FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Útoky pomocí softwarově definovaného rádia
Student:	Ondřej Vokoun
Vedoucí:	Ing. Jiří Dostál, Ph.D.
Studijní program:	Informatika
Studijní obor:	Bezpečnost a informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	Do konce letního semestru 2019/20

Pokyny pro vypracování

Softwarově definovaná rádia (SDR) se často používají pro testování zabezpečení bezdrátové komunikace. S jejich cenovou dostupností tak o ně roste zájem i ze strany útočníků. Umožňují různé typy útoků. Např. jamming - zarušení frekvenčního pásma, což znemožní bezdrátovou komunikaci v daném pásmu, nebo replay attack - útok přehráním, kterým útočník může nahrát signál a následně ho použít (otevření garážových vrat apod.), nebo tampering - zásah do komunikace, a další typy útoků. Cílem práce je navrhnout a realizovat aplikaci pro příkazovou řádku, která usnadní tyto útoky pro účely penetračního testování.
Práci rozdělte na následující kroky:
Nastudujte metody digitální modulace/demodulace používané v SDR.
Navrhněte aplikaci, která bude provádět útoky na základě zadaných parametrů – typ útoku, frekvence, modulace, apod.
Návrh implementujte.
Aplikaci otestuje na zranitelném zařízení (např. meteostanice).
Výsledky vyhodnoťte.

Seznam odborné literatury

Dodá vedoucí práce.

prof. Ing. Pavel Tvrdík, CSc.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 12. února 2019

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Útoky pomocí softwarově definovaného rádia

Ondřej Vokoun

Katedra počítačových systémů
Vedoucí práce: Ing. Jiří Dostál, Ph. D.

14. května 2019

Poděkování

Děkuji vedoucímu práce Ing. Jiřímu Dostálovi, Ph. D. za cenné rady, poskytnutí potřebného vybavení a odborné vedení mé práce. Poděkování patří také mým rodičům za poskytnutí zázemí a podpory při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 14. května 2019 ……

Odkaz na tuto práci

Vokoun, Ondřej. *Útoky pomocí softwarově definovaného rádia* . Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Tato práce se zabývá zranitelnostmi bezdrátových komunikací a jejich využitím pomocí softwarově definovaných rádií. Probrány jsou základní prvky rádií, jejich analogová část a zpracování digitálního signálu. Následuje analýza útoků typu jamming, replay attack, tampering a relay attack. Dále byla provedena rešerše stávajících řešení umožňující útoky, analyzován byl zejména nástroj Univerasal Radio Hacker. S využitím všech těchto poznatků byla navržena vlastní aplikace pro podporu útoků, která je optimalizována zejména pro použití s rádii HackRF a USRP. Testování aplikace proběhlo na meteostanici TFA a dálkovém odemykání automobilu Ford Focus Mk 2. K testování bylo použito HackRF One a USRP B210.

Klíčová slova SDR, softwarově definovaná rádia, útok přehráním, rušení rádiového signálu, zasahování, I/Q data, zpracování digitálního signálu, GNU Radio

Abstract

This thesis focuses on security vulnerabilities of wireless communication and their use by software defined radios. Base elements of radios are discussed, their analog parts and digital signal processing. This is followed by attacks analysis like jamming attack, replay attack, tampering and relay attack. In addition, a review of existing solutions was conducted, the focus was particularly on Universal Radio Hacker. Using all this knowledge, own app was designed, implemented and optimized for use with HackRF and USRP. Also HackRF One and USRP B210 were used during testing. Application testing was done on the weather station TFA and remote keyless system used in Ford Focus Mk 2.

Keywords SDR, software defined radio, replay attack, jamming, tampering, digital signal processing, I/Q data, GNU Radio

Obsah

Úvod **1**

1 Úvod do softwarově definovaných rádií **3**
1.1 Základní stavební kameny SDR . . . 3
1.1.1 Anténa . . . 4
1.1.2 Směšovač . . . 4
1.1.3 Dolní propust . . . 5
1.1.4 A/D převodník . . . 6
1.2 Zpracování signálu – fázor a I/Q data . . . 7
1.3 Modulace . . . 8
1.3.1 Analogové modulace . . . 8
1.3.1.1 AM . . . 8
1.3.1.2 FM . . . 8
1.3.1.3 PM . . . 9
1.3.2 Digitální modulace . . . 9
1.3.2.1 Modulace ASK – Amplitude Shift Keying . . . 10
1.3.2.2 Modulace PSK – Phase Shift Keying . . . 10
1.3.2.3 Modulace FSK – Frequency Shift Keying . . . 11
1.4 Komunikační módy . . . 11
1.4.1 Simplex . . . 11
1.4.2 Half-duplex . . . 11
1.4.3 Full-duplex . . . 12

2 Analýza **13**
2.1 Bezdrátové útoky . . . 13
2.1.1 Jamming attack . . . 13
2.1.2 Replay attack . . . 14
2.1.3 Relay attack . . . 15
2.1.4 Tampering . . . 16

2.2 Dostupný software a existující řešení . . . 16
2.2.1 GNU Radio . . . 17
2.2.2 Inspectrum . . . 17
2.2.3 Universal Radio Hacker . . . 18
2.3 Použitý hardware . . . 19
2.3.1 RTL SDR . . . 19
2.3.2 HackRF One . . . 19
2.3.3 USRP™ B210 . . . 20
2.3.4 Další dostupná řešení . . . 20
2.4 Meteostanice . . . 20
2.5 Dálkové ovládání centrálního zamykání vozu . . . 21
2.5.1 Rolling code . . . 21

3 Návrh **23**
3.1 Výběr SDR . . . 23
3.2 Návrh nahrání signálu . . . 25
3.3 Návrh přehrání signálu . . . 25
3.4 Návrh rušení . . . 26
3.5 Narušení komunikace . . . 26

4 Implementace **27**
4.1 Použité technologie . . . 27
4.2 Struktura aplikace . . . 27
4.3 Pomocné/podpůrné třídy . . . 28
4.4 Nahrání signálu . . . 29
4.5 Přehrání signálu . . . 29
4.6 Rušení . . . 30
4.7 Shrnutí . . . 30

5 Testování **31**
5.1 Meteostanice . . . 31
5.1.1 Jamming . . . 31
5.1.1.1 Test č. 1 . . . 31
5.1.1.2 Test č. 2 . . . 32
5.1.1.3 Test č. 3 . . . 32
5.1.1.4 Shrnutí . . . 33
5.1.2 Replay attack . . . 33
5.1.3 Tampering . . . 34
5.2 Dálkové ovládání centrálního zamykání vozu . . . 37
5.2.1 Replay attack . . . 37
5.2.1.1 Test č. 1 . . . 38
5.2.1.2 Test č. 2 . . . 38
5.2.2 Shrnutí . . . 39

Závěr **41**
Navazující práce 42

Literatura **43**

A Seznam použitých zkratek **47**

B Uživatelská příručka **49**
B.1 Minimální požadavky 49
B.2 Spuštění 49
B.2.1 Nahrání signálu 50
B.2.2 Přehrání signálu 51
B.2.3 Rušení 51
B.2.4 Nahrání a okamžité přehrání signálu 51

C Obsah přiloženého flashdisku **53**

Seznam obrázků

1.1 Struktura SDR [2, převzato] 4
1.2 Porovnání vyzařovacích charakteristik 5
1.3 Zobrazení v komplexní rovině [12, převzato] 7
1.4 Kartézský souřadnicový systém pro I/Q data [12, převzato] 7
1.5 Časové průběhy signálů využívaných v úhlových modulacích [8, str. 135] 9
1.6 Zobrazení digitálních modulací na základě binárního vstupu [15] . 10
1.7 Komunikační módy [17] 12

2.1 Příklad realizace protokolu PKES [22, převzato] 15
2.2 Pracovní plocha GNU Radio Companion [23, pořízeno v GRC] . . 17
2.3 Prostředí programu Inspectrum [24, pořízeno v Inspectrum] 18

3.1 Návrh architektury aplikace 23
3.2 Příklad hierarchického flowgraphu [23, pořízeno v GRC] 24
3.3 Flowgraph pro nahrání signálu do souboru [23, pořízeno v GRC] . 25
3.4 Flowgraph na rušení signálu [23, pořízeno v GRC] 26

5.1 Umístění útočníka, teploměru a stanice [32, 33] 33
5.2 Demonstrace úspěšně provedeného útoku 34
5.3 GUI nástroje Spectrum Analyzer [25, pořízeno v URH] 35
5.4 Nastavení modulace v GUI URH [25, pořízeno v URH] 37
5.5 Identifikace frekvence na spektrálním analyzátoru 38
5.6 Rušení signálu [25, pořízeno v URH] 39

B.1 Nápověda aplikace 50

Seznam výpisů kódu

1 Analýza vstupu 28
2 Nastavení parametrů ovladače SDR 29

Úvod

Bezdrátové komunikace jsou všude kolem nás a dnešní moderní život si bez nich nedokážeme představit. Ať už se jedná o Wi-Fi sítě, navigační systémy nebo třeba garážová vrata, používáme je v každodenním životě, protože nám přinášejí větší pohodlí a nové možnosti použití oproti standardní „drátové komunikaci“. Ta se z praktických důvodů nehodí na některé aplikace jako mobilní sítě, datové sítě apod. V takových případech je použití bezdrátové komunikace na místě.

Bezdrátové komunikace mají ale i svá úskalí, která spočívají ve složitějším návrhu, nižší propustnosti dat, spolehlivosti komunikace a také zabezpečení. Protože se bezdrátové komunikace přenášejí volným prostorem, každý může přenášený signál odposlouchávat a zabezpečení se stává nedílnou součástí komunikace.

Práce se zaměřuje na zařízení, u kterých je při návrhu kladen důraz na jednoduchost a energetickou nenáročnost. Z výše uvedených důvodů zařízení nemají dostatečnou úroveň zabezpečení a mohou být zranitelná vůči různým typům útoků. Typicky se může jednat o nejrůznější Internet of Things (IoT) zařízení, zařízení pro chytrou domácnost a další.

V posledních letech softwarově definovaná rádia (SDR) zaznamenala velký vývoj, což se projevuje na jejich dostupnosti a ceně. Umožňuje to komukoliv si SDR pořídit a otevírají se úplně nové možnosti v oblasti útoků na zabezpečení bezdrátových komunikací. Z těchto důvodů by měla být SDR považována za hrozbu a je důležité se zabývat problematikou zabezpečení bezdrátových komunikací.

Cílem práce je analyzovat stávající úroveň zabezpečení bezdrátové komunikace u běžně dostupných zařízení a zranitelnosti těchto zařízení za využití různých typů útoků, kterými mohou být jamming, replay attack a tzv. tampering neboli zásah do komunikace. Na základě této analýzy by měla být navrhnuta a implementována aplikaci pro operační systém Linux, která bude umožňovat tyto typy útoků. K implementaci bude využit framework GNU Ra-

dio. Aplikace by měla být otestována na zranitelných zařízeních a tím demonstrovat, že úroveň zabezpečení bezdrátové komunikace je často na velmi špatné úrovni. Taková aplikace může v praxi sloužit pro penetrační testování nebo např. etické hackování. Díky použití frameworku GNU Radio může být snadno rozšířena o nové funkcionality.

Práce je rozdělena na několik kapitol, *Úvod do softwarově definovaných rádií* a *Analýza* tvoří teoretický základ pro praktickou část práce. Ta je rozdělena na návrh aplikace a její implementaci, kapitolu *Návrh*, respektive *Implementace*. Na závěr je v kapitole *Testování* popsáno testování na zranitelných zařízeních.

V první kapitole a jejích podkapitolách jsou popsány základy SDR, jejich analogová část a následné zpracování digitálního signálu. Dále jsou zde stručně popsány nejčastěji používané typy modulací a komunikační módy zařízení. Druhá kapitola pokračuje v teoretickém základu a v úvodu popisuje různé druhy útoků. Na to navazuje dostupný software pro analýzu signálů a framework GNU Radio pro vytváření aplikací. Dále jsou v kapitola rozebrána rádia použitá v této práci, tedy RTL SDR, HackRF One a USRP B210.

Praktická část začíná třetí kapitolou, tj. návrhem aplikace za použití frameworku GNU Radio. Na to navazuje implementační část, která popisuje implementace stěžejních částí aplikace, zejména potom jednotlivé útoky. Praktickou část uzavírá pátá kapitola, kde je popsáno, jak probíhalo testování.

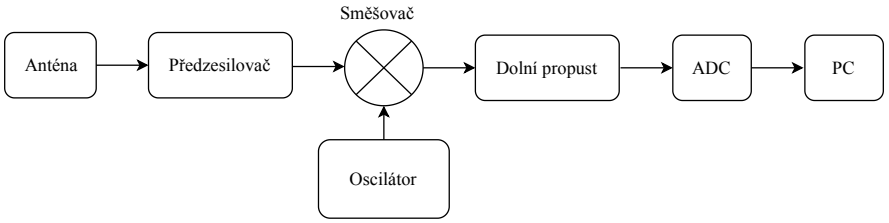
Kapitola 1

Úvod do softwarově definovaných rádií

Softwarově definované rádio (SDR) je rádiové zařízení, které umožňuje přijímat a/nebo vysílat rádiové signály a ve kterém jsou některé funkce definované softwarem. Jedná se zejména o funkce, které jsou v běžné radiotechnice implementované v hardwaru, jako např. analogové zpracování signálu. Tyto komponenty jsou nahrazeny přeprogramovatelným zpracováním digitálního signálu (DSP), díky kterému je možné měnit za běhu frekvenci, použitou modulaci, nastavení filtrů, vzorkovací frekvenci apod. To dělá z SDR velmi silný nástroj pro zájemce o DSP, radiofrekvenční analýzu a výzkumníky v oblasti radiokomunikačních technologií [1]. Tato kapitola se zabývá popisem základních komponent SDR, digitálním zpracováním signálu a komunikačními módy mezi zařízeními.

1.1 Základní stavební kameny SDR

Dle [2] by SDR pro příjem v nejjednodušší podobě obsahovalo pouze anténu, dolní propust (anglicky Low Pass Filter, LPF), A/D převodník (ADC) a zpracující zařízení, např. počítač. Obdobně by to bylo s vysílačem, který by obsahoval anténu a naopak D/A převodník (DAC). Takový přístup by kladl vysoké požadavky na ADC, proto se z praktického hlediska toto řešení nepoužívá. Praktičtějším přístupem je přidání a využití směšovače (mixeru) v kombinaci s předzesilovačem. Tuto architekturu je možné vidět na obrázku 1.1.



Obrázek 1.1: Struktura SDR [2, převzato]

1.1.1 Anténa

Anténa je zařízení k příjmu nebo k vysílání rádiových signálů a pro tento účel je i vhodně přizpůsobena. Přijímající anténa přeměňuje energii elektromagnetických vln na elektrickou energii a vysílající anténa naopak elektrickou energii na energii elektromagnetických vln [3]. Principiálně však všechny antény mohou přijímat i vysílat. Antény můžeme rozdělit do několika kategorií podle směru vysílání:

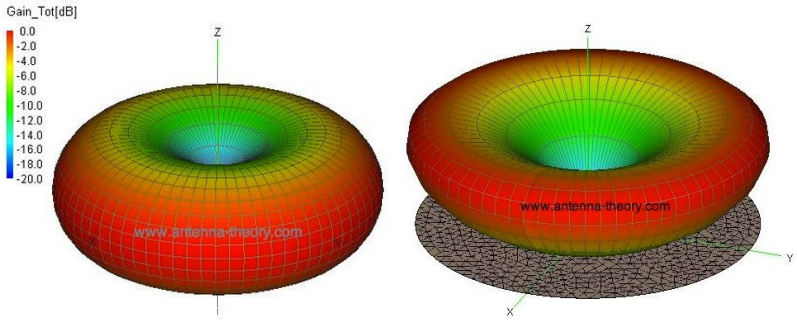
- všesměrové,
- směrové.

Důležitým parametrem antén je jejich zisk udávaný v decibelech (dB), což je bezrozměrná logaritmická jednotka. Zisk je poměr výkonu konkrétní antény vůči výkonu dipólu, tento poměr se označuje dBd. Nebo je možné uvádět zisk vůči izotropnímu zářiči, což je ideální všesměrový zářič, který v reálném světě neexistuje. Takový zisk se potom označuje dBi. [7]

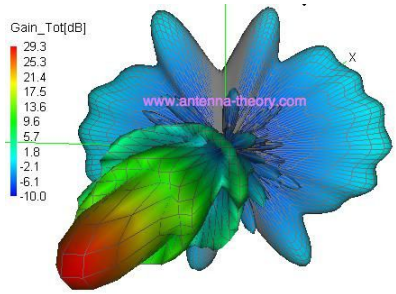
Dalším neméně důležitým parametrem antén je jejich směrovost, respektive jejich vyzařovací diagram. Na obrázcích 1.2a a 1.2b jsou zobrazeny vyzařovací diagramy všesměrových (anglicky omnidirectional) antén. Tento typ antén vyzařuje do všech směrů rovnoměrně a má tak ve všech směrech stejný zisk. Tím se také nejvíce přibližuje izotropní anténě. Naproti tomu směrové antény se vyznačují tím, že vrhají výkon do určité omezené výseče prostoru. Ten by měl být nejvyšší jen v hlavním směru vysílání a v ideálním případě všude jinde by byla „tma“. Jak je však vidět na diagramu na obrázku1.2c parabolické antény, často zde vznikají postranní laloky.

1.1.2 Směšovač

Ve směšovači dochází k přeměně vstupního signálu f_s, který je často označován jako RF (Radio Frequency), a signálu f_o, který je generován lokálním oscilátorem a označuje se LO (Local Oscilator). Ve vlastním směšovači získáváme mezifrekvenční signál f_{mf}, který má součtovou nebo rozdílovou



(a) Jednovlnný dipól [4] (b) Monopól [5]



(c) Parabolická anténa [6]

Obrázek 1.2: Porovnání vyzařovacích charakteristik

frekvenci těchto signálů. Tento vztah můžeme zapsat jako:

$$f_{mf} = f_s \pm f_o \tag{1.1}$$

V případě součtové frekvence se jedná o Up-Convertor, v případě rozdílové potom o Down-Convertor. V SDR se směšovač nejčastěji využívá jako Down-Convertor, tedy signál f_{mf} o nižší frekvenci. To umožňuje signál vzorkovat běžným A/D převodníkem. Detailnější informace může čtenář získat v [8, str. 444]

1.1.3 Dolní propust

I když máme zaručeno, že je frekvence signálu shora omezena, může docházet k aliasingu vlivem rušení signálu od různých zdrojů, jako je napájecí adaptér nebo lokální rádiová stanice. Proto je nutné zajistit, že tyto signály nebudou rušit sledovaný signál. Tyto zdroje mohou obsahovat vyšší frekvence, než je Nyquistova frekvence, tj. frekvence o vyšší frekvenci, než jsme schopni navzorkovat. Tím může vznikat aliasing a proto je vhodné využít ještě před vstupem signálu do A/D převodníku nízkofrekvenční filtr neboli dolní propust (anglicky Low Pass Filter, LPF). Tento filtr propouští nízké frekvence, tlumí vliv vysokých frekvencí a tím nedochází k aliasingu. Protože je LPF

ještě před ADC, jedná se o analogový filtr. Další informace je možné se dočíst v článku [9].

1.1.4 A/D převodník

Při převodu signálu se využívá Fourierovy transformace, která slouží pro převod mezi časovou a frekvenční doménou. Funkce $X(f)$ se nazývá *Fourierova transformace* FT a její vztah vůči aperiodické funkci $x(t)$ je

$$X(f) = \int_{-\infty}^{+\infty} x(t)\mathrm{e}^{-j2\pi ft}\mathrm{d}t, \tag{1.2}$$

respektive funkce $x(t)$ je zpětnou (inverzní) transformací *Fourierovy transformace* IFT funkce $X(f)$. Tento vztah lze vyjádřit jako:

$$x(t) = \int_{-\infty}^{+\infty} X(f)\mathrm{e}^{j2\pi ft}\mathrm{d}f \tag{1.3}$$

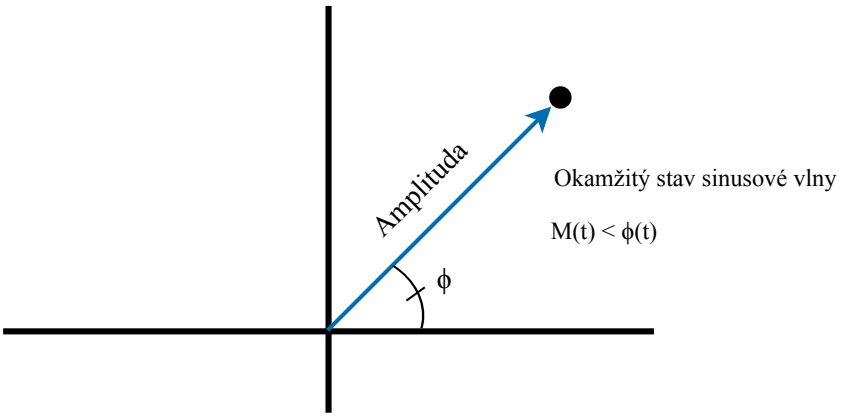
Oba tyto vztahy jsou popsány v knize *Moderní radiotechnika* [8, str. 33].

Pro práci se signálem je potřeba, aby ho bylo možné integrovat pomocí *Fourierova integrálu*. Tedy aby bylo možné nalézt konvergující výsledek integrálu a dále je důležité, aby frekvence byly shora omezeny [10].

Převod z časové do frekvenční domény se nazývá vzorkování. Vzorkovací frekvence je definována jako počet vzorků za jednotku času načítaných ze spojitého analogového signálu při přeměně na diskrétní signál. Ta musí být větší než je dvojnásobek nejvyšší frekvence harmonických složek obsažených ve vzorkovaném signálu, tzv. Shannonův teorém (někdy také nazývaný Nyquistův nebo Shannon-Nyquistův teorém). [11] Pokud není splněna podmínka vzorkovacího teorému, dochází k překrytí frekvenčních spekter vzorkovaného signálu a tedy ke ztrátě informace, resp. aliasingu. Z tohoto důvodu je nezbytné, aby byly frekvence shora omezeny.

Vzorkování se nejčastěji provádí ve stejných časových intervalech takových, že časový interval je násobkem periody T. Tím se signál navzorkuje uniformně. Teorie se zabývá náhodným vzorkováním, které má v některých ohledech lepší vlastnosti než uniformní vzorkování. Náhodné vzorkování je ale náročné převést do praxe, protože vzorkovat v náhodných intervalech je prakticky velmi těžko dosažitelné. [10]

Fourierova transformace platí pro všechny funkce takové, které mají v daných diskrétních bodech stejné hodnoty. Tím se vysvětluje, proč vzniká aliasing. Vzniká podvzorkováním signálu a tudíž funkce není jednoznačně určena. Tuto vlastnost je ale možné využít. Pokud se signál cíleně podvzorkuje, frekvence se „stáhne dolů“ a nedoje ke ztrátě informace. [10]

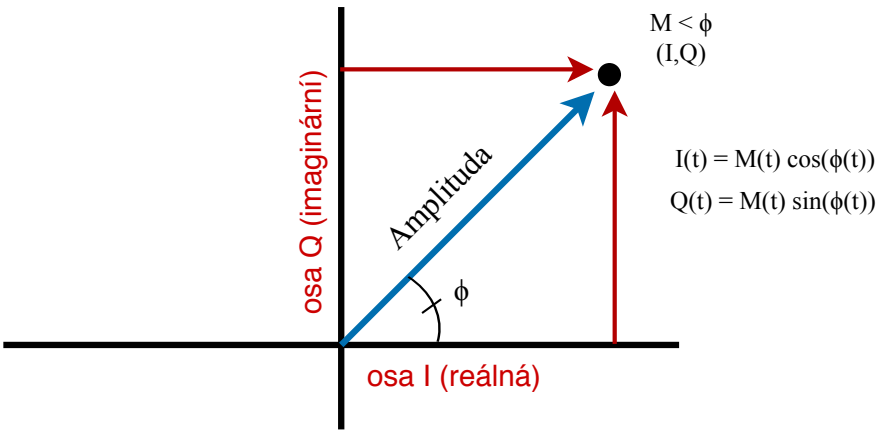


Obrázek 1.3: Zobrazení v komplexní rovině [12, převzato]

1.2 Zpracování signálu – fázor a I/Q data

Sinusovou vlnu lze vyjádřit jako $A_c cos(2\pi f_c t + \phi)$, kde A_c je amplituda, $2\pi f_c t$ je frekvence a ϕ fáze, ty tvoří fázový úhel. Okamžitý stav sinusové vlny je možné zobrazit v komplexní rovině pomocí fázoru, jako je to vidět na obrázku 1.3. Fázor je rotující orientovaná šipka umístěná v počátku soustavy souřadné, velikost fázoru je rovna amplitudě vlny a fáze je odchylka od rovnovážné polohy. Fáze bodu se mění podle aktuálního stavu sinusové vlny [13].

Podobně lze reprezentovat I/Q data. I/Q data jsou pouze převodem amplitudových a fázových dat z polárního souřadnicového systému do kartézského souřadnicového systému. Pomocí trigonometrie mohou být informace o sinusových vlnách převáděny mezi těmito systémy [12]. Z obrázku 1.4 je vidět, že obě tato vyjádření jsou ekvivalentní. Dle [14] použitím vektorového I/Q modulátoru lze vyjádřit velkou část modulačních metod. Díky nim je možné jakýko-



Obrázek 1.4: Kartézský souřadnicový systém pro I/Q data [12, převzato]

liv vysokofrekvenční signál o konstantní úhlové frekvenci a libovolné časově proměnné fázi i amplitudě zobrazit v komplexní rovině jako fázor. Fázor se skládá ze dvou kvadraturních složek se stejnou frekvencí a se vzájemnou fází 90 stupňů. První složkou je reálná složka I (In-Phase), druhou potom imaginární Q (Quadrature-Phase). Toho využívá tzv. klíčování. Více o klíčování v kapitole 1.3.2 věnované digitálním modulacím.

1.3 Modulace

1.3.1 Analogové modulace

Spojitá harmonická vlna $u_c(t) = U_c cos(2\pi f_c t)$ se moduluje obecným modulačním signálem $m(t)$. Ten může mít charakter napětí nebo proudu. [8, str. 108] Podle toho, který parametr se moduluje, se jedná o amplitudovou, frekvenční, nebo fázovou modulaci.

1.3.1.1 AM

Amplitudová modulace AM je jednou z nejjednodušších analogových modulací. Může vzniknout například tak, že pokud se mění okamžitá amplituda napětí modulované nosné vlny lineárně s modulačním napětím $m(t)$, a zároveň její relativní fáze vůči fázi nemodulované nosné vlny zůstává konstantní. Má obě postranní pásma, horní a dolní, a nepotlačenou nosnou vlnu. Frekvence nosné vlny se nemění, stejně tak jako její fáze. Informace jsou přenášeny pomocí změny amplitudy. U AM se její okamžitá amplituda mění okolo své střední hodnoty U_c lineárně s modulačním signálem $m(t)$. [8, str. 108-109]

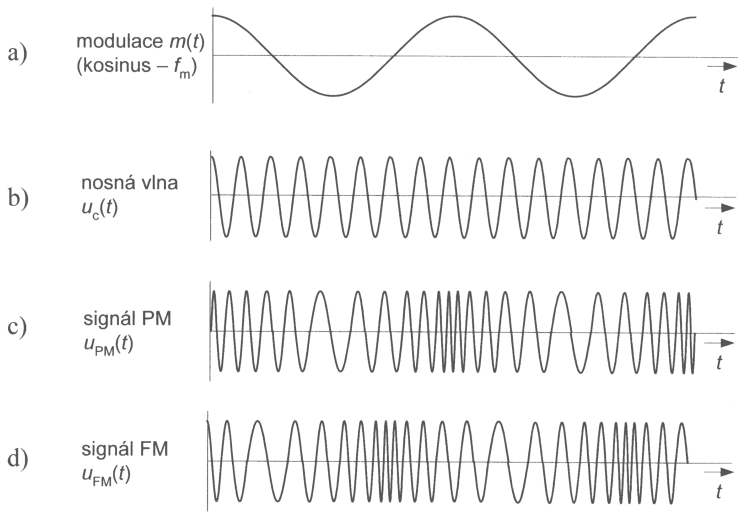
1.3.1.2 FM

Frekvenční modulace FM je podstatně výhodnější modulací, nežli starší AM. Spadá do kategorie úhlových modulací a její amplituda je konstantní, stejně tak jako u fázové modulace PM. Principem je závislost okamžité frekvence nosné vlny na změnách amplitudy modulačního signálu. Informace se přenáší změnou frekvence nosné vlny. Okamžitou frekvenci $f_i(t)$ můžeme vyjádřit jako součet f_c a časově proměnné složky $k_{FM}m(t)$, kde $m(t)$ je modulační napětí a k_{FM} je frekvenční citlivost modulátoru FM, což nám dává vztah popsaný v [8, str. 133]:

$$f_i(t) = f_c + k_{FM}m(t) \tag{1.4}$$

Zintegrováním podle času t a vynásobením 2π dostaneme vztah pro okamžitou fázi signálu FM a pokud budeme předpokládat, že v čase $t = 0$ je fáze modulované nosné vlny též nulová, získáváme obecný vztah popsaný v knize [8, str. 134] pro frekvenčně modulovaný signál v časové oblasti:

$$u_{FM}(t) = U_c cos[2\pi f_c t + 2\pi k_{FM} \int_0^t m(t)\mathrm{d}t] \tag{1.5}$$



Obrázek 1.5: Časové průběhy signálů využívaných v úhlových modulacích [8, str. 135]

1.3.1.3 PM

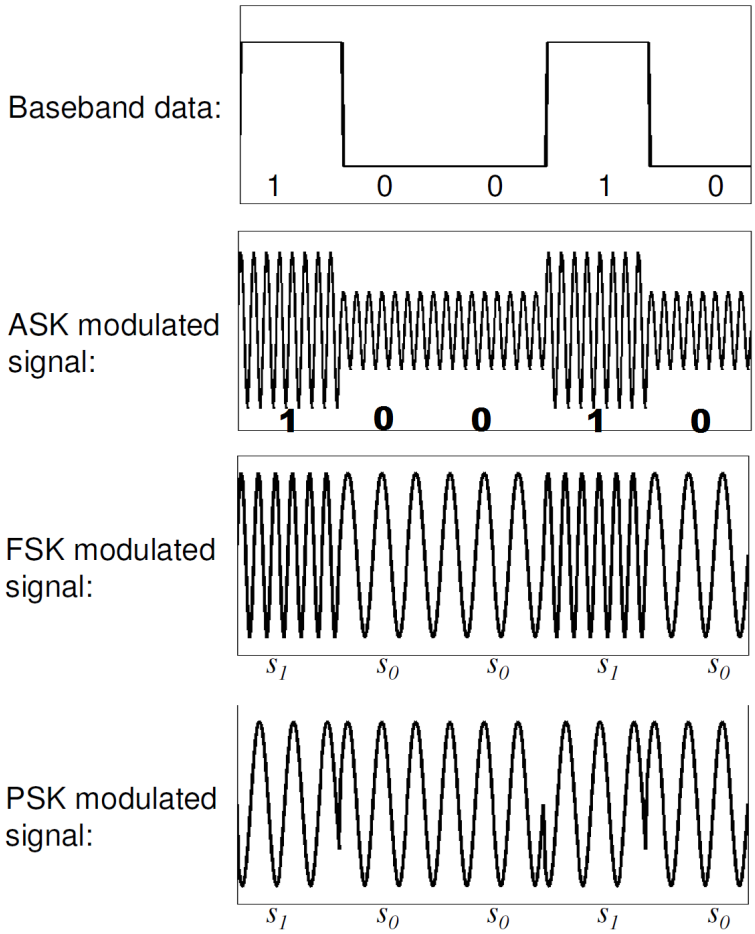
Fázová modulace PM je variantou úhlové modulace, stejně jako FM. Okamžitý fázový úhel modulovaného signálu je roven součtu fázového signálu nemodulované nosné vlny a časově proměnné složky, přímo úměrné modulačnímu napětí [8, str. 134].

Z obrázku 1.5 je patrné, že časové průběhy modulovaných signálů FM a PM jsou při sinusové modulaci prakticky stejné. K jejich jednoznačné identifikaci je proto nutné tyto časové průběhy signálů porovnávat s modulačním průběhem [8, str. 135].

1.3.2 Digitální modulace

Dle [8, str. 90] digitální modulace vznikají tak, že se vysokofrekvenční nebo mikrovlnná sinusová nosná vlna moduluje signálem některé diskrétní modulace v základním pásmu. Protože modulační signál byl již modulován, mluví se zde o dvojnásobné modulaci. Tím prvním modulačním signálem často bývá binární signál PCM.

Stejně jako u analogových modulací je možné modulovat amplitudu, frekvenci, anebo fázi nosné vlny. U dvoustavových (binárních) modulací se modulovaný parametr mění mezi dvěma diskrétními stavy, tedy mezi binární 0 a binární 1. Obecně se tyto stavy u digitálních modulací označují jako symboly. Pro uvažované digitální modulace se používá rovněž termín klíčování.



Obrázek 1.6: Zobrazení digitálních modulací na základě binárního vstupu [15]

1.3.2.1 Modulace ASK – Amplitude Shift Keying

Modulace ASK, tj. amplitudové klíčování, často také označovaná jako modulace OOK (On-Off Keying), je nejčastěji dvoustavovou modulací BASK (2ASK). Informace se přenáší pomocí změn amplitudy modulovaného signálu. Pokud je nosný kmitočet přítomný, odpovídá to logické jedničce a naopak, pokud přítomný není, vysílá se logická nula. Takový přenos je možné pozorovat na obrázku 1.6. Frekvence ani fáze se u této modulace nemění.

1.3.2.2 Modulace PSK – Phase Shift Keying

Modulace PSK neboli modulace s klíčováním fázovým posuvem je v nejjednodušší podobě BPSK (2PSK) dvoustavovou modulací. Její amplituda je konstantní, informace jsou přenášeny změnou fáze v modulovaném signálu,

kde fáze nabývá dvou diskrétních hodnot, např. 0 a 180 stupňů. [8, str. 229] Tuto modulaci je možné opět vidět na obrázku 1.6.

1.3.2.3 Modulace FSK – Frequency Shift Keying

Modulace FSK známá též jako klíčování frekvenčním posuvem nebo kmitočtové klíčování je dvoustavovou modulací. Proto se pro ni někdy používá název BFSK (Binary Frequency Shift Keying) nebo 2FSK. Má konstantní amplitudu a informaci přenáší změnou frekvence. Ta se mění v rytmu digitálního binárního modulačního signálu mezi dvěma signalizačními frekvencemi $f_1 = f_c - \Delta f$ a $f_2 = f_c + \Delta f$, kde f_1 značí první nosnou vlnu a f_2 druhou. Změna může být buďto spojitá, anebo nespojitá. Nespojitosti v průběhu signálu s nespojitou fází se projeví nežádoucími postranními složkami jejího frekvenčního spektra, proto je frekvenční spektrum značně širší než spektrum, které zabírají nosné vlny. [8, str. 225]

1.4 Komunikační módy

Komunikační mód specifikuje možnosti zařízení mezi sebou komunikovat ve smyslu přijímání a odesílání dat. Celá tato podkapitola čerpá z knihy *Network Communications Technology* [16, str. 25]. Rozlišujeme několik druhů komunikace v závislosti na tom, jak mezi sebou mohou zařízení komunikovat a to na módy:

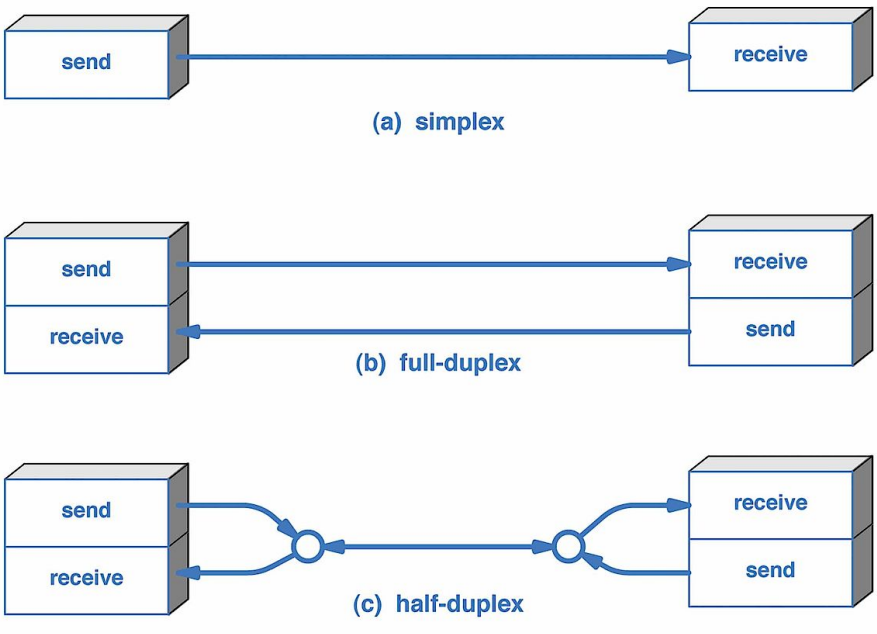
1. simplex,
2. half-duplex,
3. full-duplex.

1.4.1 Simplex

V simplex módu jsou data přenášena jen jedním směrem. Jedná se tak o jednosměrnou komunikaci, kdy vysílající zařízení nikdy neočekává odpověď od přijímajícího zařízení. To je tedy pouhým přijímačem a neobsahuje vysílající hardware. Typickým příkladem je rádio nebo TV vysílání.

1.4.2 Half-duplex

Half-duplex mód implementuje komunikaci oběma směry. Nicméně limitujícím faktorem je, že v jednu chvíli mohou být data přenášena pouze jedním směrem. Zde typickým příkladem mohou být např. vysílačky, walkie-talkie apod.



Obrázek 1.7: Komunikační módy [17]

1.4.3 Full-duplex

Ve full-duplex módu mohou obě zařízení vysílat i přijímat ve stejnou chvíli. Příkladem mohou být dnešní telefony, které umožňují oběma stranám jak mluvit, tak i poslouchat. Graficky je komunikace zobrazena na obrázku 1.7.

Kapitola 2

Analýza

Tato kapitola je věnována seznamu útoků, které se běžně provádějí, a jejich popisu. Dále se zabývá přehledem dostupného softwaru pro analýzu radiových signálů, popisu frameworku GNU Radio. V dalších podkapitolách je popsán hardware použitý v této práci, příp. další zařízení, která mohou být použita pro tyto účely.

2.1 Bezdrátové útoky

Vzhledem k dnešní dostupnosti různých SDR řešení je provádění útoků na bezdrátovou komunikaci velice dostupné. V této podkapitole jsou popsány útoky od toho nejjednoduššího, tzv. zarušení až po nejtěžší část útoků, tampering.

2.1.1 Jamming attack

Jamming attack, do češtiny volně přeloženo jako útok rušením, je zaměřený na narušení fyzické vrstvy. Rušení může být definováno jako vysílání nesrozumitelného signálu, který způsobuje zarušení frekvenčního pásma a znemožňuje tak legitimním zařízením v komunikaci [1]. Existuje několik způsobů útoků, které jsou popsány v práci *Detekce útoku úmyslného rušení v bezdrátových senzorových sítích* [18].

Neustálý útočník – Constant jammer vysílá nepřetržitě, nejčastěji pomocí generátoru průběhu vlny (waveform generator).

Klamavý útočník – Deceptive jammer vysílá regulérní pakety.

Náhodný útočník – Random jammer přepíná mezi stavy klidu a rušením.

Reagující útočník – Reactive jammer je oproti ostatním „neaktivní". Pokud na kanálu není žádný provoz, tak nevysílá. Ve chvíli kdy zaznamená aktivitu, začne rušit. Šetří tím energii a je velmi špatně detekovatelný.

Tato práce se zabývá zejména útoky typu neustálého rušení. Proti tomuto typu útoku je velmi obtížné se bránit.

2.1.2 Replay attack

Replay attack (též známý jako playback attack), volně přeloženo do češtiny jako útok přehráním, je dalším jednoduše proveditelným útokem. Jeho podstata spočívá v odposlouchávání a zachycení šifrovaného signálu. Ve chvíli, kdy má útočník uložený signál u sebe, může ho kdykoliv opakovaně přehrát. Obrovskou výhodou z pohledu útočníka je, že nepotřebuje pokročilé znalosti k dešifrování zprávy. Útok může být úspěšný díky pouhému přeposlání zachyceného signálu. [19]

Obrana proti tomuto typu útoku je popsána v článku [19] a může být implementována několika způsoby.

1. Jak odesílatel, tak příjemce si zřídí náhodný klíč relace, který je validní jen pro tuto jedinou relaci a nemůže být dále použit.

2. Současně se zprávou se odesílá i časová známka. Snižuje se tak doba, kdy může útočník přijmout a znovu odeslat nahranou zprávu.

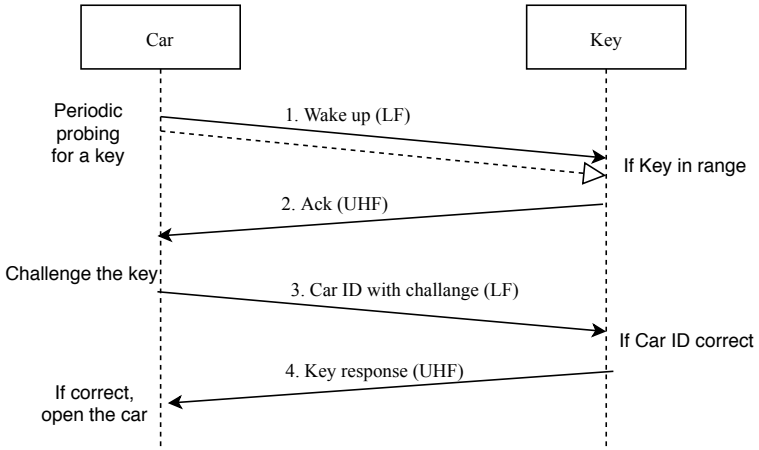
3. Použít heslo, které je unikátní pro každou transakci a po použití je vyřazeno. Tím se opět zajistí, že při pozdějším přehrání už zpráva není validní.

4. Použití tzv. rolling code. Jak příjemce, tak odesílatel obsahují kryptograficky bezpečný PRNG pro generování kódů. Pokud příjemce přijme signál, porovná ho s posledním ověřeným číslem synchronizace a pokud kódy odpovídají, přijme zprávu [20]. Tyto systémy se často používají v bezklíčových systémech aut nebo u garážových vrat.[1]

Protože se často jedná o zařízení s nízkou spotřebou, žádný z výše zmíněných přístupů nebývá implementován a zařízení jsou zranitelná tímto typem útoku.

Pokročilejším útokem je kombinace replay attacku a jammingu zároveň. Útočník zaruší frekvenční pásmo a zároveň se snaží zachytit legitimní signál pro další použití. Ve chvíli, kdy se oběť pokusí opětovně signál odeslat, útočník odešle svůj uložený signál a opět zachytí nový signál. Tím dojde k akci, kterou oběť očekává, ale zároveň útočník získá poslední platný signál. Praktické provedení takového útoku je popsáno v podkapitole 5.2.1.2.

[1] Příklad implementace takového systému od firmy Atmel `http://ww1.microchip.com/downloads/en/AppNotes/Atmel-2600-AVR411-Secure-Rolling-Code-Algorithm-for-Wireless-Link_Application-Note.pdf`



Obrázek 2.1: Příklad realizace protokolu PKES [22, převzato]

2.1.3 Relay attack

Relay attack je zaměřen na tzv. *Passive keyless entry and start* (PKES) systémy, které jsou podobné běžným dálkovým ovládáním zamykání u aut, nevyžadují však žádný stisk tlačítka na klíči a klíč tak může zůstat v kapse uživatele. Jak je vidět na obrázku 2.1, jedná se klasický challenge-response protokol.

Existují dvě možnosti, jak takový útok provést. První možností je samotný relay attack, druhou potom amplified relay attack. Princip útoků je stejný a využívají stejných zranitelností, liší se však realizací útoku.

Relay attack využívá dvě zařízení, která mezi sebou komunikují. První zařízení je umístěno v blízkosti cíle (držitele klíče) a přenáší signály od klíče do druhého zařízení. To je umístěno v blízkosti auta a umožňuje útočníkovi posílat signál autu. Je tak vytvořen tunel mezi útočníkovými zařízeními, který může používat odlišný typ přenosu, než používá samotný klíč a auto. To útočníkovy umožňuje použít kanál, který dovoluje přenos dat na daleko větší vzdálenosti.

Amplified relay attack využívá pouze jedno zařízení, které funguje jako zesilovač. Útočník je v tomto případě v blízkosti auta a zesiluje signál mezi autem a klíčem. Zde je vhodné využít např. směrové antény pro dosažení nejlepších výsledků, tj. dosáhnout na co největší vzdálenost. Pokud se klíč nachází v tomto dosahu, je opět možné auto odemknout a nastartovat.

Princip útoků je tedy stejný, přijímač auta si myslí, že je klíč v jeho blízkosti a umožní odemknout či nastartovat auto. Více o těchto útocích se může čtenář dočíst v [21] a [22].

2.1.4 Tampering

Tampering je nejnebezpečnějším typem útoku a zároveň pro útočníka tím nejnáročnějším. Jedná se o změnu přenášených dat skrze neautorizovaný přístup. Prvním krokem útočníka je zachycení přenášené komunikace, to přináší hned několik obtíží. Protože jsou data přenášena na sdíleném médiu a zařízení typicky používají frekvence, která nejsou zatížena licenčními poplatky, může docházet k rušení a interferencím s ostatními signály. Proto je potřeba použít směrové antény k zachycení signálu, signál dále filtrovat a odstranit tak nežádoucí šum.

Následně je potřeba zjistit použitou modulaci, aby bylo možné získat binární data. K tomu mohou být použity nástroje pro analýzu signálů popsané v podkapitole 2.2, příp. v USA musí mít každý výrobek, používající rádiové frekvence, unikátní FCC[2] ID a musí projít testováním v nezávislé laboratoři, aby bylo zajištěno, že splňuje FCC standardy. Výsledné testy a reporty často obsahují použité frekvence, modulace a další užitečné informace a jsou dostupné veřejnosti na stránkách `https://www.fcc.gov/oet/ea/fccid`.[3]

Po úspěšné demodulaci a získání binárních dat následuje analýza protokolu přenášené zprávy. Neboť protokol většinou není znám, je potřeba reverzním inženýrstvím identifikovat přenášená data. Nejjednodušším způsobem je dlouhodobě sledovat komunikaci, porovnat jednotlivé zprávy a identifikovat měnící se části zprávy.

Dalším krokem po identifikaci protokolu je změna dat, která může být provedena změnou jednotlivých bitů. Často také bývá přítomný nějaký typ kontrolního součtu, který je nutné přepočítat. Po provedení změny dat následuje opětovná modulace binárních dat, která není obtížná, díky předešlé analýze. Na závěr jsou data opět odeslána. Pokud celá komunikace není šifrovaná, je velmi obtížné se bránit proti takovému útoku.

2.2 Dostupný software a existující řešení

Níže jsou zmíněny některé programy, které byly využity během psaní této práce. Jedná se zejména o nástroje vhodné pro analýzu frekvenčního spektra, příjem dat, dekódování apod.

- GNU Radio
- GQRX
- Inspectrum
- Baudline

[2] Federal Communications Commission

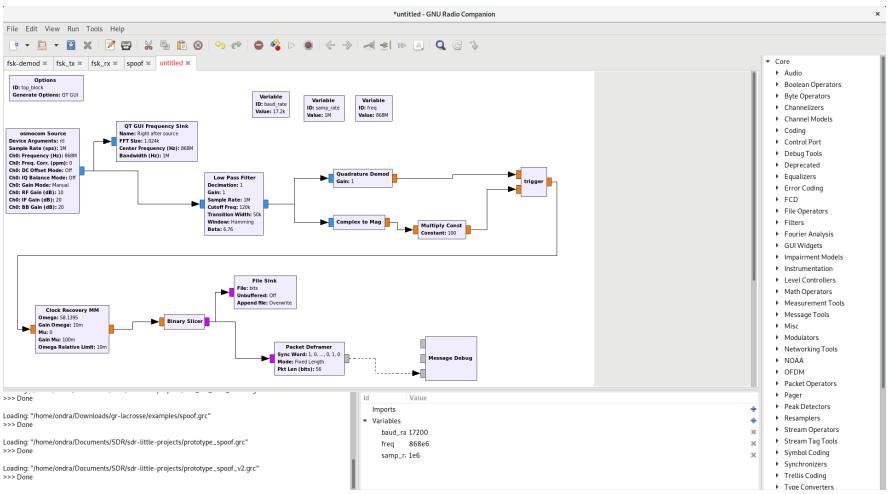
[3] Alternativně lze použít stránky s příjemnějším rozhraním `https://fccid.io/`

- Universal Radio Hacker
- RF Analyzer

2.2.1 GNU Radio

Jedná se o vývojový nástroj pro aplikace na zpracování signálu, který nabízí framework pro vytváření systémů pro SDR. Je distribuovaný pod GNU GPL licencí a většina kódu je licencována jako svobodný software. GNU Radio je kompletně napsané v C++ a podporuje též jazyk Python, který se často využívá pro přidávání vlastních nástrojů a modulů. Aplikace vytvářené nad tímto frameworkem jsou známy jako „flography“ a mohou být napsané v C++ nebo v Pythonu.

GNU Radio Companion (GRC) je grafické rozhraní pro vývoj aplikací nad frameworkem GNU Radio a je jeho součástí od verze 3.2.0 [23]. GRC obsahuje tlačítko „Generate the flowgraph“, které vygeneruje kód v Pythonu pro daný flowgraph. Ten může být následně dále upravován. Grafické rozhraní se dělí na několik částí, výchozí rozložení je vidět na obrázku 2.2. Pravá část obsahuje

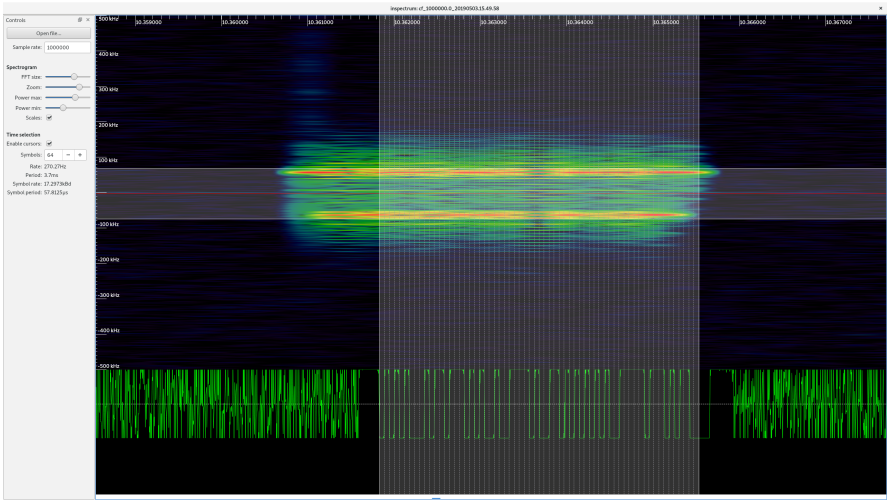


Obrázek 2.2: Pracovní plocha GNU Radio Companion [23, pořízeno v GRC]

všechny dostupné bloky roztříděné do kategorií podle typu, spodní část obrazovky zabírá konzole pro výpis spolu se správcem proměnných. Zbylou část plochy zabírá tzv. „canvas“ neboli plátno pro samotný vývoj aplikace.

2.2.2 Inspectrum

Inspectrum je nástroj pro analýzu zachycených signálů, který dovoluje nahrávat soubory o velikosti větší než 100 GB a vizualizovat je na spektrogramu. Prostředí programu je vidět na obrázku 2.3.



Obrázek 2.3: Prostředí programu Inspectrum [24, pořízeno v Inspectrum]

Inspectrum umožňuje zobrazit různé typy grafů:

- amplitudy
- frekvence
- fáze
- I/Q vzorků

Další velmi užitečnou vlastností je použití kurzoru pro měření periody, symbol rate a extrahování symbolů stejně tak jako export vzorků a demodulovaných dat. [24]

2.2.3 Universal Radio Hacker

Universal Radio Hacker (URH) je nástroj pro analýzu bezdrátových protokolů. Umožňuje komplexní analýzu signálů v několika krocích.

1. Interpretace
2. Analýza
3. Generování
4. Simulace

Zatímco GNU Radio je hodně nízkoúrovňové, URH nabízí uživateli možnost provádět analýzu signálu s minimem znalostí z oblasti zpracování digitálního signálu [25].

2.3 Použitý hardware

Pro příjem signálu bylo použito zejména zařízení RTL SDR, později i HackRF One. Pro aktivní útoky (vysílání) bylo použito HackRF One a USRP B210.

2.3.1 RTL SDR

Původem se jedná o laciný DVB-T TV přijímač, který je založený nejčastěji na A/D převodníku RTL2832U (odtud název RTL SDR) od firmy Realtek a tuneru Rafael Micro R820T. Existují i webové stránky věnované téměř výhradně RTL [26], jsou zde popsány technické specifikace RTL stejně jako mnoho dalších informací.

Chip RTL2832U umožňuje přenos surových I/Q vzorků do počítače a lze tudíž použít jako SDR. Možnost získání právě surových I/Q dat byla objevena Ericem Fry. Na jeho práci navázal Antti Palosaari a za dnes nejrozšířenějším ovladačem stojí potom tým Osmocom. Největším nedostatkem RTL je použití pouze 8 bitového A/D převodníku, což se ale dá pochopit vzhledem k ceně. Ta se během psaní práce pohybuje okolo 160 Kč za nejlevnější kusy.[4]

V této práci je použito zařízení s tunerem R820T2, který je jen novější verzí R820T. To nabízí možnost příjmu v rozsahu od 24 do 1 766 MHz s teoretickou propustností až 3,2 Msps (milion vzorků za vteřinu). Aby se předešlo ztrátě vzorků, je doporučená maximální použitelná datová propustnost 2,4 Msps. Po vyzkoušení a odladění se jako nejlepší osvědčilo použití 2,8 Msps, při kterém ještě tuner fungoval stabilně. Použitý tuner disponuje anténním konektorem MCX. Ten nabízí schopnost širokopásmového připojení od DC až po 6 GHz, což pro účely této práce v kombinaci s použitým RTL postačuje.

2.3.2 HackRF One

Nejdostupnějším řešením jak pro příjem, tak zejména pro vysílání, je zařízení HackRF One. Jedná se o open source projekt vytvořený Michaelem Ossmannem. Zařízení nabízí half-duplex transceiver (transmitter a receiver zároveň), který umožňuje pracovat ve frekvenčním rozsahu od 1 MHz až po 6 GHz. Vzorkovací frekvence je od 2 do 20 Msps. HackRF disponuje 8 bitovými A/D a D/A převodníky a komunikaci s hostitelským zařízením obstarává USB verze 2. Anténní konektor je zde použit standardní SMA s odporem 50 ohmů. Dále HackRF nabízí SMA vstup a výstup pro synchronizaci hodin, tlačítko Reset pro jednoduchý restart a tlačítko DFU pro programování zařízení. Uvnitř se potom nachází piny pro případná další rozšíření [27].

[4] Např. na Ebay `https://www.ebay.co.uk/sch/i.html?_nkw=r820t&_sop=15`

2.3.3 USRP™ B210

USRP B210 pochází ze série *Bus* od Ettus Research. Níže jsou sepsány technické parametry ze stránek Ettus Research [28]. Jedná se o jednodeskové zařízení s frekvenčním rozsahem od 70 MHz po 6 GHz, které nabízí šířku pásma až 56 MHz. B210 umožňuje full-duplex komunikaci a to dokonce v konfiguraci 2x2 MIMO.[5] O převod signálu se starají A/D a D/A převodníky s rozlišením 12 bitů a připojení k počítači probíhá přes rozhraní USB verze 3.

Díky MIMO konfiguraci tak B210 nabízí na přední straně dva SMA konektory pro vysílání a dva konektory pro příjem. Na zadní straně se potom nachází SMA konektory pro anténu GPS a časovou synchronizaci. Všechny konektory mají odpor 50 Ohm.

2.3.4 Další dostupná řešení

Dalším vhodným adeptem bylo zařízení z rodiny Universal Software Radio Peripheral od Ettus Research, konkrétně zařízení N210 dostupné v RFID laboratoři na naší fakultě. To umožňuje jak příjem, tak vysílání. Bohužel limitujícím faktorem je instalovaná deska pracující v rozsahu od 0 až do 30 MHz a tudíž nepoužitelná pro požadované účely. Jinou desku se během vzniku této práce nepodařilo sehnat.

Obdobným zařízením jako je HackRF, je i bladeRF, které oproti HackRF nabízí full-duplex konektivitu, propojení přes USB verze 3 s vyšší propustností a A/D a D/A převodníky s 12 bitovým rozlišením.

2.4 Meteostanice

Jako cíl útoku byla vybrána meteostanice a to z několika důvodů. Bezdrátová čidla jsou nejčastěji napájena bateriemi a tudíž je kladen velký důraz na spotřebu zařízení. Z toho plyne, že kvůli energetické náročnosti není implementováno žádné zabezpečení komunikace. A pokud si představíme situaci, kdy bude základnová stanice připojena na další systémy jako topení, protipožární systém apod., a bude do těchto systémů odesílat sesbírané informace z čidel, hrozí reálné napadení útočníkem a napáchání škod.

Cílem útoku se stala meteostanice TFA 30.3032.01.IT[6] a bezdrátový teploměr TFA 30.3143.IT[7] od výrobce TFA Dostmann/Wertheim. Teploměr pro komunikaci se základnovou stanicí používá protokol Instant Transmission (IT+) na frekvenci 868 MHz. Jedná se o simplexní komunikaci, tedy teploměr vysílá ve směru ke stanici. Použitá modulace je 2FSK, bitrate cca 17 200 b/s a přenášený paket má 64 bitů [29].

[5] Multiple-input multiple-output s možností 2 Tx a 2 Rx zároveň

[6] `http://www.mkoptik.cz/doplnkovy-sortiment/detail/termometr-easy-go-tfa-30_3032_01_it/240`

[7] `https://www.tfa-dostmann.de/en/produkt/temperature-transmitter-3/`

2.5 Dálkové ovládání centrálního zamykání vozu

Dalším použitým zařízením v této práci je dálkové ovládání centrálního zamykání vozu Ford Focus Mk 2 (FF2). Klíč používá pro odemknutí či zamknutí vozidla bezdrátovou komunikaci na frekvenci okolo 433,92 MHz. Opět se zde jedná o simplexní komunikaci, klíč je vysílačem a vozidlo přijímačem. To umožňuje odemykat/zamykat auto bez použití fyzického klíče. Takový systém je často označován jako *remote keyless entry system* (RKE) [30]. Nejčastěji se v takových systémech pro simplexní komunikaci používá protokol rolling code, jehož cílem je zabránit v provádění útoků typu replay attack.

2.5.1 Rolling code

Jak ve své práci [31, str. 26-29] uvádí Timo Kasper, v typické implementaci rolling code systému transponder obsahuje unikátní a veřejné sériové číslo, fixní diskriminační hodnotu, vnitřní čítač a klíč K pro šifrování. Diskriminační hodnota slouží jako další úroveň ochrany před zasíláním náhodných kódů. Vnitřní čítač je inkrementován či generován PRGN během každého vyslání signálu.

Při každém přenosu tak dochází k inkrementaci čítače, který je spojen s diskriminační hodnotou a funkční hodnotou, tj. pokud transponder obsahuje více než jednu funkci (transponder u FF2 obsahuje celkem tři funkce). Celá tato sekvence je zašifrována s použitím šifrovacího klíče K a výsledný šifrový text se nazývá rolling code. Ten se spojí s nezašifrovaným sériovým číslem klíče a celá zpráva se odešle k přijímači.

Když přijímač přijme signál, porovná přijaté sériové číslo s vlastním seznamem uložených sériových čísel a pokud najde shodu mezi sériovými čísly, zjistí klíč K. Poté je šifrový text dešifrován a porovná se diskriminační hodnota s uloženou diskriminační hodnotou. Pokud jsou hodnoty stejné, zjistí se hodnota čítače, která musí být v rozsahu platných hodnot. Jestliže je hodnota čítače platná a všechny předchozí podmínky tak byly splněny, přístup je povolen a rozsah platných hodnot v přijímači se posune. V opačném případě, pokud některá z podmínek selže, je přístup odepřen.

Kapitola 3

Návrh

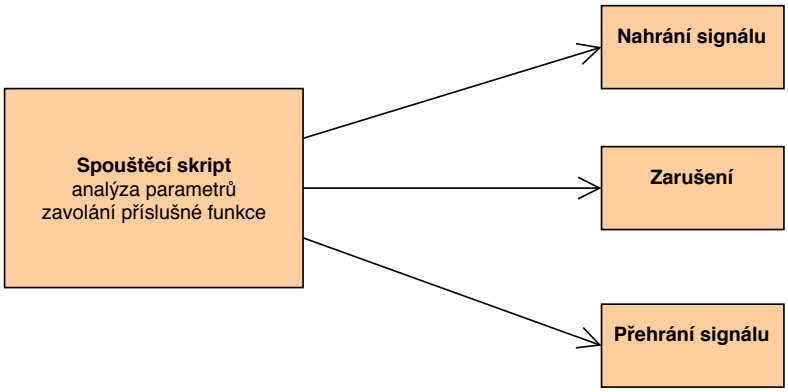
Základní principy fungování SDR byly zmíněny v kapitole 1, prostředí GNU Radio popsané v kapitole 2.2.1 a různými typy útoků prováděných pomocí softwarově definovaných rádií popsaných již dříve v kapitole 2.1.

Tato kapitola se zabývá návrhem jednotlivých funkcí v prostředí GNU Radio Companion, které budou použity jako základ ve výsledné aplikaci. Na obrázku 3.1 je vidět, jak by mohla vypadat architektura aplikace.

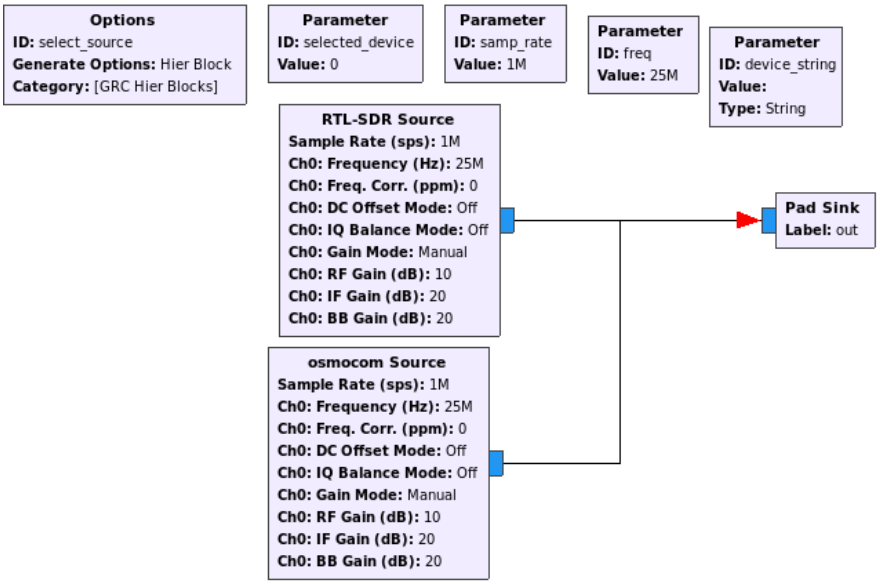
3.1 Výběr SDR

Aby aplikace nabízela možnost použití s různými typy SDR, je potřeba zajistit, že bude vždy vybrán správný ovladač pro konkrétní typ SDR. Ceny SDR se pohybují od desítek dolarů za ta nejlevnější až po tisíce dolarů za profesionální rádia, tomu odpovídá i funkční vybavenost rádií a tedy možnosti nastavení.[8]

[8] Kompletní seznam komerčně dostupných rádií je zde `https://en.wikipedia.org/wiki/List_of_software-defined_radios`.



Obrázek 3.1: Návrh architektury aplikace



Obrázek 3.2: Příklad hierarchického flowgraphu [23, pořízeno v GRC]

GNU Radio podporuje zejména amatérská a poloprofesionální rádia, níže je seznam několika dostupných ovladačů:

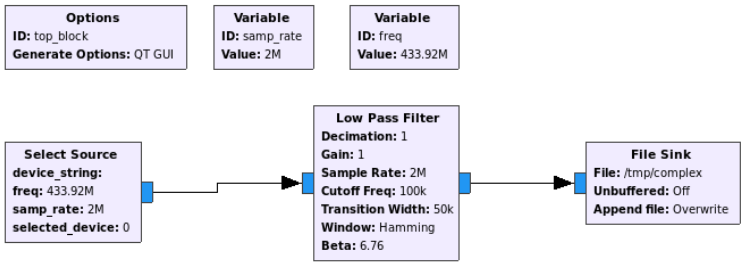
- UHD, USRP Hardware Driver™
- Osmocom[9]
- Funcube Dongle
- LimeSDR[10]

Pro vývoj vlastního modulu z již dostupných bloků nabízí GNU Radio Companion užitečný nástroj, díky kterému po vygenerování kódu vznikne nový modul. Toto nastavení je dostupné v bloku Options, kde je možné pod položkou Generate Options zvolit Hier Block. Po vygenerování flowgraphu vznikne tzv. hierarchický blok, který se po spuštění přidá do GRC a je okamžitě k dispozici pro používání. Příklad takového flowgraphu je vidět na obrázku 3.2.

Červená šipka na obrázku indikuje chybu, protože zapojení více výstupů do jednoho vstupu není možné. Připojení pouze jednoho výstupu je tak nutné zajistit v kódu, to je detailně popsáno v kapitole 4.3. Obdobným způsobem je možné vytvořit výběr SDR pro vysílání, kde je vstupem komplexní signál a na výstupu je vybráno správné SDR.

[9] Seznam podporovaných SDR skrze ovladač Osmocom je dostupný na stránkách `https://osmocom.org/projects/gr-osmosdr/wiki`

[10] Pro podporu LimeSDR je nutné doinstalovat ovladač dostupný na stránkách. `https://wiki.myriadrf.org/Gr-limesdr_Plugin_for_GNURadio`



Obrázek 3.3: Flowgraph pro nahrání signálu do souboru [23, pořízeno v GRC]

3.2 Návrh nahrání signálu

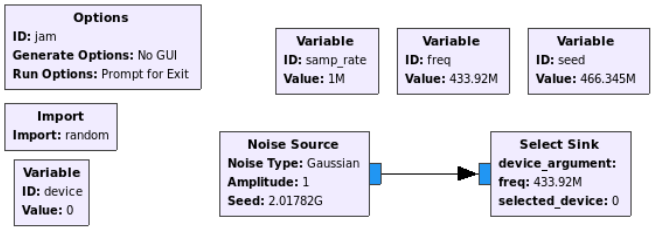
Skrze blok Select Source, vytvořený v předešlé kapitole 3.1, je vybráno správné SDR a komplexní signál je přiveden do dolní propusti (Low Pass Filteru). Odtud je signál veden do bloku pro uložení do souboru, tzv. File Sinku.

Je zde několik povinných parametrů pro nastavení SDR. Těmi jsou frekvence, vzorkovací frekvence a zvolený typ SDR. Na to navazují další nepovinné parametry, např. zisk, sériové číslo SDR nebo cesta pro uložení souboru.

3.3 Návrh přehrání signálu

Návrh na přehrání signálu je podobný předešlému návrhu. Parametrem je určena cesta k souboru, který má být přehrán. Následuje opět několik povinných parametrů pro nastavení a volbu SDR.

Pro výběr souboru je zde blok File Source, kterému je předána cesta k souboru. Protože v GRC není možné všechny argumenty specifikovat parametricky, některé budou muset být předány do bloku přímo v kódu. Mohou jimi být např. vstupní typ dat nebo opakované přehrání signálu. File source je potom spojen s blokem pro výběr SDR, tzv. Select Sink.



Obrázek 3.4: Flowgraph na rušení signálu [23, pořízeno v GRC]

3.4 Návrh rušení

Schéma pro rušení signálu je obdobné se schématem pro přehrání signálu. Zdrojem signálu je zde tzv. Noise Source a výstupem je opět blok Select Sink. U zdroje rušení je opět možné nastavit několik parametrů jako typ rušení nebo vstupní seed. Toto zapojení je možné vidět na obrázku 3.4.

3.5 Narušení komunikace

Po provedení analýzy a domluvě s vedoucím bylo rozhodnutu, že není možné navrhnout univerzální aplikaci pro tento typ útoku. Vždy je nutné analyzovat každý unikátní signál zvlášť a existuje zde mnoho proměnných, proto návrh takové aplikace není prakticky možný.

Kapitola 4

Implementace

Aplikace je kompatibilní s operačním systémem Linux a její chování bylo již detailně popsáno v kapitole 3. Tato kapitola se zabývá vlastní implementací aplikace a funkcí. Ty stěžejní jsou v podkapitolách popsány dopodrobna.

4.1 Použité technologie

Pro vývoj aplikace posloužil jazyk Python ve verzi 2.7.15 a framework GNU Radio, resp. aplikace GNU Radio Companion (GRC) ve verzi 3.7.13.4. GNU Radio je detailně popsáno v podkapitole 2.2.1.

Pro vývoj vlastních modulů do GRC byl použit nástroj `gr_modtool`, který usnadňuje vývoj a instalaci do GNU Radio Companion. Jak vytvářet bloky v jazyce Python pomocí tohoto nástroje je detailně popsáno na stránkách GNU Radio Guided Tutorial, GNU Radio in Python. Pro definici bloků GRC používá jazyk Extensible Markup Language (XML). Přehledný seznam dostupných modulů pro GNU Radio nabízí stránka `http://cgran.hopto.org/`

4.2 Struktura aplikace

Aplikace je rozdělena na spouštěcí skript `main.py`, který se stará o volání funkcí, a několik dalších podpůrných tříd. Spouštěcí skript má na starost syntaktickou analýzu vstupu z konzole, vytvoření objektu pomocné třídy SoftwareDefinedRadio popsané v podkapitole 4.3 a dle zadaných parametrů zavolání příslušné části kódu. Aplikace má několik povinných argumentů, mezi které patří frekvence, vzorkovací frekvence, zvolený mód a typ SDR zařízení. Na ukázce 1 je úryvek kódu zodpovědný za jejich analýzu. Mezi nepovinné argumenty potom patří zisk, cesta k souboru pro uložení signálu nebo vysílání ze souboru.

```
parser = argparse.ArgumentParser(description='Cli App')
    optional = parser._action_groups.pop()
    required = parser.add_argument_group('required arguments')

    required.add_argument('-m', '--mode',
                          choices=['jam', 'record', 'replay',
                                   'tamper', 'transmit'],
                          help='Set mode',
                          type=str.lower, required=True)
    required.add_argument('-f', '--frequency',
                          help='Set center frequency',
                          required=True)
    required.add_argument('-s', '--samp-rate',
                          help='Specify sample rate',
                          required=True)
    required.add_argument('-d', '--device',
                            choices=[constant.RTL, constant.HACKRF,
                            constant.BLADERF, constant.USRP],
                          type=str.lower,
                          required=True)
```

Výpis kódu 1: Analýza vstupu

4.3 Pomocné/podpůrné třídy

Vedle tříd implementujících hlavní funkce aplikace ještě obsahuje několik dalších tříd:

- `software_defined_radio.py`,
- `select_source.py`,
- `select_sink.py`,
- `constant.py`.

Třída SoftwareDefinedRadio obsahuje většinu parametrů, které je možné rádiím přiřadit. Jedná se zejména o frekvenci, vzorkovací frekvenci, parametry zisku nebo výběr konkrétního SDR na základě sériového čísla apod. Usnadňuje tak předávání parametrů v aplikaci, tj. stačí vždy předat objekt typu SDR a není nutné předávat každý argument zvlášť.

Třída SelectSource bere jako argument objekt typu SDR a implementuje výběr zdroje signálu, tj. na základě zadaných parametrů vybere příslušný ovladač SDR a nastaví mu parametry uvedené ve třídě SoftwareDefinedRadio. Takové nastavení je možné vidět na ukázce 2.

```
elif self.selected_device == constant.BLADERF or
self.selected_device == constant.HACKRF:
    self.osmosdr_source = osmosdr.source(args="numchan="
    + str(1) + " " + self.device_argument)
    self.osmosdr_source.set_sample_rate(self.samp_rate)
    self.osmosdr_source.set_center_freq(self.frequency, 0)
    self.osmosdr_source.set_freq_corr(0, 0)
    self.osmosdr_source.set_dc_offset_mode(0, 0)
    self.osmosdr_source.set_iq_balance_mode(0, 0)
    self.osmosdr_source.set_gain_mode(False, 0)
    self.osmosdr_source.set_gain(self.rf_gain, 0)
    self.osmosdr_source.set_if_gain(self.if_gain, 0)
    self.osmosdr_source.set_bb_gain(20, 0)
    self.osmosdr_source.set_antenna('', 0)
    self.osmosdr_source.set_bandwidth(0, 0)
```

Výpis kódu 2: Nastavení parametrů ovladače SDR

Třída SelectSink funguje obdobným způsobem, argumentem je opět objekt typu SDR a místo zdroje signálu zde funguje jako výběr výstupu signálu pro vysílání.

Soubor `constant.py` nabízí seznam konstant použitých v celé aplikaci a jejich jednoduchou správu.

4.4 Nahrání signálu

Třída Record pro nahrání signálu vychází z návrhu 3.2. Tato třída implementuje samotné nahrávání do souboru. Argumentem jí je třída SoftwareDefinedRadio, která obsahuje veškeré potřebné parametry pro nastavení a není nutné předávat další argumenty.

Díky třídě SoftwareDefinedRadio bylo nutné přepsat předávání argumentů do dalších tříd. Do bloku SelectSource je předán celý objekt SDR, File Sink má naopak nastavené některé parametry z třídy SDR.

4.5 Přehrání signálu

Obdobně je implementována třída Replay, která vychází z návrhu 3.3. Zde je implementováno přehrávání ze souboru. Argumentem je třída SoftwareDefinedRadio, která usnadňuje předávání argumentů. Pro výběr SDR se používá třída SelectSink, které je předána třída SDR. Výběr souboru se provádí v bloku File Source.

4.6 Rušení

Implementace třídy Jamming pro rušení vychází z návrhu 3.4. Tato třída implementuje rušení v daném frekvenčním pásmu. Stejně tak jako u ostatních tříd, i zde je argumentem objekt typu SoftwareDefinedRadio. Pro generování šumu je použit blok Noise Source z kategorie analogových bloků Analog. Tento blok je přímo spojen se třídou SelectSink, která obstarává výběr správného SDR.

4.7 Shrnutí

Aplikace je navržena pro maximálně jednoduché použití. Nesnaží se konkurovat komplexním nástrojům jako je Universal Radio Hacker popsaný v kapitole 2.2.3, ale spíše je doplňovat. Je určena pro rychlé a snadné použití, protože jí stačí nastavit několik základních přepínačů.

Kapitola 5

Testování

Aplikaci se podařilo naprogramovat a je možné ji k některým útokům využít. Testování proběhlo na meteostanici TFA popsané v kapitole 2.4 a na dálkovém ovládání vozu Ford Focus MK2 2.5. Výsledky testování jsou popsány níže v této kapitole.

5.1 Meteostanice

5.1.1 Jamming

Útok rušením je nejjednodušším útokem na provedení, podrobně je popsán v kapitole 2.1.1. Pro tento typ útoku bylo použito USRP B210 v kombinaci s anténou Vert900.[11]Ve všech případech byla vzorkovací frekvence nastavena na 1 Msps a frekvence na 868 MHz.

5.1.1.1 Test č. 1

Vzdálenost mezi teploměrem a metostanicí byla 2 metry, rádio bylo umístěno přímo na přímce mezi nimi. Všechna zařízení byla umístěna ve volném prostoru, tj. nebyla mezi nimi žádná překážka.

Při prvním běhu byl zisk nastaven na 30 dB. Tento útok byl spuštěn příkazem (obdobně je použit i v dalších testech):

```
python main.py -f 868250000 -s 2e6 -m jam -d hackrf -r 30
```

S takovým nastavením byla pozorovatelná aktivita v pásmu, na přenos dat to však vliv nemělo. V dalším pokusu byl zisk nastaven na 40 dB, kdy sice rušení pásma bylo už dobře viditelné, ale přenos dat probíhal i nadále bez větších výpadků. Během třetího pokusu byl zisk nastaven na 50 dB. Za této

[11]Podrobnější informace lze naleznout na stránkách výrobce `https://kb.ettus.com/Antennas#Vert900`.

situace už docházelo k častému výpadku a stanice nebyla schopna přijímat data v pravidelných intervalech. Ve čtvrtém pokusu byl použit zisk s hodnotou 55 dB. V tomto případě už stanice nebyla schopna přijímat žádná data a útok byl úspěšný. Po spuštění útoku meteostanice přestala přijímat signál od teploměru a po 10 minutách začala opětovně vyhledávat vysílače. Po dalších 5 minutách, během kterých bylo rušení stále aktivní a stanice nebyla schopná přijmout signál, přestala ukazovat teplotu úplně.

5.1.1.2 Test č. 2

Při druhém testu bylo změněno umístění zařízení, vzdálenost mezi teploměrem a meteostanicí byla 3 metry a SDR leželo mimo tuto přímku. Vzdálenost SDR od přijímače byla 3 metry a od teploměru 2 metry. Zařízení byla opět ve volném prostoru.

Při prvním pokusu v tomto uspořádání byl zisk nastaven na 55 dB. S tímto nastavením však komunikace probíhala bez problému, stejně tak, jako při druhém pokusu se ziskem 65 dB. Jako další byl použit zisk s hodnotou 75 dB, zde už byl pozorovatelný občasný výpadek v příjmu dat. Proto jako další hodnota byl použit zisk 80 dB, kde bylo možné pozorovat častý výpadek. Pásmo však stále nebylo zarušeno dostatečně a meteostanice byla schopná přijímat. Proto následoval další pokus se ziskem 85 dB, kdy už stanice nebyla schopna přijímat zprávy vůbec.

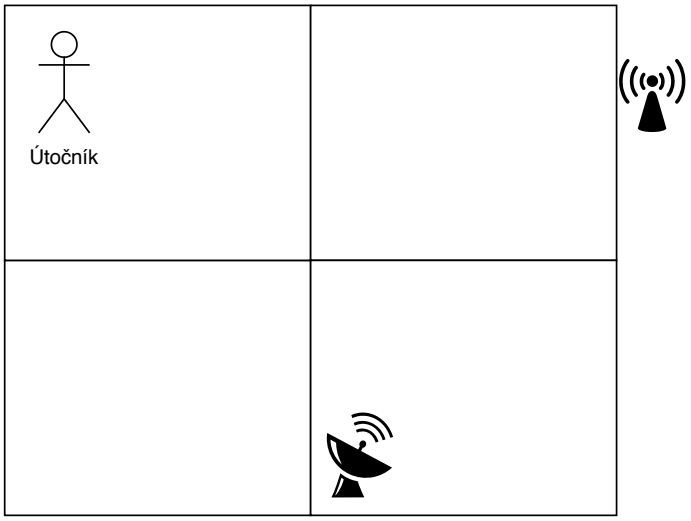
5.1.1.3 Test č. 3

Třetí test se snaží simulovat reálnou situaci, kdy útočník může mít vizuální kontakt s čidlem, ale nemá žádnou informaci o umístění přijímače, tj. meteostanice. Umístění bylo následující:

- Teploměr byl umístěn přímo za oknem.
- Meteostanice uvnitř místnosti a s teploměrem ji dělila zeď domu a jedna další zeď bytu.
- Útočník, resp. SDR bylo v místnosti oddělené dvěma stěnami od meteostanice.

Grafické znázornění je vidět na schématu 5.1.

První test proběhl rovnou s nastavením zisku na 85 dB. Při tomto rozmístění byl pozorovatelný častý výpadek signálu, přesto ale stanice v některých případech signál přijala.



Obrázek 5.1: Umístění útočníka, teploměru a stanice [32, 33]

5.1.1.4 Shrnutí

Během několika testů se ukázalo, že jamming attack je snadný na provedení. A v „laboratorních“ podmínkách byl také velmi úspěšný. Při třetím testu bylo ale vidět, že naopak v reálném prostředí účinnost může rychle klesat. Výsledky by se dalo vylepšit lepším vybavením, např. rádiem s vyšším výkonem nebo směrovou anténou.

5.1.2 Replay attack

Pro replay attack bylo použito HackRF One, které bylo detailně popsáno v podkapitole 2.3.2. S HackRF byla použita základní anténa ANT500.[12] Pro tento test bylo rádio umístěno v bezprostřední blízkosti mezi teploměrem a stanicí.

Útok na meteostanici se povedlo provést, neboť stanice v komunikaci neimplementuje žádnou ochranu proti tomuto typu útoku. Během párování teplotního čidla a základnové stanice dochází i k synchronizaci času, tj. čidlo vysílá ve 4 sekundových intervalech a stejně tak i přijímá stanice. Proto pro úspěšné provedení útoku je nutné, aby i útočník vysílal v tomto intervalu.

To by vyžadovalo, aby útočník sledoval provoz v daném spektru a data vysílal v přesně stanoveném čase, což je však také poměrně náročné. Výhodou takového postupu je, že takto provedený útok je špatně detekovatelný.

Další možností, jak útok provést, je vysílat data neustále. S vysokou pravděpodobností se útočník „trefí“ dříve či později do okna, kdy stanice přijímá. Z toho plynou nevýhody této varianty, tedy že může nastat situace,

[12] Detailnější informace o anténě je možno naleznout na stránkách `https://greatscottgadgets.com/ant500/`

Obrázek 5.2: Demonstrace úspěšně provedeného útoku

kdy útok nebude úspěšný, protože útočníkem vyslaný paket nebude vyslaný v požadovaný čas. Druhou nevýhodou je, že útočník může být snadno odhalen díky neustálému vysílání.

Vzhledem k demonstračnímu charakteru byla zvolena varianta neustálého vysílání, která by však v reálném prostředí byla méně vhodná. Útok by pravděpodobně byl úspěšný, ale snadná odhalitelnost útočníka je v tomto případě obrovskou nevýhodou.

Prvním krokem bylo zjištění přesně frekvence, na které teploměr vysílá. Zde byl použit nástroj Spectrum Analyzer v URH. Po zjištění správné frekvence stačilo signál nahrát vytvořenou aplikací příkazem:

```
python main.py -f 868250000 -s 2e6 -m record -d rtl
```

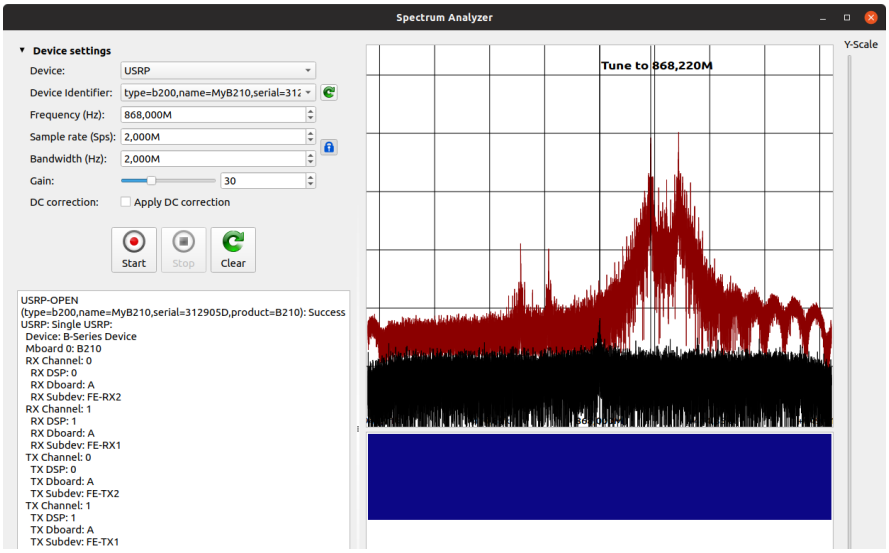
Pro neustálé vysílání zprávy bylo potřeba signál upravit, aby obsahoval pouze tuto zprávu, a uložit. K tomu je možné použít např. Audacity nebo signál upravit v URH. Takto upravený signál je možné začít vysílat a to příkazem:

```
python main.py -f 868250000 -s 2e6 -m transmit -d hackrf -r 40
-p /tmp/meteo.complex
```

Úspěšný útok je možné vidět na obrázku 5.2.

5.1.3 Tampering

Tampering, neboli narušení komunikace, je nejobtížnějším typem útoku, proto byl také proveden pouze na meteostanici popsané v kapitole 2.4. Pro testování



Obrázek 5.3: GUI nástroje Spectrum Analyzer [25, pořízeno v URH]

bylo použito opět USRP B210 s anténou Vert900 Po identifikování signálu, je nutné analyzovat celou zprávu a rozluštit jednotlivé části použitého protokolu.

Pro přesnou identifikaci frekvence opět posloužil Spectrum Analyzer v URH. Frekvenci bylo nutné opět zjistit zejména proto, že při restartu zařízení se může změnit kanál a tedy i frekvence, na které teploměr vysílá.

Následovalo nahrání signálu pomocí nástroje Record Signal. Po nahrání se signál automaticky otevře na záložce *Interpretation* v URH. Nahraný signál je možné oříznout pouze na tu část, která útočníka zajímá, tj. odeslaná zpráva. Pro snazší práci z této zprávy je možné vytvořit nový signál a dále pracovat jen s ním.

URH nabízí možnost volby několika parametrů, ale ve výchozím nastavení se je pokusí automaticky odhadnout. Pro přesnější analýzu byl použit ještě nástroj Inspectrum, díky kterému bylo možné parametry upravit a nastavit přesněji, než to udělal program URH. Po nastavení všech parametrů bylo vidět jednotlivé bity, které tvoří paket.

Nyní bylo potřeba přijít na to, co které bity znamenají. Rozsáhlou analýzu již provedl *Chritophe Jacquet*, kterou popisuje v článku [29]. Zpráva začíná preambulí ve tvaru `10101010`, tj. `0xaa` hexadecimálně, následovaná pravděpodobně synchronizačním slovem, které je neměnné. To má tvar `0x2dd4` v hexadecimální podobě. Za ním následuje délka odesílaných dat až do konce rámce včetně kontrolního součtu, má vždy hodnotu 9.

Jako první je identifikátor senzoru, který má tvar např. `0xbf` a mění se při každém restartu. Následuje ho teplota, která je kódovaná v BCD[13]. Navíc, aby teploměr mohl ukazovat i záporné teploty, je vždy k teplotě přičteno 40.

[13]Binary Coded Decimal, dvojkově reprezentované dekadické číslo

Číslo kvartetu	Příklad v hexa	Pole
0	a	Preambule
1	a	
2	2	Synchronizační slovo
3	d	
4	d	
5	4	
6	9	Délka dat
7	b	Identifikátor senzoru
8	f	
9	6	Desítky teploty
10	3	Jednotky teploty
11	3	Desetiny teploty
12	6	Vlhkost vzduchu
13	a	
14	e	CRC8
15	9	

Tabulka 5.1: Rámec celé zprávy

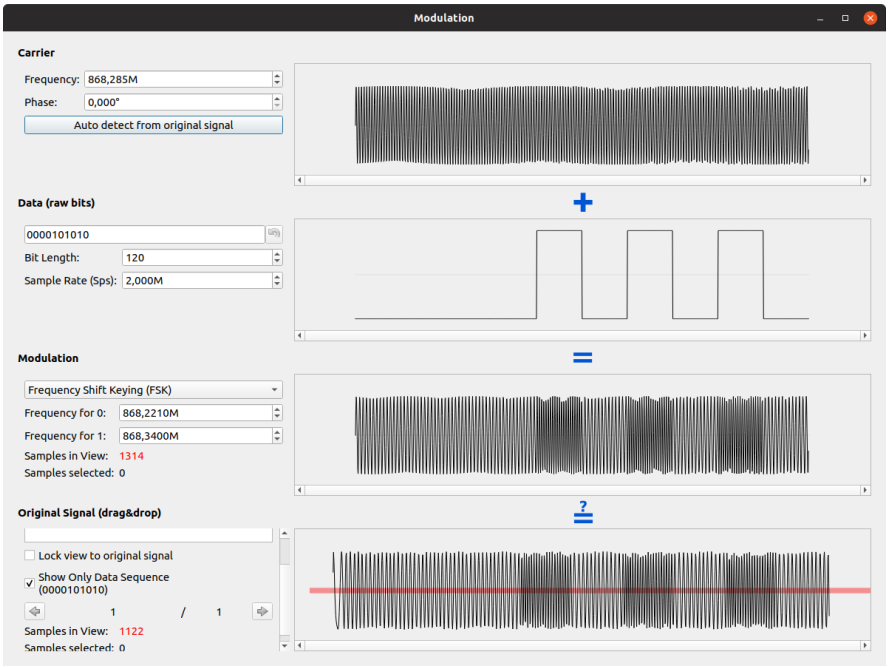
Např. teplota kódovaná jako `0x633` odpovídá teplotě 23,3°C. Po teplotě je další v pořadí vlhkost vzduchu. Ta už nepoužívá BCD a je kódovaná standardně binárně. Protože však senzor ani stanice použité v této práci vlhkost vzduchu nepodporují, je odesílaná hodnota vždy `0x6a`. Ta zároveň značí, že vlhkoměr není dostupný [34].

Poslední část rámce tvoří kontrolní součet CRC8. CRC je počítáno od 6. kvartetu po 13. Jako polynom je použit $x^8+x^5+x^4+x^0$, tj. `0x31`, inicializační hodnota je 0 a na konci se neprovádí žádný XOR. Celý paket je možné vidět v tabulce 5.1.

V URH celou tuto analýzu usnadňuje záložka *Analysis*, díky možnosti vytvářet štítky (tzv. labels) pro jednotlivé části zprávy a strukturu celého protokolu si uložit.

Nyní, když byly identifikovány veškeré odesílané údaje, je možné přejít k samotnému útoku. Na záložce *Generator* je možné data upravit. Zde byla upravena teplota, podle ní bylo nutné přepočítat CRC a tuto hodnotu též upravit. Posledním krokem před odesláním dat je správné nastavení modulace. To je možné provádět též na této záložce po kliknutí na tlačítko *Edit*.

Zde může uživatel měnit veškeré nastavení potřebné pro správné nastavení modulace, jako např. frekvenci nosné, fázi, výběr z několika modulací a další nastavení. Všechny tyto možnosti jsou vidět na obrázku 5.4. Zde opět posloužily získané údaje z prvního kroku, jedná se o modulaci FSK, na obrázku 5.3 je vidět frekvence pro 0 a frekvence pro 1. Z těchto údajů byla zvolena nosná frekvence. Také díky již dříve provedené analýze v programu Inspectrum



Obrázek 5.4: Nastavení modulace v GUI URH [25, pořízeno v URH]

bylo možné nastavit délku jednoho bitu a to na 120 při 2 Msps.

Po zadání všech parametrů je vidět náhled vytvořeného signálu, které je možné hned porovnat s jiným signálem, kupříkladu tím, který útočník přijal. Vytvořený signál lze uložit do souboru pro další použití, nebo okamžitě odeslat. Při odeslání stačí vybrat SDR, nastavit zisk a příp. další parametry, a přejít k útoku. Zde byl princip stejný jako při útoku přehráním, to znamená, že data byla vysílána neustále a po chvíli stanice začala tato data přijímat.

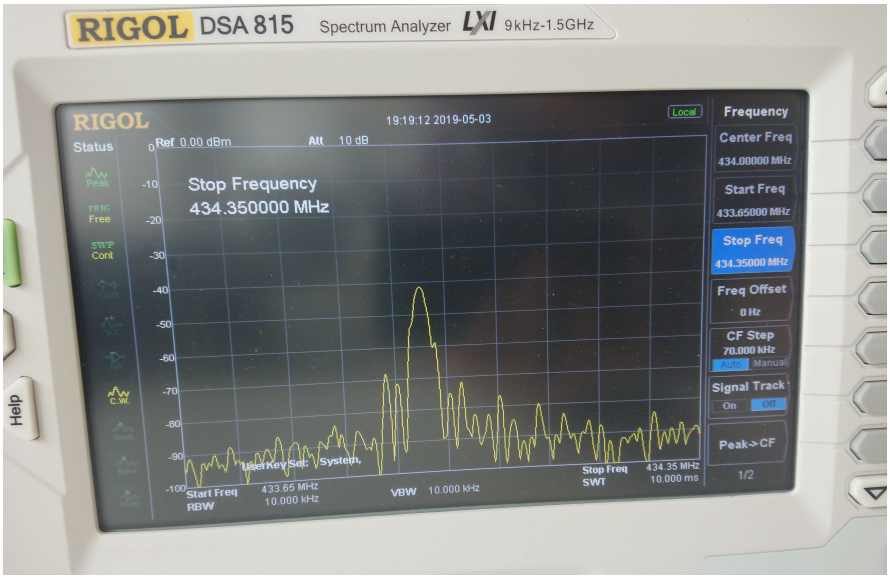
5.2 Dálkové ovládání centrálního zamykání vozu

Cílem dalšího útoku se stalo vozidlo Ford Focus Mk 2 popsané v kapitole 2.5. Z útoků byl proveden replay attack na centrální dálkové ovládání.

Auto s klíčem používá pravděpodobně některou z implementací rolling code (též označované jako hopping code). Oproti fixním kódům je při každém stisknutí tlačítka odeslán jiný kód. To má právě zabránit proti snadnému provádění replay attacku.

5.2.1 Replay attack

Pro tento útok bylo použito jak HackRF s anténou ANT500, tak i USRP B210 s anténou Vert900. Provedení replay attacku je v tomto případě možné jedině v případě, že útočník získá poslední platný kód. Respektive nemusí to



Obrázek 5.5: Identifikace frekvence na spektrálním analyzátoru

být poslední vygenerovaný kód, ale nesmí po něm následovat již použitý kód. V takovém případě kód přestává být platným.

5.2.1.1 Test č. 1

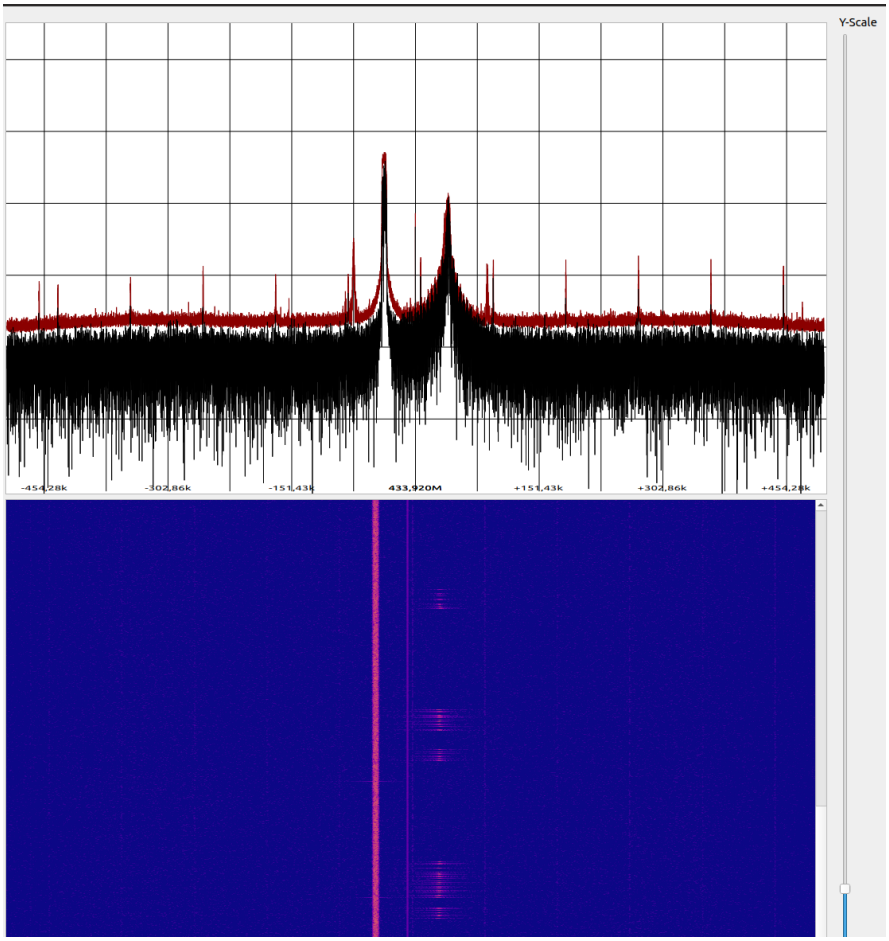
První test proběhl opět v „laboratorních“ podmínkách. Nejdříve byla zjištěna přesná frekvence, kterou klíč s vozidlem používají. To je možné vidět na spektrálním analyzátoru na obrázku 5.5, pro zkontrolování byl použit i Spectrum Analyzer v URH. Komunikace probíhala na frekvenci 433,55 MHz.

Po identifikování bylo možné zachytit vysílaný signál mimo dosah automobilu a uložit ho pro pozdější použití. Protože přijímač v autě „poslouchá“ neustále, nebylo potřeba nahraný signál jakkoliv upravovat. Stačilo ho v blízkosti auta opět přehrát a vozidlo se odemklo.

5.2.1.2 Test č. 2

Druhý test byl již reálným testem, kdy probíhalo rušení pásma a nahrávání signálu zároveň. Tento útok využívá nedokonalosti v použitých součástkách v autě a díky tomu auto přijímá signál na širším pásmu než je nutné a je možné ho zarušit tak, že přijímač není schopný identifikovat validní signál, ale samotné pásmo na kterém vysílá klíč, není ovlivněno. Tato situace je vidět na obrázku 5.6. Útok je znám už řadu let, ale první demonstraci provedl až v roce 2015 Samy Kamkar.[14]

[14]Útok popsal na DEF CONu v roce 2015 `https://samy.pl/defcon2015/`



Obrázek 5.6: Rušení signálu [25, pořízeno v URH]

Bohužel při testování se nepodařilo dosáhnout situace, při které by zachycený signál dokázal auto odemknout. Na vině bylo pravděpodobně nedostatečné odfiltrování rušícího signálu a signál pro odemknutí tak byl znehodnocen. Jak je vidět ale z obrázku 5.6, útok je teoreticky možný díky širšímu pásmu, na kterém auto přijímá.

5.2.2 Shrnutí

Replay attack na dálkové ovládání zamykání je i dnes nebezpečným útokem s vysokou pravděpodobností na úspěch. Nejenže je v České republice průměrné stáří osobních automobilů více než 14 let[15] a zabezpečením je nanejvýš technologie rolling code, ale některé automobilky ještě před několika lety používaly fixní kódy.[16]

[15] `http://portal.sda-cia.cz/clanek.php?id=6304&v=m`
[16] `https://calebmadrigal.com/hackrf-replay-attack-jeep/`

Závěr

Cílem této práce bylo analyzovat zabezpečení bezdrátové komunikace u běžně dostupných zařízení a principy zabezpečení. Dále byly zmapovány typy útoků, jejich principy a na základě této analýzy navrhnuta aplikace, pomocí které jsou tyto útoky demonstrovány.

Cíl práce se podařilo splnit, aplikace byla úspěšně naprogramována a je spustitelná na OS Linux. Pro návrh posloužily tzv. flowgraphy v GNU Radio Companion, díky kterým je návrh funkcí velmi pohodlný a vygenerovaný kód je pěkně strukturovaný. Některé části bylo potřeba naprogramovat, přesto si aplikace zachovává stejnou strukturu, jako aplikace vytvořené v GRC. To je důležité pro další vývoj aplikace.

Na začátku práce je čtenář uveden do problematiky softwarově definovaných rádií, práce ho seznámí se základními vlastnostmi SDR, jejich analogovou částí a principy zpracování digitálního signálu. Dále je popsán framework GNU Radio, který je de facto standardem při vývoji aplikací pro softwarově definovaná rádia. Následuje popis dostupných aplikací, které mohou sloužit pro tyto účely.

Aplikace byla úspěšně otestována na meteostanici, kde posloužila jako vhodný nástroj pro rušení, protože nevyžaduje mnoho nastavení a práce s ní je velice intuitivní. Stejně tak byla úspěšně použita na replay attack.

Dále byla použita pro replay attack na dálkové odemykání vozu Ford Focus, který však používá pro zabezpečení tzv. rolling codes. Ty mají vozidla zabezpečit právě proti tomuto typu útoku. Přesto pokud útočník získá poslední platný signál, může být i přes toto zabezpečení útok úspěšný.

Útok se zásahem do komunikace, tzv. tampering, v aplikaci realizován není, neboť z praktických důvodů nelze vytvořit univerzální aplikaci, která by dokázala jednoduše analyzovat použitý protokol a pozměňovat data. Naopak během psaní práce se objevil projekt Universal Radio Hacker, popsaný v kapitole 2.2.3, který je velmi komplexní a usnadňuje analýzu signálů, stejně tak jako nabízí některé nástroje vhodné pro útok typu tampering. Pro jeho složitost se však nehodí na rychlé použití nebo skriptování.

Z kapitoly testování je vidět, že pro útoky na zařízení bez jakéhokoliv zabezpečení nejsou potřeba hluboké znalosti zpracování digitálního signálu. Díky nástrojům jako je GNU Radio nebo Universal Radio Hacker jsou tak útoky dostupné široké veřejnosti.

Navazující práce

Aplikace je plně funkční, ale protože během psaní této práce je GNU Radiem podporována pouze verze Pythonu 2.7, dalo by se do budoucna uvažovat o přepsaní do Pythonu verze 3.6, která bude podporována od GR verze 3.8. Také některé komponenty by mohly být pro větší efektivitu implementovány v jazyce C++. V této fázi aplikace slouží jako jednoduchý nástroj pro rychlé použití. Díky zachované struktuře, která je použitá při generování kódu v GRC, je možné tuto aplikaci považovat za základní stavební kámen pro budoucí rozšíření a může se tak stát základem daleko vetší a komplexnější aplikace.

Literatura

1. DUARTE GARCÍA, Jorge. *Software Defined Radio for Wi-Fi Jamming* [online]. 2016 [cit. 2019-04-08]. Dostupné z DOI: 10.13140/RG.2.2.23772.90240.
2. INSTALLFEST. *Softwarově definované rádio (Jan Hrach)* [online] [cit. 2019-04-08]. Dostupné z: https://www.youtube.com/watch?v=i1ZB70nPF-g.
3. *Wikipedie*. Anténa [online]. 2019 [cit. 2019-04-09]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=Ant%C3%A9na&oldid=16921501 Page Version ID: 16921501.
4. BEVELACQUA, Peter Joseph. *The Dipole Antenna* [obrázek online] [cit. 2019-04-06]. Dostupné z: http://www.antenna-theory.com/antennas/dipole.php.
5. BEVELACQUA, Peter Joseph. *Monopole Antenna* [obrázek online] [cit. 2019-04-17]. Dostupné z: http://www.antenna-theory.com/antennas/monopole.php.
6. BEVELACQUA, Peter Joseph. *Radiation Pattern* [obrázek online] [cit. 2019-04-17]. Dostupné z: http://www.antenna-theory.com/basics/radpattern.php.
7. LAGARTO. *FCC průmyslové systémy* [online]. 2017 [cit. 2019-04-21]. Dostupné z: http://www.fccps.cz.
8. DOBEŠ, Josef; ŽALUD, Václav. *Moderní radiotechnika*. 1. vyd. Praha: BEN – technická literatura, 2006. ISBN 80-7300-132-2.
9. *Anti-Aliasing Filters and Their Usage Explained – National Instruments* [online] [cit. 2019-04-08]. Dostupné z: http://www.ni.com/cs-cz/innovations/white-papers/18/anti-aliasing-filters-and-their-usage-explained.html.

10. ROSA, Tomas. 101 RF Hacking with SDR – From Beautiful Equations to Real Threats [online], s. 60 [cit. 2019-04-08]. Dostupné z: http://crypto.hyperlink.cz/files/mkb-rosa-2018.pdf.

11. *Wikipedie*. Vzorkovací frekvence [online]. 2018 [cit. 2019-04-09]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=Vzorkovac%C3%AD_frekvence&oldid=16429291 Page Version ID: 16429291.

12. *What is I/Q Data? – National Instruments* [online] [cit. 2019-04-06]. Dostupné z: http://www.ni.com/tutorial/4805/en/.

13. KEKULE, Jaromír. *Obvody se střídavým proudem* [online] [cit. 2019-04-06]. Dostupné z: http://lucy.troja.mff.cuni.cz/~tichy/elektross/elektrina/el_proud/stridavy_proud/rlc_obv/rlc.html.

14. HOREVAJ, Michal. *Vektorový generátor* [online] [cit. 2019-04-06]. Dostupné z: http://www.elektrorevue.cz/clanky/02034/index.html.

15. *Question about digital modulation* [Electrical Engineering Stack Exchange] [obrázek online] [cit. 2019-05-02]. Dostupné z: https://electronics.stackexchange.com/questions/360296/question-about-digital-modulation.

16. ELAHI, Ata. *Network Communications Technology* [online]. Cengage Learning, 2001 [cit. 2019-04-08]. ISBN 978-0-7668-1388-5. Google-Books-ID: g6Vu7vu20rgC.

17. *Full Duplex* [obrázek online] [cit. 2019-04-17]. Dostupné z: https://www.networxsecurity.org/members-area/glossary/f/full-duplex.html.

18. KLOUČEK, Luboš. Detekce útoku úmyslného rušení v bezdrátových senzorových sítích [online], s. 32 [cit. 2019-04-08]. Dostupné z: https://is.muni.cz/th/nagj5/bp.pdf.

19. *What is a Replay Attack and How to Prevent it | Kaspersky Lab* [online] [cit. 2019-04-18]. Dostupné z: https://www.kaspersky.com/resource-center/definitions/replay-attack.

20. *Protocol design – How does a rolling code work?* [Cryptography Stack Exchange] [online] [cit. 2019-04-18]. Dostupné z: https://crypto.stackexchange.com/questions/18311/how-does-a-rolling-code-work.

21. *The Car Hacker's Handbook* [online] [cit. 2019-05-14]. Dostupné z: http://opengarages.org/handbook/ebook/.

22. CAPKUN, Srdjan; FRANCILLON, Aurélien; DANEV, Boris. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. *ETH Zurich* [online]. 2011 [cit. 2019-05-14]. Dostupné z DOI: 10.3929/ethz-a-006708714.

23. *GNURadioCompanion – GNU Radio* [online] [cit. 2019-04-27]. Dostupné z: https://wiki.gnuradio.org/index.php/GNURadioCompanion.

24. WALTERS, Mike. *Offline radio signal analyser.* [online]. 2019 [cit. 2019-04-14]. Dostupné z: https://github.com/miek/inspectrum. original-date: 2015-06-01T00:15:48Z.

25. POHL, Johannes; NOACK, Andreas. Universal Radio Hacker: A Suite for Analyzing and Attacking Stateful Wireless Protocols. In: *12th USENIX Workshop on Offensive Technologies (WOOT 18)* [online]. Baltimore, MD: USENIX Association, 2018 [cit. 2019-04-21]. Dostupné z: https://www.usenix.org/conference/woot18/presentation/pohl.

26. *About RTL-SDR* [rtl-sdr.com] [online]. 2013 [cit. 2019-05-04]. Dostupné z: https://www.rtl-sdr.com/about-rtl-sdr/.

27. OSSMANN, Michael. *low cost software radio platform.* [online]. 2019 [cit. 2019-04-08]. Dostupné z: https://github.com/mossmann/hackrf. original-date: 2012-03-10T18:31:42Z.

28. *B200/B210/B200mini/B205mini – Ettus Knowledge Base* [online] [cit. 2019-04-21]. Dostupné z: https://kb.ettus.com/B200/B210/B200mini/B205mini.

29. JACQUET, Christophe. *Christophe Jacquet – Décodage de capteur thermo-hygro TFA* [online] [cit. 2019-04-18]. Dostupné z: http://www.jacquet80.eu/blog/post/2011/10/Decodage-capteur-thermo-hygro-TFA.

30. *Wikipedia*. Remote keyless system [online]. 2019 [cit. 2019-05-02]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Remote_keyless_system&oldid=887154896 Page Version ID: 887154896.

31. KASPER, Timo. *Security analysis of pervasive wireless devices* [online] [cit. 2019-05-13]. Dostupné z: https://www.emsec.ruhr-uni-bochum.de/media/attachments/files/2012/11/timo_phd_thesis.pdf.

32. *SVG > tower broadcasting broadcast communication* [obrázek online] [cit. 2019-05-02]. Dostupné z: http://svgsilh.com/image/297697.html.

33. *SVG > atmosphere space nasa aerial* [obrázek online] [cit. 2019-05-02]. Dostupné z: http://svgsilh.com/image/304946.html.

34. BOSSARD, Fred. *TX29 Protocol* [online] [cit. 2019-04-25]. Dostupné z: http://fredboboss.free.fr/articles/tx29.php/.

Příloha A

Seznam použitých zkratek

ADC Analog to Digital Converter

BCD Binary Coded Decimal

CRC Cyclic Redundancy Check

DAC Digital to Analog Converter

DC Direct Current

FCC Federal Communications Commission

GPS Global Positioning System

GR GNU Radio

GUI Graphical User Interface

IoT Internet of Things

LO Local Oscilator

MIMO Multiple Input, Multiple Output

PKES Passive Keyless Entry and Start

PRNG Pseudo Random Number Generator

RF Radio Frequency

Rx Receive

SDR Software Defined Radio

Tx Transmit

USB Universal Serial Bus

USRP Universal Software Radio Peripheral

URH Universal Radio Hacker

XML Extensible Markup Language

XOR Exclusive OR

Příloha B

Uživatelská příručka

B.1 Minimální požadavky

Pro správný chod programu je nutné mít nainstalované GNU Radio ve verzi alespoň 3.7.13.4 a Python verze 2.7.15. Pro zjištění nainstalované verze GNU Radio lze použít příkaz:

```
gnuradio-config-info -v
```

Nainstalovanou verzi Pythonu lze zjistit příkazem:

```
python --version
```

Pokud se zobrazí Python ve verzi 3.x, je nutné ověřit, jestli je nainstalovaný i Python verze 2:

```
python2 --version
```

V takovém případě je potřeba nastavit buďto Python 2 jako výchozí, anebo vždy spouštět program příkazem přímo pro spuštění Pythonu ve verzi 2.

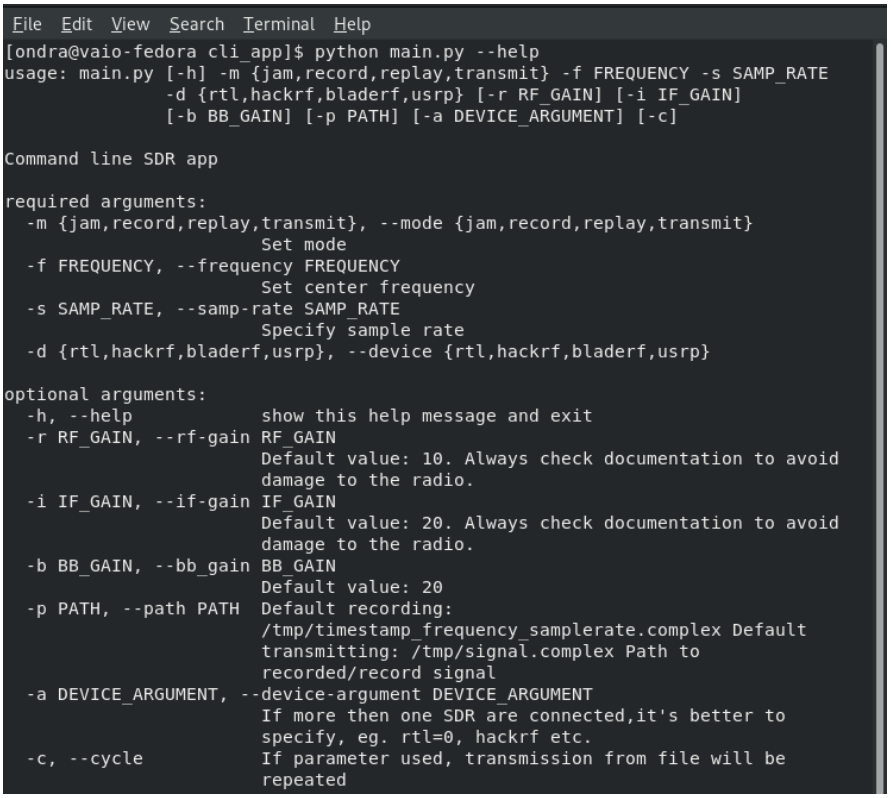
B.2 Spuštění

Na přiloženém flashdisku je spouštěcí skript aplikace. Ta může být spuštěna příkazem:

```
python main.py
```

Při spuštění bez parametrů se zobrazí nápověda. Pro zobrazení podrobné nápovědy stačí použít příkaz:

```
python main.py --help
```



Obrázek B.1: Nápověda aplikace

Aplikace nabízí několik režimů:

- Nahrání signálu
- Přehrání signálu
- Rušení
- Nahraní a okamžité přehrání signálu

B.2.1 Nahrání signálu

Pro nahrání signálu slouží příkaz:

```
python main.py -m record -d usrp -s 2e6 -f 433955e3
```

Volitelně může uživatel změnit výchozí cestu pro uložení souboru a nastavit jednotlivé parametry pro zisk. Nahrávání se přeruší stiskem klávesy Enter a signál je uložen.

B.2.2 Přehrání signálu

Přehrání signálu ze souboru je možné provést příkazem:

```
python main.py -m transmit -d usrp -s 2e6 -f 868e6
   -p /tmp/my_signal.complex -r 60
```

Pro přehrávání ve smyčce může uživatel přidat přepínač `--cycle`.

B.2.3 Rušení

Rušení pásma je aplikováno po zavolání příkazu:

```
pythom main.py -m jam -d hackrf -s 8e6 -f 315e6 -r 40
```

Rušení se ukončí po stisku klávesy Enter.

B.2.4 Nahrání a okamžité přehrání signálu

Tento mód supluje módy pro nahrání a přehrání signálu. Po ukončení nahrávání se signál ihned přehraje. Použijí je následující:

```
pyton main.py -m replay -d usrp -s 4e6 -f 433e6 -r 50
```

Příloha C

Obsah přiloženého flashdisku

```
readme.txt ..... stručný popis obsahu flashdisku
main.py ..... spouštěcí skript implementace
src
    impl ..... zdrojové kódy implementace
    doc ..... adresář s dokumentací
    thesis ..... zdrojová forma práce ve formátu LaTeX
text ..... text práce
    thesis.pdf ..... text práce ve formátu PDF
```