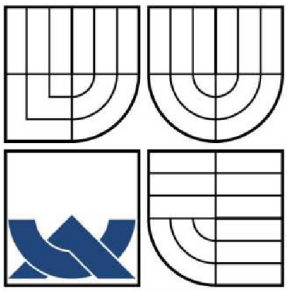
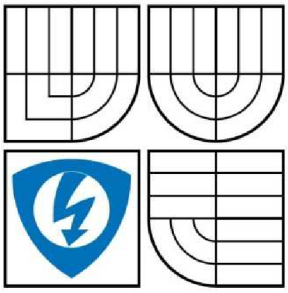


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

APLIKACE PRO BEZPEČNÉ UKLÁDÁNÍ DAT DO PAMĚTI MOBILNÍCH ZAŘÍZENÍ

APPLICATION FOR SECURE DATA STORAGE FOR MOBILE DEVICES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MAREK KOCÁB
AUTHOR

VEDOUCÍ PRÁCE — Ing. TOMÁŠ MÁCHA
SUPERVISOR

BRNO 2010

Aplikace pro bezpečné ukládání dat do paměti mobilních zařízení

Originální zadání práce

ANOTACE

Bakalářská práce je zaměřena na vytvoření aplikace pro bezpečné ukládání dat do paměti mobilních zařízení. Teoreticky jsou popsány operační systémy podle oblíbenosti u uživatelů a programovací jazyky. V další části práce je vysvětlen šifrovací standard AES a jeho princip šifrování a dešifrování dat. V praktické části je vytvořen program PINapplication. Aplikace umožní uživateli ukládat jeho citlivá data pod svým zvoleným heslem. Uložená citlivá data je možno libovolně odebírat a přidávat nová. Data vložená uživatelem jsou zašifrována standardem AES. Celá tato aplikace je naprogramována v jazyce Java ve verzi pro mobilní zařízení (J2ME).

Klíčová slova

Operační systém, programovací jazyk, J2ME, symetrická šifra, AES

ABSTRACT

This bachelor's thesis is focused on the creation of application for secure data storage for mobile devices. The paper theoretically describes operating system according to users popularity and programming languages. The following section explains AES encryption standard and its principle of data encryption and decryption. In the practical part is created a PINapplication programm. The application allows user to store sensitive data using chosen password. It is possible to save new data or erase current data. These data are encrypted by mentioned AES standard. The application is programmed in Java language with the version for mobile devices (J2ME).

Keywords

Operating system, programming language, J2ME,symmetric cipher, AES

KOCÁB, M. *Aplikace pro bezpečné ukládání dat do paměti mobilních zařízení.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 42 s. Vedoucí bakalářské práce Ing. Tomáš Mácha.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Aplikace pro bezpečné ukládání dat do paměti mobilních zařízení jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne …….. .....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Tomáši Máchovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování bakalářské práce.

V Brně dne ……. .....

podpis autora

SEZNAM ZKRATEK

AAC	Advanced Audio Coding
AES	Advanced Encryption Standard
A-GPS	Assisted - Global Positioning System
AJAX	Asynchronous Java Script and XML
AMR	Adaptive Multi Rate
API	Application Programming Interface
ARM	Advanced RISC Machine
AWT	Abstract Window Toolkit
CDC	Connected Device Configuration
CDMA	Code Division Multiple Access
CISCO IPsec	CISCO Internet Protocol security
CLDC	Connected Limited Device Configuration
DBMS	Database Management Systém
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
DVB-H	Digital Video Broadcasting- Handheld
EDGE	Enhanced Data Rates for GSM Evolution
EKA	EPOC Kernel Architecture
EV-DO	Evolution - Data Optimized
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
GF	Galois Field
GPRS	General Packet Radio Service
GPS	Global Positioning Systém
GSM	Global System for Mobile communications
GUI	Graphical User Interface
HSDPA	High- Speed Downlink Packet Access
HTML	HyperText Markup Language
HTTP	Hyper Transfer Protocol
IDEA	International Data Encryption Algorithm
IMP	Information Module Profile
IrDA	Infrared Data Association
ISDB-T	Integrated Services Digital Broadcasting – Terrestrial
JPEG	Joint Photographic Experts Group
JRE	Java Runtime Enviromment
KVM	Kilobyte Virtual Machine
MIDI	Musical Instrument Digital Interface
MIDP	Midlet Information Device Profile
MIPS	Microprocessor without Interlocked Pipeline Stages
MPEG	Moving Picture Experts Group
OPEN GL	Open Graphics Library
OPL	Open Programming Language
OS	Operating System
PDA	Personal Digital Assistant
PNG	Portable Network Graphics
PPTP	Point-to-Point Tunneling Protocol
QVGA	Quarter Video Graphics Array
RAM	Random Access Memory

RSA	Rivest Shamir Adleman
SDK	Software Development Kit
STL	Standard Temple Library
SVG	Scalable Vector Graphics
TCP/IP	Transmision Control Protocol/ Internet Protocol
Triple DES	Triple Data Encryption Standard
UI	User Interface
UIQ	User Interface Quartz
UMTS	Universal Mobile Telecommunications Systém
USB	Universal Serial Bus
VGA	Video Graphics Array
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
WVGA	Wide Video Graphics Array
XML	eXtensible Markup Language
XMLDOM	eXtensible Markup Language Document Object Model
XSL	eXtended Sparse Linearization

Obsah

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

V posledních letech se zvyšují nároky na mobilní telefony a jejich funkce. Každý uživatel chce mít ve svém mobilním telefonu stejné nebo aspoň podobné funkce, které stále nabízí běžné počítače. Každá nově vytvořena aplikace se k těmto požadavkům přibližuje a snaží se z každého mobilního telefonu udělat multifunkční zařízení.

Cílem této práce je seznámit se, popsat operační systémy a programovací jazyky pro mobilní zařízení. Dále pak navrhnout a zrealizovat aplikaci, která umožní uložit soukromá data pod zvoleným heslem tak, aby je nepovolaná osoba nemohla zneužít ke svému prospěchu. Tato data budou zašifrována pomocí symetrického algoritmu AES (Advanced Encryption Standard).

V první kapitole jsou popsány operační systémy a jejich verze. Operační systémy tvoří důležitou část každého mobilního zařízení. Plní pokyny uživatele a řídí přístup k jednotlivým zdrojům. Správným výběrem operačního systému si můžeme práci při vývoji aplikace ulehčit nebo také ztížit. Důležitou roli, zde hraje taky rozšířenost operačního systému a jeho dostupnost. V práci jsou popsány operační systémy podle dostupnosti a oblíbenosti u uživatelů.

V druhé kapitole jsou popsány jazyky pro vývoj aplikací. Obsahují posloupnosti příkazů a postupů, podle kterých se vyvíjený program chová. Jsou zde popsány tři základní programovací jazyky: Python, C++ a Java. Dále pak jaký jazyk je vhodný pro vytváření aplikací v libovolně zvoleném operačním systému. Je zde uveden přehled výhod a nevýhod těchto programovacích jazyků. S důkladně promyšlenou volbou jazyka se může vývoj aplikace pro daný typ mobilního zařízení usnadnit nebo zkomplikovat.

Třetí kapitola popisuje šifrovací standard AES. Je zde vysvětleno jakými operacemi, lze libovolná data zašifrovat a dešifrovat.

Poslední kapitola zahrnuje praktickou ukázku naprogramované aplikace. Je zde ukázáno, jaký programovací jazyk byl použit při vývoji aplikace a zobrazení požadovaných cílů při návrhu. Popsaná aplikace umožňuje uživateli přidávat a odebírat libovolně zvolený záznam teprve po jeho úspěšném přihlášení. Aplikace může být libovolně rozšiřována o nové funkce. Dále je na aplikaci implementován šifrovací standard AES. Pro šifrování a dešifrování dat je použit 128 bitový šifrovací klíč. Tato velikost je dostačující, protože nebyl ještě prolomen.

Uložením soukromých dat si každý může v případě zapomenutí zkontrolovat, zda data souhlasí nebo ne. Je důležité si své osobní údaje chránit před nepovolanými osobami, protože nikdo z nás nechce, aby nám někdo narušoval naše soukromí a zneužíval jej.

1 OPERAČNÍ SYSTÉMY

V dnešní době mobilní telefony už neslouží jako dorozumívací prostředek, ale jejich využitelnost se rozšířila na zábavu, zdroj informací a také mobilní kancelář. Všechny tyto vymoženosti lze provozovat na některých starších telefonech, ale pro mobilní telefony s operačním systémem jsou naprostou samozřejmostí.

Mobilní telefony, které poskytují například video přehrávač, MP3 přehrávač, navigaci, Internet, synchronizovaný e-mail s přílohami, editor kancelářských dokumentů a další zajímavé funkce se nazývá „smartphone“. Označení smartphone, v překladu znamená chytrý telefon, který má otevřený operační systém umožňující instalaci nových aplikací z různých odvětví a tím lze rozšířit jeho funkčnost. Rozšiřitelnost znamená, pokud bude v mobilním telefonu chybět nějaká funkce, lze ji snadno doinstalovat. Lze tedy říct, čím víc bude mobilní telefon rozšířitelným tím bude větší zájem u lidí, kteří by byly ochotni si za kvalitnější telefonní přístroj i připlatit a koupit si ho. Dalším zajímavým znakem u telefonů smartphone je, že využívají tzv. multitasking neboli běh více aplikací na pozadí. Má sdílenou vnitřní paměť, díky této vlastnosti nebudete mít žádné konkrétní omezení pro ukládání položek kontaktů, SMS zpráv, různých událostí, kalendáře, vytváření archivů atd. Nezbytnou součástí chytrých mobilních telefonů je velký displej a podpora paměťových karet. Umožňují také široké možnosti konfigurace, ať už jde o uživatelské prostředí, menu či hardwarových tlačítek, ale také bezproblémovou synchronizaci a zálohy dat s počítačovým softwarem.

Nevýhodou těchto chytrých mobilních telefonů je poměrně vysoká cena, jsou větší, těžší a slabá výdrž baterie. Menší výdrž baterie je dána tím, že mají výkonnější a energetický náročnější hardware.

Trh s operačními systémy pro telefony smatphone je otevřený a dochází stále k vývoji nových operačních systémů. V této práci budou operační systémy popsány podle toho, jakou mají úspěšnost na našem trhu. Mezi nejpoužívanější a nejoblíbenější patří operačním systém pod názvem iPhone od Apple Inc. Za iPhone se propadl operační systém Symbian od firmy Symbian Ltd., který prosazuje největší výrobce mobilních telefonů Nokia. Tyto dva úspěšné operační systémy sledují z povzdálí Windows Mobile od společnosti Microsoft a Android od Google Inc.

1.1 IPHONE OS

Operační systém iPhone (Obrázek 1.1) je operační systém s otevřeným zdrojovým kódem pro telefony smartphone, je vyvíjen společností Apple Inc. Dříve byl tento operační systém znám pod jménem OS (Operating System) X iPhone. V zařízeních ve kterých je tento operační systém nainstalován se používají centrální procesorové jednotky ARM (Advanced RISC Machine).

OS iPhone je složen z několika vrstev. Nejvyšší vrstva je dotyková (grafické ovládání uživatelem), následují vrstva multimediálních služeb (videopřehrávání, komunikační služby), vrstva Core Services a nejnižší vrstva je Core OS. Pro uživatele vrstvy Core Services a Core OS jsou nejméně dosažitelné. Obsahují základní rozhraní iPhone OS (přístup k souborům, nízkoúrovňovým datovým typům, síťové zásuvky). Tyto části vrstvy jsou napsány v jazyce C a zahrnuje technologie SGLite, přístup k UNIX zásuvkám, centrální uložiště atd. OS zabírá v centrálním uložišti méně než 240 MB paměti. Ve vrstvě multimediálních služeb se používá jak jazyk C tak i objektově orientované programování v C. Mezi její základní vlastnosti patří 2D a 3D vykreslování, audio a video. V nejvyšší vrstvě tzv. dotykové se hlavně využívá objektové orientované programování v C. Důležitou součástí na této vrstvě jsou knihovny, které poskytují základní infrastrukturu požadované aplikace. Knihoven je spousta, ale jako příklad jsou uvedeny knihovny Foundation a UIKit. Knihovna Foundation se stará o správu souborů sítových operací a objektově orientovanou podporu kolekcí. Knihovna UIKit

poskytuje vizuální podobu požadované aplikace, včetně tříd pro tvorbu oken, pohledů, ovládacích prvků a manažerů těchto tříd. Ostatní knihovny umožňují přístup ke kontaktním informacím, fotografiím uživatele a ostatním hardwarovým modulům.

Uživatelské rozhraní OS je založeno na dotykovém ovládání Multi Touch, díky kterému se může ovládat spousta funkcí telefonu smartphone. Ovládání může být prováděno klepnutím, švihnutím nebo pohybem prstů od sebe a k sobě. Tento styl ovládání umožňuje rychleji spouštět aplikace nebo si vyhledat telefonní číslo a vytočit ho. Operační systém reaguje na držení přístroje tzn., pokud přístroj položíme na šířku, změní se orientace v některých aplikacích. Tato vlastnost je výhodná například pro psaní zpráv SMS, klávesnice se zvětší a je čitelnější nebo pro hraní her.

Na operačním systému iPhone lze vytvářet vlastní aplikace pomocí jednoduchých a snadno dostupných šablon v XCode. Nejvýkonnější aplikace na iPhone OS budou, pokud se využívají dostupné knihovny a výkonnost celé aplikace lze sledovat pomocí nástroje Shark. Je možnost také vytvářet aplikace pomocí podpory SDK (Software Development Kit), která běží nativně na iPhone OS, podporuje grafickou tvorbu aplikací, která může uživatelovi tvorbu aplikace usnadnit. Aplikace, které sám uživatel vytvoří jsou umístěny na Home obrazovce spolu s dalšími systémovými aplikacemi jako jsou například Photos, Weather, Clock. Při spuštění vytvořené aplikace se zobrazí celá obrazovka a uživatel bude mít přístup ke všem systémovým prostředkům.

Výhodou tohoto OS systému je, že podporuje multitasking, ochranu paměti, aplikace spolupracují bez problému a je možnost je synchronizovat s osobním počítačem. Mezi velkou nevýhodu patří zabezpečení, protože lze převzít kontrolu nad tímto OS a poslat si jeho uložené údaje na třetí mobilní telefon.

Obrázek 1.1: Smartphone s OS iPhone

1.1.1 Verze OS iPhone

IPhone OS má několik verzí. První verze pro iPhone OS je 1.0 a v dnešní době nabízí sérii verze 3. Všechny tyto verze pracují na procesoru ARM, velikost displeje pro všechny verze je 480x320 pixelů. Všechny aplikace, které se spouštěli, se ukládali do paměti RAM

(Random Access Memory), protože systém nepodporoval přídavnou paměť. Systém používal pro udržení uložených informací obnovovací impulzy v paměti RAM.

Verze iPhone OS 1 podporovala zpracování multi-line adres v mapách, protokol TCP/IP, e-maily a různé přílohy bylo možné prohlížet jak na výšku displeje, tak i na šířku. Byly zde také podporovány aplikace (kalendář, jazyková podpora, zapnutí nebo vypnutí EDGE/GPRS (Enhanced Data Rates for GSM Evolution/General Packet Radio Service) při volání ze/do zahraničí). Možnost zobrazení poslední doby hovoru, kapacita pro ukládání SMS zpráv byla 75000, podpora TV výstupu.

Ve verzi iPhone OS 2 byla do OS přidána podpora Wi-Fi připojení, Cisco IPsec (Internet Protocol security) VPN (Virtual Private Network), SVG (Scalable Vector Graphics. Vylepšené možnosti e-mailových zpráv (otevírání příloh z Microsoft Office, mazání více e-mailu najednou), zlepšené vyhledávání v telefonních kontaktech, možnost ukládání telefonních kontaktů na SIM kartu. Zlepšila se výkonnost systému, komunikace s 3G sítěmi. Verze iPhone OS 2.1 umožnila ukládat data pod heslo (je možnost mít až deset pokusů pro zadávání hesla). Po pěti špatně zadaných pokusech se přístroj na jednu minutu vypnul, po šesti špatných pokusech se přístroj vypnul na deset minut a po deseti pokusech se všechny uložená data vymazaly. Při aktualizování si zanechala aplikace původní místo, kde byla uložena. Verze iPhone OS 2.2 má výrazné zlepšení kvality zvuku pro vizuální hlasové zprávy. Systém byl výkonnější a jeho výkonnost lze vidět hlavně při užití poznámek a fotoaparátu. Zlepšení zabezpečení uložených dat, využití USB portu k připojení k PC a externí paměti (paměťové karty).

1.1.2 Verze 3.0 a novější

IPhone OS 3 umožňuje zvětšení libovolné aplikace na celou obrazovku tzn. zvětšení může být provedeno až pětkrát, než je normální velikost. Je zde možnost uložit až 180 aplikací, pro přenos dat se dá využít Bluetooth. Při hovoru pomocí sluchátek se dá zvolit, zda poslech hovoru bude prováděn levým, pravým zvukovým kanálem nebo oběma zvukovými kanály. Verze iPhone OS 3.1 umožňuje rychlejší kopírování, vyjmutí a vložení textu z jedné aplikace do jiné, vylepšené vyhledávání libovolných položek v mobilním telefonu.

Každá verze iPhone OS, která přijde na trh je kvalitnější, rychlejší a výkonnější, než její předchozí verze, ale v každém nově uvedeném OS se najdou chyby. Proto se vydávají tzv. opravné balíčky Service Pack, které mají tyto chyby zacelit a zpříjemnit uživatelům práci s těmito operačními systémy. [2]

1.2 SYMBIAN

Symbian (Obrázek 1.2) je otevřený operační systém, který se používá především na telefonech smartphone. Předtím než vznikl operační systém Symbian, byl jeho předchůdcem OS EPOC, který vyvíjela společnost Psion a fungoval na ARM procesorech. OS Symbian je vyvíjen společnosti Symbian Ltd. O jeho prosazení na trhu vděčí největšímu výrobci mobilních telefonů na světě společnosti Nokia. Můžeme se s ním setkat i u jiných výrobců mezi, které patří Ericsson, Motorola, Samsung, Panasonic a Psion.

OS Symbian je strukturován jako jiné stolní operační systémy, s preemptivním multitaskingem, multithreadingem (umožňuje provádění více paralelních výpočtů současně) a ochranou paměti. Jeho hlavní prioritou je šetření paměti a zdroje systému. Pravidelně kontroluje, zda je aplikace využívána, pokud není, procesor ji odpojí. Tímto způsobem se docílí zvýšení životnosti baterie.

OS Symbian je složen z několika vrstev. Nejvyšší je vrstva grafického rozhraní pro uživatele, dále je vrstva média (multimediální služby, všeobecné služby a služby na zabezpečení), vrstva základních služeb a nejnižší je vrstva služeb jádra. Vrstva základních

služeb, kterou využívá uživatel je nejméně dosažitelná ze všech. Je tvořen souborovým serverem, poskytuje správu paměti, výběr požadované knihovny pro potřebu uživatele, rozhraní pro zásuvné moduly, DBMS (Database Management System) databázi a centrální uložiště (pro nainstalování OS).

Během svého působení na trhu se OS Symbian vyvíjel a zlepšoval. Vzniklo několik verzí. Základem jeho vývoje se stal OS EPOC, který později změnil své jméno na současné.

Obrázek 1.2: OS Symbian

1.2.1 Historie OS Symbian

Předchůdcem OS Symbian byl OS EPOC, který byl vyvinut společností Psion. První z verze OS EPOC se jmenoval EPOC16. Byl určen pro 16 bitové procesory, napsán z části v programovacím jazyku C a z části v assembleru. Následovaly verze EPOC Release 1-4, byly to 32 bitové verze, které byly naprogramovány v jazyku C++ a označují se jako EPOC32. Ve verzi EPOC Release 5 byla označována jako Symbian OS 5 a toto označení se příliš neujalo. Používali ho zařízení netBook, Ericsson R380, MC218, netPad atd. Následovala verze EPOC Release 5u, byl to OS, který neměl otevřený zdrojový kód, nešlo do něho instalovat aplikace a používal se v mobilních telefonech Ericsson R380 . Přípona „u“ v názvu znamenala, že tento systém podporoval Unicode. Byla to také poslední verze před nástupem OS Symbian. OS Symbian je dnes velmi známý a nabízí spoustu verzí.

1.2.2 Verze OS Symbian

První verze pod novým názvem je OS Symbian v6.0. Tato verze OS měla otevřenou platformu, podporovala Bluetooth a byla nainstalována na Nokiích 9210. Při vývoji UI (User Interface), které podporovaly chytré telefony a komunikátory, se dospělo ke dvěma UI. Rozhraní se jmenovaly DFRDs a Device Family Reference Designs a později byly rozšířeny o rozhraní Quartz a Crystal. Tyto rozhraní se spojily s designem Ronneby od společnosti Ericsson a vytvořili základ pro rozhraní UIQ (User Interface Quartz), které změnilo později název na Nokia Series 80 UI. Verze 6.1 se používala na telefonech Nokia 7650 a byla první, která používala zabudovaný fotoaparát s rozlišením 0,3 Mpx.

Ve verzi 7.0 bylo poprvé použito uživatelské rozhraní UIQ. Rozhraní UIQ se použilo pro mobilní telefony v sériích 60, 80, 90. Byla přidána podpora IPv6 a přenos dat pomocí EDGE.

Další verze, jejíž označení je 7.0s, byla kompatibilní s verzí 6.0 a s komunikátory 9500 a 9210. V této verzi se ukázalo, že i OS pro mobilní telefony lze napadnout virem.

Verze 8.0 umožnila vybírat si mezi dvěma jádry EKA1 (EPOC Kernel Architecture 1) a EKA2. Jádro EKA1 mělo tu vlastnost, že bylo kompatibilní se staršími ovladači a jádro EKA2 podporovalo real-time. Běžný uživatel nepoznal rozdíl mezi těmito jádry, protože se chovali naprosto identicky. U této verze byla přidána podpora CDMA (Code Division Multiple Access), 3G, obousměrný datový přenos, DVB-H (Digital Video Broadcasting-Handheld) a OpenGL (Open Graphics Library). Následovali ještě verze 8.1a, 8.1b. Verze 8.1a používala jádro EKA1 a verze 8.1b EKA2. Verze 8.1b podporovala jednočipové zařízení, ale neměla bezpečnostní vrstvu a nepodporovala instalaci otevřených aplikací.

Verze 9.0 se využívala jen pro vnitřní potřeby společnosti Symbian, přestali se využívat jádra EKA1, protože mezi jádrami se přestalo rozlišovat a jejich vývoj byl ukončen. Došlo ke zlepšení bezpečnosti a nastavitelnosti uvnitř OS, je možnost přecházet z ARMv4 na ARMv5 a zpět tzn., že při tomto přechodu nebyla narušena kompatibilita. Verze 9.1 obsahuje mnoho funkcí, jejíž prioritou byla bezpečnost. Tato vlastnost neumožňovala využívat některé aplikace a velmi tím omezila i vývojáře. Verze 9.1 se používá na platformě S60 3rd Edition. První verze obsahovali chybu, která způsobila zaseknutí mobilního přístroje poté, co uživatel odeslal řádově sto SMS. Chybu odstranila společnost Nokia, která vydala malý jednoduchý program. Byla zde možnost přenášet data pomocí Bluetooth. Verze 9.2 se používá na platformu S60 3rd Edition Feature Pack1 a podporuje zařízení OMA Management (Open Mobila Alliance Management). Verze 9.3 podporuje bezdrátové připojení pomocí Wi-fi 802.11b/g, HSDPA (High-Speed Downlink Packet Access) a vylepšenou správu paměti. Verze 9.4 má vylepšenou metodu stránkování, umožňuje aplikacím pracovat rychleji, podpora SQL. Verze 9.5 nabízí využití v oblasti mobilní digitální televize pro formáty DVB-H a ISDB-T (Integrated Services Digital Broadcasting-Terrestrial), kompatibilní se všemi ostatními verzemi. Verze 9.5 je zatím poslední verzí na trhu OS Symbian.[4][5][6]

1.3 WINDOWS MOBILE

Windows Mobile je operační systém, který vytvořila společnost Microsoft. OS je založen na Windows CE (Windows Embedded CE). Windows CE je OS reálného času s hybridním jádrem, využívá sadu základních aplikací WIN 32 API (Application Programming Interface), je určen pro mobilní telefony například PDA (Personal Digital Assistant), smartphone a pro kapesní počítače Pocket PC. Při vývoji OS Windows Mobile se kladl důraz na to, aby byl podobný stolní verzi.

Uživatelské rozhraní (Obrázek 1.3) je odlišné od stolní verze. Můžeme zde vidět datum, čas, e-mailové zprávy, ikony pro použití Bluetooth a Wi-fi. Je zde také nabídka Start (zobrazuje aplikace, které byly nedávno spuštěny, různé odkazy na seznamy programů a nápovědu) a kontakty, tohle vše se nachází na hlavním panelu, který je podobný ze stolní verze. OS obsahuje klasické programy od firmy Microsoft například kancelářský balík Microsoft Office, Windows Media Player, Internet Explorer, tvorba virtuálních sití VPN, PPTP (Point-to-Point Tunneling Protocol) atd.

OS Windows Mobile má tři edice pro různá hardwarová zařízení (Windows Mobile Classic pro PDA, Windows Mobile Standard, Windows Mobile Professional pro mobilní telefony smartphone s dotykovou obrazovkou. Celý OS pracuje na procesoru ARM.

1.3.1 Historie Windows Mobile

První OS, který položil základ pro Windows Mobile se jmenoval Pocket PC. Tento OS využíval různé procesorové architektury například ARM, MIPS (Microprocessor without Interlocked Pipeline Stages).

První verze Pocket PC 2000 měl jednoduché grafické rozhraní, podporoval rozlišení 230x320 bodů (QVGA-Quarter Video Graphics Array), možnost využití přídavných karet (zejména CompactFlash a MultiMediaCard), přenos dat pomocí infraportu IrDA (Infrared Data Association). V této verzi byly naistalovány aplikace typu Pocket Office (Pocket Word, Excel), Pocket Internet Explorer, Windows Media Player atd.

Verze Pocket PC 2002 hlavně podporovala rozlišení QVGA, objevila se i na telefonech smartphone hlavně v oblasti GSM (Global System for Mobile communications) a mezi nové vylepšení v této verzi bylo podpora VPN sití, MSN Manager (Microsoft Network Manager), Windows Media Player s funkcí stream a terminálové služby.

Obrázek 1.3: Uživatelské rozhraní Windows Mobile

1.3.2 Verze Windows Mobile

V roce 2003 došlo ke změnu názvu OS Pocket PC na Windows Mobile, je zde využívána procesorová architektura ARM.

První verze se jmenovala Windows Mobile 2003 a byla vydána ve čtyřech edicích (Windows Mobile 2003 pro Pocket PC Premium Edition, Windows Mobile 2003 pro Pocket PC Professional Edition, Windows Mobile 2003 pro telefony smartphone a Windows Mobile 2003 pro Pocket PC Phone Edition). V této verzi byly přidány tyto funkce například přenos dat pomocí Bluetooth,Windows Media Player 9.0 s optimalizací pro funkci stream, přídavná klávesnice, MIDI (Musical Instrument Digital Interface) soubory pro funkci vyzvánění a použití bezdrátového headsetu. Následovala verze Windows Mobile 2003 SE, měla pár nových vylepšení oproti předcházející verzi. Podporovala tyto funkce například změna displeje na šířku a potom zpět na výšku (využití jen pro Pocket PC), rozlišení VGA (Video Graphics Array) 640x480 a Wi-Fi připojení k Internetu.

Verze Windows Mobile 5.0 umožňovala využívat služby Microsoft Exchange Server, který pracoval s Exchange 2003 s opravným balíkem Service Pack 2. Výhodou této verze je, že se zvýšila životnost baterie. Dříve se data ukládala do paměti RAM a díky tomu se rychle snižovala životnost baterie. Všechny požadovaná data se zálohovala v paměti Flash a nemohlo dojít ke ztrátě dat. Tato verze se mohla pravidelně aktualizovat pomocí implementované funkce Adaption kit upgrades. Byly zde nainstalovány nové funkce mezi, které patří například nová verze kancelářského balíku pod jménem Office Mobile (PowerPoint Mobile, Excel Mobile atd.), rozhraní pro GPS (Global Positioning System), Qwerty klávesnice a pro rychlejší připojení k PC funkce ActiveSync.

Verze Windows Mobile 6 byl vydán ve třech edicích (Windows Mobile 6 Standard pro telefony smartphone bez dotykové obrazovky, Windows Mobile 6 Professional pro Pocket PC

a Windows Mobile 6 Classic pro Pocket PC bez GSM lokalizace. Verze hlavně využívá služeb Windows Live a Exchange Server 2007, používá rozlišení 320x320 a 800x480 (WVGA-Wide Video Graphics Array), podpora VoIP (Voice over Internet Protocol), aplikace Outlook Mobile, která využívá HTML e-mailů, Internet Explorer Mobile podporuje Java Script, AJAX (Asynchronous JavaScript and XML) a XMLDOM (Extensible Markup Language Document Object Model). Verze 6.1 zvyšuje výkonnost systému, hlavní obrazovka se může zobrazit i v horizontálním provedení, možnost zvětšení stránky v internetovém prohlížeči, program Microsoft OneNote (umožňuje vytvářet textové, zvukové a grafické poznámky) a registrace k rozhraní System Center Mobile Device Manager 2008 pro správu mobilního zařízení. Nejnovější verze 6.5 podporuje vylepšený GUI (Graphical User Interface) pro ovládání prsty, nový prohlížeč Internet Explorer Mobile pojmenovaný IE 6, nabídku Start, která se zobrazí na samostatnou obrazovku a službu Microsoft My Phone (umožňuje zálohovat data a přistupovat k datům přes webový prohlížeč).

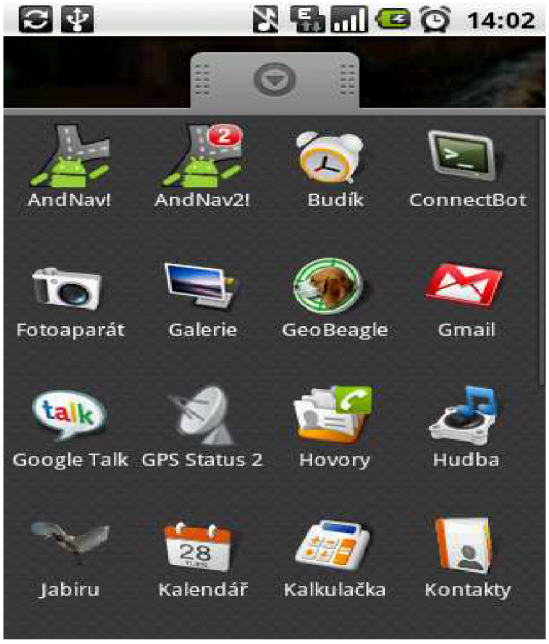
Společnost Microsoft vyvíjí verzi 6.5.1, která bude mít výkonnější systém, zlepšení psaní SMS zpráv, podpora A-GPS (Assisted - Global Positioning System) a bude přidáno více tlačítek pro lepší ovládání. Další verze, která se vyvíjí, se jmenuje Windows Mobile 7. Bude podporovat uživatelské rozhraní založené na SilverLight (je to ovladač, který vytváří zdokonalené webové aplikace), obsahovat podobné prvky jako ve verzi 6.5 a funkci Zune (možnost koupení hudby pomocí Internetu).[7][8]

1.4 ANDROID

Android (Obrázek 1.4) je otevřený operační systém pro mobilní zařízení (smartphone, PDA), který běží na jádru Linux. Vyvinula ho společnost Google Inc. Všechny aplikace pro OS Android jsou napsány v programovacím jazyku Java a pro jejich vývoj mohou vývojáři používat knihovny od společnosti Google.

OS Android podporuje moderní technologie připojení přes GSM/EDGE, CDMA, UMTS (Universal Mobile Telecommunications System), EV-DO (Evolution - Data Optimized), 3G, Bluetooth a Wi-fi. Pro grafické rozhraní používá VGA, 2D grafické knihovny a 3D grafické knihovny založené na OpenGL ES 1.0. Data ukládá do databáze SQLite. Umožňuje psát textové zprávy SMS a posílat zprávy MMS. Na prohlížení webových stránek používá Webkit. Pro multimediální přehrávání podporuje formáty MPEG4 (Moving Picture Experts Group 4), H.264 (kodek pro video), MP3, AAC(Advanced Audio Coding), AMR (Adaptive Multi-Rate), JPEG (Joint Photographic Experts Group), PNG (Portable Network Graphics), GIF (Graphics Interchange Format). Je zde možnost využit kameru pro nahrávání, GPS a dotykový displej.

OS Android je nejmladší OS na trhu a mobilní zařízení, na kterých pracuje, přicházejí stále rychleji na trh a získává oblíbenost u lidí. [3]



Obrázek 1.4: OS Android

1.5 OSTATNÍ OS

Na trhu se nachází spousta jiných OS. Mají zastoupení v různých částech světa, ale jejich podíl na trhu nelze srovnávat se světovými výrobci OS. Mezi ně patří například BlackBerry OS, OS Linux a Palm OS. OS Blackberry je vyvinut společnosti RIM, jeho hlavní využití je v oblasti administrativní a manažerské činnosti. OS Palm je vyvinut společností Palm Inc a OS Linux má největší zastoupení v Asii, především v Číně.

1.6 POROVNÁNÍ OPERAČNÍCH SYSTÉMŮ

V předchozí části byly popsány OS jejich vlastnosti. Je zde sestavena tabulka (Tabulka 1.1), která porovnává jednotlivé OS pro tuto práci. V tabulce jsou uvedeny položky displej, ovládání, otevřenost, programovací jazyky, využití a typ telefonu.

Položka displej obsahuje velikosti rozlišení displeje, položka ovládání definuje jaký druh ovládání je použit. Položka otevřenost hodnotí vytváření aplikací z třetích stran. Položka programovací jazyky určuje, v jakých programovacích jazykách lze vytvářet aplikace pro daný OS. Položka využití na, jakým telefonu lze OS najít a položka typ telefonu uvádí, s jakým zařízením OS komunikuje.

Tabulka 1.1: Parametry OS

OS	iPhone	Symbian	Windows Mobile	Android
Displej	- 480x320 px (pixelů) dotykový	- 360x640 px, - 800x352 px dotykový	- 320x320 px, - 640x320 px, - 800x480 px dotykový	- 480x320 px, - 800x480 px dotykový
Ovládání	- dotykové	- QWERTY, - klasická tlačítka, - dotykové	- QWERTY, - klasická tlačítka, - dotykové	- QWERTY, - klasická tlačítka, - dotykové
Otevřenost	- velký výběr aplikací zdarma nebo placeně, -některé jsou otevřené a uzavřené, - vytváření aplikací v XCode	- je částečně otevřený (jen pro vývojáře), - málo open source aplikací pro OS Symbian	- podporuje všechny druhy aplikací	- umožňuje nahrávat vytvořené aplikace do jiných OS, - podpora cizích aplikací
Progr. jazyky	- Java, -.NET, - C/C++, - SDK	- C/C++, - Python, - Java, -.NET, - SDK	- C, - C++, -.NET, - Java	- Java, - SDK
Využití	- jen na vlastních mobilních telefonech iPhone	- nachází se hlavně na telefonech Nokia	- vyskytuje se na zařízeních HTC a Palm	- nachází se na telefonech T-Mobile G1, T-Mobile G2 Touch,Samsung
Typ telefonu	- smartphone	- smartphone	- smartphone, - PDA, Pocket PC	- smartphone, - PDA

2 PROGRAMOVACÍ JAZYKY

Programovací jazyk je nástroj pro zápis algoritmů a předem definovaných postupů, podle kterých počítač pracuje. Algoritmus napsaný v požadovaném programovacím jazyku se jmenuje program. Pro překlad programu a následného spuštění se používá překladač, který přeloží program tak, aby jej počítač pochopil. Tento překladač se jmenuje kompilátor.

Všechny programovací jazyky jsou napsány v anglickém jazyce, proto je důležité dodržovat správnost příkazů a úkonu. Programovací jazyky mají svou vlastní syntaxi a gramatiku.

Programovací jazyky se dělí na nižší (používají symbolické adresy například Assembler) a vyšší (většina dnešních moderních jazyků například C++, Java atd).

Podle způsobu překladu a spuštění je můžeme dělit na kompilované (před spuštěním přeloženy kompilátorem) a nekompilované (překládají se do mezikódu a potom při spuštění do zdrojového kódu). Zmíněné postupy mohou být kombinovány.

Vyšší programovací jazyky lze dále dělit na procedurální a neprocedurální. Procedurální programování popisuje přesný postup a posloupnost příkazu, jak požadovaný problém řešit. Dělí se na strukturované a objektově orientované programování. Strukturované programovaní rozdělí algoritmus na menší dílčí úlohy a objektově orientované programování pracuje s objekty. Neprocedurální programování je založeno na principu, který určuje daný cíl, a algoritmy jsou ponechány v programu. Dělí se na funkcionální a logické programování. Funkcionální programování se hlavně zaměřuje na programy, které jsou složeny z funkcí, a také nato co se má vypočítat. Logické programování používá matematickou logiku.

2.1 PROGRAMOVACÍ JAZYKY PRO OS

Operační systémy, které jsou v této práci uvedeny, umožňují vytvářet nové aplikace a implementovat do sebe tak, aby při spuštění neohrozily daný OS. Při vytváření aplikací se používají různé programovací jazyky nebo grafická rozhraní, která umějí uživatelovi návrh zjednodušit.

OS iPhone používá pro tvorbu aplikací programovací jazyky Java, .NET, C/C++. Pro vizuální a klasický návrh aplikaci je určeno prostředí Interface Builder nebo NetBeans IDE.

OS Symbian realizuje aplikace v programovacích jazycích C/C++, Java ME, Python, Ruby, .NET, Web Runtime (WRT) a Visual Basic. Programovací jazyk C/C++ používá kompilátor CodeWarrior a může pracovat na OS Windows, Macintosh a Linux. Python, Java a C++ budou popsány v dalších kapitolách této práce. Programovací jazyk Ruby používá podobnou syntaxi jako Perl.

OS Windows Mobile pro vývoj svých aplikací používá vývojové prostředí Microsoft Visual C++. Je to integrované vývojové prostředí (IDE) od firmy Microsoft pro programování v jazykách C a C++. Microsoft Visual C++ hlavně obsahuje nástroje pro tvorbu a ladění C++, využít se dá i pro programy vytvořené v Microsoft Windows API, DirectX API a Microsoft .NET.

Dříve bylo zmíněno, že OS Android pro vývoj svých aplikací používá programovací jazyk Java. V dnešní době se pro realizaci aplikací používá Android SDK. Obsahuje všechny důležité prvky například ladění, knihovny, emulátor, dokumentaci, jednoduché zdrojové kódy a příručky. Android SDK umožňuje pracovat na OS Windows XP, Vista, Macintosh a Linux. Vývojové prostředí se jmenuje Eclipse, který používá předem nainstalované vlastnosti Android Development Tools.

2.2 PYTHON

Python je jednoduchý objektově orientovaný programovací jazyk. Jeho hlavní prioritou je, aby se rychleji vytvářeli nové aplikace a zdrojové kódy byly kratší a jednodušší, než ve standardních jazycích například C, C++ a Java. Neobsahuje žádné nové prvky, ale přebírá a spojuje rysy z ostatních programovacích jazyků. Lze zde najít například dynamické určování typů z jazyka Lisp, objektovou orientaci z jazyka Smalltalk a práci s výrazy z UNIXových shellů. Všechny tyto prvky spojuje do balíku napsaného v nezávislé platformě jazyka C a bývá také označován jako CPython. Python patří mezi hybridní jazyky tzn., že při psaní programu lze využívat nejen objektově orientované, strukturální, ale i funkcionální programovaní. Python je volně šiřitelný, použitelný a to i pro komerční účely, je možnost ho stáhnout z adresy www.python.org.

2.2.1 Knihovna

Python obsahuje velkou knihovnu pro vytváření programů. Knihovna lze doplňovat novými moduly, které mohou být napsány v jazyce C nebo Python. Je přizpůsobená tak, aby podporovala různé formáty (HTML, XML), protokoly (FTP, HTTP) pro psaní aplikací, které mohou spolupracovat s Internetem. Umožňuje vytvářet grafická uživatelská rozhraní, připojovat se k relačním databázím a pracovat s aritmetikou na libovolný počet desetinných míst.

2.2.2 Vlastnosti jazyka Python

Programovací jazyk Python umožňuje vytvořené programy vkládat do jiných jazyků a tím ho lze použít jako rozšiřující jazyk pro jiné aplikace, které potřebují nastavit programovatelné rozhraní. Automaticky spravuje paměť (stará se o přidělení a uvolnění paměti) a mezi jeho důležitou vlastnost patří rychlost psaní programů, přístup k vývojářským nástrojům, manipulace s obrázky webové skripty apod. Zhlediska výkonnosti je na velmi dobré úrovni, protože obsahuje knihovnu Psyco, která nastaví kód programu tak, aby se zvýšila jeho výkonnost na co nejvyšší úroveň. Programy jsou kratší, protože obsahují vysokoúrovňové datové typy, bloky se identifikují pomocí odsazení řádků, proměnné a argumenty se nedeklarují.

Mezi charakteristické rysy jazyka patří, že každá proměnná se definuje jako odkaz na objekt, dojde pouze ke svázání nového názvu proměnné s původním objektem. Všechny funkce se chovají jako běžný objekt až do té doby pokud nejsou zavolány. S funkcemi lze manipulovat, ukládat do předem definovaných proměnných, polí, objektů tzn. manipulace pouze s odkazem na objekt funkce. Složené datové struktury mohou ukládat pouze odkazy na objekty například libovolný seznam, může obsahovat objekty různých typů. U vytváření objektů se mohou používat i členské proměnné (proměnné uvnitř objektu). Operátory nejsou vázány na určité datové typy a jejich použití na požadované operandy se kontroluje za běhu programu. Jazyk Python lze spustit v interaktivním režimu, tento režim se používá na rychlé pokusy.[1][9]

2.3 C++

C++ je objektově orientovaný programovací jazyk, který rozšířil jazyk C. Jazyk C++ není čistě objektovým, ale dokáže využívat i jiné programovací techniky mezi ně patří například generické, objektově orientované a procedurální programování. Řadí se mezi jazyky střední úrovně, který tvoří kombinaci vyšších a nižších úrovní jazyka. Začal se využívat jako vylepšené C, umožňoval vytvářet třídy, virtuální funkce, vícenásobné dědičnosti, šablony, operátory přetížení a kontroly výjimek. Starší verze se jmenovala C with Classes. Je

definován standardem ISO/IEC 14882-xxxx, kde xxxx určují rok vydání. Dnešní vydané standardy mají označení C++0x, kde x značí rok vydání. Například rok vydání 2009 bude zapsán jako C++09. Jazyk C++ se hlavně používá v softwarovém průmyslu nebo také pro vytváření klientských aplikací. V dnešní době patří mezi nejpoužívanější a nejrozšířenější programovací jazyk.

2.3.1 Knihovna

Knihovna C++ byla přejata ze standardní knihovny jazyka C a upravena. Knihovna jazyka C obsahuje například standardní vstup a výstup, konverzi základních datových typů, práce s paměti a zásobníkem, operaci se soubory atd. Důležitou knihovnou v jazyku C++ je STL (Standard Template Library). STL podporuje lepší využití datových struktur a algoritmů například spojové seznamy, vektory (zlepšené pole) a ukazatele. Je možnost psát knihovny i v jiných jazycích a potom je propojit s knihovnou C++. Mezi jazyky, ve kterých se mohou psát nové knihovny jsou Fortran, Pascal, Basic a samozřejmě i v samotném jazyku C.

2.3.2 Vlastnosti jazyka C++

Programovací jazyk C++ umí implicitně přidělovat paměť, ale ne velmi dobře. Pro správnou funkcí požadovaného programu je dbát na to, aby paměť pro určité naprogramované operace byla alokována. Správným vymezením prostoru a uvolněním paměti se může zvýšit výkonnost vytvořeného programu. Mezi charakteristické vlastnosti jazyka C++ patří šablony, objekty, přetěžování funkcí a operátorů.

Šablony umožňují generické programování tzn., pokud chceme vytvořit šablonu, musí kompilátory nahradit požadované parametry šablony určitými hodnotami a potom pro námi vytvořenou šablonu vygenerovat funkci nebo třídu. Jsou velmi užitečným pomocníkem, ale jejich využití zvyšuje velikost kódu, protože každá šablona si vytvoří svoji vlastní kopii.

Objekty jsou instance tříd, které se vytváří za běhu programu a obsahují tyto vlastnosti abstrakci, enkapsulaci, dědičnost a polymorfizmus. Abstrakce obsahuje definované objekty, které vykonávají svou požadovanou práci a udržují komunikaci s jinými objekty v daném programu. Enkapsulace umožňuje skrývat informace. V těle definované třídy se může nastavit, aby členové dané třídy byly deklarovány veřejně, chráněně nebo soukromě. Veřejní členové mají dovolený přístup ke všem funkcím. Soukromý členové mají pouze přístup k těm funkcím, které se nacházejí v dané třídě podle přístupových práv. Chránění členové mají přístup ke všem třídám, které byly zděděny Dědičnost umožňuje vytvářet nové třídy na třídách, které byly už vytvořeny. Tato vlastnost dovoluje napsat část požadovaného programu obecněji a potom ho děděním využívat. Dědění dovoluje nastavit přístup pro dané třídy. Přístupy mohou být veřejné, soukromé a chráněné. Využívá se i virtuální dědičnost, která zajistí, aby byla použitá pouze jedna třída v daném dědění. Je možnost používat i vícenásobné dědění, protože některé třídy mohou být odvozeny z více, než jedné naprogramovaných tříd. Polymorfizmus definuje danému rozhraní více druhů implementací a objektům určuje jejich chování za různých podmínek. Programovací jazyk C++ podporuje dva druhy polymorfizmu statický a dynamický. Statický polymorfizmus podporuje přetěžování funkcí, které dovoluje programům deklarovat více funkcí pod stejnými názvy. Dynamický polymorfizmus se dělí na dědičnost a virtuální funkce členů. Dědičnost pracuje s ukazateli deklarovaných tříd, které mohou ukazovat na odvozené třídy daného typu. Kontejnery mohou tímto způsobem obsahovat ukazatele na objekty různých typů. Funkce virtuálních členů podporuje přesně požadovanou implementaci volaných funkcí, podle předem definovaného objektu.

Další důležitou vlastností je přetěžování funkcí a operátorů. Přetěžování funkcí dovoluje deklarovat více funkcí pod stejným názvem. Jejich použití se určí podle počtu a typů

parametrů. Přetěžování operátorů nemění svou vlastnost při výpočtech a ani počet využívaných operandů (žádný operand není ignorován).[1][10][21]

2.4 JAVA

Programovací jazyk Java je objektově orientovaný programovací jazyk, vyvinutý společností Sun Microsystems Inc. Většina syntaxe je odvozena z programovacích jazyků C a C++, ale obsahuje jednodušší modely a objekty. Mezi velkou výhodu při programování v Javě patří její přenositelnost, která umožní spouštět aplikace na všech operačních systémech. Všechny aplikace se kompilují do bytového kódu. Java byte kód obsahuje instrukce, které jsou podobné strojovým kódům, ale jejich instrukce musí implementovány v tzv. Virtual Machine. Pro běžné zobrazení různých Java aplikací se používá program JRE (Java Runtime Enviromment).

Platforma Java je rozdělena na tři verze. Mezi její verze patří Java SE (Standard Edition), Java EE (Enterprise Edition) a Java ME (Micro Edition). Verze Java SE se používá pro vývoj aplikací na osobním počítači, verze Java EE je nástavba Java SE a slouží pro tvorbu složitějších aplikací, verze Java ME je nejmenší verze a podporuje vytváření aplikací pro mobilní telefony. V této práci bude popsána platforma Java ME (J2ME), která je pro tuto oblast nejvhodnější.

2.4.1 Vlastnosti jazyka Java

Programování v jazyce Java je jednodušší, než v ostatních programovacích jazycích například C, C++. Mezi velkou výhodu patří, že se nemusíme při programování aplikace starat o paměť. Program ji za nás přidělí a po skončení aplikace zase uvolní. Programovací jazyky C a C++ tuto vlastnost umožňovali, ale neefektivně.

Jazyk Java patří mezi jednoduché jazyky, protože má jednodušší syntaxe oproti jazykům C a C++. Mezi charakteristické vlastnosti patří distribuovanost, která podporuje aplikace v sítí například práce se vzdálenými soubory, vytvoření klientských aplikací. Interpretovanost vytváří mezikód ze zdrojového kódu, dále také robustnost, která vytvoří spolehlivé aplikace a vyvaruje se chyb například správa paměti, použití ukazatelů. Další charakteristickým rysem je bezpečnost, umožní vytvořit ochranu kolem počítače při vývoji aplikací v Internetu a výkonnost tzn., že kompilátory překládají pouze ty části zdrojového kódu, které jsou právě zapotřebí. Samozřejmostí je také víceúlohovost (vícevláknové zpracování aplikací) a dynamičnost, kde mohou být knihovny libovolně rozšiřovány.

Nevýhoda jazyka Java je, že vývojové prostředí musí vytvořený program přeložit do zdrojového kódu a potom spustit, celá operace vede ke zpomalení systému.

2.4.2 Java ME

Verze Java ME (J2ME) je určena pro nejmenší zařízení s menším výkonem, mezi které patří například PDA, mobilní telefony, GPS navigace, pagery, čipové karty a dalších zařízení. Je rozdělena na konfigurace a profily. Konfigurace definují softwarovou skupinu pro požadované zařízení a profily doplňují konfigurace pro konkrétnější vlastnosti určitého zařízení. Mezi konfigurace, které se používají v dnešní době patří CLDC (Connected Limited Device Configuration) a CDC (Connected Device Configuration). Každý z těchto z konfigurací obsahuje také svoje profily. Nejznámější a nejpoužívanější profil pro CLDC se jmenuje MIDP (Midlet Information Device Profile), ale taky sem patří i profil IMP (Information Module Profile). Pro CDC jsou vytvořeny tři typy profilů. Patří sem Foundation Profile, Personal Basis Profile a Personal Profile.

Konfigurace CLDC je určena pro zařízení s menší dostupnou pamětí (160 kB až 512 kB). Podporuje bezdrátovou komunikaci, má jednoduché uživatelské rozhraní, menší nároky na výkon, lze nahrát do každého zařízení, minimální počet knihoven, pracuje na 16 nebo 32 bitových procesorech s taktovací frekvencí 25 MHz a využívá virtuální nástroj KVM (Kilobyte Virtual Machine). Virtuální nástroj KVM pro svou činnost potřebuje jen několik kilobajtů paměti. V této konfiguraci, jak již bylo zmíněno, patří i profily. Profil MIDP obsahuje API, knihovny a další funkce pro kvalitnější vytváření aplikací. Mezi nové vlastnosti patří připojení k síti, podpora pro uložení dat a vylepšené uživatelské rozhraní. Na tomto profilu se mohou vytvářet tzv. MIDlet aplikace. MIDlet se skládá ze dvou souborů. První soubor má příponu .jar, je složen z tříd, podprogramů, vytvořených obrázků atd. Lze tedy říct, že tento soubor obsahuje celou naprogramovanou aplikaci. Druhý soubor s příponou .jad má v sobě uložené informace o název souboru .jar, jméno MIDlet aplikace, číslo MIDlet verze, typ konfigurace (CLDC nebo CDC) a jméno MIDP profilu. Tyto dva soubory musí být nahrány do mobilního telefonu, aby byla zajištěna správná funkčnost celé naprogramované aplikace. Dalším profilem této konfigurace je IMP. Je to profil, který se hlavně používá pro jednoduchá zařízení s displejem nebo bez displeje a málo využívá připojení k Internetu. Profil IMP můžeme nalézt na zařízeních typu automaty nebo bezpečnostních systémů.

Konfigurace CDC je určena hlavně pro zařízení s vyšším výkonem a častým připojením k síti. Mezi tyto zařízení patří GPS navigace, PDA, různé komunikátory a další. Je kompaktibilní s CLDC, má vylepšené uživatelské rozhraní, pro vytváření aplikací vyhrazena paměť 2 až 16 MB, pracuje na 16 a 32 bitových procesorech, používá virtuální nástroj Java Virtual Machine a podporuje API. Tato konfigurace obsahuje taky profily a jejich vlastnosti jsou následující. Foundation Profile je pro zařízení, která potřebují ke své funkci minimální nebo kompletní podporu verze Java SE, profil Personal Basis má podobné vlastnosti jako Foundation Profile, ale navíc obsahuje podporu knihovny AWT (Abstract Window Toolkit). Posledním profilem je Personal Profile, který má složitější komponenty AWT.[1][11]

2.5 OSTATNÍ PROGRAMOVACÍ JAZYKY

Pro tvorbu aplikací existuje řada dalších programovacích jazyků. Mezi programovací jazyky pro mobilní telefony patří Perl, Visual Basic a OPL (Open Programming Language).

Programovací jazyk Perl je interpretovaný a dynamický. Je vytvořen tak, aby měl co nejširší oblast využití. Jeho syntaxe je podobná jazyku C. Visual Basic je jazyk, který je řízený událostmi od společnosti Microsoft. Jeho cílem je rychle vytvářet aplikace s grafickým uživatelským rozhraním. Programovací jazyk OPL je interpretovaný a hodně se podobá jazyku Visual Basic. Používá se pro přenosná zařízení (Nokia, Sony Ericsson, Psion) a požaduje pro svou funkčnost OS Symbian.[1]

2.6 SDK

SDK byl mnohokrát již zmíněn a určitě si zaslouží být v této kapitole zahrnut. SDK obsahuje sadu vývojářských nástrojů pro vytváření aplikací pro daný software. Nepatří mezi programovací jazyky, ale používá programovací jazyky pro jednodušší vytváření aplikací. Velmi často obsahuje ukázky kódu nebo jiné technické poznámky. Prostředí SDK je volně dostupné ke stažení z Internetu.

2.7 VOLBA PROGRAMOVACÍHO JAZYKA

V této kapitole byly popsány programovací jazyky, které umožňují vytvářet aplikace pro mobilní telefony. Pro jejich přehlednost je, zde uvedena následující tabulka (Tabulka 2.1).

V tabulce se nachází tyto položky, mezi které patří komplexnost, zastoupení, verze a využití. Položka komplexnost definuje možnosti programovacího jazyka pro vývoj aplikací

(software). V položce zastoupení je ukázáno, který jazyk je využíván na našem trhu. Umožňuje-li programovací jazyk vytvářet aplikace pro mobil, blíže určuje položka verze. Položka využití popisuje, kde a jak se programovací jazyk používá.

Tabulka 2.1: Porovnání programovacích jazyků

Progr. jazyky	Python	C++	Java
Komplexnost	- přebírá rysy z ostatních progr. jazyků, - kratší a jednodušší kódy	- kombinuje vyšší a nižší úrovně jazyků, - velká oblíbenost	- výborná přenositelnost, -tři podverze pro různé oblasti
Zastoupení	- menší rozšířenost oproti jiným jazykům	- velké	- velké
Verze	- ano - Python for S60, - SDK	- ne - SDK	- Java ME, - SDK
Využití	- vkládání vytvořených programů z jazyka Python do jiných program. jazyků - tvorba aplikací pro mobilní telefony Nokia	- vývoj software, - seznámení při studiu na školách	- vývoj software, ale v menším poměru než v C++, - seznámení se při studiu na školách

2.8 SHRNUTÍ

Po seznámení se s programovacími jazyky, je teď na řadě vybrat vhodný programovací jazyk pro danou aplikaci. Podle verzí zaostává jazyk C++, který nemá verzi pro tvorbu aplikací v mobilních telefonech. Dalším kritériem je hlavně využití, protože umožní zjednodušit si práci při vývoje aplikace na požadovaný typ mobilního telefonu. Ostatní vlastnosti programovacích jazyků jsou přibližně na stejné úrovni.

V této práci se bude používat pro vývoj aplikace programovací jazyk Java ME. Jazyk Java dnes podporují všechny mobilní telefony, lze tedy nahrát aplikaci do každého mobilního telefonu s různým OS. Jako vývojové prostředí se použije NetBeans IDE, podporuje verzi Java ME a poslouží jako simulátor telefonu pro danou aplikaci.

3 ŠIFROVACÍ STANDARD AES A JEHO PARAMETRY

Zabezpečení informací je nedílnou součástí dnešní společnosti. Pomocí různých způsobů a metod, je umožněno zakódovat jednotlivé informace, které se přenesou pomocí komunikačních prostředků ke dvěma nebo více uživatelům. Je kladen důraz na to, aby zakódované přenášené informace nemohl nikdo získat a v nejhorším případě zneužít ve svůj vlastní osobní prospěch.

Jednotlivými způsoby zabezpečení se zabývá kryptografie. Stará se hlavně o to, aby zakódovaná data byla čitelná s určitou znalostí. Pro zabezpečení používá šifru známou jako kryptografický algoritmus, který převede požadované informace na šifrovaný text. Důležitou součástí šifry je klíč, bez kterého by zašifrovaná data nešla dešifrovat.

V současnosti existuje mnoho způsobů šifrování informací, ale mezi ty hlavní patří symetrické a asymetrické šifry. Symetrická šifra bude popsána v další části práce. Do oblasti symetrických šifer patří například DES (Data Encryption Standard), Triple DES, IDEA (International Data Encryption Algorithm), AES (Advanced Encryption Standard) a další. Asymetrická šifra je šifra, která používá pro šifrování a dešifrování různé klíče. Pro šifrování informací se používá veřejný klíč. Veřejný klíč uvolní sám uživatel a každý kdo bude mít zájem, může šifrovat jeho informace (zprávy). Dešifrování se provádí soukromým klíčem. Uživatel jej nezveřejní a může podle něj informace (zprávy) dešifrovat. Tento způsob šifrování je pomalý a nedoporučuje se pro zabezpečení velkých objemů dat. Příkladem pro asymetrické šifry jsou RSA (iniciály autorů Rivest, Shamir, Adleman), EIGamal a DSA (Digital Signature Algorithm).

3.1 STANDARD AES

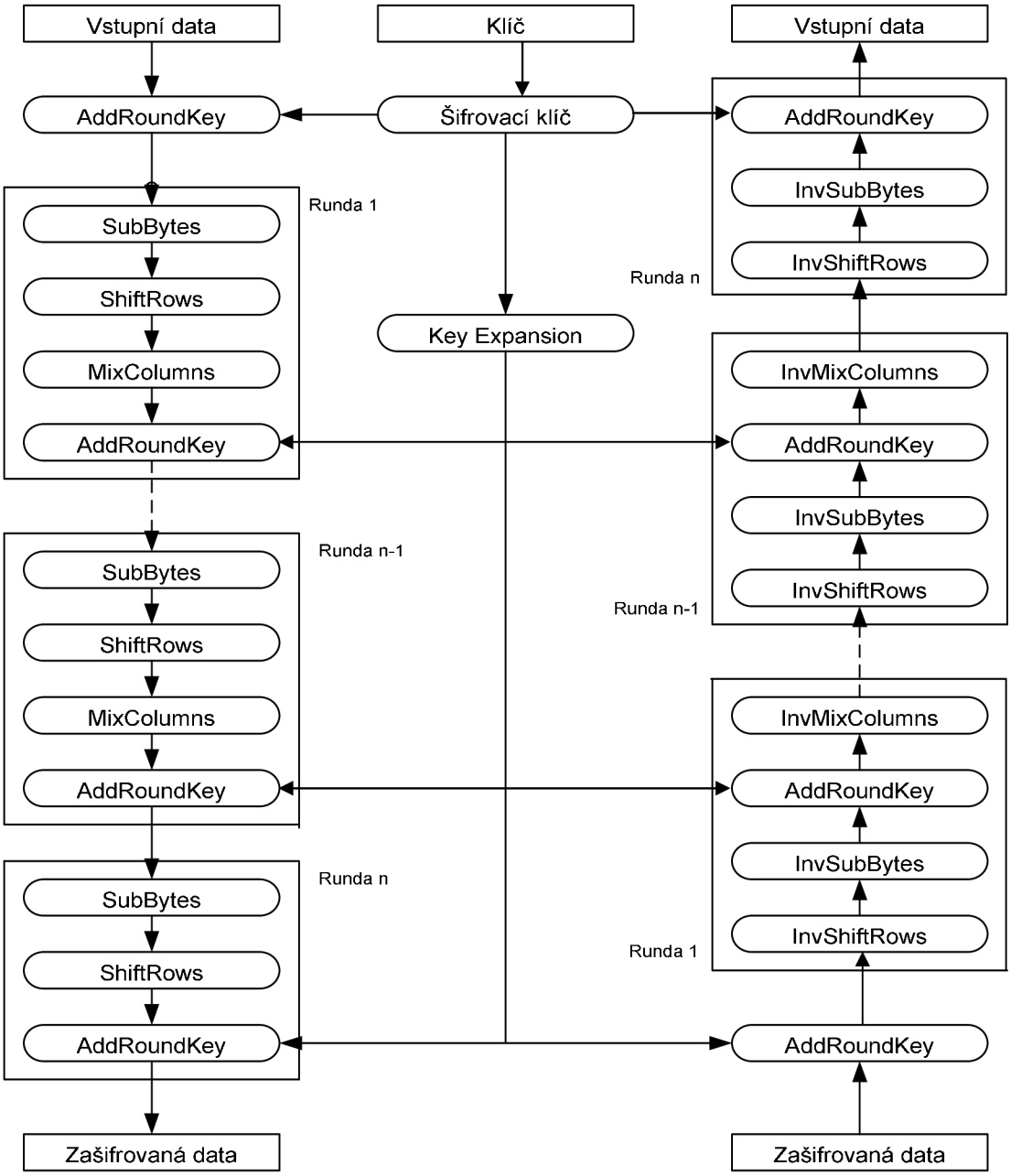
Šifrovací standard AES je symetrická bloková šifra. K šifrování a dešifrování používá stejný klíč. Mezi výhody symetrického šifrování patří, že je velmi rychlé a umožňuje zpracovat větší množství požadovaných informací. Nevýhoda této metody spočívá ve sdílení klíče. Klíč musí být dostatečně dlouhý a náhodný, aby šifra nebyla snadno prolomitelná. Protože je klíč sdílený musí provést bezpečný přenos od odesílatele k příjemci. Jinak by tento způsob šifrování byl nepraktický a mohlo by dojít k získání šifrovacího klíče jiným (nežádoucím) příjemcem.

Dříve se využíval standard DES, který používal klíč o velikosti 64 bitů. Šifrovací algoritmus DES obsahuje bezpečnostní slabiny a lze ho prolomit za několik hodin. Nahradil jej v roce 2002 standard AES. AES má definovanou délku bloku dat 128 bitů a používá šifrovací klíče o délkách 128, 192 nebo 256 bitů. Data určená pro šifrování a dešifrování se uspořádají do matice o velikosti 4x4 byty (State array). Jestliže data nesplňují požadovaný rozměr matice, musí se doplnit. Pro nejjednodušší doplnění se používají nuly, ale jsou i jiné složité algoritmy například PKCS#7 (detailnější popis lze najít v RFC2315, jinak je tento algoritmus implementován .NET Framework). Všechna vstupní data se převedou do šestnáctkové soustavy. Blok dat se zpracovává v tzv. rundách. Počet rund je závislý na velikostí vstupního šifrovacího klíče. Pro velikosti klíče 128, 192 a 256 bitů je dáno 10, 12 a 14 rund. V každé rundě se musí aplikovat rundovní klíče, které jsou získány ze šifrovacího klíče a potom z předchozích rundovních klíčů.

Dříve než vstoupí data do první rundy, je použitá operace XOR se šifrovacím klíčem. Dále rundy aplikují operace SubBytes, ShiftRows, MixColumns a AddRoundKey. Poslední runda provede všechny předchozí operace kromě MixColumns. Po těchto krocích jsou data zašifrována.

Při dešifrování je použit obrácený postup. Vytvořené rundovní klíče jsou stejné, ale jsou aplikovány v opačném pořadí. Rundy provedou operace InvShiftRows, InvSubBytes,

AddRoundKey a InvMixColumns. Poslední runda opět vynechá operaci InvMixColumns a data jsou dešifrována (Obrázek 3.1). [15][16][17]



Obrázek 3.1: Princip šifrování a dešifrování AES

3.2 PRINCIP ŠIFROVÁNÍ

V této části podkapitoly je uveden proces, jak probíhá zašifrování dat pomocí jednotlivých operací SubBytes, ShiftRows, MixClomuns a AddRoundKey. Dříve než dojde k samotnému zašifrování dat, je provedena operace XOR mezi vstupním blokem dat a šifrovacím klíčem.

3.2.1 Operace SubBytes

Operace SubBytes nahradí všechny byty v bloku dat substituční tabulkou S-Box. Každý byte je rozložen na 2×4 bity. Nejvyšší hodnoty čtyř bitů určí adresu řádku a zbytek definuje adresu sloupce. Společným průnikem obou adres se nahradí hodnota v bloku dat hodnotou zjištěnou substituční tabulkou (Tabulka 3.1).

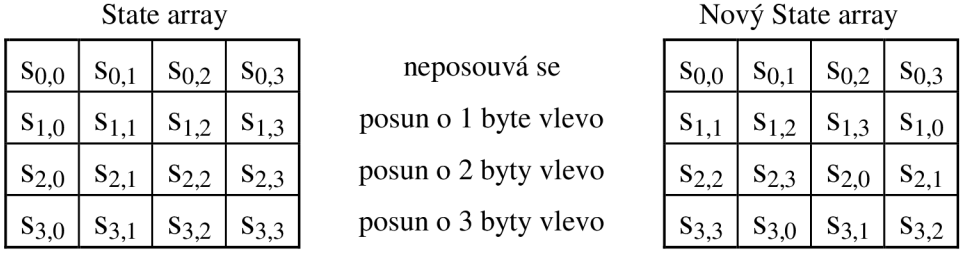
Tabulka 3.1: Substituční tabulka S-Box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

7A po substituci DA

3.2.2 Operace ShiftRows

Tato operace aplikuje v bloku dat po jednotlivých řádcích posun o 0, 1, 2 a 3 byty vlevo (Obrázek 3.2).



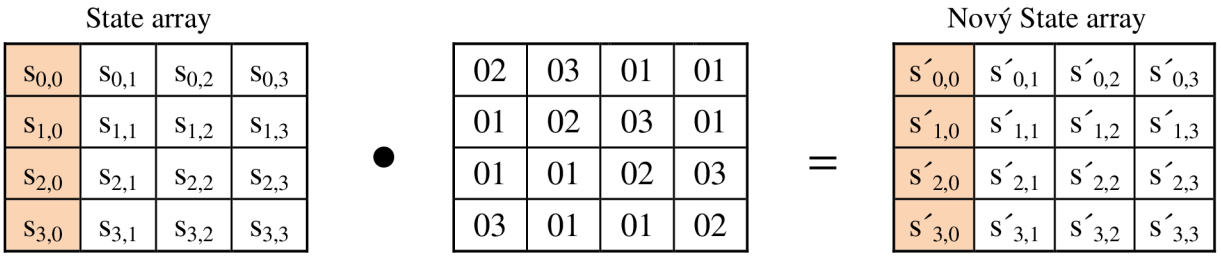
Obrázek 3.2: Operace ShiftRows

3.2.3 Operace MixColumns

Operace MixColumns je použita na všechny sloupce v bloku dat State array. Každý sloupec v bloku dat je chápán jako polynom třetího stupně v tělese GF(2^8). Jednotlivé byty sloupce budou mít po transformaci novou hodnotu. Použije se násobení sloupce polynomu s polynomem

$$a(x) = 03x^3 + 01x^2 + 01x + 02 \text{ modulo } m(x)= x^4+1.$$

Výsledkem je polynom třetího stupně a nová hodnota je funkcí všech bytů daného sloupce (Obrázek 3.3).



Obrázek 3.3: Operace MixColumns

3.2.4 Operace AddRoundKey

Každý byte v bloku dat State array, provede operaci XOR s odpovídajícími byty v rundovním klíči (Obrázek 3.4). [15][16][18]

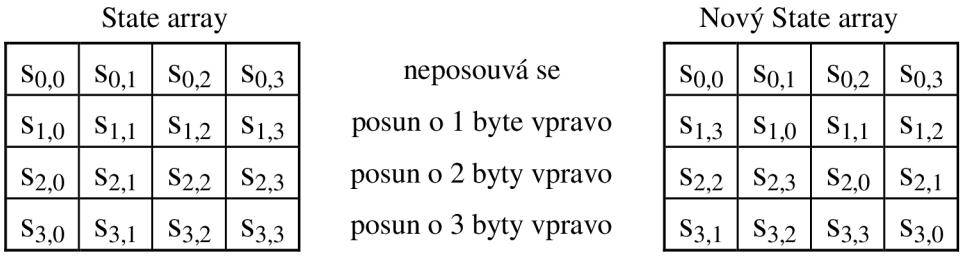
Obrázek 3.4: Operace AddRoundKey

3.3 PRINCIP DEŠIFROVÁNÍ

Po předchozích krocích v minulé podkapitole se vstupní data zašifrovala. Pro dešifrování je použit stejný postup, ale pořadí jednotlivých operací se změní. Operace jsou InvShiftRows, InvSubBytes, AddRoundKey a InvMixColumns. Před dešifrováním zašifrovaných dat je použita operace XOR pro vstup do první rundy. Tato operace je provedena mezi zašifrovanými daty a posledním rundovním klíčem v šifrování.

3.3.1 Operace InvShiftRows

Operace InvShiftRows, posouvá v bloku dat State array jednotlivé řádky o 0, 1, 2 a 3 byty doprava (Obrázek 3.5).



Obrázek 3.5: Operace InvShiftRows

3.3.2 Operace InvSubBytes

Byty v bloku dat se nahradí pomocí inverzní substituční tabulky (Tabulka 3.2). Substituce se provádí stejně jako u operace SubBytes (Tabulka 3.1).

Tabulka 3.2: Inverzní substituční tabulka InvS-Box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

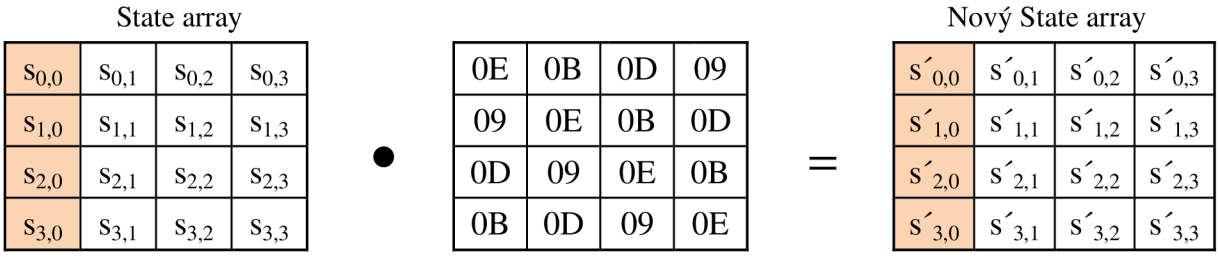
3.3.3 Operace AddRoundKey

Tato operace pracuje na stejném principu, jako u šifrování to znamená, že v jednotlivých bytech v bloku dat State array, je aplikována operace XOR s odpovídajícími byty v rundovním klíči (Obrázek 3.4).

3.3.4 Operace InvMixColumns

Operace InvMixColumns (Obrázek 3.6) využívá stejného postupu jako u MixColumns, ale polynom a(x) je nahrazen za inverzní polynom

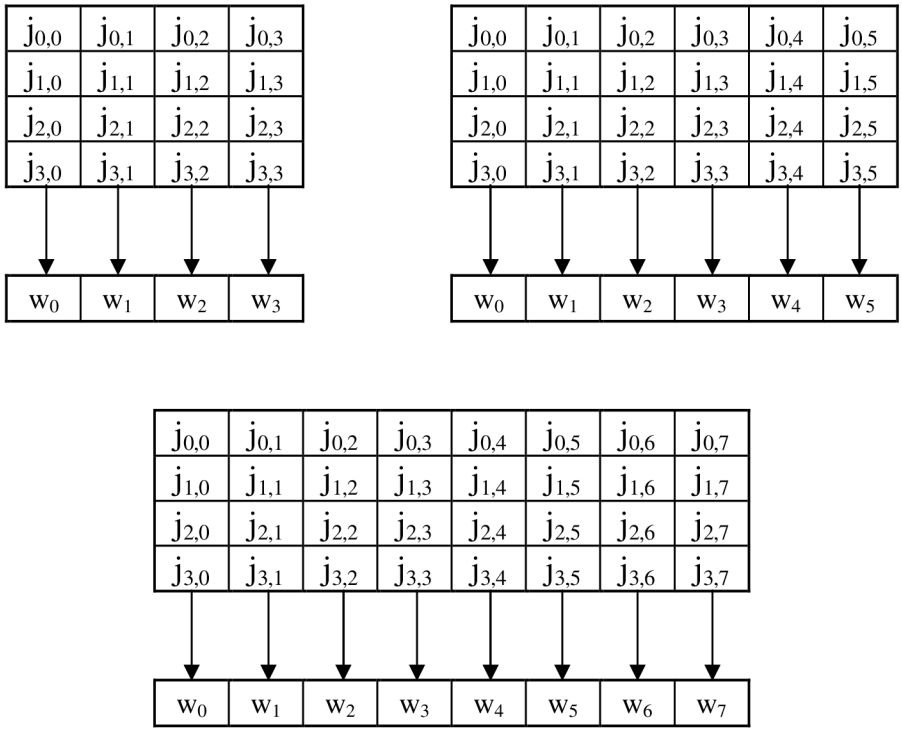
$$a^{-1}(x)= 0Bx^3 + 0Dx^2 + 09x + 0E.$$



Obrázek 3.6: Operace InvMixColumns

3.4 ŠIFROVACÍ KLÍČ

Šifrovací klíč je vyjádřen ve tvaru matice o velikosti K x 4, kde K je rovno 4, 6 nebo 8 pro délku klíče 128, 196 nebo 256 bitů. Každý sloupec klíče obsahuje 4 byty, které vytvoří 32 bitové slovo w_0, w_1, w_2 atd. (Obrázek 3.7)



Obrázek 3.7: Šifrovací klíč velikosti 128, 196 a 256 bitů

3.4.1 Rundovní klíče

Podle velikosti šifrovacího klíče se nadefinuje počet rund. Jednotlivé rundy pro správnou funkci potřebují rundovní klíče. Každý klíč je získán rozšířením vstupního šifrovacího klíče. Všechny jednotlivé kroky popisuje operace Key Expansion.

3.4.2 Key Expansion

Operace Key Expansion vytvoří ze šifrovacího klíče rundovní klíče pro jednotlivé rundy. Z každé předchozí rundy je vytvořen nový rundovní klíč. Předtím, než jsou aplikovány nové rundovní klíče jsou provedeny operace RotWord, SubWord a XOR. V operaci RotWord je provedena rotace posledního sloupce daného klíče o jeden byte (Obrázek 3.8).

Obrázek 3.8: Operace RotWord

Dále je po rotaci posledního sloupce použita operace SubWord. Tato operace nahradí sloupec jednotlivými byty pomocí substituční tabulky S-Box (Tabulka 3.1). K výsledkům z předchozího kroku je aplikována operace XOR a dále je přidána konstanta pro daný krok z tabulky Rcon (Tabulka 3.3). V této poslední operaci je ještě přidáno první slovo získané ze šifrovacího klíče a dále z rundovních klíčů. Hodnoty tabulky Rcon jsou získány ze vztahu $[x^{i-1}, \{00\}, \{00\}, \{00\}]$, kde x^{i-1} jsou mocniny x. Zde x vyjadřuje {02} v tělese $GF(2^8)$. Ve zmíněném vztahu „i“ představuje právě aktuální krok klíče podle tabulky Rcon (Tabulka 3.3). [15][16][18]

Tabulka 3.3: Tabulka Rcon pro klíč délky 128 bitů

Krok:	1	2	3	4	5	6	7	8	9	10
	01	02	04	08	10	20	40	80	1B	36
	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00	00	00

3.5 SHRNUTÍ

Šifrovací standard AES dnes patří mezi oblíbené šifrovací algoritmy. AES ještě nebyl prolomen, ale každým rokem jsou zkoušeny nové způsoby prolomení. Prvním útokem proti AES bylo převést algoritmus na soustavy rovnic (AES-128 by obsahoval přibližně 8000 rovnic o 1600 neznámých). Tento způsob útoku dostal název XSL (eXtended Sparse Linearization). Při prováděných simulacích bylo zjištěno, že převod algoritmu na soustavy rovnic je špatný a nešlo by tento útok v praxi zrealizovat. V roce 2006 byl proveden útok na postranní kanály šifrovacího standardu AES. Útok byl prováděn ne na algoritmus, ale na implementaci. Výsledkem těchto útoků je, že standard AES patří zatím mezi neprolomitelnou šifru.

4 PROGRAM PINAPPLICATION

V předchozích kapitolách byly popsány operační systémy, které se používají v telefonech smartphone a programovací jazyky, používané pro vývoj aplikací. Vytvořená aplikace bude uživateli ukládat jeho soukromé informace pod zvoleným (vlastním) heslem. Heslem v tomto případě je libovolně zvolená kombinace čísel. Do aplikace se bude muset uživatel nejdříve přihlásit, a provede se kontrola zadaného hesla. V případě, že heslo bude zadáno nesprávně, zobrazí se varovné hlášení. Na přihlášení jsou tři pokusy, jestli i na třetí pokus nebude heslo souhlasit, pak se aplikace ukončí. Přihlašovací heslo lze libovolně měnit. Po splnění přihlašovacích údajů se zobrazí uživateli nabídka, kde může vkládat svá soukromá data. Data, která budou zobrazena na ploše, mohou být libovolně smazána.

4.1 VÝVOJOVÉ PROSTŘEDÍ

Na vývoj aplikace je použito prostředí NetBeans IDE, které podporuje programování v jazyce Java. Pro vytvoření projektu v NetBeans IDE se musí použít oblast Java ME, dále se pojmenuje programovaná aplikace, lze vybrat i na jakém typu telefonu budou prováděny simulace. A poslední důležitou vlastností je správná volba rozhraní. Při nesprávně zvoleném rozhraní může být importovaná aplikace do mobilu nefunkční. Doporučuje se nastavit konfigurační rozhraní CLDC-1.1 a profil MIDP-2.0.

Program se jmenuje PINapplication, podle daného názvu se vytvoří midlet s třídou, která se nazývá *PINapplication*. V této třídě se nachází už předdefinované metody *startApp()*, *pauseApp(), destroyApp()* a *PINapplication()*. Do třídy *PINapplication()* se vkládají jména proměnných, dále v metodě *PINapplication()* se vytváří objekty předem definovaných proměnných. Metoda *pauseApp()* dovoluje přerušit prováděný úkon na aplikaci a poslední metoda *destroyApp()* celou aplikací ukončí.

Java ME používá pro svou práci knihovny, které vývojáři vytvořili. Byly použity tyto knihovny *io*, *rms, midlet*, *lcdui* a *lang*. Knihovna *io* slouží pro zadávání vstupních a získávání výstupních dat. Následuje knihovna *rms*, je to systémová databáze pro ukládání a načítání dat pro mobilní telefony. Mezi její zajímavé vlastnosti patří, že na simulátoru ve vývojovém prostředí se data neukládají, ale v mobilním telefonu ano. Používá posloupnost číslování od jedničky, a jestli se vymaže záznam z databáze, je posloupnost narušena. Knihovna *midlet* obsahuje celý chod programované aplikace. Je nastavena sama už při vytváření projektu. Pro vytváření barevných pozadí na mobilních telefonech je realizováno pomocí knihovny *lcdui*. Poslední knihovnou, která byla použitá pro tuto aplikaci je *lang*, umožňuje pracovat s převody a kódování řetězců.

Soubory s příponou *.jad* a *.jar*, ve kterých je aplikace vytvořena i s konfiguračními údaji se naimportují do mobilního telefonu pomocí rozhraní Bluetooth. Po nahrání aplikace do mobilního telefonu by se měla aplikace spustit. Jestli se aplikace nespustí, mohou být špatně nastavena rozhraní, jak již bylo v této části zmíněno.

4.2 APLIKACE

Po spuštění aplikace se zobrazí úvodní obrazovka, po stisknutí tlačítka *Pokračovat* se zobrazí přihlašovací okno. Je nastaveno základní čtyřmístné heslo *1234* pro vstup na plochu s přístupovými hesly, stisknutím tlačítka *Login* se přihlašuje (Obrázek 4.1). Pokud je zadáno přihlašovací heslo nesprávně, zobrazí se varovné hlášení. Je možnost zadat jen třikrát nesprávné heslo. Po třetím se aplikace ukončí.

Obrázek 4.1: Přihlašovací obrazovka

Po úspěšném přihlášení se zobrazí obrazovka s názvem *Přístupová hesla*, která bude prázdná. Pro přidání záznamu na plochu se stiskne tlačítko *Přidat údaj*, po jeho stisknutí se zobrazí obrazovka s nápisem *Vložte osobní údaje* (Obrázek 4.2). Pro uložení nového záznamu se stiskne tlačítko *Uložit*. Tlačítkem *Zpět* se lze vrátit na obrazovku s přístupovými hesly.

Obrázek 4.2: Přidání nového záznamu

Na obrazovce s přístupovými hesly se pak zobrazí uložený záznam. Příkladem mohou být záznamy zobrazené na (Obrázek 4.3).

Obrázek 4.3: Přístupová hesla

Všechny záznamy, které budou zobrazeny na obrazovce s přístupovými hesly, se mohou libovolně odebírat a znovu přidávat nové záznamy. Pro ukázku, je zde předvedeno odebírání libovolného záznamu (Obrázek 4.4). Stisknutím tlačítka *Odhlásit* se uživatel odhlásí a aplikace se ukončí.

Obrázek 4.4: Smazání záznamu

Aby nebylo možné zneužít citlivé informace, existuje zde možnost změny přístupového hesla. Nejdříve se zadá staré heslo, potom nové a pro kontrolu nově zadaného hesla je požadavek na zapsání nového hesla zopakován. Pokud je vše v pořádku, zobrazí se hlášení *Kontrola hesla proběhla úspěšně* (Obrázek 4.5). Tlačítkem *Zpět* se lze vrátit a použít nové heslo pro přihlášení. Po opětovném spuštění aplikace se bude uživatel přihlašovat svým nově

uloženým heslem. V případě nesprávného zadání hesla se zobrazí varovné hlášení *Zadal jste nesprávné heslo*.

Obrázek 4.5: Změna přihlašovacího hesla

4.3 IMPLEMENTACE AES DO PROGRAMU PINAPPLICATION

V předchozí kapitole byl popsán a vysvětlen šifrovací standard AES. Vytvořená aplikace bude rozšířena o zabezpečení citlivých dat vložených uživatelem tímto šifrovacím standardem. Pojmenování aplikace zůstane beze změny, bude se nadále jmenovat PINapplication. Pro šifrování a dešifrování citlivých dat bude použito vstupní přihlašovací heslo. Po přihlášení může uživatel vkládat svá důvěrná data. Přidaný záznam bude zašifrován a uložen do databáze v mobilním zařízení. Pro správnou činnost šifrovacího standardu AES bude záznam, který nesplňuje požadovanou velikost, doplněn o nuly. Dešifrováním se vložené nuly zase odstraní a uživatel uvidí jen svá uložená data.

4.3.1 Aplikace se standardem AES

Všechny funkce a ovládací prvky aplikace zůstaly beze změny, viz. podkapitola 4.2. Výjimku tvoří přihlašovací heslo, které bylo upraveno tak, aby uživatel nezadával jen číselnou kombinaci čísel. Je možno vytvořit vstupní přihlašovací heslo z kombinace číslic a písmen.

Šifrovací standard AES je v aplikaci použit teprve, až uživatel vloží svůj citlivý záznam. Potom dochází ke kontrole velikosti záznamu a v případě, kdy záznam nesplňuje požadovanou velikost, doplní se nulami a je vložen do databáze zašifrovaně. Celý tento proces je proveden automaticky po stisknutí tlačítka *Uložit* (Obrázek 4.2). Dešifrování citlivých dat umožňuje stisknutí tlačítka *Zpět* na obrazovce pro přidání nového záznamu (Obrázek 4.2). Dále k dešifrování dat dochází i po úspěšném přihlášení.

Pro praktickou ukázku jsou použita data přístupových hesel z kapitoly 4 (Obrázek 4.3). Tato přístupová hesla a jejich zašifrování lze vidět na (Obrázek 4.6). Pro úplnost jsou zde zobrazeny dešifrovaná data i s doplněním nulami (Obrázek 4.7).

Obrázek 4.6: Zašifrovaná data

Obrázek 4.7: Data s doplněním nul

ZÁVĚR

Cílem bakalářské práce bylo vytvořit aplikaci, která by ukázala, že je v dnešní době důležité chránit citlivé informace před nepovolanými osobami. K ochraně těchto citlivých informací slouží bezpečnostní hesla a klíče. Bez těchto bezpečnostních prvků by docházelo k pravidelnému zneužívání našeho soukromí.

V práci jsou popsány operační systémy podle oblíbenosti na našem trhu a programovací jazyky, které mohou vytvářet nové aplikace pro dané mobilní zařízení. Operační systémy jsou programy, které se snaží co nejvíce ulehčit uživateli práci při splnění jeho požadavkům. Každá jejich verze přináší něco nového a hlavním cílem je být co nejúspěšnější ze strany uživatelů.

Programovací jazyky, které jsou vybrány pro programování mobilních zařízení, patří k oblíbeným mezi vývojáři. Jde o objektově orientované programovací jazyky, které nabízejí spoustu funkcí. Umožňují rychle psát zdrojové kódy a tím zpříjemnit práci při tvorbě nových aplikací. Jazyk Java, který je vybrán pro vytvoření praktické ukázky, lze vyzkoušet na jakémkoliv mobilním zařízení. Důležité je, aby mobilní zařízení tento jazyk podporoval, jinak nemá smysl aplikaci vůbec nahrávat (importovat).

V další části práce je popsán symetrický algoritmus AES, jeho způsoby zašifrování a dešifrování dat. Patří mezi symetrické šifry, které používají k zašifrování a dešifrování stejný klíč. Je důležité tuto informaci vědět, jinak při zadání jiného klíče nebudou uložená citlivá data dešifrována správně, zobrazí se jiná a popřípadě i nečitelná data.

V praktické části je naprogramována aplikace pro bezpečné ukládání důvěrných dat. Program vyzve uživatele, aby se nejdříve přihlásil. Po správném přihlášení může uživatel ukládat svá důvěrná data, například heslo k běžnému účtu, bezpečnostní schránky atd. Přidané záznamy lze libovolně smazat a potom i přidávat nové záznamy do již uložených. Existuje možnost změnit původní přihlašovací heslo za nové. Aplikace se dá doplňovat o nové funkce, které umožňují získat mnohem kompaktnější a kvalitnější program. Zdrojový kód se nemusí přepisovat, stačí vytvořit nové třídy nebo metody a vhodně na ně odkázat. Tato data pak mohou být například různě řazena, vyhledávaná a také je možnost zvětšit prostor potřebný pro uložení citlivých informací.

Naprogramovaná aplikace byla potom rozšířena o šifrovací standard AES. Pro velikost vstupního šifrovacího a následně dešifrovacího klíče je použita velikost 128 bitů. Citlivá data zadaná uživatelem jsou kontrolována, zda splňují požadovanou velikost. Jestli tomu tak není, doplní se důvěrná data nulami, a následně jsou teprve provedeny operace zašifrování a dešifrování. Menšími úpravami zdrojového kódu, lze rozšířit velikost klíče na 192 a 256 bitů. Velikost šifrovacího klíče 128 bitů je dnes dostačující, protože klíč nebyl ještě prolomen. Aplikace může sloužit i jako ukázka, že je možno šifrovací algoritmus AES implementovat na mobilní telefony.

SEZNAM LITERATURY

[1] FITZEK, Frank H.P.; REICHERT, F. *Mobile Phone Programming and its Application to Wireless Networking*: Springer, 2007. 474 s. ISBN 978-1-4020-5968-1.

[2] *IPhone* [online]. 2009 [cit. 2009-11-6]. Dostupný z WWW: <http://www.developer.apple.com/iphone>.

[3] *Android* [online]. 2009 [cit. 2009-11-12]. Dostupný z WWW: <http://www.developer.android.com/ guide/basics/what-is-android.html>.

[4] *Symbian* [online]. 2009 [cit. 2009-11-8]. Dostupný z WWW: <http://www.symbian.com>.

[5] *Symbian Portal* [online]. 2009 [cit. 2009-11-8]. Dostupný z WWW: <http://www.symbianportal.cz>.

[6] *All About Symbian* [online]. 2009 [cit.2009-11-9]. Dostupný z WWW: <http://www.allaboutsymbian.com>.

[7] *Microsoft* [online]. 2009 [cit. 2009-11-14]. Dostupný z WWW: <http://www.microsoft.com/cze/windowsmobile/meet/version-compare.aspx>.

[8] *Microsoft* [online]. 2009 [cit. 2009-11-14]. Dostupný z WWW: <http://www.microsoft.com/windowsmobile/en-us/default.mspx>.

[9] *Python* [online]. 2009 [cit. 2009-11-16]. Dostupný z WWW: <http://www.python.org>.

[10] *C++* [online]. 2009 [cit. 2009-11-20]. Dostupný z WWW: <http://www.cplusplus.com>.

[11] *Java* [online]. 2009 [cit. 2009-11-21]. Dostupný z WWW: <http://www.java.sun.com>.

[12] *OS iPhone* [online]. 2009 [cit. 2009-11-7]. Dostupný z WWW: <http://www.apple.com/iphone/compare-iphones>.

[13] *OS Symbian* [online]. 2009 [cit. 2009-11-9]. Dostupný z WWW: <http://www.mynokia.cz/f/s60/o/Telefony/E60/Displeje/Screenshot0355.jpg>.

[14] *UI Windows Mobile* [online]. 2009 [cit. 2009-11-15]. Dostupný z WWW: <http://www.nokia-mobile-tone.com/wp-content/uploads/2008/03/3-26-08-winmo61.jpg>.

[15] *AES* [online]. 2010 [cit. 2010-02-22]. Dostupný z WWW*:* <http://csrc.nist.gov/archive/aes/index.html>.

[16] *Federal Information Processing Standards 197 - AES* [online]. 2001 [cit. 2010-02-18]. PDF Dokument: Dostupný z WWW:

<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.

[17] *Rijndael* [online]. 2010 [cit. 2010-02-10]. Dostupný z WWW: <http://www.cs.bc.edu/~straubin/cs381-05/blockciphers/rijndael_ingles2004.swf>.

[18] DAEMEN, Joan; RIJMEN, V. *The Design of Rijndael: AES – The Advanced Encryption Standard*: Springer, 2002. 255 s. ISBN 978-3-5404-2580-9.

[19] HARRISON, Richard. *Symbian OS C++ for Mobile Phones. Programming with Extended Functionality and Advanced Features.* England: WILEY, 2003. 439 s. ISBN:0-470-87108-3.

[20] HARRISON, Richard. *Symbian OS C++ for Mobile Phones. Professional Development on Constrained Devices.* England: WILEY, 2003. 798 s. ISBN:0-470-85611-4.

[21] PRATA, Stephen. *Mistrovství v C++*. Třetí, přepracované vydání. Computer Press, 2007. 1120 s. ISBN 978-80-251-1749-1.

SEZNAM PŘÍLOH

CD

- Bakalářská práce formát pdf,
- program PINapplication (zdrojový kód včetně komentářů),
- komprimovaný balíček PINapplication se soubory jad, jar pro nahrání aplikace do mobilního telefonu,
- zadání práce formát pdf.