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

ELEKTRONICKÝ GEOCACHING

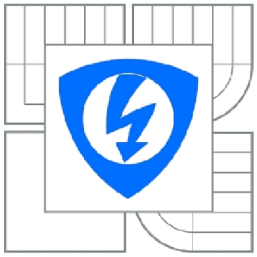
ELECTRONIC GEOCACHING

DIPLOMOVÁ PRÁCE
MASTERS'S THESIS

AUTOR PRÁCE — Bc. LUKÁŠ PAVEL
AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. KAREL BURDA, CSc.
SUPERVISOR

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Lukáš Pavel ***ID:*** 125280
Ročník: 2 ***Akademický rok:*** 2013/2014

NÁZEV TÉMATU:

Elektronický geocaching

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte koncept elektronického geocachingu, kde skrýš (cache) bude realizována bezdrátovou kartou a komunikace s touto kartou bude uskutečňována pomocí telefonu s rozhraním NFC. Navrhnete kryptografické zabezpečení elektronického geocachingu a tento geocaching prakticky zrealizujte. Realizace bude sestávat z aplikace na kartě, z aplikace na mobilním telefonu a z webového hráčského serveru.

DOPORUČENÁ LITERATURA:

[1] Burda K.: Aplikovaná kryptografie. VUTIUM, Brno, 2013.
[2] Gemalto .NET v2/v2+ Smart Card - User Guide. Gemalto, Amsterdam 2008.
[3] Průcha J.: Elektronický geocaching. VUT v Brně, Brno 2013.

Termín zadání: 10.2.2014 ***Termín odevzdání:*** 28.5.2014

Vedoucí práce: doc. Ing. Karel Burda, CSc.
Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ANOTACE
Cílem práce je navrhnout a realizovat koncept elektronického geocachingu s pomocí bezkontaktní karty a telefonu s rozhraním NFC (Near Field Communication). Realizace se skládá z aplikace na bezkontaktní kartě, Android aplikace pro telefon s NFC a webového hráčského serveru. V první kapitole popisuji geocaching a snažím se čtenáře seznámit s touto turistickou hrou. V druhé kapitole rozebírám problematiku čipových karet, věnuju se zde popisu kontaktních i bezkontaktních karet a jejích bezpečnosti. Ve třetí kapitole popisuju rádiové komunikační rozhraní NFC jeho použití a bezpečnost. Ve čtvrté kapitole pak popisuju principy symetrické a asymetrické kryptografie a vybrané kryptografické techniky. V páté kapitole je pak rozebrán návrh řešení elektronického geocachingu pomocí elektronického podpisu. V poslední kapitole jsou popsány všechny vytvořené aplikace.

Klíčová slova: elektronický geocaching, geocaching, smart card, smart karta, čipová karta, bezkontaktní, MULTOS, NFC, near field communication, NFC štítky, bezpečnost, kryptografie, asymetrická kryptografie, RSA, digitální podpis, Programování, Android, Java, C

ABSTRACT
The goal is to design and implement a concept of electronic geocaching with contactless smart card and cell phone with NFC interface. In the first chapter I describe geocaching and try to familiarize the reader with this game. The second chapter deals with the issue of smart cards, I describe here the contact and contactless cards and their security. In the third chapter, I describe use and security of radio communication interface NFC. In the fourth chapter, I describe the principles of symmetric and asymmetric cryptography and selected cryptographic techniques. In the fifth chapter is description of proposed solution for electronic geocaching with digital signature. The last chapter describes all created applications.

Keywords: electronic geocaching, geocaching, smart card, contactles card, MULTOS, NFC, near field communication, NFC Tag, security, cryptography, RSA, digital signature, Programing, Andorid, Java, C

PAVEL, L. *Elektronický geocaching*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 64 s. Vedoucí diplomové práce doc. Ing. Karel Burda, CSc..

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma Elektronický geocaching jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této práce 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 zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

podpis autora

Poděkování

Děkuji vedoucímu práce doc. Ing. Karlu Burdovi, CSc. a Ing. Lukáši Malinovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne ....

podpis autora

Obsah

Úvod 10

1 Geocaching 11

1.1 Historie geocachingu 11

1.2 Stručný popis geocachingu 11

1.2.1 Seznámení s listingem 11

1.2.2 Geocaching v terénu 14

1.2.3 Typy keší 15

1.2.4 Předměty v keších 16

1.2.5 Software pro geocaching 16

2 Čipové karty 19

2.1 Kontaktní karty 20

2.2 Bezkontaktní karty 21

2.3 Programovatelné karty 22

2.3.1 Java Card 22

2.3.2 .NET Card 23

2.3.3 MULTOS 24

2.4 Bezpečnost čipových karet 24

2.4.1 Fyzické útoky 25

2.4.2 Logické útoky 26

2.4.3 Útoky postraními kanály 26

3 Technologie NFC 29

3.1 Použití NFC 30

3.1.1 Platby 30

3.1.2 Přenos dat 31

3.1.3 Použití NFC štítků 31

3.2 Režimy přenosu 32
3.2.1 Emulace karty (Card emulation) 32
3.2.2 Rovný s rovným (Peer-to-peer) 32
3.2.3 Čtení/zápis (Read/Write) 33
3.3 Bezpečnost NFC 33
3.3.1 Odposlech 33
3.3.2 Poškození vysílaných dat 33
3.3.3 Modifikace vysílaných dat 33
3.3.4 Přepojování 33
3.3.5 Navázání komunikace na neuzavřený komunikační kanál 34
3.3.6 Opakované přenášení odposlechnutých autentizačních dat 34
4 Kryptografie 35
4.1 Základní pojmy 35
4.1.1 Kryptoanalýza 35
4.1.2 Hashovací funkce 36
4.1.3 Digitální podpis 36
4.1.4 Certifikace veřejných klíčů 36
4.2 Symetrická kryptografie 37
4.2.1 Proudové šifry 37
4.2.2 Blokové šifry 38
4.3 Asymetrické kryptografie 38
4.3.1 Algoritmus RSA 39
5 Návrh řešení 40
5.1 Jednoduchý návrh 40
5.2 Pokročilý návrh 41
5.2.1 Doplňky návrhu 42
5.2.2 Shrnutí návrhu 42

5.3 Analýza chytrých telefonů s NFC 43
5.4 Reálný návrh 43
5.4.1 Naprogramování čipové karty 43
5.4.2 Naprogramování aplikace pro mobilní telefon 44
5.4.3 Webová aplikace 44
6 Tvorba aplikací 45
6.1 Aplikace pro bezkontaktní kartu 45
6.1.1 SmartDeck 45
6.1.2 Mutil 46
6.1.3 Popis aplikace na kartě 47
6.1.4 Práce s certifikáty 49
6.2 Aplikace pro Windows 49
6.3 Aplikace pro Android 50
6.4 Webová aplikace 53
6.4.1 Popis webové aplikace 54
6.4.2 Popis databáze webové aplikace 54
6.5 Postup vytvoření nové keše 55
6.5.1 Vytvoření keše v databázi 55
6.5.2 Vytvoření keše na kartě 56
7 Závěr 58
8 Literatura 59

Seznam obrázků

Obr. 1.1: Logo Geocaching.com 11
Obr. 1.2: Mapa s bublinou informací o Snezka Cache 12
Obr. 1.3 Listing keše Snezka Cache 13
Obr. 1.4 Různé velikosti keší 15
Obr. 1.5 Garmin chirp 16
Obr. 1.6 Snímky obrazovek aplikace c:geo 17
Obr. 1.7 Snímky obrazovek aplikace Geocaching 17
Obr. 1.8 Snímky obrazovek aplikace Locus 18
Obr. 2.1 Různé použití čipových karet 19
Obr. 2.2 Snímek čipu pod kontaktní oblastí 20
Obr. 2.3 Rozložení pinů 20
Obr. 2.4 Schéma čipové karty 21
Obr. 2.5 Cívky bezkontaktní karty 13,56MHz a 125kHz 21
Obr. 3.1 NFC anténa na zadním krytu telefonu Samsung Nexus S 29
Obr. 3.2 NFC anténa zabudovaná v baterii telefonu Samsung Galaxy Nexus 29
Obr. 3.3 Bezkontaktní platba telefonem s NFC 31
Obr. 3.4 NFC / RFID štítek 32
Obr. 5.1 Přečtený lokační a textový záznam a obrazovka aplikace NFC TagWriter 40
Obr. 5.2 Struktura elektronického geocachingu 41
Obr. 6.1 Záložka Exchange APDU v Mutil 46
Obr. 6.2 Aplikace pro Windows 50
Obr. 6.3 Android aplikace a logování elektronickým podpisem 51
Obr. 6.4 Komunikace mobilní aplikace s aplikací na kartě 52
Obr. 6.5 Ověření nálezu ve webové aplikaci 54
Obr. 6.6 Databáze webového aplikace 55

Úvod

Cílem práce je navrhnout a realizovat koncept elektronického geocachingu s pomocí bezkontaktní karty a telefonu s rozhraním NFC (Near Field Communication). Realizace se skládá z aplikace na bezkontaktní kartě, Android aplikace pro telefon s NFC a webového hráčského serveru.

V první kapitole popisuji klasický geocaching a snažím se čtenáře seznámit s touto turistickou hrou. Je zde popsán způsob jak si vybrat hledanou keš, jak taková keš vypadá a kde ji najít. Také je zde stručný popis aplikací pro geocaching. V druhé kapitole rozebírám problematiku čipových karet, věnuju se zde popisu kontaktních i bezkontaktních karet, jejích vnitřní struktury a bezpečnosti. Také zde popisuji jednotlivé platformy programovatelných čipových karet. Ve třetí kapitole popisuju rádiové komunikační rozhraní NFC jeho použití, přenosové režimy a bezpečnost. Ve čtvrté kapitole pak popisuju principy symetrické a asymetrické kryptografie, vybrané kryptografické techniky a základní kryptografické pojmy jako je hashovací funkce a digitální podpis. V páté kapitole je pak rozebrán návrh řešení elektronického geocachingu pomocí elektronického podpisu. V poslední kapitole jsou popsány všechny vytvořené aplikace. Je zde popsána aplikace pro bezkontaktní kartu MULTOS, představující elektronickou keš, dále aplikace pro operační systém Android, která přes rozhraní NFC komunikuje s kartou a také popis webové aplikace a její databáze.

1 Geocaching

Geocaching [džiokešing] je celosvětová hra spojující turistiku a moderní technologie. Cílem hry je najít s pomocí navigačního přístroje GPS ukrytou schránku (anglicky cache [keš]). Tato hra je vhodná jak pro jednotlivce, tak i pro rodiny s dětmi. Málokteré dítě odmítne výlet s pokladem na konci. Geocaching je také ideální pro milovníky záhad a šifer, které se u určitých typů keší musí vyluštit. Je to radost z pokoření určité výzvy co pomáhá hráčům geocachingu v honu za další keší.

Obr. 1.1: Logo Geocaching.com

1.1 Historie geocachingu

Zásadním krokem bylo rozhodnutí vlády Spojených států amerických vypnout chybu uměle zaváděnou do systému GPS (Global Positioning System). Do té doby byla přesnost GPS přijímačů řádově v desítkách metrů, což byla nedostatečná hodnota pro jakékoliv přesnější určování pozice. Na základě tohoto rozhodnutí se v noci z 1. na 2. května 2000 přestala tato umělá chyba do systému zavádět a přesnost GPS se zvýšila na jednotky metrů. [2]

Jistý Dave Ulmer následující den umístil do lesa v Oregonu schránku a zveřejnil její souřadnice. Kdo ji našel, mohl si z ní něco vzít a zanechat něco svého – knihy, CD, software. Během několika dnů se v USA objevily další podobné schránky a do měsíce byl na internetu vytvořen web, kde se lidé mohli podělit o své dojmy a zážitky z hledání. Slovo geocaching bylo poprvé použito 30. května 2000. Skládá se z předpony geo, označující činnost související se Zemí, a slova cache (schránka). [2]

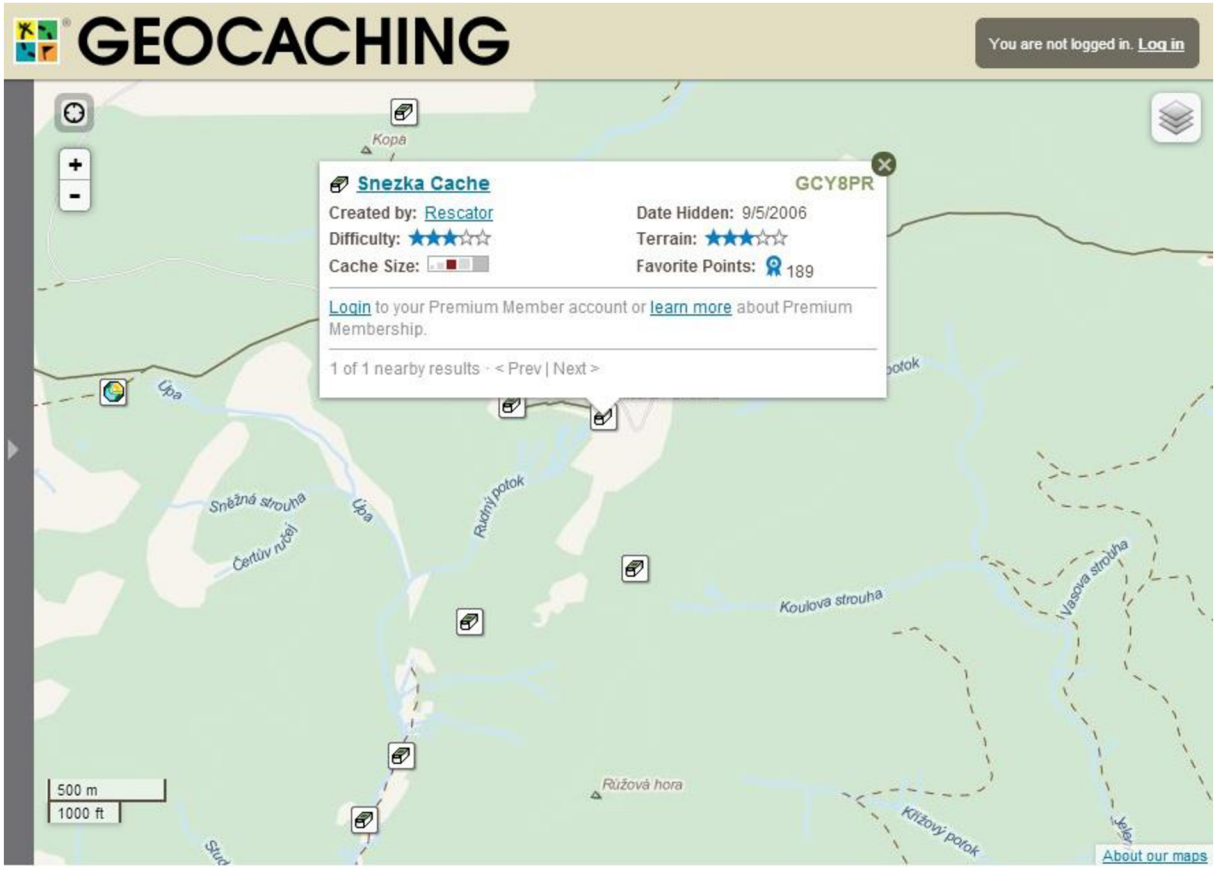
Do dnešního dne je po celém světě ukryto přes 6 000 000 keší, které hledá 2 265 997 aktivních hráčů.[3] V České Republice je geocaching velice populární je zde ukryto přes 36 000 keší, což je zhruba 46 keší na 100km^2 to řadí Českou Republiku na 7. místo s nejhustší sítí schránek na světě!

1.2 Stručný popis geocachingu

1.2.1 Seznámení s listingem

Nejdůležitější je bezplatná registrace na oficiálních stránkách www.geocaching.com. Po registraci si již můžeme na mapě (www.geocaching.com/map/) najít nějakou keš, ke které se chceme vydat. Pro tuto práci jsem jako vzorovou keš vybral nejvýše položenou keš na území České Republiky, která je uložena na Sněžce.

Po kliknutí na obrázek keše na mapě se zobrazí bublina se základními informacemi o keši, viz Obr. 1.2. Pro zobrazení kompletních informací o keši stačí kliknout na odkaz skrývající se pod názvem keše. Tato stránka s informacemi o keši se nazývá listing.



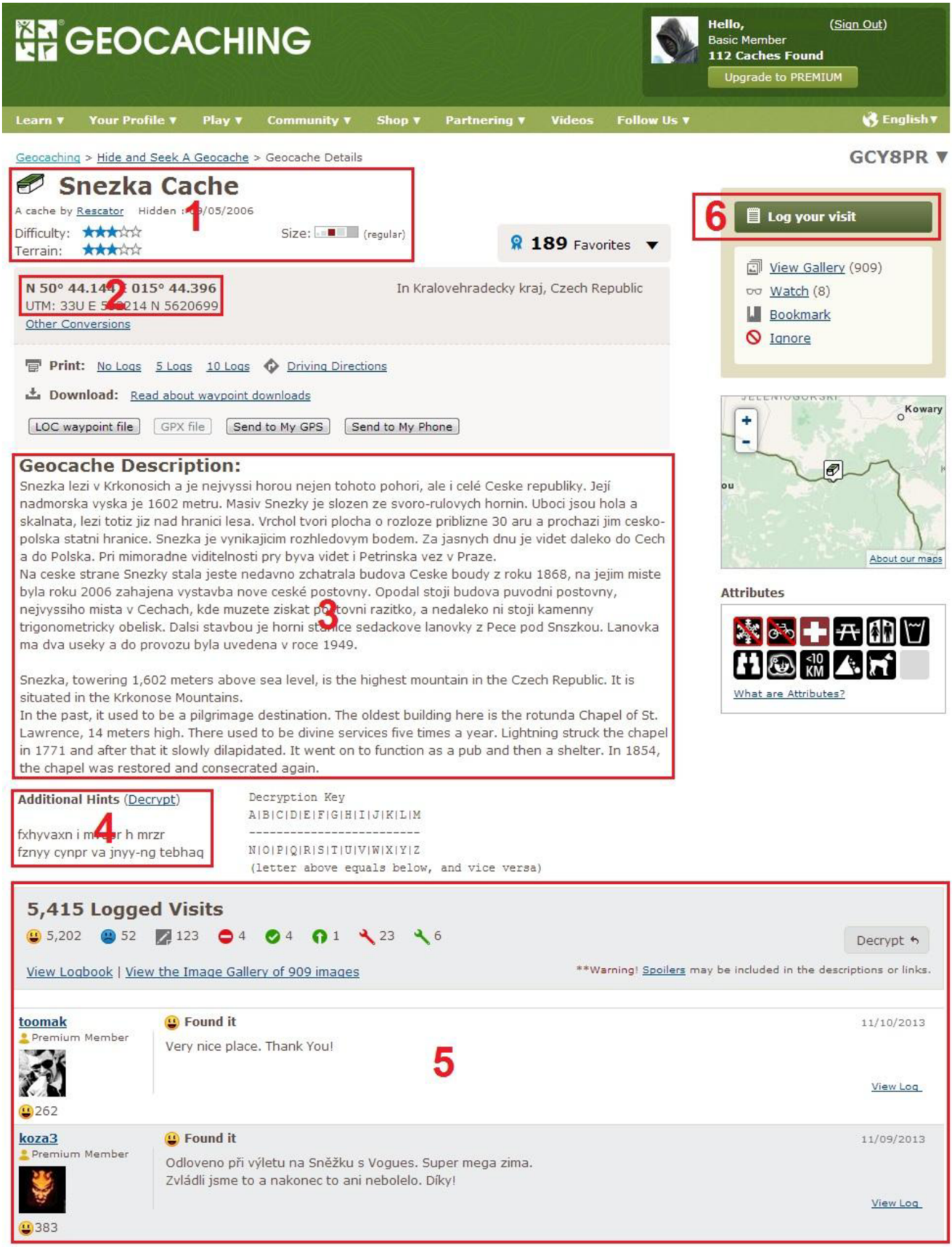
Obr. 1.2: Mapa s bublinou informací o Snezka Cache

Celý listing můžeme vidět na Obr. 1.3. Na obrázku jsem červeně vyznačil nejdůležitější oblasti webové stránky, které jsou potřeba k úspěšnému nalezení keše. V prvním označeném poli se nachází název keše a její typ (typy keší jsou probrány v kapitole 1.2.3), který je znázorněn ikonou. Dále je zde informace o velikosti keše a obtížnosti jejího nalezení. Položka *Difficulty* nám říká jak dobře je keš schovaná a položka *Terrain* jak náročný terén musí hráč zdolat, aby se ke keši dostal. Platí, čím více hvězdiček tím obtížnější.

V druhém označeném poli jsou souřadnice keše v systémech WGS84 a UTM. Po kliknutí na odkaz *Other Conversions* je možno vybrat i jiné souřadnicové systémy dle preferencí hráče. Souřadnice jsou vidět až po přihlášení.

Pod třetím označeným polem je samotný popis keše. Kromě samotného popisu jak nalézt keš je zde obvykle i kratší text, který nám o zdejším místě řekne historii, zajímavost nebo povídku která se k místu váže.

Ve čtvrtém poli je takzvaný *Hint* neboli nápověda, která se používá při dohledávání keše. Tato nápověda se zavedla z důvodu špatného signálu v určitých lokacích, kde je přesnost GPS příliš špatná na přesné určení pozice keše. Je zde napsáno např. v pařezu, pod kameny atd.



Obr. 1.3 Listing keše Snezka Cache

Z důvodu výzvy pro některé hráče, kteří nechtějí nápovědu používat a chtějí keš najít jen s pomocí souřadnic je nápověda zašifrována pomocí jednoduché šifry s posouváním abecedy (Caesarova šifra). Ostatní si mohou nápovědu zobrazit buď kliknutím na slovíčko *Decrypt*, nebo ho rozluštit pomocí již zmíněného dosazení písmen v posunuté abecedě. Pro tento případ je vedle zašifrované nápovědy i klíč k jejímu rozluštění.

Pátým označeným polem jsou tzv. *logy*. Jsou to zápisky hráčů, kteří keš našli a zde se s ostatními můžou podělit o zážitky z hledání keše a poděkovat za keš jejímu zakladateli. V logu by se neměla objevit žádná nápověda jak keš najít, aby se nepokazila zábava těm, co keš ještě nenašli. Je možno zde také zjistit kolik lidí už keš našlo, jejich počet je napsán vedle žlutého smajlíka nad logy 😀.

Pod posledním šestým polem je tlačítko *Log your visit*. Toto tlačítko používá hráč pro vyplnění logu. Zde může popsat svoje zážitky a poděkovat, vyplněním tohoto pole je hráči připsán nález keše a obvykle je to označováno jako *zalogování*. Do logu lze také napsat současný stav keše jako informaci pro jejího správce, např. že do keše zatekla voda a je potřeba vyměnit logbook (bude o něm zmínka v příští kapitole).

1.2.2 Geocaching v terénu

Předtím než vyrazíme do terénu, je vhodné mít sebou listing. Celý listing je možno si vytisknout na papír nebo stáhnout do mobilní aplikace v chytrém telefonu. (O aplikacích se budeme bavit v kapitole 1.2.5). Další důležitou věcí je mít zařízení s GPS přijímačem nebo alespoň mapu. V městských oblastech je možné najít keš jen s pomocí leteckých map a nápovědy, ale hledat keš v otevřeném prostranství nebo v lese je bez GPS téměř nemožné. Určitě je vhodné mít sebou tužku nebo propisku, občas se může hodit i baterka.

Když dorazíme na cílové souřadnice, musíme samotou keš najít. Vedle nápovědy nám v hledání můžou pomoct i různé znaky v okolí. K frekventovaným keším vedou často tzv. *geocestičky*, což jsou vyšlapané cesty v trávě nebo jehličí od předchozích hráčů. Keše jsou obvykle dobře maskovány, aby je nenašli nezasvěcení. Nejčastější ukryty keší jsou různé dutiny ve stromech, skalách nebo ukryty pod kameny, kořeny stromů a v pařezech. V městských oblastech se taky dost často používá magnetů, k přichycení malých keší zespodu laviček a různých kovových konstrukcí. Nicméně existují i keše uložené pod vodou uprostřed rybníka nebo pověšené vysoko na stromech a skalách.

K jejich nalezení nám může pomoct také velikost, zmíněná v kapitole 1.2.1. Nejčastější velikosti keší jsou *regular* a *micro*. Mikro keše reprezentují nejčastěji obaly od fotografických filmů. Jako regular keše se nejčastěji používají vodotěsné svačinové krabice s objemem 1-3 litry. Různé vzhledy keší jsou zobrazeny na Obr. 1.4.

Nejdůležitější součástí keše je *logbook*. Logbook je taková návštěvní kniha keše. Do logbooku se zapisují hráči, kteří keš našli. Můžou zde napsat krátkou zprávu o svém hledáni a poděkovat za keš, ale hlavně se sem píše přezdívka hráče a čas jejího nalezení. V keši si dále

může nacházet propiska a případně nápověda k další hledané keši pokud se jedná o Multikeš. Dále zde mohou být různé předměty na výměnu. Můžou zde být nálepky, přívěsky, hračky, mince a různé další poklady. Pokud si hráč chce nějakou věc vzít, musí do keše na oplátku vložit věc podobné hodnoty, aby zůstala hodnota keše zachována. Proto je vhodné si sebou na hledání keší brát i svojí krabičku s poklady na výměnu. Pokud hráč něco v keši vymění, zapíše výměnu do logbooku s označením IN: (vložil jsem) OUT: (vzal jsem).

Obr. 1.4 Různé velikosti keší [6]

Hráč by měl být při vyzvednutí nebo ukládání keše opatrný, aby nevzbudil pozornost a nikdo ho při vyzvednutí/ukládání keše neviděl. Je také dobré keš dobře zamaskovat. Důvodem této opatrnosti je ochrana schránek proti poškození, přemístění nebo odcizení.

1.2.3 Typy keší

Tradiční keš - Nejjednodušší typ keše, v listingu jsou zapsány přesné souřadnice jejího uložení. Nemusí se plnit žádné další úkoly, stačí keš najít, zapsat se do logbooku v keši a zalogovat se na internetu. Tento typ keší je nejpočetnější.

Multikeš - souřadnice z listingu zavedou hráče na určité místo, kde získá další informace jak se dostat k finální keši. Může zde najít krabičku s dalšími souřadnicemi, šifrou nebo hádankou, která ho po vyluštění nasměruje správným směrem. Těchto zastávek může být více než jedna, takže hledání Multikeší zabere více času.

Mystery keš - tento typ keše se vyznačuje tím, že neznáte žádné souřadnice a je potřeba na ně přijít už doma. Obvykle je potřeba vyluštit nějakou hádanku, rébus nebo odpovědět správně na několik otázek, přičemž každá odpověď má přiřazené číslo, které se dosadí do vzorce pro určení souřadnic. Stačí jedna špatná odpověď a souřadnice mohou ukázat na místo vzdálené stovky metrů.

Earth keš - je typ keše u které se nehledá krabička, ale cílem je dostat se na nějaké místo s geologickou či jinou zajímavostí týkající se naší Země. Splnit zde nějaký úkol, například se vyfotit před skalním útvarem nebo odpovědět na pár jednoduchých otázek z okolí keše.[5]

1.2.4 Předměty v keších

Travel bug, geocoin -Travel bug je libovolná věc doplněná o kovový identifikační štítek s unikátním číslem. Jeho cílem je putovat z keše do keše dle přání jeho majitele, např. cesta kolem světa. Toto cestování zajišťují hráči, kteří se plánují vydat správným směrem a TB si vezmou sebou. Geocoin je kovová nebo i jiná mince s identifikačním kódem, která má stejný účel jako TB. Současnou pozici TB nebo geocoinu je možno zjistit pomocí Tracking Number na stránkách geocacing.com.[5]

Garmin Chirp - je speciální zařízení určené pro geocaching. Do okolí v průměru 10m vysílá signál, který mohou zachytit kompatibilní GPS navigace. Po zachycení signálu se hráči v navigaci zobrazí majitelem naprogramovaná nápověda nebo další souřadnice. Garmin Chirp je voděodolný a baterie by měla napájet zařízení až jeden rok.[1][4]

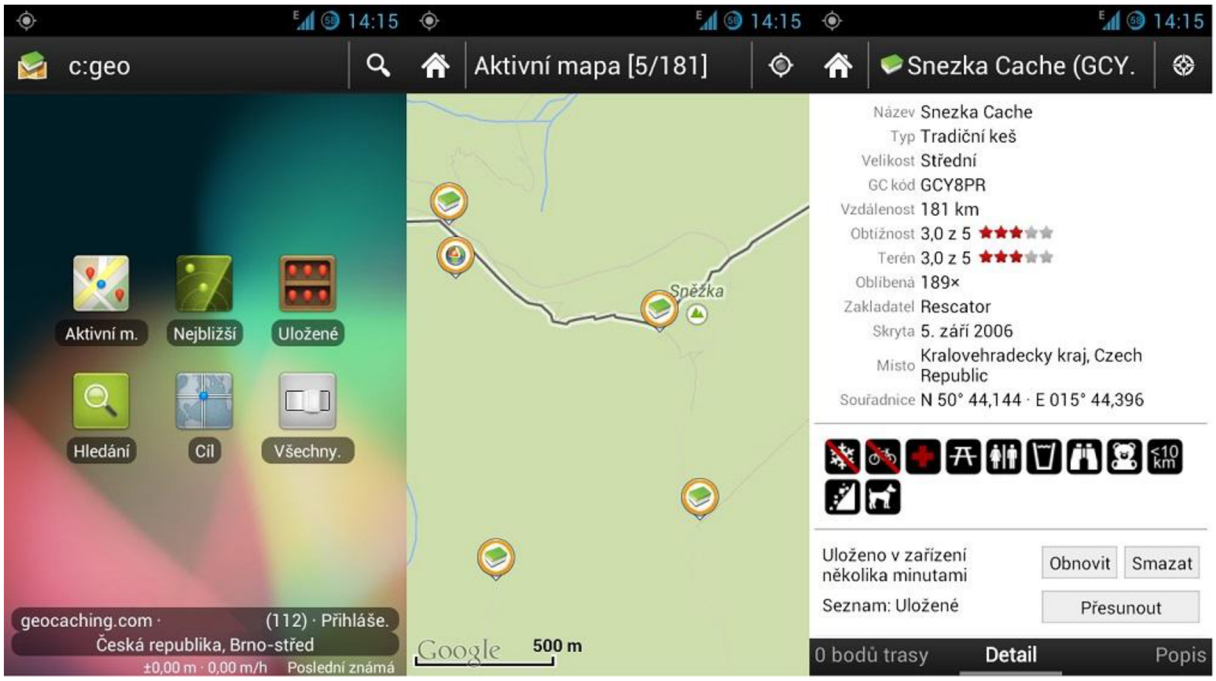
Obr. 1.5 Garmin chirp

1.2.5 Software pro geocaching

Pro geocaching existuje plno aplikací, na českých webových stránkách GeoWiki [5] lze nalézt seznam těch nejpoužívanějších. Jsou zde aplikace pro různé platformy od PC, PDA přes webové aplikace až po aplikace pro operační systémy Android, Symbian a iOS. V práci se zmíním jen o aplikacích pro Android, protože je to nejrozšířenější OS pro přenosná zařízení a je nejvhodnější pro práci s rozhraním NFC.

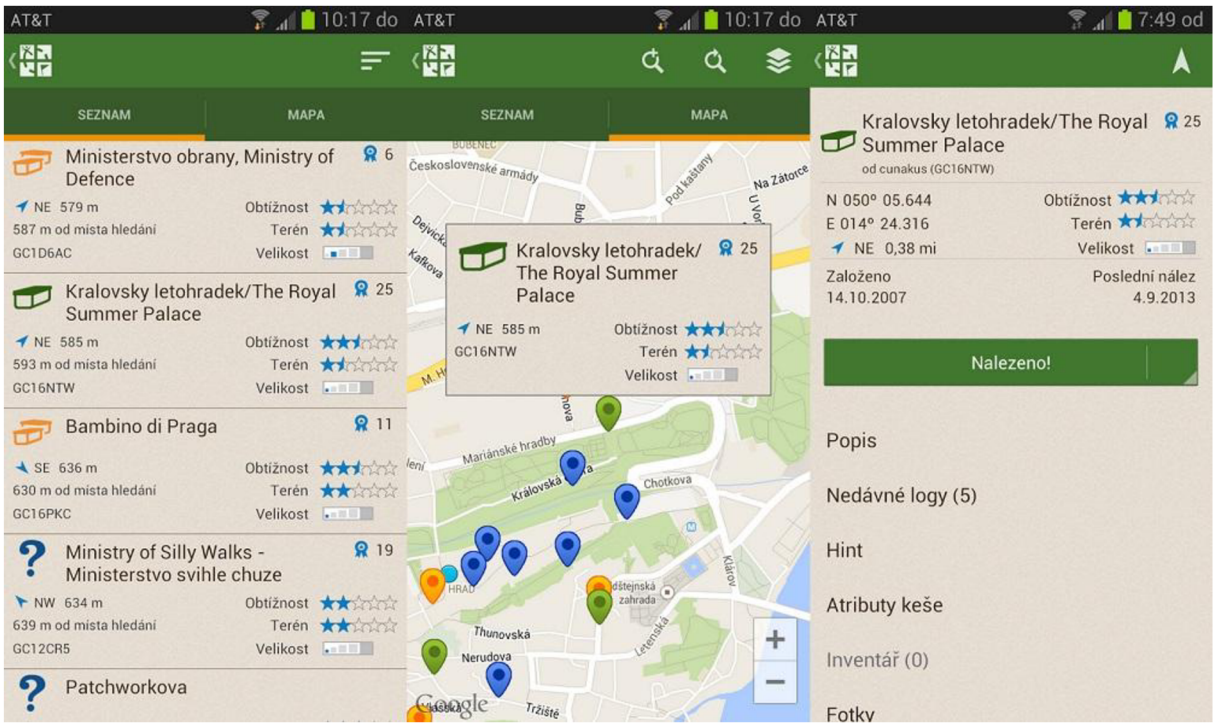
Dle počtu stažení na stránkách Google Play je nejpoužívanější aplikací pro geocaching bezplatná aplikace c:geo.[7] Snímky vybraných obrazovek této aplikace jsou na Obr. 1.6. C:geo obsahuje tzv. Aktivní mapu, která zobrazuje všechny keše ve vybrané oblasti přímo na mapě, takže není potřeba souřadnice zadávat do zařízení ručně. Ovšem k provozu aktivní mapy je potřeba internetové připojení. Pro uživatele bez mobilního internetového připojení je zde možnost stáhnout listingy vybraných keší pro offline použití. Stažený listing obsahuje

vše, co obsahuje webová verze. Od informací o obtížnosti přes popis keše a nápovědu až po samotné logy hráčů. V aplikaci je možno vyhledat keše nejbližší současným GPS souřadnicím nebo je vyhledat ručně podle zadaných GPS souřadnic nebo Adresy. Aplikace také počítá s dohledáváním finálních Multikeší, takže obsahuje funkci zadání souřadnic ručně, dokáže také ke keši navigovat a po jejím nálezu je možné si keš zalogovat.



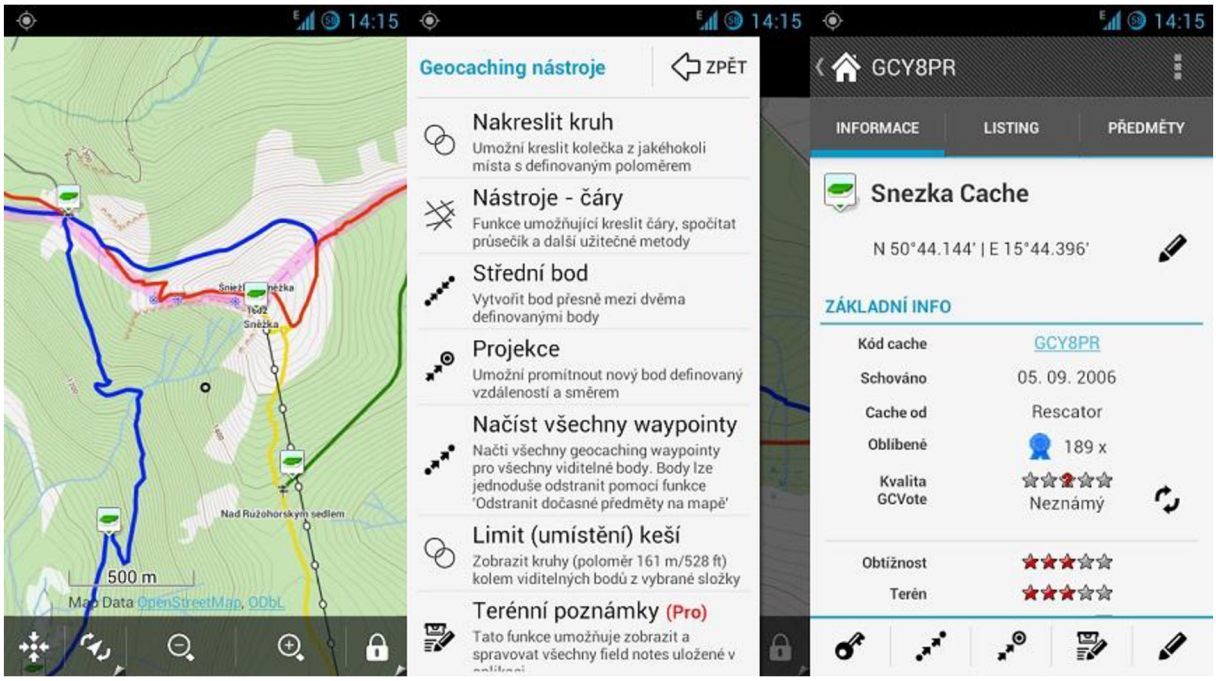
Obr. 1.6 Snímky obrazovek aplikace c:geo

Druhou nejpoužívanější aplikací je oficiální placená mobilní aplikace Geocaching, obsahuje stejnou funkcionalitu jako výše zmíněné c:geo. Snímky obrazovky jsou na Obr. 1.6.



Obr. 1.7 Snímky obrazovek aplikace Geocaching [8]

Třetí nejpoužívanější aplikací a poslední aplikace, kterou v práci stručně zmíním je Locus. Aplikace je dostupná v bezplatné verzi Free a placené verzi Pro. Její výhodou je možnost pracovat s offline vektorovými mapami, spolupráce s aplikací c:geo a implementovanými nástroji pro geocaching. Jako alternativa k c:geo může pro Locus sloužit i addon Geocaching4Locus.



Obr. 1.8 Snímky obrazovek aplikace Locus

2 Čipové karty

První kartu s integrovaným mikroprocesorem (čipem) vyrobila firma Bull v roce 1979, ale masového rozšíření se čipové karty dočkali až v 80. letech 20. století, převážně ve Francii jako platební karty do veřejných telefonů. Čipová karta je plastová destička obsahující integrovaný obvod. Její rozměry jsou 85,6 mm × 54 mm × 0,76 mm, jak udává norma ISO 7816, rozměry jsou tedy totožné s rozměry platební karty. V angličtině je pro ně vžitý název Smart Card neboli chytré karty. Jejich název plyne z jejich širokého využití. [9]

Čipové karty se používají v následujících oblastech:

- identifikace uživatelů - elektronické pasy, řidičské průkazy
- zabezpečení - přístupové karty do budov a elektronických systémů
- bankovnictví - kreditní a debetní karty
- telekomunikace - SIM karty, předplacené karty do veřejných tel. [10][14]

Samotná karta je vlastně takový malý počítač, obsahuje mikroprocesor (CPU), operační paměť (RAM), vstupní a výstupní obvody (I/O) a dále pevnou paměť ROM s nahraným SW od výrobce a paměť pro zápis EEPROM. Některé čipové karty navíc obsahují kryptografický koprocesor, který zpracuje kryptografické výpočty rychleji než obyčejné čipové karty. Díky této architektuře lze čipové karty použít plno různými způsoby.

Karty můžeme rozdělit podle komunikačního rozhraní, na kontaktní, bezkontaktní, hybridní a duální. Kontaktní karty obsahují pozlacenou plošku s vodivými kontakty. Pro čtení této karty je potřeba kartu zastrčit do kontaktní čtečky. V bezkontaktních kartách je zalita cívka / anténa, která zajišťuje komunikaci karty se čtečkou po jejich vzájemném přiblížení. Hybridní karta obsahuje oddělené kontaktní a bezkontaktní rozhraní. Každé rozhraní má svoje vlastní prostředky (CPU, RAM, ROM) ve vlastním čipu.

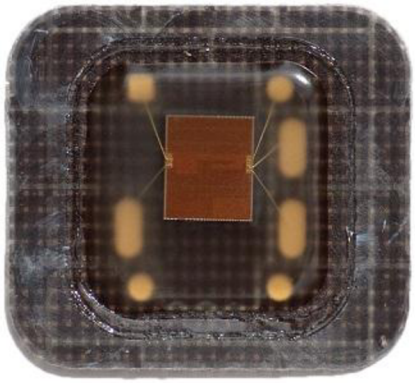
Obr. 2.1 Různé použití čipových karet

Duální karty taktéž obsahují kontaktní i bezkontaktní rozhraní, avšak obsahují jen jeden čip. Kontaktní a bezkontaktní karty si podrobněji popíšeme v kapitolách 2.1 a 2.2. Dále můžeme

karty rozdělit na statické a dynamické. Statické karty umožňují jen zápis a čtení dat z karty, do této kategorie patří většina kryptografických karet, které jsou dnes na trhu. Do dynamických karet lze nahrát programový kód, podle kterého pak karta reaguje. Mezi dynamické karty patří hlavně Java Card, .NET Card a MULTOS. Podrobnější rozbor jednotlivých platforem je v kapitole 2.3. [12][14]

2.1 Kontaktní karty

Mechanické a elektrické vlastnosti kontaktních karet popisuje norma ISO 7816, v normě je mimo jiné popsán i komunikační protokol a předepsané umístění kontaktů na kartě. Kontaktní oblast karty má plochu přibližně $1cm^2$ a je rozdělena do 6 nebo 8 sekcí. Přímo pod kontaktní oblastí se nachází samotný čip karty, viz Obr. 2.2.



Obr. 2.2 Snímek čipu pod kontaktní oblastí [16]

Pro komunikaci s kartou je nutné kartu vložit do čtecího zařízení, tak aby došlo ke spojení vodivých kontaktů. Samotné napájení karty zajišťuje čtecí zařízení. Význam jednotlivých pinu dle normy ISO 7816-2 je znázorněn na Obr. 2.3. Pin VCC se používá k napájení čipu napětím 5V s tolerancí ±10%, u SIM karet do mobilních telefonů se používá 1,8V nebo 3V. Pin RST se používá pro tzv. warm reset. Ke cold resetu karty dojde při odpojení napájení nebo vyjmutím karty ze čtečky. Pinem CLK se do čipu přivádí externí hodinový signál. Pin GND je pin pro uzemnění, tento pin je spojen i se středem kontaktní plochy jak je vidět na obrázku, avšak střed se nijak nevyužívá. Pin VPP je vyžadován pro zápis do paměti u dřívějších smart karet. Pin I/O se používá při výměně dat nebo příkazů v half-duplex modu (V jednom čase jdou data jen jedním směrem). Dva spodní kontakty jsou rezervovány pro budoucí využití. [10][11][13]

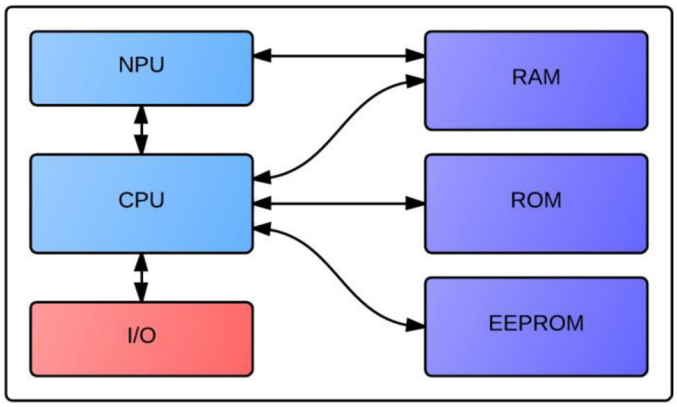
Kontaktní plocha čipových karet:

VCC - napájení čipu
RST - reset signál
CLK - hodinový signál
GND - uzemnění
VPP - programovací vstup
I/O - sériový vstup/výstup pro half duplex

Obr. 2.3 Rozložení pinů

Součásti čipových karet

Hlavní součástí čipových karet je CPU, nejčastější jsou 8 bitové mikroprocesory s kmitočtem 5Mhz. Dále karta obsahuje operační paměť RAM, která slouží k ukládání dočasných dat, se kterými pracuje mikroprocesor. Velikost paměti RAM se pohybuje v řádu jednotek kB. Po odpojení napájení se veškerý obsah paměti RAM vymaže. Další pamětí, kterou čipové karty obsahují, je paměť ROM. Jedná se o nepřepisovatelnou paměť o velikosti desítek maximálně stovek kB, z výroby je zde uložen operační systém karty a testovací procedury.

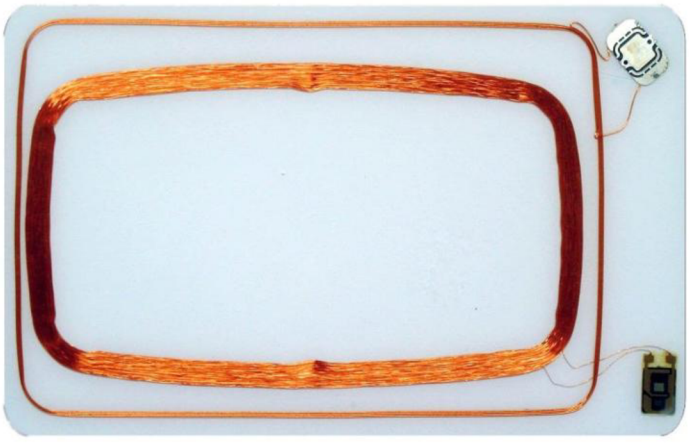


Obr. 2.4 Schéma čipové karty

Poslední pamětí je paměť typu EEPROM (Electrically Erasable Programmable Read Only Memory) data v této paměti nejsou trvalá, je tedy možno je přepisovat a vymazávat. Jsou zde uloženy uživatelská data nebo aplikace. Velikost paměti EEPROM se pohybuje od 2kB do 74kB. Při odpojení napájení nedojde ke ztrátě dat. Nepovinnou součástí čipových karet je numerický koprocesor (NPU). Koprocesor nejčastěji slouží k šifrování dat a komunikace karty kryptografickými algoritmy. Tyto algoritmy jsou výpočetně náročné a koprocesor jejich provedení urychluje. Patří mezi ně například algoritmus RSA (Rivest Shamir Adleman). [9][13][14]

2.2 Bezkontaktní karty

Parametry bezkontaktních karet popisuje norma ISO 14443. Rozměry karty se shodují s rozměry kontaktních karet. Pokud se nejedná o hybridní nebo duální karty, tak bezkontaktní karty neobsahují kontaktní plochu. Na místo kontaktní plochy je v plastovém pouzdru karty zalita cívka, jak je vidět na Obr. 2.5.



Obr. 2.5 Cívky bezkontaktní karty 13,56MHz a 125kHz [17]

Podobnou cívku obsahuje i čtečka. Komunikace karty se čtečkou probíhá na normou předepsané frekvenci 13,56MHz. Nicméně existují i karty inspirované RFID s komunikační frekvencí 125 kHz. Pomocí elektromagnetické indukce cívky ze čtečky je samotná karta i napájena. Vzdálenost potřebná ke komunikaci karty se čtečkou se pohybuje od 2mm do 1m. U smart karet je nejběžnější vzdálenost do 10cm. Čipy bezkontaktních karet mají stejné součásti jako karty kontaktní, tedy CPU, RAM, ROM, EEPROM, jak je popsáno v kapitole 2.1. Použití bezkontaktních karet je pohodlnější a samotné karty jsou i spolehlivější a mají delší životnost než karty kontaktní. Avšak rádiová komunikace činí karty méně bezpečné, jelikož je možno komunikaci se čtečkou zachytit. [9][14][15]

Samotná komunikace čtečky s kartou na úrovni bitů probíhá následovně: Čtečka nepřetržitě generuje harmonický signál, který se po přiblížení karty začne indukovat v její cívce. K cívce je napojen kondenzátor, který se z indukovaného proudu začne dobíjet a tak napájet čip karty. V obvodu je paralelně zapojen čipem ovládaný spínač se zátěžovým odporem.

Pokud karta vysílá úroveň H (bit 1) je spínač rozepnut a zátěžový odpor je odpojen od obvodu cívky. Karta ze čtečky tedy odebírá jen energii pro napájení svého čipu a v cívce čtečky je jen malý proud. Pokud je vysílán stav L (bit 0), tak je spínač sepnut a zátěžový odpor je zapojen do obvodu. Zapojení odporu zvýší celkový odpor karty a v důsledku transformátorové vazby se zvýší i proud v cívce čtečky. Čtečka tedy detekuje změny stavu vysílané kartou pomocí měření proudu ve své cívce. [15]

2.3 Programovatelné karty

V této podkapitole si popíšeme různé platformy programovatelných karet. Jak již bylo zmíněno na začátku druhé kapitoly, dynamické karty dovolují do karty nahrát programový kód, který lze na kartě spustit. To umožňují operační systémy Java Card, .NET Card a MULTOS. Operační systém karty se stará o řízení komunikace mezi aplikací a čipem. Také spravuje souborový systém a poskytuje rozhraní pro práci s kryptografickým HW. Karty obsahují tzv. běhové prostředí (Runtime Environment) a zavaděč. Běhové prostředí se stará o běh aplikací na smart kartě a také o její knihovny. Zavaděč se používá při nahrávání a mazání aplikací z karty. Programovací jazyk se odvíjí od operačního systému použitého na dané smart kartě. Příkazy pro aplikace jsou čtečkou vysílány pomocí APDU (application protocol data unit). [10]

2.3.1 Java Card

Technologie Java Card byla vytvořena v roce 1996 společností Oracle Corporation. Je to otevřená platforma, která umožňuje spouštět aplikace napsané v jazyce Java na čipových kartách. Java Card není závislá na zařízení, je tedy univerzální. Tato univerzálnost umožňuje rychlý vznik a uvedení aplikace do praxe. Je také možné, aby na kartě běželo více aplikací naráz. Svoje bezpečností prvky čerpá ze samotného jazyka Java. Mezi bezpečností prvky patří například analýza před instalací, verifikace kódu, oddělení aplikací (applet firewall), autentizace a užití kryptografických algoritmů. Vzhledem k omezeným možnostem čipových karet, neobsahuje technologie Java Card kompletní možnosti jazyka Java. Chybí zde například podpora pro velké datové typy, řetězce, vícerozměrná pole a vlákna.

Technologie Java Card obsahuje tři základní komponenty, jedná se o Java Card Virtual Machine, Java Card Runtime Enviroment a Java Card API (Application Programming Interface).

- Java Card Virtual Machine (VM) - poskytuje instrukční set pro virtuální stroj Java Card, popisuje podporované množiny jazyka Java a souborové formáty používané k instalaci appletů a knihoven do zařízení podporující technologii Java Card. [18]
- Java Card Runtime Environment (RE) - definuje chování běhového prostředí při jakékoliv implementaci technologie Java Card. Běhové prostředí obsahuje virtuální stroj Java Card, třídy Java Card API a podporované služby, jako je výběr a deaktivování appletů. [18]
- Java Card API - aplikační programové rozhraní Java Card doplňuje specifikaci běhového prostředí a popisuje aplikační programové rozhraní technologie Java Card. Obsahuje definice tříd potřebné pro podporu Java Card VM a Java Card RE. [18]

Kvůli omezeným prostředkům čipových karet je virtuální stroj rozdělen na interpret a konvertor. Interpret je obsažen na kartě a konvertor se nachází v PC nebo terminálu. Aplikace se na kartu dostane následujícím způsobem: nejdříve se zkompiluje kód v jazyce Java, následně proběhne konverze do CAP (converted applet) souboru a jeho kontrola, kterou provádí Off-Card Verifier. Ověřený kód je pak nahrán na smart kartu a poté je již samotný applet na kartu nainstalován. [10][12]

2.3.2 .NET Card

Technologie .NET Card byla vytvořena v roce 2002 na žádost firmy Microsoft. Implementaci .NET Framework pro čipové karty provedla firma HiveMinded. Účelem bylo vytvořit konkurenční platformu pro populární Java Card, postavenou na technologiích Microsoftu. .NET Card má na rozdíl od Microsoft.NET Framework několik omezení: Datové typy nepracují s plovoucí desetinnou čárkou, aplikace nemohou pracovat s více vlákny, nejsou povolena více rozměrná pole a není podporován asynchronní přenos. Taky je zde upravené běhové prostředí CLR (Common Language Runtime) pro řízení běhu aplikací na smart kartě. Prostředí CLR také obsahuje CLI (Common Language Infrastructure), které dovoluje programovat aplikace pro smart karty ve vysokoúrovňových jazycích. Aplikace pro platformu .NET Card lze tedy programovat v jazycích C++, C# nebo Visual Basic.

Technologie .NET Card také umožňuje psát aplikaci v několika programovacích jazycích současně, protože výsledný kód je překompilován do univerzálního bytekódu, který se nazývá CIL (Common Intermediate Language). CIL je platformě nezávislý soubor instrukcí, které mohou být realizovány v jakémkoli prostředí podporující CLI (Common Language Infrastructure).

.NET Card pracuje stejně jako Java Card s virtuálním strojem, aby zamezil nežádoucímu sdílení dat mezi aplikacemi. Každá aplikace uložená na kartě představuje server s běžícími službami. Každá běžící služba má přiděleno své URI (Uniform Resource Identifier). Skrze tyto služby komunikuje klientská aplikace na PC s aplikaci uloženou na kartě.[10][12]

2.3.3 MULTOS

MULTOS je celosvětově hojně využívaná platforma. Její využití najdeme v elektronických dokladech, přístupových systémech nebo bankovnictví. Aplikace pro MULTOS je možné psát v programovacích jazycích C a Java, výsledný kód je překompilován do MULTOS Executable Language (MEL).

MULTOS je platforma založená na maximální bezpečnosti. Její hlavní výhoda spočívá v potřebě schválení a certifikování programového kódu před nahráním aplikace na kartu. To firmě MULTOS umožňuje mít maximální kontrolu nad svojí platformou. Průběh certifikace vypadá následovně: nejdříve se vytvořená aplikace přeloží pomocí aplikace SmartDeck Debugger, při tomto procesu vznikne soubor s příponou *.alu*. Pomocí tohoto souboru a programu Application Registration File Generator vytvoříme další soubor s příponou *.aif*. Tento soubor již nahrajeme na certifikační server www.stepxpress.com, který nám vygeneruje následující certifikáty: Application Load Certificate (ALC) a Application Delete Certificate (ADC). První certifikát se použije při nahrání aplikace na kartu, druhý při jejím odstranění.

Druhým krokem zabezpečení MULTOS karet je striktní oddělení aplikací na kartě. Každá aplikace má při nahrávání přesně vyčleněn svůj prostor, který může využívat a do kterého nemohou zasahovat ostatní aplikace. Neoprávněné zásahy do prostoru cizích aplikaci hlídá firewall, při detekci takového zásahu firewall problémovou aplikaci ukončí. Samotná oblast určená pro jednu aplikaci je rozdělena na kódovou a datovou oblast. V kódové oblasti je uložen kód aplikace. Z této oblasti není možné číst ani do ní zapisovat, je možné z ní jen spustit uloženou aplikaci. V datové oblasti jsou uložena aplikační data, je možno z ní číst i do ní zapisovat. Do datové oblasti patří statická paměť uložená v EEPROM a dynamická uložená v RAM. [12][19]

2.4 Bezpečnost čipových karet

Čipové karty jsou navrhovány tak, aby se zajistila co největší bezpečnost a ochrana dat uložených na kartě. Nicméně jako u většiny systému se ani při návrhu čipových karet a jejich OS mohou vyskytnout chyby, které mohou být po nalezení zneužity k ovládnutí karty nebo dat na ní uložených. Žádný návrh tedy není dokonalý a může být zneužit potencionálními útočníky. Při využití nemalých prostředku, pokročilých znalostí o systému a dostatečně dlouhé doby, je možné prolomit ochranu každého systému a manipulovat s jeho daty. Útočníci si většinou vybírají cíle a metody útoku, při kterých není zapotřebí velkého úsilí nebo nákladů. Motivem útočníků může být odcizení informací s vysokou hodnotou nebo snaha se zviditelnit. Informace z karty mohou sloužit ke krádeži finančních zdrojů uživatele, krádeži identity nebo mohou mít velkou hodnotu pro konkurenční společnosti. I samotná informace o možnosti prolomení systému, může způsobit velké problémy společnosti, která produkt vyrobila.

Nejdůležitějším prvkem ochrany je již návrh čipových karet, který by měli provádět jen prověřené a důvěryhodné osoby. Toto opatření zabrání v zanesení škodlivého kódu nebo „zadních vrátek“ do výsledného návrhu. Vhodné je také zveřejnění nových kryptografických technik, před jejich implementací. Při nezávislém průzkumu veřejností, může dojít k odhalení drobných nedostatků, které mohou být napraveny ještě před uvedením výsledného návrhu a

tím zvýšení jeho bezpečnosti. Při návrhu je také vhodné myslet na to, aby jeden úspěšný pokus o prolomení nezkompromitoval celý systém. Bezpečnostním prvkem může být i častější obměna čipových karet za karty s implementovanými novými bezpečnostními prvky a větším kryptografickým výkonem. Při využití nejnovějších technik je možné po dobu životnosti karty zajistit maximální ochranu a minimalizovat tak šance na prolomení jejich ochran.

Útoky na čipové karty lze zařadit do několika kategorií. Řadí se mezi ně fyzické útoky, logické útoky a útoky postraními kanály. Při fyzických útocích dochází k přímému napojení na čip karty a je k němu tedy potřeba množství technického vybavení. Fyzická analýza čipu se dělí na statickou a dynamickou. Při statické analýze je analyzován čip, který není v provozu. Při dynamické analýze je naopak čip v provozním stavu a zpracovává mnoho informací. Při tomto druhu analýzy je potřeba, aby útočník vlastnil velmi rychlé záznamové zařízení, které dokáže všechny tyto informace zpracovat. Logické útoky jsou při útoku na čipové karty v dnešní době nejpoužívanější. Jedná se o nedestruktivní metody, při kterých není potřeba manipulace s čipem karty. K provedení takového útoku také není potřeba žádného drahého vybavení, stačí čtečka karet a PC. Tyto útoky se zaměřují na softwarové chyby v operačním systému a dešifrování dat na kartě. Poslední kategorií jsou útoky postraními kanály. Při tomto typu útoku se k získání informaci používá metod sledování vnějšího chování karty při zpracování dat. Patří mezi ně časová analýza, napěťová proudová analýza nebo Elektromagnetická analýza. Jednotlivé formy útoků si popíšeme v několika následujících podkapitolách. [14][20]

2.4.1 Fyzické útoky

Fyzické útoky se zaměřují na manipulaci s čipem karty. Útoky se konkrétně zaměřují na procesor, paměť nebo datovou sběrnici. Při těchto útocích se používá reverzního inženýrství a různých modifikací čipu. Protože je k těmto technikám potřeba kvalitní vybavení, jsou tyto útoky finančně velice nákladné. Mezi zařízení potřebné k těmto útokům patří např. elektronový mikroskop, laserový nůž, mikromanipulátory nebo výkonné osciloskopy a počítače. Před samotným útokem se nejprve čip z karty vyjme nebo vyřízne nožem. Dále je nutné z čipu odstranit epoxidovou pryskyřici a čip očistit (čip zalitý epoxidovou pryskyřicí je na Obr. 2.2). Na takto připravený čip je již možné provést útok. Útoky můžeme rozdělit na aktivní a pasivní. Aktivní útoky jsou destruktivní. Manipuluje přímo se spoji na čipu. Mezi aktivní útoky patří:

- Odstranění pasivační vrstvy a přerušení spojů na čipu pomocí laserového nože
- Oklamání senzorů testujících přítomnost pasivační vrstvy pomocí suchého leptání
- Vytvoření nových spojů na čipu za pomocí iontového paprsku

Pasivní útoky jsou naopak nedestruktivní a řadí se mezi ně různé druhy sledování. Jsou to například:

- elektrické sledování datového toku na sběrnici pomocí mikroskopických jehel
- sledování paměťových buněk nebo signálu na sběrnici pomocí elektronového mikroskopu
- sledování tranzistorů v čipu pomoci spodního rentgenování s vlnovou délkou, pro kterou je křemíkový substrát průhledný

Pro lepší pozorování čipu může útočník snížit taktovací frekvenci procesoru. Snížením taktovací frekvence dojde ke zpomalení veškerých operací čipu a budou tak lépe detekovatelné změny stavů.

Aby se zamezilo útokům, jsou čipy chráněny množstvím ochran. Ochrany se dělí na pasivní a aktivní. Pasivní ochrany jsou založeny na technologiích výroby polovodičů, jedná se o vrstvy bránící snadnému přístupu k čipu a jeho obvodům. Tyto pasivační vrstvy musí útočník před samotným útokem odstranit, což pro útočníky vybavené laserovým nožem není problém. Mezi další pasivní obranné mechanizmy patří i součásti bránící různým typům analýz. Mezi aktivní ochrany můžeme zařadit různé typy senzorů, které jsou přímo integrovány na křemíkovém čipu. Patří mezi ně např. senzory světla, tepla nebo UV záření. O zpracování a vyhodnocení informací ze senzorů se stará operační systém čipu. V případě narušení těchto senzorů jsou vymazána veškerá citlivá data. OS může také monitorovat taktovací frekvenci a zamezit provádění výpočtů pokud se frekvence liší od té tovární. Nicméně všechny aktivní prvky ochrany jsou závislé na napájení čipu a správném fungování operačního systému. [14][20]

2.4.2 Logické útoky

Logické útoky spočívají v nalezení softwarových chyb v operačním systému karty, a jejich zneužití pro prolomení jeho ochrany. Jedná se o nedestruktivní metody, při kterých není potřeba manipulace s čipem karty. Softwarové chyby zneužívané při těchto útocích mohou být chyby přehlédnuté při bezpečnostních testech, nebo při normálním používání. Může se jednat i o nekorektní reakce karty na chybné příkazy nebo chyby v kryptografických algoritmech, které není lehké odhalit. Zneužití těchto chyb pak může vést ke zmatení karty, která následně odhalí citlivé informace nebo k otevření cesty pro nahrání škodlivého kódu na kartu. Škodlivé aplikace pak mohou používat operace, ke kterým je za normálních okolností potřeba zvláštní oprávnění. Logické útoky se v dnešní době řadí mezi nejčastější útoky na čipové karty. Je to dáno tím, že k provedení takového útoku je zpravidla zapotřebí jen počítače a čtečky čipových karet. Tyto útoky jsou sami o sobě jen málo úspěšné, ale v kombinaci s fyzickým útokem mohou být více nebezpečné. [14][20]

Mezi logické útoky patří:

- Zneužití objevených chyb v OS
- Zneužití chyb transportního protokolu
- Objevení neplatných a nepovolených požadavků
- Objevení skrytých příkazu
- Průzkum souborového systému
- Získání tajné informace kryptografickou analýzou
- Použití nelegálních aplikací využívajících nepovolených instrukcí nebo parametrů

2.4.3 Útoky postraními kanály

Principem útoku na postraní kanály je skutečnost, že výstupní chování karty je závislé na právě zpracovávaných datech. Za pomocí různých analýz je tedy možné z tohoto chování získat citlivé informace. Postraní kanál je nežádoucí výměna informací mezi kartou a jejím okolím. Postraním kanálem může být jakákoliv fyzikální veličina, kterou je možné změřit

v okolí karty nebo na kartě. Může se jednat například o měření spotřeby karty nebo měření času potřebného k provedení různých operací. Analýzou takto naměřených dat je pak možné zrekonstruovat zpracovávaná data. Mezi nejpoužívanější typy analýz se řadí již zmíněná časová a napěťově proudová analýza měřící spotřebu. Napěťová analýza byla v posledních letech výrazně rozšířena. Mezi další typy analýz můžeme zařadit elektromagnetickou analýzu, která se zabývá získáním informací z elektromagnetických vlastností karty při její činnosti. Dalším druhem úniku dat postraním kanálem je úmyslné zavádění chyb. Analýzou chybných výpočtů je možné získat informace o systému karty. Z útoků postraními kanály se v posledních letech stala vážná hrozba pro zabezpečení karet. Důvodem může být i skutečnost, že útoky mohou být úspěšné už při použití pouhého zlomku uniklých informací. Ovšem vytvoření protiopatření proti takovým útokům nepředstavuje snadný úkol. K tvorbě účinných protiopatření je potřeba rozsáhlých znalostí HW a pochopení principu těchto útoků. V současné době se v čipových kartách používají procesory s několika obrannými mechanizmy proti takovým to útokům. Dále stručně popíšu jednotlivé druhy útoků postraními kanály. [20]

Časová analýza

Časová analýza vychází z monitorování doby výpočtu mezi odesláním příkazu čtečkou a odpovědí od čipové karty. Jelikož je v některých případech doba tohoto výpočtu závislá na tajném klíči, je možné za pomocí této časové informace klíč odhalit. Všechny moderní karty jsou proti tomuto útoku odolné. Jejich obrana spočívá v použití složitějších kryptografických algoritmů, u kterých není šifrovací a dešifrovací čas závislý na tajném klíči. Tyto algoritmy jsou navrženy tak, aby byla délka cesty přes algoritmus stejná pro různé kombinace textu a klíčů. Algoritmus nejprve zjistí nejdelší cestu a následně všechny ostatní cesty upraví tak, aby odpovídali délce nejdelší cesty. [20]

Napěťově proudová analýza

Tento druh analýzy využívá informací z měření spotřeby elektrické energie v průběhu vykonávání kryptografického výpočtu na kartě. Analýzou těchto informací můžeme odhalit prováděné operace, které mohou vést k objevení tajného klíče. Spotřebu karty není obtížné změřit, protože většina karet je napájena přímo ze čtečky. Napěťově proudovou analýzu můžeme rozdělit do dvou skupin a to na jednoduchou a diferenciální. Při jednoduché napěťově proudové analýze se měří velikosti proudu na rezistoru zapojeného v sérii s napájecím napětím. Průběh velikostí proudu je zaznamenáván A/D převodníkem a následně ukládán do počítače. Aby byl průběh zaznamenán dostatečně detailně je potřeba, aby A/D převodník vzorkoval hodnoty s frekvenci až desítek Mhz. Aby byl průběh zaznamenán dostatečně detailně je potřeba, aby A/D převodník vzorkoval hodnoty s frekvenci až desítek Mhz. Naměřené hodnoty jsou následně zaneseny do grafu. Zkušený krypto-analytik může z charakteristických průběhů grafu odvodit informace o provedených instrukcích nebo použitém klíči. Tyto informace pak mohou zjednodušit útok hrubou silou. Diferenciální napěťově proudová analýza využívá k odhalení tajného klíče statistických metod. Tato metoda je výkonnější než předešlá metoda. Výkonnost této metody plyne z toho, že je možné ji automatizovat. Před analýzou je ale nutné změřit velké množství dat. Tyto data pak slouží jako podklady pro odfiltrování šumu a k využití statistických funkcí pro korekci chyb, které nám odhalí provedené operace. [20]

Elektromagnetická analýza

Elektromagnetická analýza využívá pole vznikajícího kolem čipu karty. Logické změny stavů v čipu generují krátké elektrické impulzy, které ve svém okolí způsobí změnu elektromagnetického pole. Tyto elektromagnetické změny lze zaznamenat impulzní sondou. Analýzu naměřených elektromagnetických změn je možné provést stejným způsobem jako u napěťově proudové analýzy. Nevýhodou elektromagnetické analýzy je vyšší úroveň šumu ve srovnání s předešlými metodami měřenými přímo přes napájecí vedení. Další nevýhodou je i skutečnost, že moderní karty jsou obvykle chráněny kokovými mřížkami, které snižují úroveň vyzařované energie. [20]

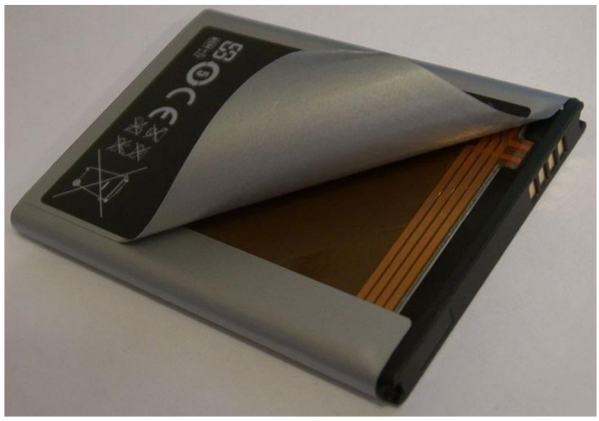
Útok zaváděním chyb

Při útoku zaváděním chyb se útočník snaží různými technikami přimět čip, aby prováděl chybné výpočty. Analýzou těchto chyb se pak snaží zjistit nějaké informace o systému. Útoky zaváděním chyb často vyházejí z norem pro výrobu a funkci karet. Útočník se snaží přimět kartu pracovat v podmínkách těsně za hranicí normy, tak aby byla funkční ale vykazovala chyby. Typicky se jedná například o změnu napětí o ±10 % od hodnot udávaných normou. Dalším příkladem může být změna taktovací frekvence těsně nad mez udávanou normou. Útočník může kartu také vystavit nízkým nebo vysokým teplotám, které jsou mimo rozsah obvyklých pracovních teplot. Pro ovlivnění karty se také můžou použít různé druhy záření. Mikrovlnné záření může ovlivnit chyby výpočtu a UV záření může vymazat paměťové buňky EEPROM a flash pamětí. [20]

3 Technologie NFC

Near Field Communication je technologie bezdrátové komunikace s přenosem dat na velmi krátkou vzdálenost. Pracuje na frekvenci 13,56Mhz a její přenosové rychlosti se pohybují od 106kbit/s do 424kbit/s. Tato technologie vychází ze standardů smart karet a RFID ISO/IEC 14443 a FeliCa. NFC je převážně určeno pro použití v přenosných zařízeních jakou jsou mobilní telefony, tablety nebo přenosné čtecí terminály. Přenosová vzdálenost této technologie je teoreticky až 10cm, ale v praxi NFC reaguje obvykle v rozmezí 1 - 4 cm. Standard NFC byl schválen v roce 2003, následující rok založily firmy Nokia, Philips a Sony neziskovou organizaci NFC Fórum, která navrhuje standardy pro NFC. Počáteční specifikace byli vytvořeni v roce 2006 a už ve stejném roce uvedla společnost Nokia první mobilní telefon s integrovaným NFC. NFC v té době ale nemělo velký úspěch, převážně z důvodů malé podpory trhu a nákladnosti její implementace. V posledních několika letech se ale technologie NFC stává stále více populární díky rychlému nástupu chytrých telefonu a velké podpoře společnosti Google, která NFC integrovala do svého mobilního operačního systému Android v roce 2010. V telefonech se zabudovaným NFC je většinou NFC anténa přilepena zevnitř zadního krytu přístroje (Obr. 3.1). Možné je také zabudování antény do baterie, jak je vidět na Obr. 3.2.

Obr. 3.1 NFC anténa na zadním krytu telefonu Samsung Nexus S



Obr. 3.2 NFC anténa zabudovaná v baterii telefonu Samsung Galaxy Nexus [23]

Do telefonů bez NFC je možné tuto technologii dostat v podobě upravené microSD nebo SIM karty. Nevýhodou takového řešení ale je, že většina telefonů má tyto karty uloženy v kovovém rámečku nebo pod baterií, takže je většina signálu odstíněna ještě než se dostane mimo zařízení. [21][22]

3.1 Použití NFC

NFC slouží v první řadě jako náhrada různých čipových karet a čtecích zařízení. Použití NFC můžeme rozdělit do následujících skupin:

Platby

- provádění plateb u bezkontaktních terminálů v obchodech
- placení jízdenek v městské hromadné dopravě
- rychlé placení vstupenek do zábavních parků, ZOO apod.
- placení parkovného na parkovištích s přesným odečtením času při odjezdu

Výměna informací

- jednoduché spárování Wi-Fi nebo Bluetooth dotykem dvou zařízení
- sdílení kontaktu, webových odkazů a dalšího obsahu

Přístup

- odemykání domu nebo hotelového pokoje
- odemykání automobilu a jiných dopravních prostředků
- přístup do osobního nebo pracovního počítače
- jednoduché připojení k místní Wi-Fi síti
- přístup do vyhrazených prostor v práci nebo škole

Identifikace

- náhrada občasného, řidičského nebo jiného identifikačního dokladu

Usnadnění při použití NFC štítků (tagů)

- otevření webových stránek přiložením zařízení k chytrému plakátu
- vyhledání ve štítku uložených GPS souřadnic na mapě
- získání doplňkových informací na vhodně vybavených informačních místech
- spuštění nadefinované akce v mobilním telefonu přiblížením k NFC štítku

Ze seznamu je zřetelné, že použití NFC značné usnadňuje práci uživateli a snaží se o centralizování jeho přístupových a platebních údajů do jednoho zařízení. Dále bych rád upřesnil některé možnosti použití z výše uvedeného seznamu.[22]

3.1.1 Platby

Mezi nejdůležitější možnosti použití NFC patří bezkontaktní platby. K provedení takové platby je potřeba, aby byla v telefonu nahraná aplikace pro správu debetních a kreditních karet. Patří mezi ně například aplikace Google Wallet nebo O2 Wallet. V ČR je současně nejrozšířenější aplikace pro NFC platby právě O2 Wallet. Jak napovídá název, jedná se o aplikaci mobilního operátora O2, který je v současné době nejaktivnější v zpřístupnění NFC plateb pro své zákazníky. Testovací provoz letos (2013) zahájil i mobilní operátor T-Mobile ve spolupráci s ČSOB. Všichni velcí mobilní operátoři v současnosti pracují na projektu Czech Wallet, který by měl sjednotit aplikaci pro všechny operátory a umožnil by v této

aplikaci spravovat platební, věrnostních a dopravní karty uživatele. Testovací provoz Czech Wallet by měl začít v létě 2014. Další text se bude vztahovat pouze k použití O2 Wallet, ale použití ostatních aplikací bude velmi podobné. Aby mohl zákazních použít NFC platby u O2 je potřeba aby vlastnil jeden z osmi podporovaných telefonů, byl zákazníkem u Komerční banky nebo GE Money Bank a nechal si od O2 vyměnit svojí SIM kartu za SIM s integrovaným ochranným prvkem (secure element) ve kterém je emulována platební karta.

Obr. 3.3 Bezkontaktní platba telefonem s NFC [24]

Aplikace O2 Wallet nabízí automatický a manuální režim placení. V manuálním režimu je nutné spustit aplikaci, vybrat metodu „Platba se zadáním PIN“, zadat PIN a až pak je možné přiložit telefon k terminálu a zaplatit. Automatický režim naopak funguje, i když je telefon zamčený a v režimu spánku. Pro platby do 500 Kč s automatickým režimem není nutné zadávat PIN, při platbě nad 500 Kč je PIN vyžadován vždy. Bezkontaktní platby s telefonem vybaveným NFC se tedy velmi podobají platbám s bezkontaktními kartami. [22][25][26]

3.1.2 Přenos dat

Dalším bodem ze seznamu, který bych chtěl podrobněji probrat je přenos dat přes NFC. NFC vychází z technologie RFID, která umí přenášet jen ASCII znaky nebo identifikační číslo (URI - Uniform Resource Identifier). Aby bylo možné posílat skrze NFC jakákoliv data (např. obrázky), zavedla organizace NFC Fórum standard NDEF (NFC Data Exchange Format), který definuje formát zapouzdření zpráv pro přenos dat mezi NFC zařízeními. NDEF sice umožňuje přenos jakýchkoli dat, ale vzhledem k rychlosti NFC (106 - 424 kbit/s) je častější použití NFC jen jako prostředníka ke spárování dvou zařízení pomoci rychlejších bezdrátových technologií jako je Bluetooth (až 24Mbit/s) nebo Wi-Fi Direct (až 300Mbit/s), přes které se následně data posílají.[22]

3.1.3 Použití NFC štítků

Dále bych chtěl přiblížit použití NFC štítků (tagů). NFC štítky se velice podobají RFID štítkům, které se používají v obchodech pro identifikaci zboží. NFC / RFID štítek je znázorněny na Obr. 3.4. Jedná se převážně o pasivní prvky, které jsou napájeny stejným způsobem jako bezkontaktní karty. S tím rozdílem, že čtečku nahrazuje zařízení vybavené

NFC. Přiblížením NFC zařízení ke štítku se v cívce štítku začne indukovat proud, který napájí jeho čip. Ihned po získání napájení štítek odešle do NFC zařízení v sobě uloženou informaci. Může se jednat o odkaz na webovou stránku, GPS souřadnice nebo prostý text.

Obr. 3.4 NFC / RFID štítek [27]

Tyto štítky mohou být přilepeny na plakátech nebo pohlednicích. Přiložením telefonu k NFC štítku může být uživatel jednoduše a rychle odkázán na webové stránky události nebo si prohlédnout v mapové aplikaci místo odkud byl odeslán pohled. Pomocí různých aplikací lze taky navolit reakce telefonu při přiblížení ke konkrétnímu štítku. Štítky mohou být nalepeny na různých místech a uživatel si k těmto štítkům nadefinuje svoje akce. Jako příklady mohu uvést:

- štítek na okně spustí aplikaci s informacemi o aktuálním počasí
- štítek na rádiu spustí aplikaci na rozpoznání hudby
- štítek na nočním stolku zapne tichý režim a nastaví budík na 7.00
- štítek na držáku telefonu v autě zapne GPS, navigační aplikaci a spáruje palubní handsfree s telefonem pomocí Bluetooth

3.2 Režimy přenosu

3.2.1 Emulace karty (Card emulation)

Jak již napovídá název, v tomto režimu dochází k emulaci karty na NFC zařízení. V tomto režimu se zařízení s NFC chová jako běžná bezkontaktní karta, která reaguje jen na příkazy od čtečky. NFC v tomto režimu tak lze použít jako debetní nebo kreditní karta. Protože se pracuje s citlivými informacemi, je přenos a uložení dat u tohoto režimu zabezpečen. Kartu bylo do nedávna nutné emulovat v HW čipu (secure element), ale společnost Google uvedla s novou verzí 4.4 svého operačního systému Android funkci nazývanou Host-based Card Emulation, která dovoluje bezpečné emulování karty jen za pomocí SW. [22][28]

3.2.2 Rovný s rovným (Peer-to-peer)

Režim Peer-to-peer slouží pro komunikaci mezi dvěma aktivními zařízeními. Umožňuje obousměrnou komunikaci v režimu half-duplex, vždy tedy vysílá jen jedno zařízení a druhé naslouchá. Zařízení se ve vysílání a naslouchání rychle střídají, takže navenek vypadají jako by komunikovali současně. Maximální rychlosti přenosu v tomto režimu dosahují 424 kbit/s.[22]

3.2.3 Čtení/zápis (Read/Write)

Režim Čtení/zápis se používá při čtení a zápisu do NFC štítků. Je založen na standardech ISO/IEC 14443 a FeliCa. Protože ve štítcích se většinou nenacházejí citlivé informace, není potřebné, aby byl tento přenos zabezpečen. Maximální rychlosti přenosu v tomto režimu dosahují 106 kbit/s. [22]

3.3 Bezpečnost NFC

Technologie NFC není v základu nijak zabezpečená a spoléhá na zabezpečení komunikace na vyšších vrstvách v klientských aplikacích. Jediným zabezpečením definovaným ve standardu je zabezpečení peer-to-peer režimu NFC-SEC, mezi dvěma aktivními zařízeními. Toto zabezpečení umožňuje bezpečnou výměnu klíčů a sestavení zabezpečeného kanálu. Režimy čtení/zápis a emulace karty jsou nezabezpečeny a jediným bezpečnostním prvkem je tak velmi krátká komunikační vzdálenost a malé výkony komunikujících zařízení. Vzhledem k malé komunikační vzdálenosti a výkonům vysílání je odposlech nebo narušení komunikace obtížné. Dále uvedu různé možnosti útoku na NFC zařízení. [10][21][22]

3.3.1 Odposlech

I přes krátkou komunikační vzdálenost NFC je možné tuto komunikaci odposlechnout. K odposlechu musí útočník vlastnit anténu, zesilovač a zařízení umožňující dekódování komunikace zúčastněných NFC zařízení. Úspěšnost takového útoku je závislá hlavně na kvalitě útočníkova vybavení a vysílacího výkonu odposlouchávaných zařízení. Odposlech můžeme rozdělit do dvou kategorií a to na odposlech aktivních a pasivních prvků. Odposlech aktivních prvku je jednodušší, protože mají vlastní napájení a jejich vysílací výkon je větší. To umožňuje odposlech až ze vzdálenosti 10 metrů. U pasivních prvků je odposlech složitější, protože vysílají odpověď jen v přítomnosti čtecího zařízení, které jim dodává energii. Takto naindukovaná energie umožňuje vyslat odpověď jen s nízkým ziskem a tím se vzdálenost odposlechu snižuje na 1 metr. [21][22]

3.3.2 Poškození vysílaných dat

Principem poškození vysílaných dat je rušení komunikace na frekvenci 13,56Mhz se zařízením s vyšším výkonem. Tímto způsobem bude docházet k narušení vysílaných dat a špatnému vyhodnocení dat na straně příjemce. [22]

3.3.3 Modifikace vysílaných dat

Podobným způsobem probíhá i modifikace dat, kdy se útočník snaží pozměnit vysílaná data tak, aby se jevila jako platná. Modifikace dat je podstatně složitější než pouhé rušení. Pro takovou modifikaci je potřeba, aby se jednotlivé bity radiového signálu změnili v přesně daný okamžik, a aby útočník použil vyšší vysílací výkon než původní zařízení. Úspěšnost záměny bitů je závislá na hloubce amplitudové modulace a použitém kódování. [21][22]

3.3.4 Přepojování

Při tomto typu útoku se útočník snaží, aby veškerá komunikace mezi komunikujícími zařízení procházela přes útočníkovo zařízení. Útočník při útoku ruší přímou komunikaci mezi

zařízeními a veškeré zachycené požadavky a odpovědi přeposílá pomocí svého zařízení. Útočník může při přeposlání data jen odposlouchávat nebo je i modifikovat. Nevýhoda tohoto útoku spočívá v tom, že zařízení mohou detekovat rušení a komunikaci ukončit. [22]

3.3.5 Navázání komunikace na neuzavřený komunikační kanál

Tento typ útoku zneužívá nekorektního uzavření komunikačního kanálu po skončení komunikace. Pokud není komunikační kanál uzavřen a u čtečky se nenachází žádné vysílací zařízení, čtečka spustí časovač, který po uplynutí uzavře komunikační kanál a začne vyžadovat autentizaci k přístupu k němu. Pokud se útočníkovi podaří navázat komunikaci ještě před uplynutím této doby, získá tím přístup ke kanálu bez autentizace. [22]

3.3.6 Opakované přenášení odposlechnutých autentizačních dat

Při tomto typu útoku, muže útočník odposlechnout zahájení komunikace mezi dvěma zařízeními a tuto komunikaci si uložit. Útočník pak může tuto komunikaci bez jakékoliv znalosti opakovaně odeslat a vydávat se tak za původní zařízení. [22]

4 Kryptografie

Kryptografie je věda zabývající se problematikou utajování dat pomocí různých šifrovacích technik. Cílem kryptografie je utajit zprávu pomocí šifrování tak, aby jí mohl přečíst jen vlastník tajného klíče. Pro nepovolané osoby je tak zpráva nečitelná. Zpráva je zašifrována pomocí šifrovacího klíče a příslušného šifrovacího algoritmu. Výstupem této operace je zašifrovaná zpráva, která se nazývá kryptogram. Příjemce pro rozluštění původní zprávy potřebuje zašifrovanou zprávu (kryptogram) a dešifrovací klíč, dešifrovací algoritmus následně kryptogram pomocí dešifrovacího klíče rozluští do podoby původní zprávy. Techniky šifrování jsou založené na časové náročnosti hledání řešení nebo na obtížnosti řešení matematických problémů, kdy je velice těžké odhalit šifrovací (dešifrovací) klíč v dostatečně krátkém čase na současných výpočetních prostředcích. Současné algoritmy jsou postaveny tak, aby nebylo možné ze znalosti algoritmu získat zašifrovanou zprávu, není tedy nutné utajovat samotné algoritmy, stačí utajit jen šifrovací klíče. Kryptografie umožňuje přenášet citlivé informace přes nezabezpečené sítě nebo je ukládat v zabezpečené formě. Používá se k ochraně dat na lokálním disku, k digitálnímu podpisu elektronické pošty nebo pro přenos citlivých firemních informací přes internet. S velice dynamickým rozvojem informační techniky je stále více kladen důraz na vývoj rychlejších a spolehlivějších šifrovacích algoritmů pro síťový provoz, kde je kladen důraz právě na rychlost šifrování a dešifrování dat. S rostoucím výkonem výpočetní techniky se ale zároveň zkracuje doba potřebná pro rozluštění zašifrované zprávy, proto jsou vyvíjeny stále silnější algoritmy s delšími klíči, které mají nahradit starší méně bezpečné algoritmy. Techniky současné kryptografie můžeme rozdělit do dvou kategorií a to Symetrická kryptografie a Asymetrická kryptografie. Rozdílem mezi těmito technikami je způsob použití šifrovacích a dešifrovacích klíčů. Symetrická kryptografie používá pro šifrování i dešifrování stejný klíč, zato Asymetrická kryptografie používá pro šifrování a dešifrování různé klíče. Tyto techniky si podrobněji popíšeme v podkapitolách 4.2 a 4.3. [29][30][31]

4.1 Základní pojmy

4.1.1 Kryptoanalýza

Kryptoanalýza se snaží získat zašifrovanou zprávu z kryptogramu bez znalosti dešifrovacího klíče. Mezi nejčastější kryptoanalytické metody patří:

Útok hrubou silou - při tomto typu útoku se útočník snaží prolomit šifrování systematickým zkoušením všech možných variant klíčů. Z principu funkce algoritmu se jedná o zdlouhavou metodu, kde je zásadní délka použitého klíče. S delším klíčem se i prodlužuje doba potřebná pro vyzkoušení všech kombinací. Použití tohoto typu útoku je v dnešní době velmi časté, a to hlavně díky rostoucímu výkonu výpočetní techniky.

Lineární kryptoanalýza - před provedením tohoto útoku je potřeba, aby útočník získal několik zpráv zašifrovaných stejným šifrovacím algoritmem a klíčem. Analýzou těchto zašifrovaných zpráv a jejich aproximací lineární funkcí původní zprávy, může dojít k prolomení šifry.

Diferenciální kryptoanalýza - cílem této metody je zjistit klíč za pomocí mnoha šifrovaných zpráv, u kterých známe i původní text. Při analýze se vyberou páry zašifrovaného textu, které vykazují rozdíly oproti původnímu textu. Analýzou velkého počtu párů lze zjistit šifrovací klíč. [31]

4.1.2 Hashovací funkce

Hashovací funkce je jednocestná kompresní funkce, která vytvoří ze zprávy s libovolnou délkou krátký řetězec s pevnou délkou. Tomuto řetězci se říká digitální otisk a jeho délka je nejčastěji 128, 256 nebo 512 bitů. Hashovací funkce se zapisuje: h = H (Z), kde h je výsledná hash, H( ) je hashovací funkce a Z je vstupní zpráva. Hashovací funkce je důležitým bezpečnostním nástrojem, protože zajišťuje ověření autentičnosti a integrity digitálních dat. Postup ověření je následující: Uživatel příjme zašifrovanou zprávu s její hashí h. Tuto zprávu dešifruje a z dešifrované zprávy vypočte hash h'. Pokud se h = h' tak je zpráva autentická a nebyla nijak upravena. Hasovací funkce by měla splňovat několik požadavků:

- pro stejnou vstupní zprávu musí vždy vypočítat stejnou hash
- nesmí umožňovat výpočet původní zprávy z hashe
- hash by měla být pro každou vstupní zprávu unikátní, aby nedošlo ke kolizi hashí

Mezi nejběžněji používané hashovací algoritmy patří MD5 (128b), SHA-1 (160b) a SHA-2 (512b). [12][29][31]

4.1.3 Digitální podpis

Digitální podpis (též elektronický podpis) se používá pro ověření pravosti elektronických dokumentů a dat. Digitálně je možné podepsat elektronické dokumenty, zprávy, přístupová práva nebo další libovolná data. K digitálnímu podpisu se používá asymetrická kryptografie a hashovací funkce. Elektronická data lze digitálním podpisem podepsat dvěma způsoby. První způsob využívá jen asymetrické kryptografie se soukromým a veřejným klíčem. Odesílatel přiloží k odesílanému elektronickému dokumentu zprávu zašifrovanou pomocí svého soukromého klíče. Na obsahu této zprávy se domluví odesílatel s příjemcem, tato zpráva pak slouží jako digitální podpis. Příjemce si může jeho platnost ověřit dešifrováním zašifrované zprávy pomocí veřejného klíče. Pokud je obsah zprávy neporušen, dokument je pravý. Druhý způsob je podobný s tím rozdílem, že místo domluvené zprávy se jako podpis použije hash odesílaného dokumentu. Postup je tedy následující: Nejdříve se vypočítá hash odesílaného dokumentu, tato hash se zašifruje pomocí soukromého klíče odesílatele. Zašifrovaná hash je tedy digitální podpis, který se přiloží k odesílanému dokumentu a odešle se příjemci. Příjemce nejprve vypočítá hash z přijatého dokumentu a následně dešifruje zašifrovanou hash (podpis) za pomocí veřejného klíče. Pokud se vypočítaná hash shoduje s dešifrovanou hashí z digitálního podpisu je dokument pravý. [12][29]

4.1.4 Certifikace veřejných klíčů

Digitální podpis ověřuje pravost odeslané zprávy, ale už neumožňuje ověření totožnosti odesílatele. Tento problém řeší Infrastruktura správy a distribuce klíčů. Součástí této infrastruktury je Certifikační autorita, která na základě žádosti uživatele uděluje Certifikáty, které ověřují totožnost uživatele. Tento certifikát je elektronický dokument obsahující

identifikační údaje uživatele a je podepsaný soukromým klíčem Certifikační autority. Tento certifikát vlastní jen určitá osoba a znesnadňuje tak útočníkům se vydávat za někoho jiného. Certifikační autorita má uloženy certifikáty všech registrovaných uživatelů. Všechny tyto certifikáty se dají ověřit pomocí veřejného klíče certifikační autority. K tvorbě certifikátů se používá symetrické i asymetrické kryptografie. Každá z metod má své přednosti, certifikáty vytvořené symetrickou kryptografií jsou rychleji dešifrovány a je možné je dešifrovat offline. Naopak certifikáty šifrované asymetrickou kryptografií jsou bezpečnější, protože je potřeba přístupu k veřejnému klíči na internetu. Pro získání certifikátu uživatel odešle svoje identifikační údaje a veřejný klíč Certifikační autoritě, ta tyto údaje podepíše svým soukromým klíčem a předá certifikát uživateli. Ověření probíhá tak, že si komunikující strany vyžádají od Certifikační autority vzájemné certifikáty, které následně ověří veřejným klíčem Certifikační autority. [12][29]

4.2 Symetrická kryptografie

Symetrické šifry používají pro šifrování i dešifrování zprávy stejný klíč. Případně je dešifrovací klíč snadno vypočitatelný. Proto je nutné držet klíč v tajnosti a zabezpečit jeho doručení tak, aby se ke klíči nedostala neoprávněná osoba. Pro distribuci klíčů je možné využít asymetrického šifrování. Je také vhodné při šifrování používat takové operace a parametry, které při znalosti vstupního i zakódovaného textu znemožňují jednoduché odhalení klíče. Dalším bezpečnostním prvkem je použití delšího klíče, v dnešní době se doporučuje používat alespoň 128 bitové klíče. Symetrické šifrování je nejrozšířenější šifrovací metoda pro rychlé šifrování, používá se hlavně pro šifrování zpráv na nešifrovaném kanále. Algoritmy symetrické kryptografie se dělí na dva typy: proudové algoritmy šifrují data postupně po jednotlivých bitech a blokové algoritmy dělí data do bloku a až pak je šifrují. Mezi nejpoužívanější metody šifrování patří algoritmy AES a DES. [12][29][31]

4.2.1 Proudové šifry

Proudové šifry zpracovávají data po jednotlivých bitech. Výhodou tohoto zpracování je vysoká rychlost a malá HW náročnost. Další výhodou proudových šifer je minimální šíření chyb, protože případná chyba ovlivní jen šifrovaný znak. Pokud se nedodržují bezpečností opatření, můžou být proudové šifry lehce prolomitelné. Proudové šifry se rozdělují podle způsobu, jakým generují proud znaků (keystream) a to na synchronní a asynchronní.

• Synchronní - u synchronních šifer je keystream generován na základě šifrovacího klíče a šifrovacího algoritmu. Keystream je tedy závislý na aktuálním stavu algoritmu, pokud se mezi odesílatelem a příjemcem ztratí synchronizace nelze šifrovaný text dešifrovat. Protože na rozdíl od Asynchronního šifrování není dešifrování závislé na předešlých znacích, není při chybě v přenosu chybně ovlivněn zbytek zprávy. To je výhoda v situacích kdy dochází během přenosu k častým chybám.

• Asynchronní - u asynchronních šifer je keystream generován na základě šifrovacího klíče a několika předchozích znaků šifrovaného textu. Synchronizace není u asynchronního šifrování tak důležitá jako u synchronního, po přerušení synchronizace by se měla šifra po několika znacích sama synchronizovat. Nevýhodou asynchronního šifrování je, že při chybnému přenosu některého znaku je chybně ovlivněno i několik následujících znaku. [12][31]

4.2.2 Blokové šifry

U blokových šifer dojde během šifrováním k rozdělení zprávy na bloky s pevnou velikostí, které se následně šifrují. Tento proces je složitější než zpracování proudových šifer, proto je zpracování blokových šifer náročnější na výkon a jsou pomalejší. Obvyklá velikost šifrovacích bloků je 64, 128 nebo 256 bitů. Velikost bloků má bezpečnostní význam, při použití příliš malého bloku by bylo možné vytvořit pro daný klíč slovník a šifrování tak prolomit. Výhodou blokových šifer je schopnost detekovat změny znaků v bloku při dešifrování, takže není možné do bloku vložit jiný znak bez odhalení. Použití bloků ale vytváří i nevýhody, jednou z nich je šíření chyb v celém bloku při chybě v jednom znaku. Mezi nejrozšířenější algoritmy založené na blokových šifrách patří algoritmy DES (Data Encryption Standard) s 56 bitovým klíčem, dále novější 3DES s klíčem dlouhým 112 nebo 168 bitů a algoritmus AES s délkou klíče, 192 nebo 256 bitů. [12][31]

Algoritmus DES

Algoritmus DES (Data Encryption Standard) byl vyvinut v 70. letech. Velikost šifrovaného bloku je 64 bitů. Algoritmus používá 56 bitový klíč s dalšími 8 kontrolními bity. Výhodou tohoto algoritmu je, že postup dešifrování je přesně opačný než při šifrování. Není tedy potřeba zvlášť programovat šifrovací a dešifrovací část. Dnes je tento algoritmus považován za nedostatečný a nedoporučuje se ho používat. Je to dáno hlavně délkou klíče. DES s 56 bitovým klíčem je dnes možné prolomit při útoku hrubou silou za méně než jeden den. Proto se začal nahrazovat novějšími algoritmy jako je například 3DES (Triple DES) nebo AES (Advanced Encryption Standard). [31]

Algoritmus AES

Tento algoritmus vznikl jako náhrada za zastaralý DES a pomalý 3DES. V současnosti je jedním z nejpoužívanějších symetrických algoritmů. Svými autory byl původně pojmenován Rijndael, ale vžil se pro něj název AES (Advanced Encryption Standard). Jedná se o blokovou šifru s délkou bloku 128, 192 nebo 256 bitů a délkou klíče 128, 192 nebo 256 bitů. V současnosti není znám útok, který by mohl ohrozit bezpečnost tohoto algoritmu. Nepředpokládá se, že by mohl být v příštích 20-30 letech prolomen. Šifrování probíhá ve čtyřech krocích. V prvním kroku je každý bit nahrazen jiným bitem dle šifrovacího klíče, tím je zaručena ochrana před jednoduchými algebraickými útoky. V druhém kroku se všechny bity v jednotlivých řádcích matice posouvají o několik sloupců vlevo. V každém řádku o jinou hodnotu. Ve třetím kroku jsou proházeny jednotlivé sloupce a následně jsou všechny vynásobeny stejným polynomem. V posledním kroku je každý bajt šifry zkombinován s bajty sub-klíče získaného pomocí původního klíče a Rijndaelovy tabulky. Tyto čtyři kroky jsou provedeny několikrát v několika kolech podle velikosti klíče. Pro 128bitový klíč je to 10 kol pro 192 a 256 bitový klíč je to 12 a 14 kol. Výsledná šifra vznikne po ukončení posledního kola. [31]

4.3 Asymetrické kryptografie

Asymetrická kryptografie používá na šifrování a dešifrování různé klíče. To je hlavním rozdílem mezi Symetrickou a Asymetrickou kryptografií. Klíče rozdělujeme na soukromé a veřejné. Soukromý klíč je tajný a vlastní ho jeho majitel, který se také musí postarat o to, aby

nebyl klíč odhalen. Naopak k veřejnému klíči má přístup kdokoli. Klíče jsou generovány tak, aby nebylo možné jeden od druhého odvodit. Při asymetrickém šifrování použije odesílatel veřejný klíč příjemce k zašifrování zprávy, zašifrovanou zprávu pak odešle příjemci, který jí pomocí svého soukromého klíče dešifruje. Odesílatel může veřejný klíč příjemce získat přímo od příjemce nebo o něj požádat Certifikační autoritu (viz kapitola 4.1.4). Pokud chce příjemce odpovědět odesílateli, musí k šifrování zprávy použít veřejný klíč odesílatele. Asymetrické algoritmy jsou mnohem náročnější, než symetrické algoritmy z toho plyne i jejich pomalost, ve srovnání se symetrickými jsou asi 100 krát pomalejší. Asymetrická kryptografie má ale své výhody, není potřeba se šifrovanou zprávou posílat dešifrovací klíč a riskovat tím odhalení zprávy, to napomáhá k většímu zabezpečení šifrované informace. V dnešní době se asymetrická kryptografie používá hlavně pro distribuci symetrických klíčů. Tímto způsobem se zkombinují dobré vlastnosti obou typů šifrování. Zpráva je zašifrována pomoci symetrického klíče, který se zašifruje veřejným klíčem a pošle se spolu se zašifrovanou zprávou příjemci. Ten pomocí soukromého klíče dešifruje zašifrovaný symetrický klíč, kterým následně dešifruje zprávu. Tím je zaručena bezpečnost asymetrického šifrování a rychlost symetrického šifrování. Asymetrická kryptografie se dále používá při tvorbě digitálních podpisů a certifikátu viz kapitoly 4.1.3 a 4.1.4. [12][29][31]

4.3.1 Algoritmus RSA

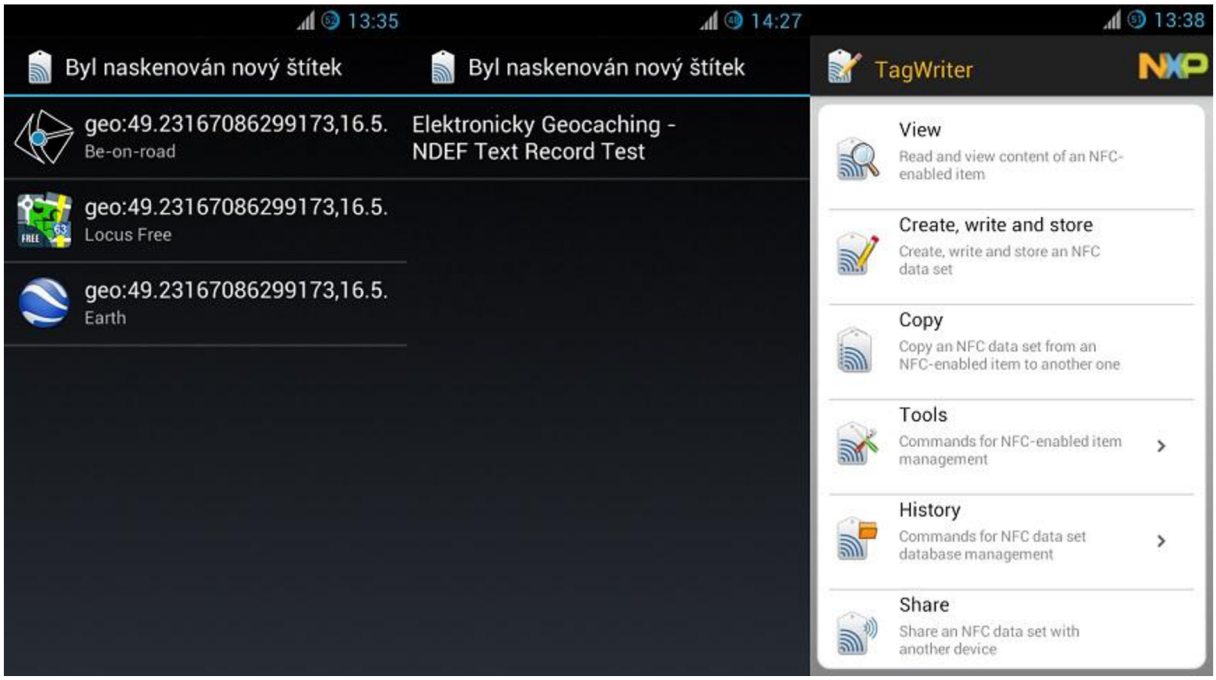
Algoritmus RSA je v dnešní době nejrozšířenějším asymetrickým algoritmem. V roce 1977 ho vytvořili Ron Rivest, Adi Shamir a Leonard Adleman podle kterých dostal i své jméno. RSA slouží pro šifrování zpráv a poskytuje autentizaci dat pomocí digitálního podpisu. Je až 1000 krát pomalejší při HW implementaci než symetrický algoritmus DES, při SW implementaci je pak pomalejší 100 krát. Algoritmus RSA je založen na obtížné faktorizaci (rozložení čísla na součin prvočísel) velkých čísel. Pro výpočet klíčů se používá součinu dvou velkých prvočísel. Sílou algoritmu je skutečnost, že není možné tyto dvě prvočísla z výsledku součinu v reálném čase odhalit. RSA používá délku klíče nejčastěji 1024, 2048 nebo 4096 bitů. Pro získání soukromého a veřejného klíče se nejdříve zvolí dvě velká prvočísla p a q, tyto prvočísla by měla být z bezpečnostních důvodů stejně velká. Vypočítá se součin těchto prvočísel $n = p*q$ a hodnota $r = (p - 1)(q - 1)$. Následně je zvolen šifrovací exponent e tak, aby byl nesoudělný s r. Dešifrovací exponent d je pak vypočítán Eukleidovým algoritmem pomocí vzorce $d \equiv e^{-1} (mod\ r)$. Veřejným klíčem je dvojice (n,e), kde n je modul a e je šifrovací exponent. Soukromým klíčem je dvojice (n,d), kde n je modul a d je dešifrovací exponent. Před šifrováním se zpráva nejdříve převede na číselný kód ASCII, následně se rozdělí na stejně velké bloky (bloky m_i musí být menší, než n). Všechny bloky pak zašifrujeme pomocí vzorce $c_i = m_i^e \ mod\ n$. Všechny bloky c_i se pak spojí do zašifrované zprávy. Příjemce zprávu během dešifrování rozloží na bloky c_i, které dešifruje pomocí vztahu $c_i = m_i^d \ mod\ n$. [1][19][29][31]

5 Návrh řešení

Cílem diplomové práce je navrhnout a realizovat koncept elektronického geocachingu. Při návrhu má být použita bezkontaktní karta reprezentující keš a má být prodiskutována možnost komunikace s touto kartou pomocí mobilního telefonu s rozhraním NFC (Near Field Communication). Návrh řešení jsem rozdělil do několika kapitol, nejdříve uvedu nejjednodušší možný návrh, který neobsahuje žádné nebo jen minimální zabezpečení. Dále popíšu nejpokročilejší návrh, ke kterému je ale zapotřebí úzká spolupráce se společností Groundspeak Inc. provozující web www.geocaching.com. Tato spolupráce je ale v blízké době nereálná jak později vyplyne z následujícího textu. Proto jsem připravil i návrh realizovatelný pro účely diplomové práce. Posledním bodem návrhu je analýza současného trhu chytrých telefonů s NFC a výběr nejperspektivnější platformy pro vývoj mobilní aplikace.

5.1 Jednoduchý návrh

Tento návrh lze realizovat pomocí paměťových bezkontaktních karet pracujících na frekvenci 13,56Mhz (např. Mifare Classic pro přístupové systémy) nebo NFC štítku. NFC formát NDEF (viz kapitola 3.1.2) totiž umožňuje do paměťových zařízení, jakou jsou výše zmíněné bezkontaktní karty nebo NFC štítky uložit záznam, který spustí po přiblížení mobilního telefonu příslušnou akci. Seznam dostupných záznamů a jejich akcí je například na [32]. Mezi pro náš účel využitelné záznamy se řadí hlavně lokační záznam (GPS souřadnice), webový odkaz případně obyčejný text. Lokační záznam může sloužit pro přesměrování hráče na souřadnice další keše (může se hodit například u Multikeše).



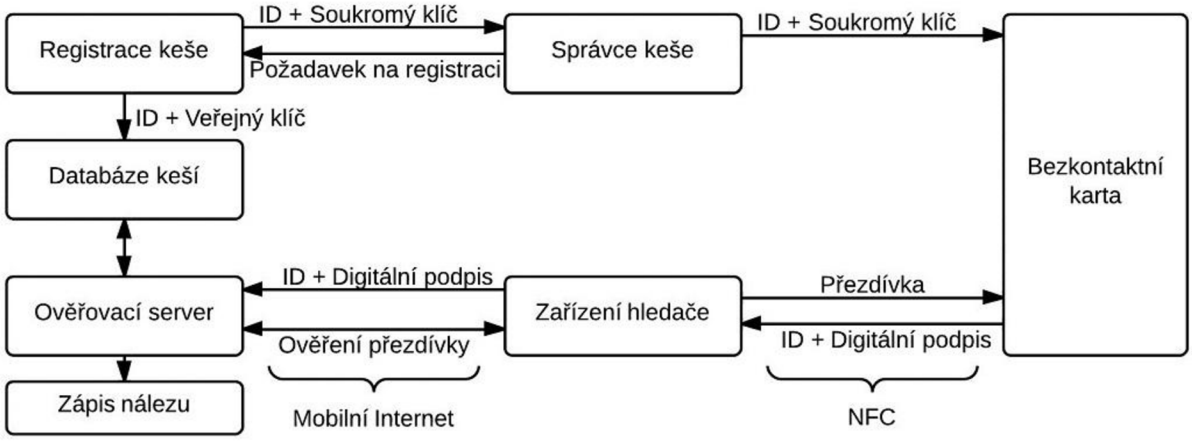
Obr. 5.1 Přečtený lokační a textový záznam a obrazovka aplikace NFC TagWriter

Webový odkaz může sloužit k otevření skryté webové stránky, na které je potřeba vyluštit nějaký rébus (použitelné u Mystery keše). Textový záznam pak může obsahovat nápovědu,

kde keš hledat (například: hledej nejbližší pařez východním směrem). Reakce telefonu na přečtení NDEF záznamu, můžeme vidět na Obr. 5.1. Výhodou použití těchto NDEF záznamů je, že jsou definované ve standardu a je tedy možné je přečíst jakýmkoliv zařízením podporujícím NFC. Nevýhodou je pak skutečnost, že jsou všechny tyto řešení nezabezpečená a informaci může přečíst kdokoli. Jediným možným zabezpečením je zašifrování textu, které by zamezilo přečtení informace nežádoucím osobám. Ovšem k dešifrování by pak byla potřeba dešifrovací aplikace. Tím bychom přišli o výhodu čtení jakýmkoliv NFC zařízením. Pro zápis těchto informací můžeme použít například aplikaci NFC TagWriter [33], kterou můžeme vidět na Obr. 5.1.

5.2 Pokročilý návrh

Pokročilý návrh je založen na digitálním podpisu. Používá se tedy asymetrické kryptografie se soukromým a veřejným klíčem. Keš by reprezentovala naprogramovaná bezkontaktní karta se soukromým klíčem. Případně by mohl být čip s anténou integrován přímo ve víku keše. Veřejné klíče by pak byly uloženy na serverech společnosti Groundspeak Inc. K digitálnímu podpisu by se používala přezdívka hráče, respektive přímo jméno účtu, které si vybral při registraci na stránkách www.geocaching.com. Protože je tato přezdívka při registraci unikátní není možné ji zaměnit s jiným hráčem. Dále je potřeba mobilní aplikace, která pomocí NFC do elektronické keše odešle hráčovu přezdívku a pak přijme digitálně podepsanou zprávu. V této aplikaci by bylo možné zadat hráčovu přezdívku, aby jí nemusel vždy při odesílání vyplňovat. Pro zvýšení bezpečnosti a zamezení v podvádění by bylo vhodné propojit uložení hráčovi přezdívky se službou Geocaching Live pomocí Geocaching.com API [34], které by zaručilo přihlášení hráče přímo na servery www.geocaching.com. Strukturu elektronického geocachingu můžeme vidět na Obr. 5.2, který zobrazuje i postup při registraci keše.



Obr. 5.2 Struktura elektronického geocachingu

Postup zalogování keše by vypadal následovně: Po nalezení keše by hráč přiložil mobilní telefon s otevřenou aplikací k víku keše (bezkontaktní kartě). Aplikace ihned po detekci elektronické keše automaticky odešle hráčovu přezdívku kartě. Elektronická keš přezdívku zašifruje (podepíše) pomocí soukromého klíče a RSA algoritmu (nebo jiného asymetrického algoritmu). Podepsanou přezdívku odešle zpět do mobilního telefonu hráče a s ní odešle i nešifrovaný identifikátor keše ve formátu GCxxxx. Následně záleží, zdali je telefon připojen k internetu nebo je v offline režimu. Protože mobilní telefony s NFC jsou z velké většiny

chytré telefony, v následujícím textu budu předpokládat, že je telefon k internetu připojený. V případě offline režimu by se následující akce odložily, dokud se telefon k internetu nepřipojí. Aplikace v telefonu po obdržení podepsané přezdívky a identifikátoru, tyto informace odešle na ověřovací server, který ve své databázi podle identifikátoru keše najde její veřejný klíč. Server pomocí veřejného klíče rozšifruje digitální podpis a výsledek porovná s přezdívkou uživatele, který požadavek na ověření odeslal (díky propojení aplikace s Geocaching Live). Pokud se přezdívky shodují, je nález hráči započten. Po započtení nálezu se v mobilní aplikaci otevře pole pro vyplnění položek IN: OUT: (viz kapitola 1.2.2), které hráč vyplní, pokud chce něco z keše vyměnit. Po vyplnění/přeskočení se pak otevře pole, kam je možné napsat poděkování za keš nebo příhody z hledání keše. Odesláním této krátké zprávy je hráč zapsán do online verze logbooku. V tomto návrhu je tedy obyčejná krabička se zápisníkem (logbookem) nahrazena krabičkou s integrovaným čipem a anténou (popřípadě bezkontaktní kartou) a online logbookem. Výhod tohoto návrhu je hned několik. První výhodou je pohodlnost a rychlost zalogování pro hráče. Dalších několik výhod souvisí s použitím online logbooku, správce keše se nemusí starat o výměnu zaplněného zápisníku, obsah keše je možno kontrolovat online a všechny nálezy jsou vidět ihned po objevení online. Nevýhody pak plynou převážně z omezení zařízení, pokud hráč nemá zařízení s NFC nezaloguje se, v případě vybité baterie telefonu taktéž. Při nedostatečném mobilním signálu může být mobilní internet nedostupný. Skupiny hráčů také nemohou používat oblíbená razítka nebo nálepky do logbooku.

5.2.1 Doplňky návrhu

Doplňkem k zabezpečení by mohlo být ověření současné polohy hráče vzhledem k logované keši. Keš by mohla být uznána, jen pokud se hráč pohybuje např. v oblasti 100m kolem webových souřadnic keše. Pokud by hráč logoval keš z jiné polohy, mohlo by to značit možné podvádění. Nevýhodou tohoto doplňkového zabezpečení je pak skutečnost, že některé keše nejsou v oblasti s dobrým GPS signálem, a bylo by tak problematické takovou keš hráči započíst.

Dalším doplňkem by mohla být krabička vybavená zámkem otevíratelným jen s pomocí telefonu s NFC a správným klíčem. K tomuto klíči by pak měl přístup jen hráč a nikdo nezasvěcený by se tak nemohl k „pokladům“ keše dostat. Problémem takové krabičky je pak napájení odemykacího mechanizmu.

5.2.2 Shrnutí návrhu

Výhody a nevýhody návrhu jsem zmínil už dříve, zde bych chtěl zhodnotit reálné možnosti implementace tohoto návrhu do současné infrastruktury na www.geocaching.com. Protože v současnosti žádný podobný systém na www.geocaching.com neexistuje, bylo by potřeba tento systém vytvořit. To by znamenalo vytvoření nové databáze pro uložení veřejných klíčů, nebo přinejmenším vytvoření nové proměnné v databázi současných keší. Dále bylo by vhodné, aby čipové karty s již uloženým soukromým klíčem a ID keše distribuovala přímo společnost Groundspeak Inc. To by umožnilo, aby byl soukromý klíč co nejvíce utajen. Alternativou je pak vyvinutí oficiální aplikace pro generování veřejného a soukromého klíče, kde by se soukromý klíč vložil do nové keše a veřejný klíč do formuláře pro schválení keše

[35]. Problémem této volby je, že takovou keš by nejdřív musel správce naprogramovat, to vyžaduje kromě odborných znalostí i vlastnictví čtečky karet. Dalším důležitým bodem pro rozšíření elektronického geocachingu by pak musela být integrace navržené aplikace s některou z nejpoužívanějších aplikací. V neposlední řadě je v současnosti NFC pořád málo rozšířené a je výhradou převážně drahých zařízení. Tomu nepomáhá ani to, že žádné mobilní zařízení od společnosti Apple neobsahuje NFC. To může být problém, protože téměř polovina [36] všech keší je umístěna v USA, kde má platforma iOS víc jak třetinový [37] podíl na trhu s chytrými telefony (viz kapitola 5.3). Proto je nepravděpodobné, že se elektronický geocaching v blízké době prosadí. Všechny tyto problémy dělají z elektronických keší menšinovou záležitost, pro kterou se pravděpodobně nevyplatí upravovat celý systém na www.geocaching.com.

5.3 Analýza chytrých telefonů s NFC

Nejdřív jsem si vyhledal statistiku používaných mobilních operačních systému.[38] Do tabulky jsem následně přidal počet zařízení s rozhraním NFC dle katalogu mobilních telefonů na serveru www.mobilmania.cz.[39]

Operační systém	Tržní Podíl Q2 2012 (%)	Tržní Podíl Q2 2013 (%)	Počet zařízení s NFC
Android	64.2	79.0	111
iOS	18.8	14.2	0
WinPhone	2.6	3.3	11
BlackBerry	5.2	2.7	4
Bada	2.7	0.4	3
Symbian	5.9	0.3	7
Ostatní	0.6	0.2	20

Tab. 5.1 Tržní podíl mobilních OS a počet zařízení s NFC

Z tabulky je zřetelné, že největší tržní podíl má operační systém Android. Také je vidět jeho velký meziroční růst. Největší počet zařízení s rozhraním NFC pracuje také na operačním systému Android, proto se jeví jako nejperspektivnější pro vývoj aplikace pracující s NFC.

5.4 Reálný návrh

Reálný návrh je rozdělen do tří částí a to, naprogramování čipové karty, naprogramování aplikace pro telefon s NFC rozhraním a zajištění kompatibility s webovou aplikací. Návrh bude tedy kvůli zpětné kompatibilitě s webovou aplikací vycházet z návrhu pana Průchy, který zpracovával podobnou práci v akademickém roce 2012/2013.

5.4.1 Naprogramování čipové karty

Na rozdíl od pana Průchy, budu pracovat s programovatelnou bezkontaktní kartou MULTOS, kterou jsem si vybral na základě doporučení konzultanta a bezpečnostních výhod zmiňovaných v kapitole 2.3.3. Platforma MULTOS zajišťuje maximální bezpečnost

uložených dat a zejména obsahuje knihovny pro práci s modulární aritmetikou, které se budou hodit při šifrování (podepisování) algoritmem RSA. Cílem tedy je naprogramovat kartu tak, aby digitálně podepisovala hráčovu přezdívku svým soukromým klíčem pomocí RSA algoritmu. Aby se zajistilo zpětné kompatibility a přijatelné rychlosti šifrování, bude použit klíč dlouhý 1024 bitů.

5.4.2 Naprogramování aplikace pro mobilní telefon

Z analýzy chytrých telefonu s NFC vyšlo najevo, že nejperspektivnějším mobilním operačním systémem je jednoznačně operační systém Android. Ostatní operační systémy nejsou tak rozšířené nebo nemají podporu pro NFC. Cílem tedy bude vytvořit aplikaci pro OS Android, aby dokázala do karty MULTOS odeslat hráčovu přezdívku a přijmout od karty digitální podpis. Ten pak bude možné zkopírovat do schránky telefonu a vložit ho do online webové aplikace přímo z terénu. Do aplikace bude možné zadat libovolnou přezdívku, pro případ, že s hráčem keš hledají jeho kamarádi, kteří nevlastní telefon s NFC.

5.4.3 Webová aplikace

Jako webová aplikace bude použita již vytvořená aplikace pana Průchy [1]. V databázi aplikace budou vytvořeny nové keše pro testování, stejně tak i uživatelské účty.

6 Tvorba aplikací

V této kapitole jsou popsány všechny vytvořené aplikace, jejich vnitřní struktura a jejich funkce. U každé podkapitoly je uveden vývojový software a stručný návod jak ho připravit pro vývoj aplikací. Všechny aplikace jsem psal pod operačním systémem Windows 7 Home Premium 64bit.

6.1 Aplikace pro bezkontaktní kartu

Pro naprogramování karty jako keše jsem si vybral platformu MULTOS. K programování jsem používal kartu MULTOS Application Developer Card ML3-80K-R1. Tato karta obsahuje kontaktní i bezkontaktní rozhraní a na rozdíl od MULTOS Live karet umožňuje nahrávat aplikace bez generování bezpečnostních certifikátu, o kterých jsem psal v kapitole 2.3.3. Kartu jsem programoval přes zapůjčenou kontaktní čtečku HID OMNIKEY 3121. Pro práci s kartou je potřeba nainstalovat programy SmartDeck a Mutil. Oba programy jsou přiloženy na DVD nebo je možné je zdarma stáhnout na stránce výrobce.[42][43]

6.1.1 SmartDeck

Program SmartDeck slouží ke kompilaci zdrojových kódů, nahrávání a mazání aplikací z karty, obsahuje debuger, simulátor karet a generátor RSA klíčů. Já jsem jej používal jen ke kompilaci kódu a generování RSA klíčů. S programem se pracuje z velké části pomocí příkazové řádky, proto je vhodné si vložit cestu k adresáři *bin* do systémové proměnné *Path*. Toho lze dosáhnout následovně: *Tento počítač* > *Vlastnosti systému* > *Upřesnit nastavení systému* > záložka *Upřesnit* a dole kliknout na tlačítko *Proměnné prostředí* > v dolní tabulce *Systémové proměnné* najít proměnou *Path* a kliknout na tlačítko *Upravit* > za poslední záznam v proměnné přidat středník a následně zadat cestu k adresáři *bin*, standardně: C:\Program Files (x86)\SmartDeck\bin > OK. Následně by měly být příkazy pro SmartDeck přístupné z příkazové řádky. Doporučil bych si udělat zástupce příkazového řádku a do kolonky *Spustit v:* vyplnit cestu ke zdrojovým souborům, aby nebylo při kompilaci nutné používat příkaz *cd* pro hledání cílového adresáře, nebo psát úplnou cestu ke zdrojovému souboru. Ke kompilaci zdrojového souboru se používají příkazy `hcl` a `halugen`, příkaz `hcl` zkompiluje soubor v jazyce C a vytvoří soubor s příponou *.hzx*, tento soubor slouží pro použití v simulátoru a debugeru, následně se použije příkaz `halugen`, který z vytvořeného souboru *.hzx* vytvoří soubor ALU, který se používá při nahrávání aplikace na kartu. Syntaxe je následující:

`hcl aplikace.c` - zkompiluje zdrojový soubor v jazyce C, vytvoří soubor aplikace.hzx

`halugen aplikace.hzx` - vytvoří soubor ALU, který se nahrává na kartu

Dalším důležitým příkazem je příkaz `hkeygen`, který slouží ke generování RSA klíčů, parametry pro nastavení vlastností generovaných klíčů jsou:

`-modsize` specifikuje délku modulu klíče v bitech (povinný)
`-exponent` specifikuje hodnotu exponentu (volitelný)
`-private` specifikuje jméno souboru se soukromým klíčem (povinný)

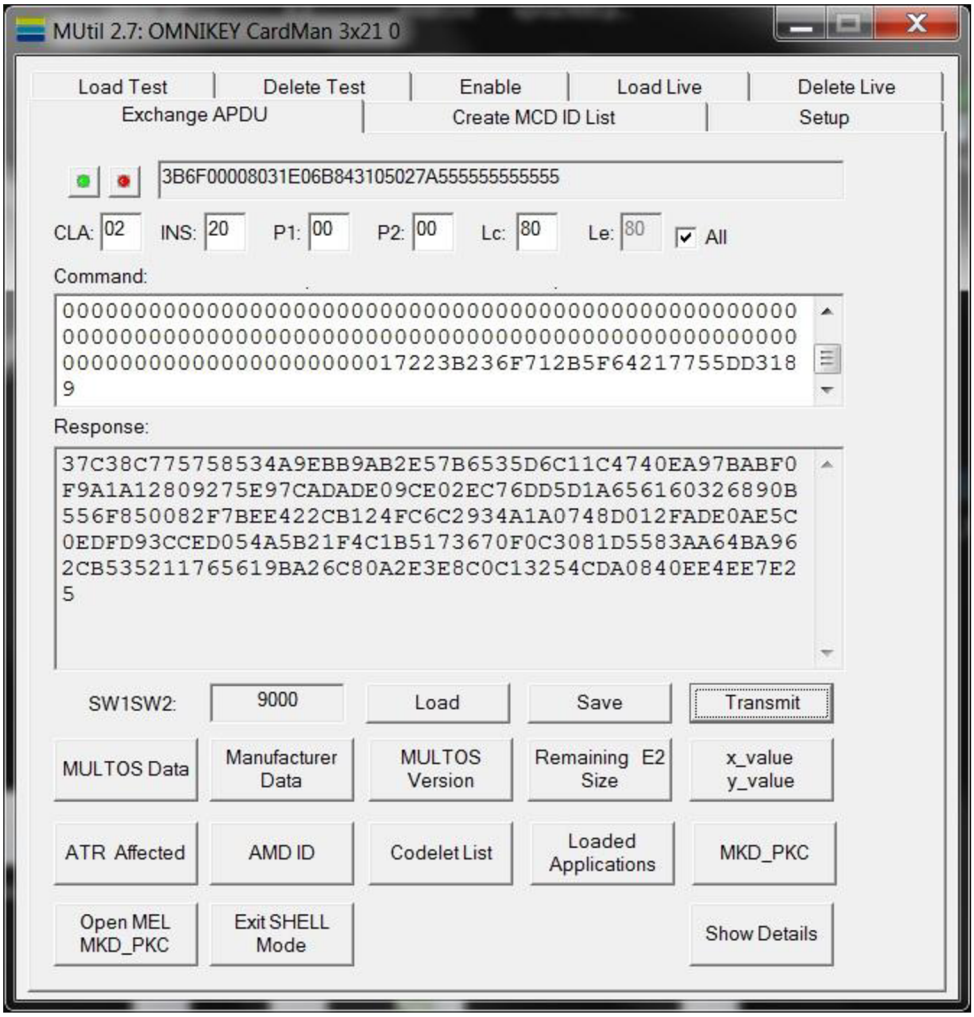
`-public` specifikuje jméno souboru s veřejným klíčem (povinný)
`-cfile` vygenerované soubory jsou ve formě hlavičkového souboru pro C (volitelný)
`-v` zobrazí vygenerované klíče v konzolovém okně (volitelný)

Syntaxe vypadá např. takto:

```
hkeygen -modsize 1024 -public public.h -private privavte.h
```

6.1.2 Mutil

Mutil je jednoduchý program s grafickým rozhraním, který slouží pro nahrávání a mazání aplikací z karty a komunikaci s kartou pomoci APDU příkazů. Podporuje nahrávání aplikací na developerské karty bez generovaných certifikátu i na Live karty s vygenerovanými certifikáty. Nejdůležitější záložky jsou *Exchange APDU*, *Load Test* a *Delete Test*.



Obr. 6.1 Záložka Exchange APDU v Mutil

Záložka *Load Test* slouží k nahrávání aplikací na developerské karty. Stačí vyhledat ALU soubor požadované aplikace, vyplnit její AID (Application ID) a stisknout tlačítko *Load*. Tlačítko *Reload* nejdříve z karty vymaže aplikaci, se stejným AID a nahradí ji vybranou aplikací. AID aplikace lze také vybírat z kolonky *Known AIDs*, kde se požadované AID objeví po modifikací položky *[AID_LIST]* v souboru *MUtil.ini* v adresáři s programem. V případě že v aplikaci používáme Session paměť, je také potřeba specifikovat velikosti

přiřazené paměti v kolonce *Session Data Size*. V záložce *Delete Test* je možno pomocí AID vymazat aplikaci z karty. Nejdůležitější záložkou je *Exchange APDU*, která je na Obr. 6.1. Tato záložka slouží pro zasílání APDU zpráv kartě a k příjímání odpovědi. Zelené tlačítko kartu zapne a vedle se zobrazí zpráva Answer to Reset (ATR). Tím můžeme ověřit, že čtečka s kartou komunikuje. Stisknutí tohoto tlačítka před komunikací s kartou není nutné, protože aplikace automaticky zapne kartu i po kliknutí na tlačítko *Transmit*. Červené tlačítko kartu vypne. Kolonky v dalším řádku slouží k vyplnění parametrů hlavičky APDU, s kartou se komunikuje výhradně v šestnáctkové soustavě. Do pole *Command* se v šestnáctkové soustavě doplňují APDU data, v poli *Response* se pak objevuje APDU odpověď. Pod polem *Response* je malé pole pro zobrazení Status Word (SW1SW2). Ve stejném řádku vpravo je tlačítko *Transmit*, které APDU odešle.

6.1.3 Popis aplikace na kartě

V této podkapitole jsou popsány nejdůležitější části aplikace, samotný kód je podrobně popsán v komentářích zdrojového souboru. Aplikaci pro kartu jsem psal v programovacím jazyce C pomocí textového editoru PSPad, který jsem přiložil i na DVD.

Základní jednotka pro komunikaci s bezkontaktními kartami je APDU (Application Protocol Data Unit). Struktura APDU vypadá takto:

CLA	INS	P1	P2	Lc	Data	Le

CLA - Class byte, identifikuje aplikaci na kartě
INS - Instruction byte, označuje instrukci, kterou chceme provést
P1, P2 - Parameter bytes, určuje další parametry pro provedení instrukce
Lc - označuje délku datové části
Data - data zasílaná kartě ke zpracování
Le - délka dat očekávaných po zpracování instrukce

Struktura APDU odpovědi:

Data	SW1	SW2

Data - data, která vrací karta, nejsou povinná
SW1, SW2 - StatusWord je vrácen vždy po zpracování APDU, označuje výsledek zpracování (např. 9000 = Úspěšné zpracování, 6F00 = Neznámá chyba)

V aplikaci jsou použity následující knihovny:

`multos.h` - knihovna pro práci s MULTOS Standard C-API
`RSA.h` - knihovna obsahuje funkce pro šifrování a dešifrování RSA algoritmem a definuje struktury pro soukromý a veřejný klíč

Parametry aplikace se nastavují pomocí atributů, nejdůležitější atributy jsou atributy pro definici, v jaké části paměti se mají nacházet datové struktury a atribut pro nastavení AID aplikace. Atributy pro definici paměti vypadají takto:

```
#pragma melstatic
#pragma melpublic
#pragma melsession
```

Data napsaná pod jednotlivými atributy se ukládají do příslušné části paměti. Statická paměť slouží k uchování dat na kartě i po odpojení napájení, zde je v našem případě uložen jen identifikátor keše. Veřejná paměť a Session paměť jsou uloženy v RAM a tak se jejich obsah po odpojení napájeni ztratí. Veřejná paměť se používá k uchovávání dat z příchozího APDU a dat pro APDU odpověď. Session paměť není v mém programu využita. Atribut pro nastavení AID vypadá v mojí aplikaci následovně:

```
#pragma attribute("aid", "f0 00 00 02")
```

Pro tuto aplikaci jsem zvolil AID F0000002, ale může zde být jakékoliv hexadecimální číslo. Pro lepší přehlednost aplikace jsou použity symbolické konstanty, které zastupují zejména CLA aplikace, jednotlivé instrukce a bajty chybových stavů StatusWord. CLA jsem zvolil 02.

```
#define ERR_BAD_INS                 0x6404
#define MYAPP_CLA                   0x02
#define CMD_CACHE_ID_INFO           0x10
```

Aby karta věděla jak má na APDU reagovat existují tzv. ISO Case, které definují jaká data či parametry má karta očekávat a jestli má posílat nějakou odpověď. Existují čtyři druhy ISO Case, můžeme je vidět v následující tabulce:

Case	APDU příkaz	APDU odpověď
1	CLA INS P1 P2	SW1 SW2
2	CLA INS P1 P2 Le	SW1 SW2 Data
3	CLA INS P1 P2 Lc Data	SW1 SW2
4	CLA INS P1 P2 Lc Data Le	SW1 SW2 Data

Tab. 6.1 Přehled ISO Case

V prvním sloupci vidíme hodnotu ISO Case, v druhém sloupci požadované parametry nebo data a ve třetím strukturu APDU odpovědi. MULTOS pro kontrolu ISO Case používá funkci `CheckCase`. Syntaxe této funkce pro ISO Case 4 vypadá následovně:

```
if (!multosCheckCase(4))
   multosExitSW(ERR_WRONGCLASS);
```

Aby bylo možné s aplikací na kartě komunikovat je nejdříve potřeba aplikaci na kartě vybrat, to se provádí speciálním Select File APDU příkazem. Ten vypadá takto:

CLA	INS	P1	P2	Lc	Le	Data
00	A4	04	00	04	00	F0000002

V datech je uloženo AID požadované aplikace a Lc vyjadřuje délku datové části v bajtech, ostatní položky jsou pro výběr jiných aplikací totožné. Po zaslání tohoto příkazu je již možné posílat aplikaci příkazy na provedení vybraných instrukcí. Po přijmutí APDU aplikace nejdříve ověří, zda je v hlavičce APDU uvedeno její CLA, v našem případě 02. Pokud souhlasí tak aplikace na základě pole INS v hlavičce APDU přepíná pomoci `switch` na požadovanou instrukci. Moje aplikace obsahuje tři následující instrukce:

INS 0x10 - v APDU odpovědi odešle identifikátor keše.
APDU pro INS 0x10:

CLA	INS	P1	P2	Lc	Le	Data
02	10	00	00	00	00	"prázdné"

INS 0x20 - Algoritmem RSA a 1024 bitovým soukromým klíčem zašifruje (podepíše) data zaslaná v APDU, výsledek odešle v APDU odpovědi.
APDU pro INS 0x20:

CLA	INS	P1	P2	Lc	Le	Data
02	20	00	00	80	80	"data z telefonu"

INS 0x30 - Algoritmem RSA a veřejným klíčem dešifruje data poslaná v APDU, výsledek dešifrování odešle v APDU odpovědi, tato instrukce není pro elektronický geocaching potřeba, ale pro ověření správného šifrování jsem ji v aplikaci zanechal.
APDU pro INS 0x30:

CLA	INS	P1	P2	Lc	Le	Data
02	30	00	00	80	80	"data z INS 20"

6.1.4 Práce s certifikáty

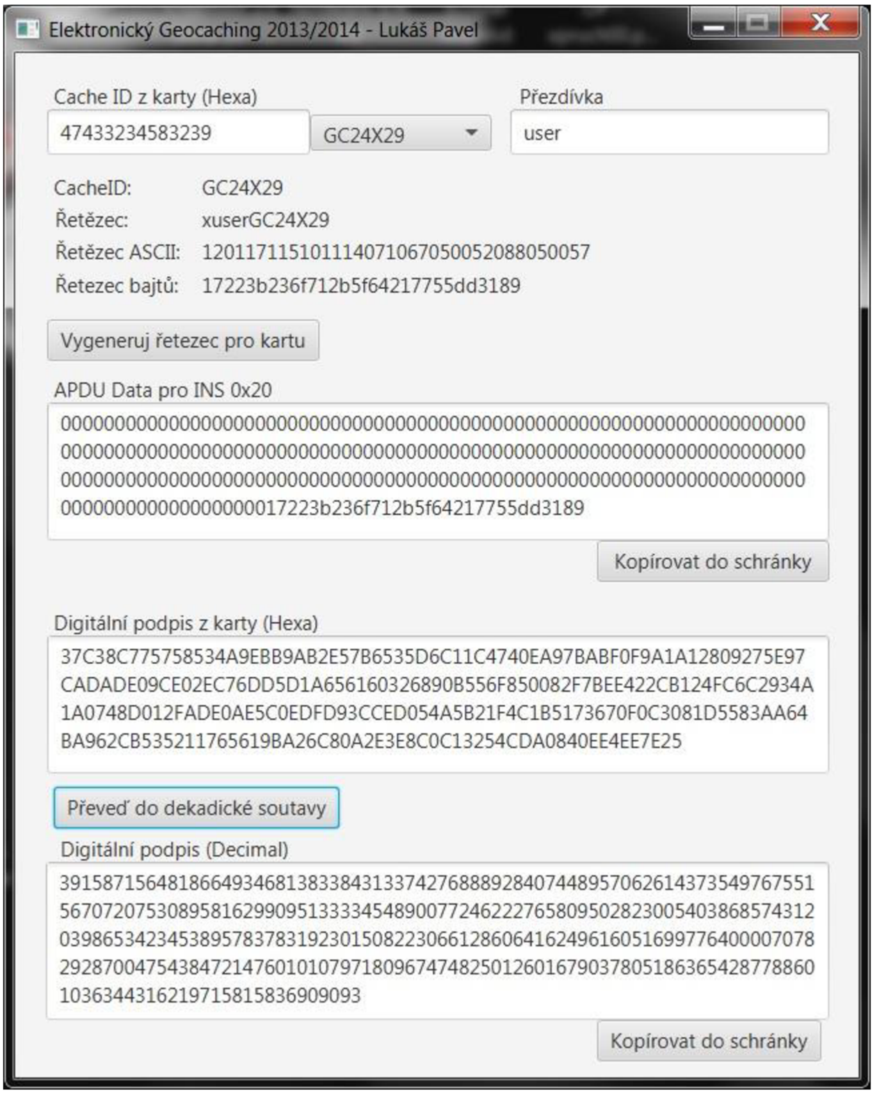
Protože jsem pracoval s developerskými kartami, nepotřeboval jsem certifikáty pro nahrávání a mazání aplikace z karty generovat, stručný popis postupu generování certifikátu je popsán v kapitole 2.3.3. Podrobnější popis lze nalézt v [19]. Program Application Registration File Generator, který je k procesu generování certifikátů potřeba jsem přesto umístil na DVD.

6.2 Aplikace pro Windows

Aplikace pro Windows je napsaná v Javě, programoval jsem ji před vývojem pro Android, abych si osvojil základy Javy a v pohodlném prostředí otestoval kód, který měl být použit na telefonu. Většina kódu pak byla v nezměněné formě použita v Android aplikaci, proto zde jen stručně popíšu význam jednotlivých polí, a samotnou strukturu kódu popíšu až u Android aplikace. Před programování jsem si stáhl Java Development Kit (JDK) obsahující vývojové prostředí NetBeans a na vytvoření okenní aplikace jsem používal JavaFX Scene Builder. Všechen jmenovaný SW je přiložen na DVD.

Aplikaci můžeme vidět na Obr. 6.2. Aplikace je navržená tak, aby šla pohodlně použít s aplikací Mutil při komunikaci s kartou. Vstupní hodnoty se kopírují přímo z APDU odpovědí z Mutilu, naopak výstupní se zkopíruji z okenní aplikace do pole *Command* v Mutilu. Na aplikaci je dobře vidět jaké kroky se provádějí při komunikaci s kartou. První pole *Cache ID* je obsah APDU odpovědi na INS 0x10, je zde možné zadat aktuální odpověď z Mutilu, nebo vybrat jednu z předem vyplněných keší ve vedlejší nabídce. Do pole napravo se zadává uživatelská přezdívka. Po kliknutí na tlačítko *Vygeneruj řetězec pro kartu* se přetvoří id keše z pole *Cache ID* na textový řetězec, zřetězí se s přezdívkou hráče a tento vytvořený řetězec se převede na desítkový ASCII kód. ASCII kód se dále převede na hexa

číslo, toto číslo se doplní nulami do 128bajtů, jak je vysvětleno v kapitole 6.3. Tento řetězec je určen k podepsání v kartě instrukcí 0x20, vloží se do pole *Command* v Mutilu. Protože webová aplikace zpracovává elektronický podpis v desítkové soustavě je potřeba APDU odpověď na instrukci 0x20 v šestnáctkové soustavě z Mutilu vložit do pole *Digitální podpis z karty* a stisknout tlačítko *Převeď do dekadické soustavy*.



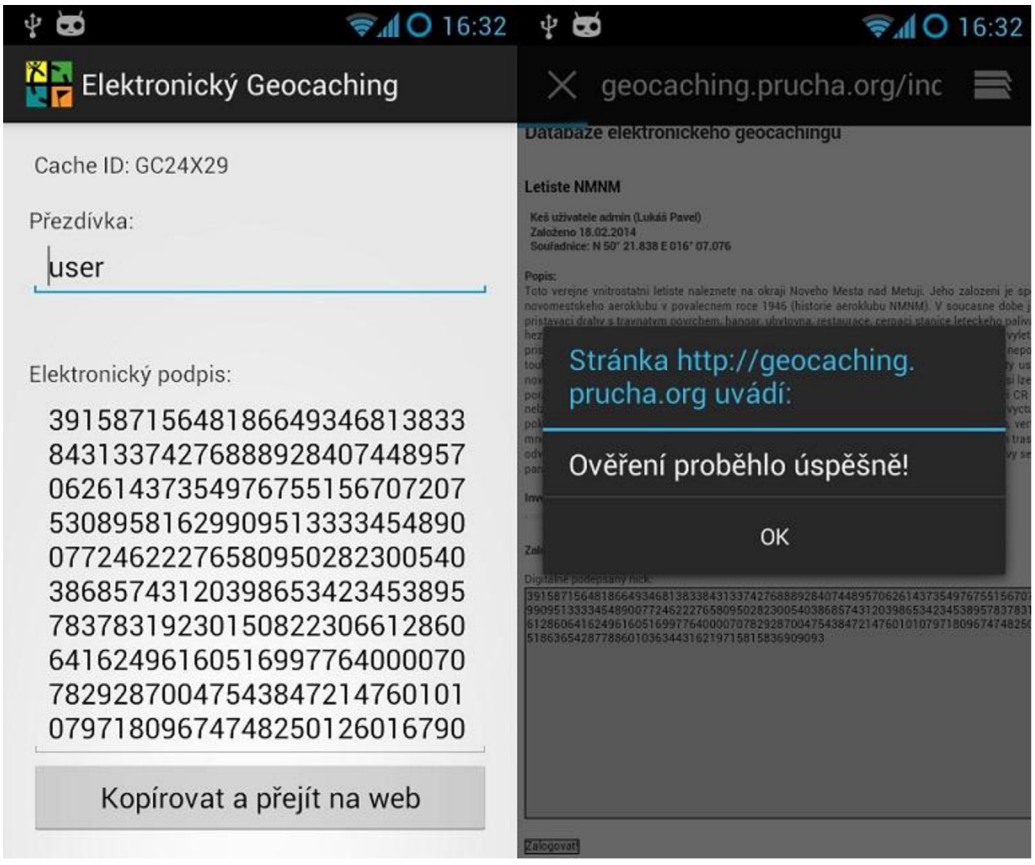
Obr. 6.2 Aplikace pro Windows

Tím se v posledním poli objeví elektronický podpis vhodný pro zkopírování do webové aplikace, kde se stačí jen přihlásit pod správnou přezdívkou a keš si zalogovat. Poslední dvě pole tedy slouží k převedení šestnáctkového čísla na číslo desítkové, to se bude ještě hodit, při tvorbě nové keše a převádění šestnáctkového veřejného klíče vygenerovaného SmartDeckem na desítkovou soustavu, protože do databáze webové aplikace se veřejný klíč vkládá také v desítkové soustavě (je podrobněji popsáno v kapitole 6.5.1).

6.3 Aplikace pro Android

K vývoji aplikací pro android je potřeba mít nainstalované JDK, a stáhnout si Android SDK, doporučuju stáhnout rovnou Android SDK Bundle [40], který obsahuje i vývojové prostředí Eclipse s nainstalovaným ADT (Android Development Tools) pluginem. Protože je

konfigurace vývojového prostředí, zprovoznění emulátoru a debugeru na skutečném zařízení zdlouhavá, tak na podrobnější návod odkážu na [41] a zmíním jen to nejdůležitější. Po instalaci SDK je potřeba stáhnout v SDK Manageru potřebné balíčky, nejdůležitější je *SDK Tools*, alespoň jednu *SDK Platform*, která obsahuje vše potřebné pro vývoj pro jednu verzi androidu, já jsem stáhl všechny SDK od Androidu 4.0 do aktuální verze 4.4.2. Dále jsem stáhl *ARM EABI v7a System Image*, který je potřeba pokud budete používat emulátor. V položce *Extra* je pak vhodné stáhnout *Intel Hardware Accelerated Execution Manager*, který zajišťuje HW akceleraci emulovaných zařízení. Nakonec jsem ještě stáhl *Google USB Driver*, abych mohl debugovat na mém Google Nexus S protože NFC aplikace není možné pod emulátorem testovat. Kromě instalace ovladačů je také potřeba v telefonu v Nastavení > *Možnosti pro vývojáře* > zaškrtnout *Ladění USB* a na PC přidat do systémové proměnné *Path* cestu do adresáře **/adt-bundle-windows/sdk/platform-tools*, podle toho kam jste Android SDK nainstalovali (postup je stejný jako v kapitole 6.1.1). Pak už stačí telefon k PC připojit a do příkazové řádky zadat `adb devices` a pokud se objeví *List of attached devices* a pod ním nějaké číslo, je debuging na skutečném zařízení funkční. Aplikaci jsem testoval na dvou různých zařízeních s dvěma různými verzemi Androidu a to Google Nexus S (i9023) s Androidem 4.1.2 a na Samsung Galaxy Nexus (i9250) s Androidem 4.3. Grafické rozhraní aplikace je vlevo na Obr. 6.3.

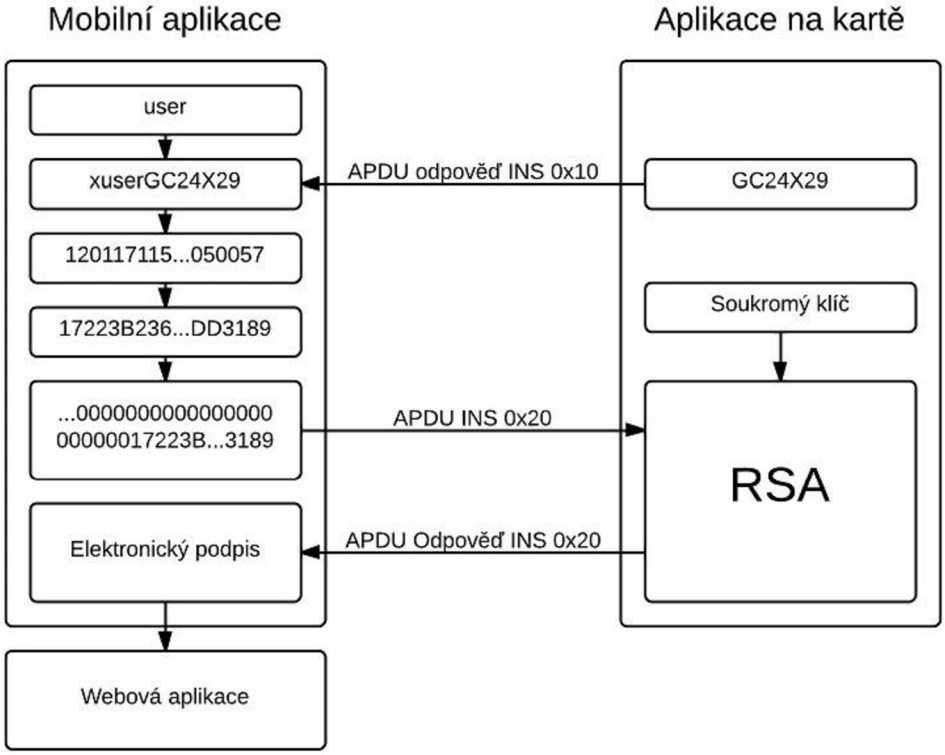


Obr. 6.3 Android aplikace a logování elektronickým podpisem

Aplikace obsahuje jen jednu obrazovku s informací o id keše, dvěma textovými poli a tlačítkem. Do prvního pole uživatel zadá svojí přezdívku, která bude sloužit k ověření nálezu na hráčském serveru, až bude elektronicky podepsána. V druhém poli se pak objevuje kartou vytvořený elektronický podpis vygenerovaný po přiblížení telefonu ke kartě. Tlačítkem se

obsah pole s elektronickým podpisem zkopíruje do schránky zařízení a v telefonu se otevře prohlížeč s online hráčským serverem. Po přihlášení zde hráč u příslušné keše vloží elektronický klíč k ověření. Pokud vše proběhne v pořádku, vyskočí na uživatele hláška o úspěšném zalogování keše, jak je vidět na Obr. 6.3 vpravo.

Tím bychom za sebou měli představení uživatelské obsluhy aplikace a následující text se bude věnovat popisu funkcí, které se dějí na pozadí. Stejně jako u aplikace pro kartu zde popíši jen nejdůležitější kroky, které aplikace provádí. Podrobný popis kódu je v komentářích zdrojového souboru. Budu předpokládat, že uživatel již vyplnil svoji přezdívku. Aplikace v základu pracuje s přezdívkou uživatele *user*, která je v aplikaci předvyplněná při startu. Pro lepší orientaci v textu je na Obr. 6.4 znázorněna komunikace mobilní aplikace s aplikací na kartě a kroky, které provádí mobilní aplikace.



Obr. 6.4 Komunikace mobilní aplikace s aplikací na kartě

Po vyplnění přezdívky tedy stačí jen přiložit telefon ke kartě. Po dostatečném přiblížení telefonu ke kartě aplikace zachytí akci `android.nfc.action.TECH_DISCOVERED` a spustí následující události. Nejdříve jsou vytvořeny dvě bajtové pole, které se naplní bajty odpovídajícími Select File APDU příkazu a APDU pro získání identifikátoru keše z karty (INS 0x10) viz kapitola 6.1.3. Aplikace se s kartou spojí a zašle vytvořený Select File APDU příkaz a následně APDU příkaz pro INS 0x10. APDU odpověď na tento příkaz je uložena do bajtového pole, které je následně přetvořeno na hexadecimální řetězec. Každá dvojice znaků v tomto řetězci je hexadecimální reprezentací znaku z ASCII tabulky. Jednotlivé dvojice tohoto řetězce jsou pak přeloženy do čitelných ASCII znaků a v telefonu vznikne čitelný řetězec identifikátoru keše (např. GC24X29), který se následně objeví v aplikaci. Při překladu na ASCII znaky jsou z původního řetězce odděleny poslední dva bajty se Status Word 9000.

Následně se začne vytvářet řetězec, který slouží k zašifrování (podepsání) RSA algoritmem v kartě. Protože se od začátku počítalo s využitím již vytvořeného webového serveru pana Průchy [1], na kterém se provádí dešifrování a ověřování elektronického podpisu. Je nutné, aby byl i vstupní řetězec pro šifrování totožný se strukturou řetězce použitého v jeho práci. Následující kroky jsou tedy nezbytné pro zaručení kompatibility s webovým serverem. Pan Průcha ve své práci označuje za formát šifrované zprávy formát:

`x + přezdívka + id keše` např.: `xuserGC24X29`

kde je každý znak převeden na desítkovou reprezentaci ASCII znaku. Každé číslo zastupující ASCII znak je trojciferné, takže u znaků s číselnou reprezentací menší než 100 je před číslem umístěna nula, jedná se hlavně o velká písmena A-Z (65-90), čísla 0-9 (48-57) a malá písmena a, b, c (97-99). Tím vznikne následující řetězec:

`120117115101114071067050052088050057`

Pro přehlednost ještě v tabulce:

120	117	115	101	114	071	067	050	052	088	050	057
x	u	s	e	r	G	C	2	4	X	2	9

V práci už ale nezmiňuje, že tento řetězec převede na číslo, které převede z desítkové soustavy do soustavy šestnáctkové a teprve pak toto číslo šifruje. Převedením na šestnáctkové číslo tedy vznikne následující řetězec:

`17223B236F712B5F64217755DD3189`

Aplikace v mobilním telefonu tedy provede všechny tyto kroky, čímž vznikne požadovaný řetězec. Následně aplikace ještě před řetězec doplní nuly, tak aby se délka zprávy pro šifrování shodovala s délkou použitého šifrovacího klíče (128bajtů pro 1024bit klíč). To se dělá, protože funkce pro šifrování v kartě, přidává za zprávy menší než je velikost šifrovacího klíče výplň ve formě nul. To při dešifrování ve webové aplikaci vedlo k chybné interpretaci původní zprávy. Webová aplikace převádí dešifrovaný řetězec na číslo, takže jakákoliv nula za původním řetězcem způsobí chybnou interpretaci původního čísla (řetězce). Takto připravený řetězec se pak převede na bajtové pole a vloží se do APDU k provedení INS 0x20. APDU se odešle kartě, ta tento řetězec zašifruje (podepíše) a v APDU odpovědi telefonu zašle elektronický podpis. Nejdříve se z přijaté zprávy oddělí poslední dva bajty reprezentující Status Word 9000 a následně je hexadecimální elektronický podpis převeden na číslo v desítkové soustavě. Toto číslo se zobrazí v druhém textovém poli v Android aplikaci, které pak může uživatel použít pro zalogování keše, jak je popsáno na začátku této kapitoly.

6.4 Webová aplikace

Jak již bylo několikrát zmíněno, jako webová aplikace byla použita aplikace vytvořená panem Průchou v Diplomové práci Elektronický geocaching v akademickém roce 2012/2013. Aplikace je dostupná online na adrese *geocaching.prucha.org*, zdrojové kódy jsou také přiloženy na DVD. V následujícím textu stručně popíšu základní vlastnosti webové aplikace a obsluhu její databáze.[1]

6.4.1 Popis webové aplikace

Aplikace je napsaná v PHP a používá databázi MySQL. Podrobnější informace lézt v [1]. Web obsahuje několik stránek, první položka menu *Schránky/keše* zobrazí všechny keše z databáze, po kliknutí na jméno keše se otevře stránka s jejím popisem, kde je možné si po přihlášení keš zalogovat. Přihlašovací údaje na web jsou v následující tabulce:

účet	heslo	práva
admin	admin	administrátorská
user	1234	uživatelská
uzivatel	1234	uživatelská

Tab. 6.2 Přehled účtů webové aplikace

Přihlášením na účet s administrátorskými právy se v menu webu objeví nová položka *Uživatelé*, kde je možné vytvořit nové účty. Další položkou menu je položka *Výpis nálezců*, kde se zobrazují uživatele (hráči), kteří si úspěšně zalogovali keš. Poslední důležitou položkou je položka *Ověření nálezu*, zde je po přihlášení možné vložit elektronický podpis, vybrat keš, která elektronický podpis vytvořila a stisknout tlačítko *Zalogovat*. Po stisknutí tlačítka se objeví jednotlivé kroky ověření a zpráva o úspěšnosti logování, viz Obr. 6.5.

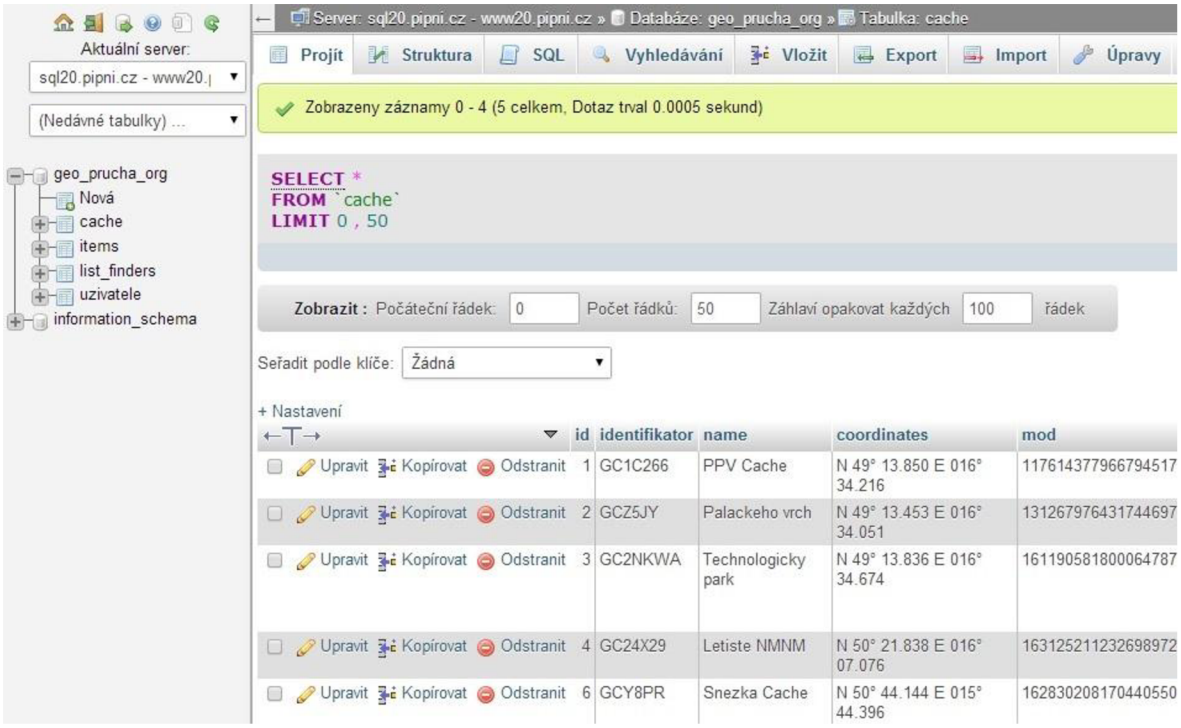
Obr. 6.5 Ověření nálezu ve webové aplikaci

6.4.2 Popis databáze webové aplikace

Databáze webové aplikace je dostupná na adrese *mysql.pipni.cz* na serveru *sql20*. Export z databáze je na přiloženém DVD. Přihlašovací údaje do databáze jsou:

Přihlašovací jméno: geo.prucha.org
Heslo: asdf

Databáze obsahuje několik tabulek. První tabulka je na Obr. 6.6, jsou zde uloženy všechny informace o jednotlivých keších, včetně identifikátoru, modulu a exponentu veřejného klíče. Tabulka keší je jediná tabulka, kterou je potřeba modifikovat manuálně a to při tvorbě nové keše. Nové keš se v databázi založí kliknutím na záložku *Vložit* a vyplněním všech polí. Modulus a exponent veřejného klíče se do tabulky vkládají v desítkové soustavě. Celý postup vytvoření nové keše je popsán v kapitole 6.5.



Obr. 6.6 Databáze webového aplikace

V databázi je dále tabulka s informacemi o uživatelích a tabulka s informacemi o nálezcích keší. Tuto tabulku je možné vyprázdnit vybráním záložky *Upravit* a kliknutím na položku *Vyprázdnit tabulku (TRUNCATE)* vpravo dole.

6.5 Postup vytvoření nové keše

V rámci práce jsem vytvořil dvě testovací aplikace pro bezkontaktní kartu reprezentující keše s identifikátory *GC24X29* a *GCY8PR*. Zdrojové soubory pro vytvoření nové keše jsou přiloženy na DVD. V této kapitole je popsán postup jak vytvořit novou keš na kartě, i v databázi webového serveru. Většina postupů již byla v různých částech práce zmíněna nicméně cílem této kapitoly je maximálně zpřehlednit celý proces vytvoření nové keše.

6.5.1 Vytvoření keše v databázi

Prvním krokem bude vytvoření keše v databázi, pro urychlení vyplňování doporučuji najít nějakou předlohu keše na www.geocaching.com/map/. Následně si otevřete databázi k webové aplikaci na webu mysql.pipni.cz zde vyberte server *sql20*:

Přihlašovací jméno: geo.prucha.org
Heslo: asdf

V postraní liště rozbalte nabídku tabulek kliknutím na *geo_prucha_org*, v seznamu klikněte na tabulku *cache*. Otevře se Vám nové okno databáze, zde se podívejte, co se do jakého sloupečku zadává (podle jména sloupce) hlavně si zapamatujte poslední použité *id* (číslo) v prvním sloupečku tabulky. Nahoře klikněte na záložku *Vložit*. Objeví se Vám plno polí, vyplňte postupně všechny pole ve sloupečku *Hodnota*, podle předlohy. Do položky *id* zapište jakékoliv číslo, které nebylo v prvním sloupečku (např. 6), položky *mod* a *exp* nechte zatím volné, do položky *founder* se zapisuje *id* uživatele v databázi (můžete zde vyplnit např. 50 - user nebo 52 - admin), jedná se o pole s informací, kdo keš založil. Teď je potřeba vygenerovat šifrovací klíče, to uděláme pomocí příkazového řádku a SmartDecku, který je přiložen na DVD. Pokud máte nastavenou Systémovou proměnnou *Path* o které jsem psal v kapitole 6.1.1, nemusíte do příkazové řádky psát celou cestu k souborům. V opačném případě je nejjednodušší pracovat se soubory v přímo adresáři SmartDecku (standardně C:\Program Files (x86)\SmartDeck\bin). Otevřete si příkazový řádek (Win+R vepsat cmd a dát Enter) dále se v příkazové řádce přesuneme do adresáře SmartDecku:

```
cd C:\Program Files (x86)\SmartDeck\bin
```

Následně můžeme použít příkaz `hkeygen` pro generování klíču:

```
hkeygen -modsize 1024 -exponent 50005 -public public.h
-private privavte.h -cfile -v
```

Ve složce, kde se nacházíme, se objeví soubory *public.h* a *private.h* s veřejným a soukromým klíčem a na obrazovce se objeví plno hexa čísel. Klikneme na příkazovou řádku pravým tlačítkem a z nabídky vybereme *Označit*, označíme všechna čísla mezi `m=+ a e=+` a zmáčkneme *Enter*, tím se nám vybraná oblast (modulus) zkopíruje do schránky. Někde si vytvoříme nový textový dokument a obsah schránky do něj vložíme. Upravíme číslo tak, aby neobsahovalo počáteční `m=+` mezery a odřádkování (aby bylo jen na jednom řádku). Celé číslo zkopírujeme do schránky, mělo by mít přesně 256 znaků, pokud ne tak bylo špatně zkopírováno z příkazového řádku. Otevřeme aplikaci pro Windows z kapitoly 6.2, která je přiložena na DVD. Do pole označeného *Digitální podpis z karty (Hexa)* vložte obsah schránky a stiskněte tlačítko *Převeď do dekadické soustavy*. V posledním poli se objeví modulus v desítkové soustavě, který můžeme konečně vložit do pole *mod* v databázi. Pokud byl při generování použit parametr `-exponent 50005`, vložíme v databázi do pole *exp* číslo 50005, v opačném případě postupujeme podobně jako u generování desítkového modulusu. Následně již můžeme stisknout tlačítko *Proveď* a keš v databázi je vytvořená.

6.5.2 Vytvoření keše na kartě

Teď přistoupíme k vytvoření samotné keše na kartě. Nejdříve si zkopírujeme některý z „Céčkových“ zdrojových souboru pro aplikaci na kartě, které jsou umístěny na DVD (např. `GC24X29.c`), soubor zkopírujeme do složky *bin* ve SmartDecku (nebo tam kde jsme si nechali vygenerovat klíče). Soubor si příhodně přejmenujte a otevřete ho v textovém editoru (např. PSPad z DVD). V kódu najděte následující část programu:

```
#pragma melstatic
static CACHE_ID cache_id = {"GC24X29"};
```

Do uvozovek napište identifikátor z Vaší keše a spočítejte počet jeho písmen, nejčastěji je to 7, u starších keší 6. V kódu vyhledejte proměnou `char id[7]`, nahází se několik řádku nad řádkem s identifikátorem keše. Do hranatých závorek následně vepište správný počet písmen identifikátoru Vaší keše. Soubor uložte a tím je zdrojový soubor vaší keše připraven ke kompilaci. Opět otevřete příkazovou řádku, přesuňte se do adresáře *bin* SmartDecku, ve kterém máte uloženy klíče a zdrojový soubor a následně napište:

```
hcl jmeno_vaseho_souboru.c
halugen jmeno_vaseho_souboru.hzx
```

Tím se Vám zkompiluje zdrojový kód a vytvoří soubor ALU pro nahrání na kartu. Příkazový řádek můžete vypnout. Zapojte čtečku karet, vložte MULTOS Developer Card, chvíli počkejte, než ji Windows detekují a pak zapněte Mutil z DVD (pokud pracujete s Live kartou, můžu jen odkázat na kapitolu 6.1.4). V Mutilu vybereme záložku *Load Test* (u Developer Card), stiskneme tlačítko *Browse* a najdeme vytvořený ALU soubor Vaší keše. Do položky AID vyplníme `F0000002` a stiskneme *Load* (nebo *Reload* pokud již na kartě nějaká keš byla). Následně je aplikace na kartě nahrána a je možné vyzkoušet její funkčnost s mobilní a webovou aplikací, jak je popsáno v kapitole 6.3.

7 Závěr

Během diplomové práce jsem nastudoval základy turistické hry geocaching, seznámil se s problematikou bezkontaktních karet a jejich bezpečnosti. Dále jsem v práci probral použití a bezpečnost rozhraní NFC. Pochopil jsem základní principy kryptografického zabezpečení, zejména pak digitálního podpisu. Získané znalosti jsem následně využil, při tvorbě návrhu elektronického geocachingu. Navrhl jsem jak jednoduché formy elektronického geocachingu, tak i pokročilou formu, která by vyžadovala spolupráci se společností provozující servery pro geocaching. Provedl jsem analýzu trhu s telefony obsahující NFC rozhraní a jako nejperspektivnější systém pro vývoj mobilní aplikace jsem zvolil operační systém Andorid. Na základě mého návrhu jsem vytvořil aplikace pro bezkontaktní kartu a operační systém Android, které jsou kompatibilní s již vytvořenou webovou aplikací pana Průchy, kterou vytvořil ve stejnojmenné práci v akademickém roce 2012/2013. Mobilní telefon s NFC a vytvořenou aplikací dokáže odeslat hráčovu přezdívku do karty, která pomocí RSA algoritmu přezdívku elektronicky podepíše a podpis odešle zpět do hráčova telefonu. Hráč může následně podpis vložit do online webové aplikace a nechat si nález ověřit. V rámci testování byly v databázi webové aplikace vytvořeny dvě nové keše s unikátními veřejnými klíči. Na konec práce jsem také sepsal postup tvorby nových keší, pro případ budoucího rozšiřování práce. Jako budoucí rozšíření konceptu elektronického geocachingu bych mohl doporučit vylepšení mobilní aplikace o historii elektronických podpisů a ukládání podpisu do textového souboru, pro hráče bez mobilního internetu. Dále vylepšení webové aplikace o mobilní zobrazení, přehlednější návštěvní knihu (logbook) a zakomponování online zápisníku s výměnou předmětu. Případně vytvoření přímého spojení mobilní aplikace s hráčským serverem bez prostředníka v podobě internetového prohlížeče, to by umožnilo pracovat se skrytým elektronickým podpisem a započtení (zalogování) keše by pro hráče bylo ještě jednodušší.

8 Literatura

[1] PRŮCHA, J. *Elektronický geocaching*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 49 s. Vedoucí diplomové práce doc. Ing. Karel Burda, CSc..

[2] LUTONSKÝ, Marek. *Geocaching: hra pro mozek, nohy a vaši GPS.* Navigovat.mobilmania.cz [online]. 12.8.2008 [cit. 2013-11-07]. Dostupné z: http://navigovat.mobilmania.cz/clanky/geocaching-hra-pro-mozek-nohy-a-vasi-gps/sc-3-a-1312930

[3] *Geocaching.com* [online]. [cit. 2013-11-10]. Geocaching - The Official Global GPS Cache Hunt Site. Dostupné z: WWW: http://www.geocaching.com/

[4] LUTONSKÝ, Marek. *Garmin Chirp: pro geocaching bez krabiček* Navigovat.mobilmania.cz [online]. 21.12.2010 [cit. 2013-11-12]. Dostupné z: http://navigovat.mobilmania.cz/clanky/garmin-chirp-pro-geocaching-bez-krabicek-test/sc-265-a-1315211

[5] *GeoWiki* wiki.geocaching.cz [online]. [cit. 2013-11-12]. Dostupné z: http://wiki.geocaching.cz/wiki/Hlavn%C3%AD_strana

[6] *A Geocaching Beginner's Guide – Geocache Container Pictures* [online]. 27.1.2013 [cit. 2013-11-13]. Dostupné z: http://blog.geocaching.com/2013/01/a-geocaching-beginners-guide-geocache-container-pictures/

[7] *Google Play - Geocaching* play.google.com [online]. [cit. 2013-11-14]. Dostupné z: https://play.google.com/store/search?q=geocaching&c=apps

[8] *Geocaching - Aplikace pro Android ve službě Google Play* play.google.com [online]. [cit. 2013-11-14]. Dostupné z: https://play.google.com/store/apps/details?id=com.groundspeak.geocaching

[9] DZURENDA, P. *Bezpečnostní rizika autentizačních metod*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 67 s. Diplomová práce. Vedoucí práce: Ing. Martin Rosenberg.

[10] HAMPL, D. *Kryptografie na výpočetně omezených zařízeních.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 50 s, 1 příloha. Vedoucí diplomové práce Ing. Lukáš Malina.

[11] MELUZÍN, I. *Multiaplikační čipové karty*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 83 s. Vedoucí diplomové práce Ing. Jiří Sobotka.

[12] KOČÍŘ, Michal *Použití smart-karet v moderní kryptografii*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 58 s. Vedoucí práce byl Ing. Jan Hajný, Ph.D.

[13] FLÉGL, J. *Systémy platebních karet.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 69 s. Vedoucí diplomové práce Ing. Ondřej Morský.

[14] MATĚJKA, J. *Útoky postranními kanály na čipové karty.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 88 s. Vedoucí diplomové práce Ing. Zdeněk Martinásek.

[15] BURDA, Karel a Ivo STRAŠIL, *Zabezpečovací systémy.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 188 s.

[16] JAILBIRD, *Smart-Card-Chip back* [online]. 16.6.2008 [cit. 2013-11-20]. Dostupné z: http://commons.wikimedia.org/wiki/File:Smart-Card-Chip_back.jpg

[17] *RFID CARD 13.56 MHz - MIFARE 1K (NXP)* [online]. [cit. 2013-11-20]. Dostupné z:http://www.weiku.com/products/12125200/RFID_CARD_13_56_MHz_MIFARE_1K_NXP_.html

[18] *Java Card Platform Specification 2.2.2*, www.oracle.com [online]. [cit. 2013-11-22]. Dostupné z:http://www.oracle.com/technetwork/java/javame/javacard/download/specs-138637.html

[19] ŠÁNEK, J. *Bezpečnost elektronických dokladů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 47 s. Vedoucí bakalářské práce Ing. Jan Hajný.

[20] POSPÍŠIL, K. *Postranní kanály u Smart Card*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 80 s. Vedoucí diplomové práce Ing. Jiří Sobotka.

[21] ŠEVČÍK, Michal *Moderní autentizace předmětem a znalostí*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2011. 48 s. Vedoucí práce byl Ing. Lukáš Malina

[22] MERTLÍK, Tomáš *Popis technologie NFC a jejího zabezpečení*: diplomová práce. BRNO: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních

technologií, Ústav telekomunikací, 2013. 116 s. Vedoucí práce byl Ing. Martin Rosenberg

[23] *Samsung Galaxy Nexus.* www.nfc.cc [online]. 29.12.2011 [cit. 2013-12-22]. Dostupné z: http://www.nfc.cc/nfc-phones/samsung-galaxy-nexus/

[24] *Orange picks Wirecard for NFC.* www.huayuansh.com [online]. [cit. 2013-12-22]. Dostupné z: http://www.huayuansh.com/admin/inc/upload/upload/201311219463069965.jpg

[25] KORB, Kryštof. *Vyzkoušeli jsme placení mobilem od O2: jak to celé funguje?.* www.nearfield.cz [online]. 12.9.2012 [cit. 2013-12-21]. Dostupné z: http://nearfield.cz/clanky/vyzkouseli-jsme-placeni-mobilem-od-o2-jak-to-cele-funguje-66

[26] PULTZNER, Martin. *Projekt Czech Wallet odstartuje později, komerčně bude spuštěn až za rok.* www.nearfiled.cz [online]. 14.11.2013 [cit. 2013-12-21]. Dostupné z: http://nearfield.cz/clanky/projekt-czech-wallet-odstartuje-pozdeji-komercne-bude-spusten-az-za-rok-128

[27] *Druhy a typy RFID štítků.* www.combitrading.cz [online]. [cit. 2013-12-22]. Dostupné z: http://www.combitrading.cz/technologie/druha-a-typy-rfid.html

[28] *Host-based Card Emulation* developer.android.com [online]. [cit. 2013-12-21]. Dostupné z: http://developer.android.com/guide/topics/connectivity/nfc/hce.html

[29] WALEK, V. *Moderní asymetrické kryptosystémy*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 63 s. Vedoucí diplomové práce Ing. Lukáš Malina.

[30] VYCHODIL, P. *Softwarová podpora výuky kryptosystémů založených na problému faktorizace velkých čísel*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 68 s. Vedoucí diplomové práce doc. Ing. Karel Burda, CSc.

[31] HAVLÍČEK, J. *Šifrovací algoritmy lehké kryptografie*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 51 s. Vedoucí bakalářské práce Ing. Vlastimil Člupek.

[32] MOPIUS. *NDEF Library for Proximity APIs / NFC.* www.codeplex.com [online]. 23.10.2013 [cit. 2013-12-28]. Dostupné z: https://ndef.codeplex.com/

[33] *NFC TagWriter by NXP* play.google.com [online]. [cit. 2013-12-28]. Dostupné z: https://play.google.com/store/apps/details?id=com.nxp.nfc.tagwriter

[34] *Geocaching public API* www.geocaching.com [online]. [cit. 2013-12-28]. Dostupné z: http://www.geocaching.com/live/apidevelopers/

[35] *Submit a New Cache Listing* www.geocaching.com [online]. [cit. 2013-12-28]. Dostupné z: http://www.geocaching.com/hide/cachebasics.aspx

[36] SCHUDISKE, E. *Celebrating Two Million Geocaches – List by Country* blog.geocaching.com [online]. 10.2.2013 [cit. 2013-12-28]. Dostupné z: http://blog.geocaching.com/2013/02/celebrating-two-million-geocaches-list-by-country/

[37] DILGER, D. *Apple's iOS takes phone sales share in US, Australia prior to iPhone 5s launch* www.appleinsider.com [online]. 4.11.2013 [cit. 2013-12-28]. Dostupné z: http://appleinsider.com/articles/13/11/04/apples-ios-takes-phone-sales-share-in-us-australia-prior-to-iphone-5s-launch

[38] *Worldwide Mobile Phone Sales* www.gartner.com [online]. [cit. 2013-12-28]. Dostupné z: http://www.gartner.com/newsroom/id/2573415

[39] *Katalog mobilů* www.mobilmania.cz [online]. [cit. 2013-12-28]. Dostupné z: http://www.mobilmania.cz/katalog-mobilu/sc-63-c-1/default.aspx

[40] *Android SDK* developer.android.com [online]. [cit. 2014-5-12]. Dostupné z: http://developer.android.com/sdk/index.html

[41] KONEČNÝ, M. *Vyvíjíme pro Android: Začínáme* www.zdrojak.cz [online]. 15.6.2012 [cit. 2014-5-12]. Dostupné z: http://www.zdrojak.cz/clanky/vyvijime-pro-android-zaciname/

[42] KUCKIR, A. *Programování v operačním systému MULTOS* Brno: Masarykova Universita, Fakulta Informatiky, 2009. 45s. Vedoucí bakalářské práce RNDr. Petr Švenda, Ph.D. [online].[cit. 2014-5-12]. Dostupné z: http://is.muni.cz/th/208391/fi_b/Programovani_v_operacnim_systemu_MULTOS.pdf

[43] *MULTOS Developer's Guide* www.multos.com [online]. [cit. 2014-5-12]. Dostupné z: http://www.multos.com/uploads/MDG.pdf

Seznam zkratek

A/D - Analog / Digital
ADC - Application Delete Certificate
AES - Advanced Encryption Standard
RSA - Rivest Shamir Adleman
ALC - Application Load Certificate
APDU - Application Protocol Data Unit
API - Application Programming Interface
ASCII - American Standard Code for Information Interchange
ATR - Answer to Reset
CIL - Common Intermediate Language
CLI - Common Language Infrastructure
CLK - Clock
CLR - Common Language Runtime
CPU - Central Processing Unit
DES - Data Encryption Standard
GND - Ground
GPS - Global Positioning System
I/O - Input / Output
INS - Instrukce
JDK - Java Development Kit
MEL - MULTOS Executable Language
NDEF - NFC Data Exchange Format
NFC - Near Field Communication
NPU - Numeric Processing Unit
OS - Operační Systém
PIN - Personal Identification Number
RAM - Random Access Memory
RE - Runtime Environment
RFID - Radio Frequency Identification
ROM - Read Only Memory
EEPROM - Electrically Erasable Programmable Read-Only Memory
RST - Reset
SDK - Software Development Kit
ADT - Android Development Tools
SIM - Subscriber Identity Module
SW - Status Word
TB - Travel Bug
URI - Uniform Resource Identifier
UV - Ultraviolet (Ultra fialová)
VCC - Power (Voltage)
VM - Virtual Machine
VPP - Programing Voltage

Obsah DVD

Aplikace pro kartu:

- GC24X29 -složka se zdrojovými soubory pro MULTOS kartu
  - GC24X29.alu -soubor vhodný pro nahrání na kartu přes Mutil
  - GC24X29.c -zdrojový soubor aplikace napsaný v jazyce C
  - GC24X29.hzx -soubor pro simulátor karet a debuger ve SmartDecku
  - private.h -soukromý klíč
  - public.h -veřejný klíč
- GCY8PR
- podpisy.txt -podpisy pro různé kombinace přezdívek a keší (hexa i decimal)
- veřejné klíče.txt -veřejné klíče pro vytvořené keše (hexa i decimal)

Aplikace pro telefon:

- Složka projektu -složka projektu pro eclipse
- Zdrojový soubor -zdrojový soubor mobilní aplikace
- elgeo.apk -balíček aplikace vhodný pro instalaci v Androidu

Aplikace pro Windows:

- Složka projektu -složka projektu pro NetBeans
- WindowApp.jar -spouštěcí soubor aplikace pro Windows

Databáze a webová aplikace:

- web -webová aplikace
- geo_prucha_org.sql -extrahovaná databáze

Použitý software:

- Android -použitý software pro programování pro andorid
- Java -použitý software pro programování v Javě
- MULTOS -použitý software pro programování pro MULTOS
- PSPad -textový editor
- Application Registration File Generator - program pro získání MULTOS certifikátů

Elektronické verze práce

postup testování.txt -soubor s postupem testování pro oponenta