro odeslání SMS či MMS na základě požadavku jiné aplikace). Tato deklarace musí být přítomna, i když aplikace de facto tyto funkcionality nepodporuje.

Naše aplikace v základním stavu podporuje pouze odesílání SMS, pro ostatní funkcionality jsou připraveny zatím neaktivní třídy. V současné verzi je též omezeno šifrování pouze na symetrickou šifru.

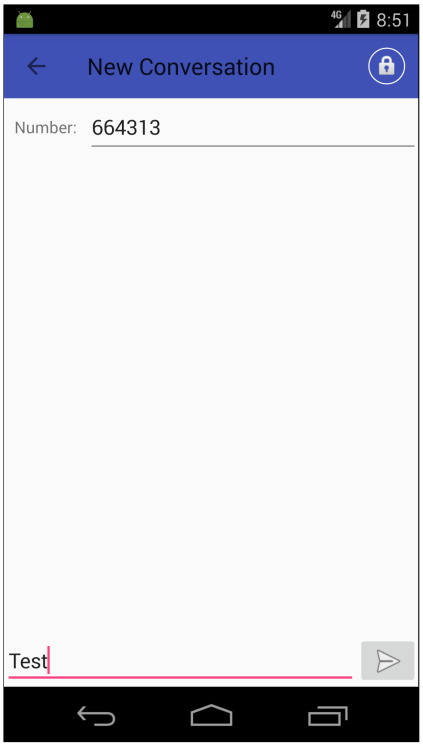
V současné verzi také nefunguje dodatečná kontrola konzistence zřetězených SMS, pokud tedy části SMS přijdou se spožděním, budou zobrazeny jako více SMS s chybějícími částmi.



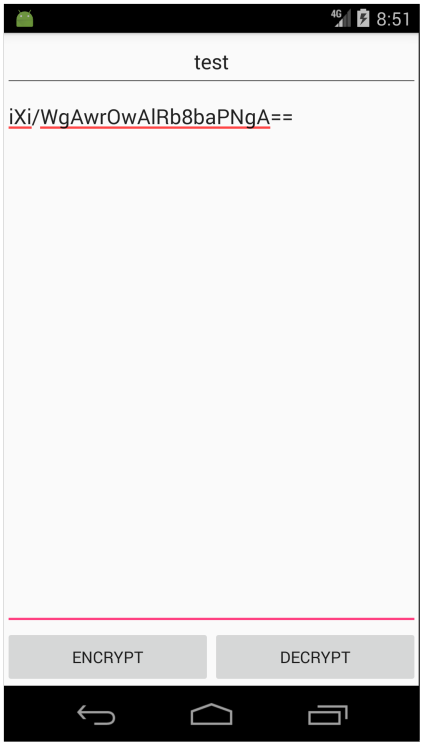
(a) Seznam konverzací



(b) Detail konverzací



(c) Nová konverzace



(d) Šifrovací utilita

Obrázek 5.3: Snímky implementace jednotlivých oken

Kapitola 6

Testování aplikace

Tato kapitola pojednává o procesu testování a získaných informací ze zpětné vazby uživatelů. Samotné testování probíhalo ve dvou fázích.

První fáze probíhala již během vývoje a jednalo se o testování funkcionality s použitím prostředí *Android Studio*, které umožňuje práci s debuggerem. Zde se jednalo především o zkoušení různých uživatelských pohybů v aplikaci. Dále byla aplikace nasazena na „živý“ telefon *Lenovo A6000*, které ukázalo především výkon a optimalizaci pro dané zařízení.

Druhá fáze započla ke konci vývojového cyklu, kdy již aplikace byla možná nainstalovat na jiná zařízení bez problému. Aplikace byla zaslána několika příbuzným s různou technickou znalostí na otestování, zpětnou vazbu a možné návrhy na zlepšení. Před osobním testováním se někteří uživatelé nedozvěděli žádné informace, z testů tedy bylo možné i usoudit, jak aplikace působí pro naprosto cizí osobu, které nebudeme moci sdělit instrukce k používání aplikace. K testování byl taktéž připojen jednoduchý dotazník, který zúčastnění po otestování aplikace vyplnili.

Dotazník obsahoval následující otázky:

- Jak dlouho Vám trvalo se v aplikaci zorientovat?
- Jak hodnotíte uživetelské prostředí?
- Jak snadné bylo zahájit šifrovanou komunikaci?
- Dokážete si představit každodenní používání této aplikace?
- Méte nějaké návrhy ke zlepšení aplikace, co vám v aplikaci chybí?

6.1 Zpětná vazba

Celkem proběhlo testování u 5 lidí, zde jsou zpracovány jejich vyhodnocení.

Dana Balvínová, pracovnice školní družiny
Orietnace byla rychlá. Po vysvětlení je aplikace velice jednoduchá a přístupná i mé věkové kategorii a technickým znalostem.

Uživatelské prostředí je intuitivní, vzhled příjemný. Ikona je nápadná a zároveň nápaditá. Pochopila jsem její význam a spojení se jménem aplikace (Tombstone - Náhrobí kámen) (tři tečky, SMS).

Snadné, pouze jsem musela dávat pozor na automatickou korekci textu telefonu. Při první zprávě automaticky doplnil háček u hesla a poté jsem se rozčilovala, že to nefunguje.

Ano, určitě ne při běžné konverzaci ale například při potřebě sdělit osobní údaje je to vhodné a pro tento případ je to skvělé.

Bylo by dobré takto posílat i obrázky či fotky, měla bych lepší pocit, že moje soukromé fotky jsou opravdu soukromé.

Anonym L, studentka

Na základní orientaci mi stačilo pár minut. Program má podobné rozložení ovládacích prvků jako jiné aplikace, které ke komunikaci běžně využívám, takže zorientování se v prostředí je snadné.

Aplikace je přehledně koncipovaná a snadno ovladatelná. Po grafické stránce jsem také spokojena.

Zahájení komunikace bylo bezproblémové, ovládání je intuitivní a zvládnou ho i méně technicky zdatní jedinci.

Pro svou běžnou komunikaci aplikaci nepotřebuji, nicméně si dovedu představit její využití při posílání „tajných“ zpráv, které nejsou určeny cizím očím. Tuto vymoženost by mohli ocenit především teenageři, protože nabízí něco nového, atraktivního.

Ocenila bych rozšíření v podobě kódování „smajlíků“, které k dnešní komunikaci neodmyslitelně patří, a také možnost posílání MMS zpráv.

Anonym H, studentka

Při prvním náhledu do aplikace bylo z mé strany velice snadné se okamžitě zorientovat. Konverzace jsou seřazené sestupně podle stáří a ikony, znázorňující jednotlivé interakce, mají jednoduchý a výstižný design, takže dle mého názoru by bylo pro uživatele velice snadné procházet konverzace a šifrovat je.

Problémem ovšem je, že bez nápovědy uživatel těžce zjistí, jak zprávu dešifrovat. Aplikaci by tudíž neuškodila ikona nápovědy, či něčeho podobného. Co se celkového designu aplikace týče, konvezace od sebe jsou děleny pouze slabou a světlou čarou a mohly by tedy splývat. Dále velikost textu, jíž jsou psané zprávy je moc malá a uživatel s vadou zraku by mohl mít problémy své zprávy přečíst. Pokud je poslední zpráva delší a okénko konverzace v hlavním menu je na ni příliž malé, spodní půlky slov nejsou vidět a nepůsobí to esteticky tak dobře. Navrhovala bych tedy aby s posledním slovem co se celé vejde do okénka nasledovaly tři tečky znázorňující, že zpráva pokračuje.

Ačkoliv Konverzace sestupně seřazené mohou působit dobrým dojmem, jsou i uživatelé, kteří dávají přednost vzestupnému řazení a možnost změny řazení zde chybí.

Anonym P, studentka

Protože jsem zvyklá s mobilními aplikacemi pracovat, zorientovat se nebyl žádný velký problém. Navíc se aplikace nijak zásadně neliší od běžných SMS aplikací. Symbol zámečku pro šifrování je také dostatečně výmluvný.

Uživatelské prostředí je přizpůsobeno účelu aplikace, a jak jsem již psala, je podobné, jako jsme u běžných SMS aplikací zvyklí. Proto mi přijde vhodné. Mohlo by být barevně o trochu více lahodící oku, to se však dá spravit v rámci dalších aktualizací.

Zahájení šifrované komunikace mi nepřipadalo těžké, zejména proto, že jsem od autora dostala pár rad, jak s aplikací nakládat. Ale předpokládám, že díky použitému symbolu zámečku by mi netrvalo dlouho na postup přijít. Přesto bych ale přidala do aplikace stručnou nápovědu.

Pokud bych chtěla své zprávy posílat šifrovaně, umím si představit, že bych aplikaci používala každodenně. Musela by ovšem ještě projít pár aktualizacemi. Co mi v aplikaci chybí, popíšu v následujícím bodu.

Jako zásadní nedostatek vnímám fakt, že aplikace není propojená s kontakty v telefonu, proto máme možnost pracovat pouze s telefonními čísly, které u většiny lidí neznáme. Také by z mého pohledu bylo vhodné aplikaci rozšířit, aby podporovala rozlišení sim karet u DualSim telefonů.

Lenka Hrubá, asistent pedagoga
Pozn: Uživatelka neodpověděla na celý dotazník, pouze okomentovala problémy s aplikací.
Osobně jsem měla problém se zahájením šifrované konverzace. Ikona zámečku byla malá a přehlédla jsem ji.

6.2 Vyhodnocení odpovědí uživatelů

Prakticky každý se v aplikaci rychle zorientoval, což připisuji blízkosti uživatelského prostředí této aplikace a defaultní SMS aplikace. Se samotným vzhledem problém nebyl, jednalo se spíše o snadno upravitelné drobnosti, nicméně bude jistě vhodné pokračovat v jeho vývoji.

Funkcionalitou splnila aplikace očekávání uživatelů, objevily se občas drobné chyby, které byly nalezeny v kódu a opraveny. Uživatelé též podali hodnotné návrhy pro zlepšení aplikace, některé z nich jsou vyčteny v sekci 7.1: Budoucnost aplikace. Některé z nich již byly zakomponovány do aplikace (například nápověda).

Kapitola 7

Závěr

Cílem této bakalářské práce bylo navrhnout, implementovat a otestovat Android aplikaci pro možnost komunikace šifrovanými zprávami. Ze studia požadavků a specifikací byl vytvořen prvotní návrh na funkcionalitu a rozhoraní aplikace, který byl neusále upravován a zlepšován, stejně jako samotná analýza požadavků.

Na základě návrhu a studia již existujících projektů bylo vytvořeno uživatelské prostředí značně připomínající defaultní SMS aplikaci, což umožňuje snadnou orientaci uživatele v nové aplikaci a snižuje čas potřebný na zaučení se. Aplikace je schopná šifrovat a dešifrovat příchozí SMS nativně, ostatní protokoly či způsoby komunikace podporuje externě pomocí samostatné šifrovací utility.

Oproti původnímu plánu podpory asymetrické šifry a uvedení na Obchod Play bylo upuštěno z důvodu časové tísně.

Aplikace byla též otestována několika uživateli a na základě jejich zpětné vazby byl upraven návrh, implementace, ale i možné budoucí vlastnosti a funkcionality aplikace.

7.1 Budoucnost aplikace

Aplikace jistě není ve své finální podobě, je potřeba na ní ještě dále zapracovat, což osobně plánuji. Po přidání několika následujících feature je možné aplikaci nahrát i na Obchod Play a aplikaci monetizovat. Plánované a možné features jsou:

- Podpora pro kontakty.
- Podpora asymetrické šifry, možnost volby šifrovacího algoritmu.
- Možnost odesílat šifrované SMS jako datové SMS.
- Nativní podpora více protokolů, například XMPP.
- Analýza textu uživatele a návrhy pro zlepšení Huffmanova kódování.
- Větší kontrola konzistence dat, například v případě zřetězených zpráv.
- Možnost logování událostí.
- Lepší možnosti upravení uživatelského rozhraní.
- Lepší struktura zdrojového kódu.

Literatura

[1] The Activity Lifecycle. [Online; navštíveno 12.05.2017].
URL https://developer.android.com/guide/components/activities/activity-lifecycle.html

[2] AsyncTask. [Online; navštíveno 13.05.2017].
URL https://developer.android.com/reference/android/os/AsyncTask.html

[3] Dashboards. [Online; navštíveno 12.05.2017].
URL https://developer.android.com/about/dashboards/index.html#Platform

[4] Fragment. [Online; navštíveno 12.05.2017].
URL https://developer.android.com/reference/android/app/Fragment.html#Lifecycle

[5] Getting Your SMS Apps Ready for KitKat. [Online; navštíveno 12.05.2017].
URL https://android-developers.googleblog.com/2013/10/getting-your-sms-apps-ready-for-kitkat.html

[6] Google Design. [Online; navštíveno 12.05.2017].
URL https://design.google.com/

[7] GSM 03.38. [Online; navštíveno 12.05.2017].
URL https://en.wikipedia.org/wiki/GSM_03.38#GSM_7-bit_default_alphabet_and_extension_table_of_3GPP_TS_23.038_.2F_GSM_03.38

[8] Platform Architecture. [Online; navštíveno 12.05.2017].
URL https://developer.android.com/guide/platform/index.html

[9] SmsManager. [Online; navštíveno 13.05.2017].
URL https://developer.android.com/reference/android/telephony/SmsManager.html

[10] Specification # 03.38. [Online; navštíveno 12.05.2017].
URL http://www.3gpp.org/ftp/Specs/archive/03_series/03.38/0338-720.zip

[11] Specification # 03.40. [Online; navštíveno 12.05.2017].
URL http://www.3gpp.org/ftp//Specs/archive/23_series/23.040/23040-e00.zip

[12] Cohen, R.; Wang, T.: *GUI Design for Android Apps*. Apress, Srpen 2014, ISBN 978-1-4842-0383-5.

[13] Goasduff, L.; Forni, A. A.: Gartner Says Worldwide Sales of Smartphones Grew 7 Percent in the Fourth Quarter of 2016. Únor 2017, [Online; navštíveno 12.05.2017].
URL http://www.gartner.com/newsroom/id/3609817

[14] Kilián, K.: Historie Androidu v kostce aneb Od verze 1.0 až po Android M. Červen 2015, [Online; navštíveno 12.05.2017].
URL https://www.svetandroida.cz/historie-androidu-201506

[15] Lockwood, A.: Content Providers & Content Resolvers. Červen 2012, [Online; navštíveno 13.05.2017].
URL http://www.androiddesignpatterns.com/2012/06/content-resolvers-and-content-providers.html

Příloha A

Tabulka kódování znaků sady GSM 03.38

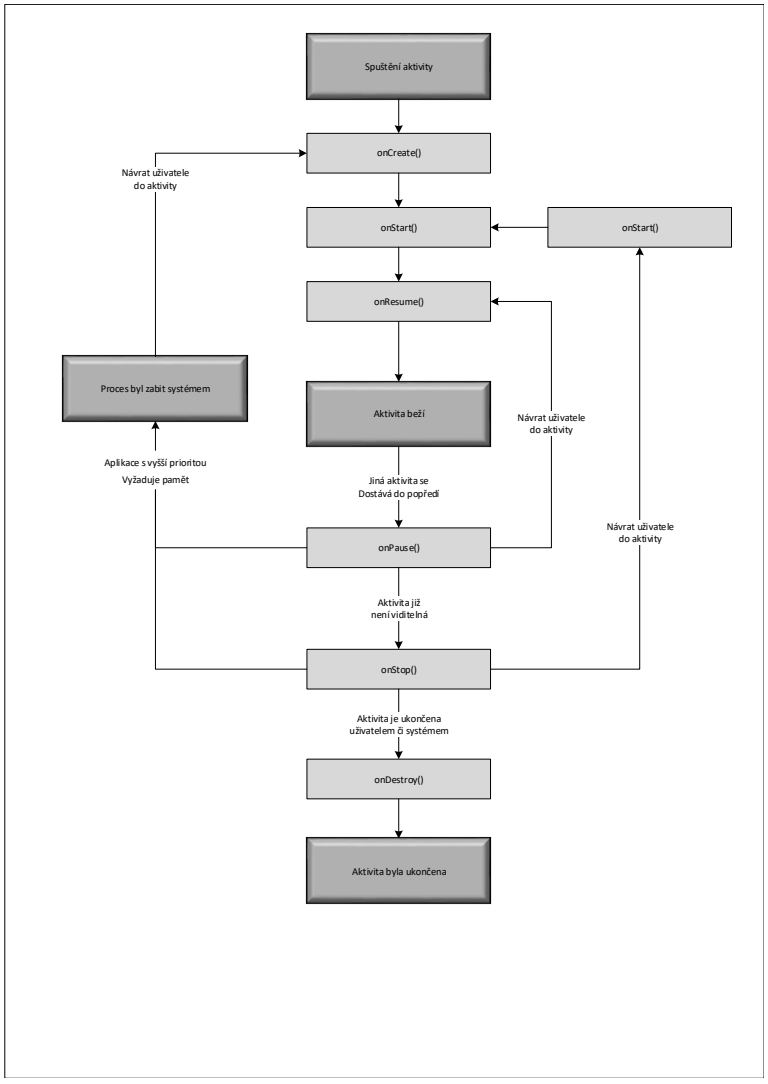
	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00	@	Δ	SP	0	¡	P	¿	p
0x01	£	_	!	1	A	Q	a	q
0x02	$	Φ	"	2	B	R	b	r
0x03	¥	Γ	#	3	C	S	c	s
0x04	è	Λ	¤	4	D	T	d	t
0x05	é	Ω	%	5	E	U	e	u
0x06	ù	Π	&	6	F	V	f	v
0x07	ì	Ψ	'	7	G	W	g	w
0x08	ò	Σ	(	8	H	X	h	x
0x09	Ç	Θ	)	9	I	Y	i	y
0x0A	LF	Ξ	*	:	J	Z	j	z
0x0B	Ø	ESC	+	;	K	Ä	k	ä
0x0C	ø	Æ	,	<	L	Ö	l	ö
0x0D	CR	æ	-	=	M	Ñ	m	ñ
0x0E	Å	ß	.	>	N	Ü	n	ü
0x0F	å	É	/	?	O	§	o	à

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00					|			
0x01								
0x02								
0x03								
0x04		^						
0x05							€	
0x06								
0x07								
0x08			{					
0x09			}					
0x0A	FF							
0x0B		SS2						
0x0C				[				
0x0D	CR2			~				
0x0E				]				
0x0F			\					

Obrázek A.1: Tabulka kódování znaků sady GSM 03.38 [7]

Příloha B

Obrázek životního cyklu Aktivity



Obrázek B.1: Životní cyklus aktivity [1]

Příloha C

Obsah CD

Na přiloženém CD se nachází adresáře:

- src – zdrojové kódy aplikace
- bin – spustitelná aplikace ve formátu APK
- pdf – text této práce ve formátu PDF
- tex – text této práce ve formátu LaTeX