VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019 Bc. Ondřej Pospíšil

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SOFTWAROVĚ DEFINOVANÉ RÁDIO PRO TECHNOLOGII LORAWAN
SOFTWARE DEFINED RADIO FOR LORAWAN TECHNOLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. Ondřej Pospíšil

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Radek Fujdiak, Ph.D.

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Semestrální práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Ondřej Pospíšil ***ID:*** 174382
Ročník: 2 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Softwarově definované rádio pro tchnologii LoRaWAN

POKYNY PRO VYPRACOVÁNÍ:

Student provede analýzu nových technologií LPWAN s bližším zaměřením na proprietární technologii LoRaWAN (rozdíly mezi LoRa, LoRaWAN 1.0.X a LoRaWAN 1.1.X), s kterou se také fyzicky seznámí. Bude provedena kompletní analýza komunikace, společně s jednotlivými procesy v síti a současným stavem. Výsledkem bude posouzení bezpečnostních aspektů technologie LoRaWAN a návrh dekódování rádiové části, založené na dostupné dokumentace, literatury a vlastním reverzním inženýrství. Bude vybráno vhodné hardwarové zařízení (868 MHz), software pro analýzu komunikace (doporučen program Wireshark) a také jeden z dostupných nástrojů pro softwarově definované rádio (doporučeno GNURadio). V neposlední řadě bude realizován tzv. sniffer pro rádiovou část technologie LoRaWAN, společně s testování jeho možností (úprava a přehrávání zprávy, aj.).
Semestrální projekt: Bude provedena analýza, výběr SW i HW, student se seznámí s vybraným SW Wireshark, GNURadio či jiné), bude zprovozněna komunikace LoRaWAN.

DOPORUČENÁ LITERATURA:

[1] „LoRaWAN What is it?: A technical overview of LoRa and LoRaWAN.“ LoRa Alliance. 2015.

[2] „GNURadio Manual and C++ API Reference.“ GNURadio.

Termín zadání: 1.10.2018 ***Termín odevzdání:*** 14.12.2018

Vedoucí práce: Ing. Radek Fujdiak, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Diplomová práce se zabývá v teoretické části popisem LPWAN technologií se zaměřením na technologii LoRaWAN. Následně je v práci popsán a rozebrán protokol LoRaWAN. Jsou zde rozebrány především bezpečnostní prvky protokolu LoRaWAN a jsou také porovnány verze LoRaWAN 1.0.2 a 1.1.x. Dále se práce zabývá odposlechem komunikace, jejím zachycením a dešifrováním a to jak na fyzické vrstvě tak na vrstvě MAC protokolu LoRaWAN. V práci je ukázáno jak využít softwarově definované radio k odposlechu komunikace LoRaWAN. Nakonec je v práci ukázka útoku přehráním a zaslání falešné zprávy na server.

KLÍČOVÁ SLOVA

LoRa, LoRaWAN, Softwarově definované radio, Bezpečnost, Šifrování, Bezdrátová komunikace, Klíč, IoT

ABSTRACT

This master's thesis deals with the description of LPWAN technologies focused on LoRaWAN technology in the theoretical part. The next part deals with description and analyzing of LoRaWAN protocol. In theoretical part also security elements of the LoRaWAN protocol are discussed and LoRaWAN 1.0.2 and 1.1.x versions are compared. The thesis also deals with LoRaWAN tapping and its decryption, both on the physical layer and the MAC protocol layer LoRaWAN. The thesis shows how to use software-defined radio to listen to LoRaWAN communication. Lastly, a replay attack and fake message over the fake session are performed.

KEYWORDS

LoRa, LoRaWAN, Software defined radio, Security, Encryption, Wireless communication, Key, IoT

POSPÍŠIL, Ondřej. *Softwarově definované rádio pro technologii LoRaWAN*. Brno, , 73 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Radek Fujdiak, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Softwarově definované rádio pro technologii LoRaWAN“ 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 diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského 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.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Radku Fujdiakovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Chtěl bych také poděkovat rodičům za dlouhodobou podporu při studiu.

Brno ....

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Semestrální práce se zabývá Low Power Wide Area Network (LPWAN) technologií LoRaWAN, která patří v současnosti mezi nejpoužívanější LPWAN technologie [1, 2]. Mezi konkurencí vyčnívá hlavně díky možnosti stavby lokální LoRaWAN sítě. Internet věcí se v posledních pár letech stal velkým trendem a LPWAN technologie jsou jeho důležitou složkou. Díky internetu věcí může být téměř jakákoli věc připojena do sítě. Tím se, ale zvětšuje požadavek na bezpečnost jednotlivých řešení, a proto musí být LPWAN technologie spolehlivé v rámci bezpečnosti.

V první části této práce jsou obecně popsány rozsáhlé sítě s nízkou spotřebou energie (LPWAN), jejich vlastnosti a požadavky. V kapitole jsou také srovnány v současnosti nejvíce používané technologie LPWAN v České republice a to LoRaWAN, NB-IoT a Sigfox. Jsou zde rozebrány výhody a nevýhody LoRaWAN oproti konkurenčním řešením. Závěrem kapitoly je použitelnost LPWAN technologií v rámci IoT.

Ve druhé kapitole je stručný obecný popis technologie LoRaWAN. Tato kapitola popisuje komunikaci v síti LoRaWAN a jednotlivé prvky, této komunikace. Nejvíce ja kapitola zaměřena na detailní popis rozdílů mezi protokoly LoRaWAN verze 1.0.2 [3] a verze 1.1.x [4]. Rozdíly jsou zaměřeny především na bezpečnost. Kapitola se celkově zabývá především bezpečností těchto dvou verzí.

Ve třetí kapitole je popsán návrh řešení softwarově definovaného radia pro LoRaWAN. V kapitole je popsán zvolený hardware a software. Je zde popsáno jak byla zprovozněna síť LoRaWAN, sestavena brána a jak byl spuštěn server LoRaWAN.

Čtvrtá kapitola se věnuje samotnému dešifrování LoRaWAN. Nejdříve je zde popsána komunikace a zachycení zpráv na fyzické vrstvě LoRa pomocí GNU Radia. Následně je zde ukázka zachycení nešifrované zprávy. Poté je v kapitole popsáno dešifrování vyšší vrstvy MAC, tedy vrstvy definující protokol LoRaWAN. Tato vrstva je již šifrována a v této kapitole je ukázáno jak tuto vrstvu dešifrovat. Je zde také prezentován program, který byl vytvořen v rámci této práce pro zjednodušení dešifrování zpráv LoRaWAN.

V poslední kapitole jsou popsány uskutečněné útoky na LoRaWAN síť. Nejdříve je v kapitole popsán útok přehráním zprávy. Následně bylo v této kapitole ukázáno jak lze změnit informace v přenášené zprávě a docílit tak změny obsahu přenášené zprávy. Je zde popsáno jaké informace a zařízení jsou k tomuto procesu potřebné.

1 Low Power Wide Area Networks

Velkým tématem současnosti je IoT (Internet of Things), který umožňuje připojit a propojit různé věci v rámci internetu [5]. IoT může ulehčit a překonat v dnešní době velmi důležitá témata jako například energetická krize, vyčerpání zdrojů, znečištění životního prostředí a další. K realizaci takovýchto cílů je nutné aby zařízení, použité k měření jednotlivých úkonů v rámci těchto témat, sdílely informace nejen mezi sebou, ale také mezi lidmi, a díky tomu je možné vytvořit optimální řešení pro jednotlivá témata.

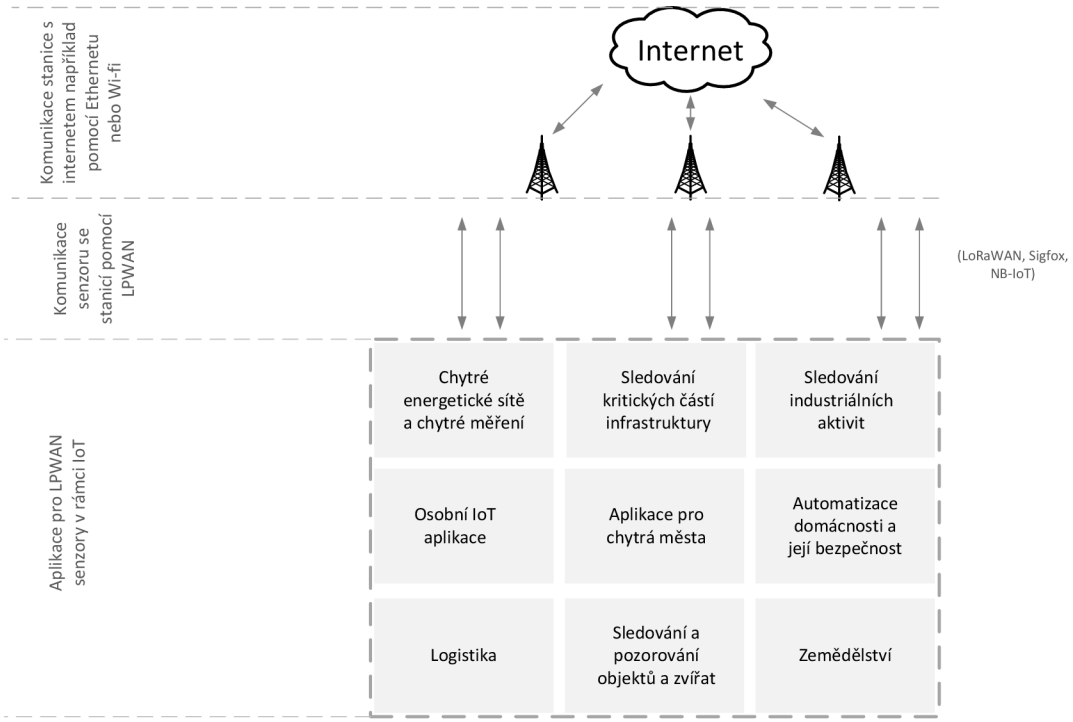
LPWAN (Low Power Wide Area Networks), tedy rozsáhlé sítě s nízkou spotřebou energie, nabízejí vhodné řešení pro IoT v oblastech chytrých měst, osobních IoT aplikací, chytrých energetických sítích, chytrého odečtu dat z různých měřících přístrojů, logistice, průmyslovém monitorování, zemědělství a v mnohých dalších oblastech. Obecně lze tedy říci, že LPWAN sítě jsou sítě pracující v geograficky rozsáhlé oblasti s nízkou spotřebou energie, ale také s malou kapacitou pro přenos dat. Díky jejich vlastnostem, jako je velký dosah (v rámci jednotek až desítek kilometrů), velké životnosti baterie (10 let), ale také nízké přenosové rychlosti (v řádech 10 kb/s) a vyšší latenci (sekundy až minuty), nejsou LPWAN samozřejmě vhodné pro všechna IoT řešení. Konkrétně jsou ideální pro řešení v oblastech, kde je určitá míra tolerance pro latenci, kde není potřeba vyšších přenosových rychlostí a tam, kde je vyžadována nízká spotřeba energie a nízké náklady na vybudování sítě. Na obr. 1.1 lze vidět v jakých oblastech lze využít senzory LPWAN a strukturu připojení do internetu [6].

1.1 Vlastnosti a požadavky na LPWAN

Dosah na velkou vzdálenost

Dosah na velkou vzdálenost je základem LPWAN sítí, je důležité, aby sítě měly geograficky velký dosah (jednotky až stovky kilometrů), ale zároveň také dobré penetrační vlastnosti, aby byli dostupné i v hůře dostupných oblastech budov. Pro dosažení tohoto cíle jsou tedy použity vhodné modulace a také je využito frekvenční pásmo menší než 1 GHz kvůli lepším penetračním vlastnostem.

Není pravidlem, aby LPWAN technologie musely využívat pásmo menší než 1 GHz, ale až na výjimky (Ingenu, Weightless-W [7, 8]) se toto pásmo využívá nejvíce. Hlavní důvody jsou ty, že signály s nižší frekvencí, oproti pásmu 2,4 GHz, vykazují méně útlumu a také větší odolnost proti vícecestnému zeslabení (multipath fading) to znamená, že se signál pomalu rozlaďuje a znovu ladí. Tyto problémy jsou způsobeny překážkami a povrchy (stěny, kopce). Další výhodou je menší vytíženost



Obr. 1.1: Oblast využití a struktura připojení.

proti pásmu 2,4 GHz. Nevýhodou je, že je na těchto pásmech kladen důraz na střídu (duty cycle) a také kvůli regulacím nemůže být využito maximální přenosové síly v pásmu.

Modulační techniky

- **Technika úzkopásmové modulace** (Narrowband): Signál je kódován v malé šířce pásma (obvykle méně než 25 kHz). Každá nosná je přiřazena úzkému pásmu a je sdíleno celkové spektrum. Úroveň šumu je v jednom pásmu minimální, to je výhoda u přijímače, jelikož není potřeba pro dekódování signálu na přijímači žádné zesílení při zpracování frekvenčního rozložení, což vede k levnému vysílači. Tuto modulaci využívají například dnes známější NB-IoT nebo Weightless-P [8].
- **Technika ultra úzkopásmové modulace** (Ultra Narrowband): Některé LPWAN technologie potlačují signál do ultra úzkého pásma o šířce 100 Hz, snižují tak šum a zvyšují počet zařízení na jednotku šířky pásma. Nevýhodou je, že přenosová rychlost koncových zařízení se zvyšuje, a tím se zvyšuje doba, po kterou musí být rádio zapnuto. Nízká přenosová rychlost v kombinaci s regulovaným spektrem snižuje maximální přenosovou frekvenci datových pa-

ketů, z toho plyne menší oblast nasazení. Příkladem technologií, které využívají tuto techniku, jsou Sigfox, Weightless-N a Telensa.

- **Technika rozprostřeného spektra** (Spread Specrtrum): Technika rozprostřeného spektra rozprostře úzkopásmový signál v širším frekvenčním pásmu, ale se stejným výkonem. Skutečně přenášený signál pak připomíná šum, který je velmi těžké odposlouchávat a který je odolnější vůči rušení. Na straně přijímače je však vyžadovaný větší zisk, aby bylo možné dekódovat signál, který je obvykle pod úrovní šumu. Rozprostření signálu přes celou šířku pásma vede k méně účinnému využití spektra. Tento problém však řeší využití více kanálů, ty můžou být dekódovány současně, takže se zvyšuje kapacita sítě. Varianty této techniky jako je DSSS (Direct Sequence Spread Spectrum) nebo CSS (Chirp Spread Spectrum) používá LoRa[9] nebo RPMA [7].

Topologie LPWAN sítě

LPWAN sítě využívají topologii na způsob hvězdy, stejně jako u celulárních technologií. Tato topologie přináší obrovskou výhodu v úspoře energie. Zařízení nemusí vybíjet energii při poslechu jiných zařízení, které chtějí přenášet provoz přes toto zařízení, jako je tomu u topologie mesh. U hvězdy je stále zapnutá základnová stanice, a tak poskytuje pohodlný a rychlý přístup pro koncové zařízení, které potřebuje zrovna komunikovat. Některé LPWAN technologie využívají i topologie mesh nebo tree, ale to pouze s komplexním protokolovým návrhem.

Nízká spotřeba

Provoz s nízkou spotřebou energie je hlavním požadavkem pro LPWAN technologie. Jsou zde velké požadavky na výdrž zařízení více jak 10 let na AA nebo knoflíkovou baterii.

Střída

Aby bylo dosaženo nízké spotřeby energie, tak se vypínají koncová zařízení, tedy vysílače. Střída právě umožňuje koncovým zařízením vypnout vysílače, když zrovna nejsou potřeba. Vysílač je zapnut pouze při přenosu dat. Střída není pouze o šetření energie, ale také je to legislativní požadavek. Regionální předpisy týkající se provozu spektra mohou omezit dobu, po kterou může vysílač spektrum obsadit, aby nedošlo s kolizí s jinými zařízeními, které sdílejí stejný kanál.

Přístup k mediím

Nejvhodnějším protokolem pro přístup k mediím (MAC vrstva) u LPWAN technologií je protokol Aloha, což je MAC protokol s náhodným přístupem v němž koncové zařízení vysílá bez snímání nosné. Ve zkratce protokol pracuje tak, že koncové zařízení vysílá zprávu právě v té chvíli, kdy potřebuje poslat zprávu a čeká na potvrzení, pokud potvrzení nedostane, vysílání opakuje. Tento protokol je využíván třeba u technologií Sigfox nebo LoRaWAN. Technologie jako NB-IoT nebo Ingenu naopak preferují protokol založený na TDMA (Time Division Multiple Access) k přidělování rádiových zdrojů za více efektivní, ale na úkor větší komplexnosti a ceně koncových zařízení. Protokol CSMA/CA není používán vzhledem k velkému počtu zařízení připojených k základnové stanici, v síti by nastala nadměrná signalizace a provoz by byl více nákladný při spolehlivém zajišťování přenosů.

Cena

Velmi důležitou složkou LPWAN sítí je cena. Hlavně díky ceně LPWAN zařízení mají LPWAN tak komerční úspěch. Konstrukce koncových zařízení za přijatelné ceny je uskutečnitelná díky použití propojení hvězda místo mesh, dále pak díky jednoduchému protokolu MAC a technikám snižujícím složitost koncových zařízení, což umožňuje výrobcům navrhnout levnější zařízení.

1.2 Srovnání technologií LPWAN

V rámci této části budou srovnány v současnosti nejrozšířenější LPWAN technologie. Jedná se o technologie LoRaWAN, Sigfox a NB-IoT. V tabulce 1.1jsou vyobrazeny technické rozdíly těchto technologií. Dále jsou technologie porovnány v rámci parametrů vhodných pro IoT a také příklady použití, vzhledem k tomu, že každá technologie se liší, a tak je také vhodnější pro jinou aplikaci v rámci IoT.

1.2.1 Porovnání parametrů v rámci IoT

Při výběru vhodné technologie LPWAN pro aplikaci v rámci IoT je potřeba zohlednit různé parametry jako například kvalita služeb, životnost baterie, latence, škálovatelnost, délka zprávy, pokrytí, rozsah, nasazení a cena. V rámci této části bylo čerpáno z článku věnujícímu se LPWAN [2].

Kvalita služeb

Sigfox a LoRa využívají bezlicenční spektrum a asynchronní komunikační protokoly. Můžou také eliminovat vícecestné zeslabení. Nemohou však nabídnout stejnou

Tab. 1.1: Technické rozdíly

	Sigfox	**LoRaWAN**	**NB-IoT**
Modulace	BPSK	CSS	QPSK
Frekvence	EU 868 MHz	EU 868 MHz	Licencované LTE
Šířka pásma	100 Hz	125–250 kHz	200 kHz
Maximální přenosová rychlost	100 b/s	50 kbit/s	200 kb/s
Velikost zprávy	12 B	240 B	1600 B
Dosah komunikace	10 km (zástavba) 40 km (venkov)	5 km (zástavba) 20 km (venkov)	1 km (zástavba) 10 km (venkov)
Autentizace šifrování	Message Authentication Code	AES 128 b	Kasumi a Snow 3G
Privátní síť	Ne	Ano	Ne
Standardizace	Sigfox, ETSI	LoRa-Alliance	3GPP

kvalitu služeb jako NB-IoT. NB-IoT využívá licencovaného spektra a synchronní protokol založený na LTE, což je výhodnější pro zaručení kvality služeb, ale za vyšší cenu [10]. NB-IoT je tedy preferováno u aplikací, které vyžadují zaručenou kvalitu služeb. Pro aplikace, nevyžadující zaručenou kvalitu služeb, je vhodnější zvolit Sigfox nebo LoRa.

Latence a životnost baterie

U všech zmíněných technologií jsou koncová zařízení většinu času v režimu spánku, což snižuje spotřebu energie, a tedy koncová zařízení mají dlouhou životnost. Koncová zařízení NB-IoT však spotřebují energii navíc kvůli synchronní komunikaci a zaručení kvality služeb. Pokud je tedy hlavním a nejdůležitějším parametrem dlouhodobá životnost koncového zařízení, tak je vhodnější Sigfox nebo LoRa ve srovnání s NB-IoT. NB-IoT nabízí výhodu vyšší latence. LoRa oproti společnosti Sigfox nabízí třídy zařízení. Zařízení třídy C mají taktéž nízkou latenci, ale na úkor vyšší spotřebě. Pro aplikace, které kladou důraz na latenci, jsou vhodnější NB-IoT nebo LoRa zařízení třídy C.

Škálovatelnost a délka zprávy

Podpora velkého množství zařízení je jednou z klíčových vlastností těchto technologií. Tyto technologie jsou připraveny na velký počet připojených koncových zařízení. NB-IoT nabízí oproti Sigfox nebo LoRa výhodu ve velmi vysoké škálovatelnosti,

umožňuje připojit až 100 tisíc koncových zařízení na jednu stanici, pro srovnání Sigfox a LoRa umožňují 50 tisíc zařízení na jednu stanici [1]. Nicméně NB-IoT nabízí také výhodu největší délky zprávy a to až 1600 bajtů. V obou těchto parametrech je NB-IoT lepší.

Rozsah a pokrytí

Největší pokrytí dle [1] má mít Sigfox a to až 40 km. LoRa má menší pokrytí maximálně 20 km a nejmenší pokrytí pak NB-IoT maximálně 10 km. NB-IoT dokáže také pracovat pouze v oblastech pokrytých LTE.

Nasazení

NB-IoT bylo specifikováno v roce 2016. Je to tedy relativně nová technologie oproti již relativně zaběhnuté konkurenci. Výhodou LoRa je také její flexibilita. Na rozdíl od Sigfox a NB-IoT nabízí LoRa kromě veřejné sítě také lokální síť (LAN). Lokální síť se dá využít například ve výrobní oblasti firmy, tedy pro větší bezpečnost firmy, která má svou vlastní síť.

Cena

Je třeba vzít v úvahu různé aspekty nákladů, jako jsou náklady na licenci spektra, náklady na síť a nasazení a také náklady na samotné zařízení. Sigfox a LoRa jsou cenově efektivnější ve srovnání s NB-IoT [1].

1.2.2 Příklady využitelnosti a aplikace

Elektrické měřící přístroje

U elektrických měřících přístrojů je většinou požadavek na častou komunikaci, nízkou latenci a vysokou přenosovou rychlost. Naopak není vyžadována nízká spotřeba ani příliš dlouhá životnost baterie vzhledem k tomu, že většina měřících přístrojů má vlastní zdroj energie. Důležitější je sledovat reálný provoz sítě, aby bylo možno okamžitě rozhodovat. Proto není pro tuto aplikaci vhodný Sigfox. LoRaWAN může být použita ale pouze třída C. Nejlepším řešením je tedy NB-IoT.

Zemědělství

V zemědělství je hlavní požadavek na životnost baterie. Vzhledem k tomu, že se podmínky nemění nijak radikálně, tak zařízení mění snímaná data jen občas. Zde je vhodnější použít Sigfox nebo LoRa i z toho důvodu, že některé oblasti nemusí být pokryty LTE.

Automatizace výroby

V tomto odvětví lze využít Sigfox, LoRaWAN či NB-IoT. Záleží na aplikaci, některé aplikace vyžadují zajištěnou kvalitu služeb a častou komunikaci a některé naopak levné senzory a dlouho životnost baterie.

Chytré budovy

Zde je většinou vyžadována nízká cena senzorů a dlouhá životnost baterie, většinou se nevyžaduje zajištění kvality služeb. Většinou aplikace jako senzory teploty,vlhkosti, bezpečnosti, průtoku vody, elektrických zástrček. Pro tyto aplikace jsou tedy vhodnější Sigfox a LoRaWAN.

Sledování palet a pohyb zboží

V této oblasti je nejdůležitější požadavek na cenu zařízení a životnost baterie. Pro tuto aplikaci se zdá vhodný Sigfox tak LoRaWAN, ale LoRaWAN poskytuje větší spolehlivost pro logistiku mimo areál, tedy při pohybu vozidel ve větší rychlosti je LoRaWAN vhodnější. V případě NB-IoT nemusí být LTE síť dostupná ve všech místech a tedy i to může být plus pro využití LoRaWAN.

1.2.3 Vyhodnocení

Každá z těchto technologií má své místo v IoT. Sigfox a LoRa slouží jako levnější zařízení s velmi dlouhým rozsahem (vysoké pokrytí) a velmi dlouhou životností baterie. LoRaWAN na rozdíl od Sigfox a NB-IoT slouží také pro lokální sítě a spolehlivou komunikaci při rychlém pohybu zařízení. Nicméně NB-IoT zajišťuje nízkou latenci a vysokou kvalitu služeb, ale za větší cenu.

Hlavně díky možnosti stavby celé vlastní LAN sítě je LoRaWAN velice zajímavou technologií pro IoT v rámci firem či výrobních prostředí, které nechtějí mít vnější vstup do sítě. Z těchto důvodů a také díky cenové výhodě oproti NB-IoT je zvolena tato technologie v práci. A tedy kvůli soukromému nasazení je důležité prověřit celkovou bezpečnost této sítě.

2 LoRaWAN

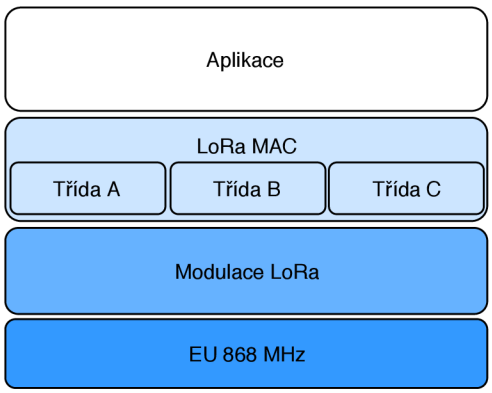
LoRa je bezdrátová modulace, založená na technice rozprostření spektra, uzpůsobená ke komunikaci na velké vzdálenosti s malou spotřebou energie a pomalým přenosem dat. LoRaWAN je protokol zajišťující správný chod celé LPWAN technologie. LoRaWAN síť je vytvořena na základě topologie hvězda, ve které brány přenáší zprávy mezi koncovými zařízeními a síťovým serverem. Síťový server směřuje pakety z každého zařízení sítě do přidruženého aplikačního serveru. Pro zabezpečení sítě LoRaWAN je využíváno symetrické šifrování, pomocí nějž se odvozují relační klíče z kořenových klíčů koncových zařízení. Správu klíčů zajišťuje připojovací server.

Brány jsou k síťovému serveru připojeny pomocí zabezpečeného IP protokolu, zatím co koncová zařízení využívají LoRa komunikaci a komunikují zároveň s jednou nebo více bránami.

Komunikace mezi koncovým zařízeními a bránami je rozprostřena na různé frekvenční kanály a různé datové rychlosti. Rychlost přenosu dat pomocí LoRa je od 0.3 kbps až po 50 kbps [11]. Aby bylo dosaženo co největší životnosti baterie koncového zařízení a také celkové kapacity sítě, lze řídit datovou rychlost pro každé koncové zařízení jednotlivě pomocí schématu adaptivní přenosové rychlosti (ADR).

Koncové zařízení musí dodržet pro vysílání některé parametry. Musí náhodně měnit kanál pro každý přenos, díky tomu se síť stane odolnější vůči rušení. Musí respektovat maximální povolenou střídu a musí také respektovat maximální dobu vysílání podle předpisů.

LoRaWAN je na MAC vrstvě, viz obr.2.1, dělena na tři různé třídy použití A,B a C, popis těchto tříd lze nalézt ve specifikaci [11]. Tato práce se zabývá nejvíce třídou A vzhledem k tomu, že to je jediná třída která má nízkou spotřebu energie, v práci je zmíněna i třída B a to z důvodu specifikace této třídy v LoRaWAN specifikaci 1.1[11].

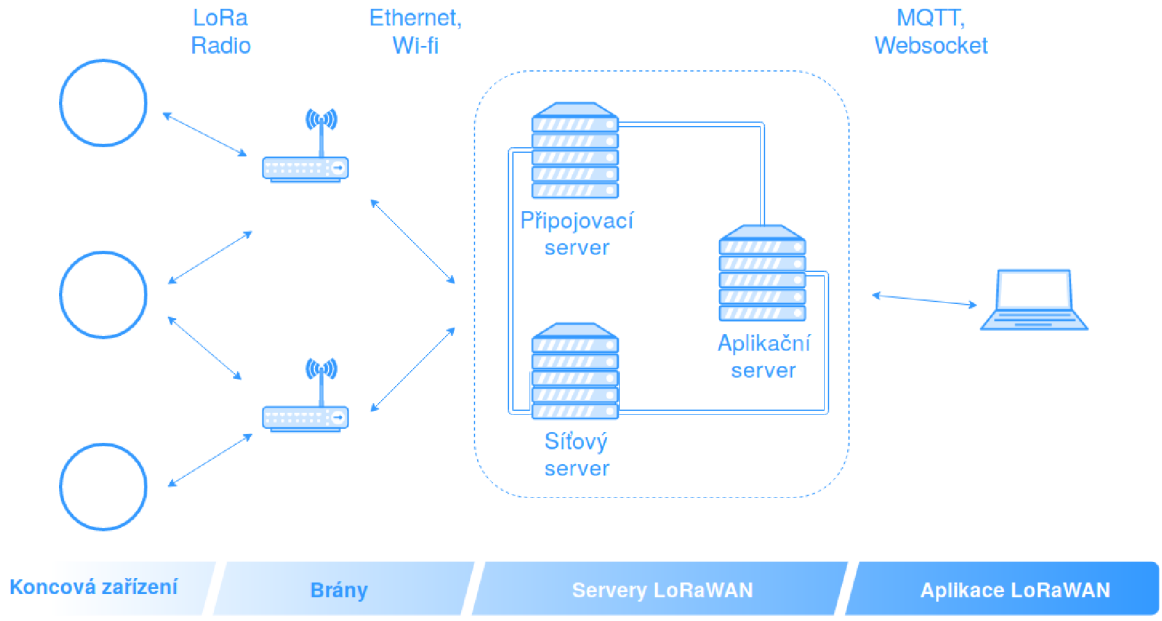


Obr. 2.1: LoRaWAN třídy na vrstvě.

V této části je detailně rozebrána komunikace a její šifrování ve verzích 1.0.x [3] a 1.1.x [4] protokolu LoRaWAN. V této části je popsán přenos zpráv při aktivaci, po aktivaci a jak jsou tyto zprávy šifrovány při přenosu.

2.1 Prvky LoRaWAN sítě a jejich komunikace

Nejdříve je nutné popsat jednotlivé prvky sítě LoRaWAN. Jsou to prvky jako koncové zařízení, brána, síťový server, připojovací server a aplikační server. Je zde také popsána stručně vzájemná komunikace těchto prvků. Na obr.2.2 lze vidět schéma komunikace LoRaWAN.



Obr. 2.2: Schéma komunikace LoRaWAN.

2.1.1 Koncové zařízení

Koncovým zařízením LoRaWAN je zařízení, které je postaveno na čipu, který umí pracovat s modulací LoRa [9], přesněji, který má vysílač a přijímač zpracovávající modulaci LoRa. Nejvíce rozšířeným čipem je RN2483 od firmy Mikrochip. Tento čip bývá nejčastěji propojen s nějakým senzorem a tedy koncové zařízení bývá nejvíce využíváno pro senzorická měření. Koncové zařízení komunikuje v síti LoRaWAN právě bezdrátovou modulací LoRa. Pomocí této modulace komunikuje s bránou. Aplikační vrstva koncového zařízení je propojena s aplikačním serverem, to znamená, že užitečná data aplikační vrstvy (např. hodnoty senzorického měření) jsou směřována na aplikační server.

Aktivace koncových zařízení, je možné dvěma způsoby ABP (Activation By Personalization) nebo OTAA (Over The Air Activation), tyto aktivace jsou více popsány v následujících částech této kapitoly.

2.1.2 Brána

Brána (gateway) slouží jako prostředník pro komunikaci mezi koncovými zařízeními a servery. Pro komunikaci se síťovým serverem používá IP protokol. Brána pracuje na fyzické vrstvě. Pakety jsou odeslané z koncového zařízení na bránu pomocí modulace LoRa a data z brány jsou odeslány na síťový server například pomocí ethernetu.

2.1.3 Síťový server

Je centrem celé LoRaWAN sítě. Funkce síťového serveru jsou kontrola adresy koncového zařízení, autentizace a kontrola čítače rámců, potvrzování a přizpůsobení datové rychlosti, reakce na požadavky přicházející od koncového zařízení, řazení zpráv mezi aplikačním serverem a koncovým zařízením, přesměrování zpráv mezi aplikačním serverem a koncovým zařízením a přesměrování zpráv mezi koncovým zařízením a připojovacím serverem.

2.1.4 Připojovací server

Slouží k bezpečnému řízení aktivace pomocí OTAA. Tento server je podporován až v rámci protokolu LoRaWAN verze 1.1.x. Připojovací server se může připojovat do více síťových serverů a také do jednoho síťového serveru může být připojeno více připojovacích serverů. Každý připojovací server je identifikován unikátní hodnotou nazvanou JoinEUI. Připojovací server poskytuje odvození relačních klíčů FNwkSIntKey, SNwkSIntKey, NwkSEncKey a AppKey. Připojovací server může obsahovat informace pro zařízení pod jeho kontrolou, jako DevEUI, AppKey,NwkKey, identifikátor síťového a aplikačního serveru a LoRaWAN verzi protokolu koncového zařízení. Kořenové klíče NwkKey a AppKey jsou uloženy pouze na připojovacím serveru a koncovém zařízení a nejsou nikdy poskytnuty síťovému nebo aplikačnímu serveru. Připojovací server slouží k zabezpečení, autenticitě a integritě komunikace.

2.1.5 Aplikační server

Pracuje pouze na aplikační vrstvě, to znamená, že pracuje s užitečnými aplikačními daty a zpracovává je. Umožňuje uživateli obsluhovat koncové zařízení na aplikační úrovni. Uchovává pouze informace jako DevEUI a AppSKey pro ověření a zpracování zpráv.

2.2 LoRaWAN verze 1.0.2

Verze tohoto protokolu byla specifikována v roce 2016 [3]. Na této verzi pracuje stále velký počet koncových zařízení a vzhledem k problému zpětné kompatibility, kdy se síť chová podle starší verze se jedná o jednu z největších bezpečnostních slabin protokolu LoRaWAN [12].

2.2.1 Uložené informace na koncovém zařízení

Uložené informace před aktivací:

Na koncovém zařízení musí být před aktivací uloženo několik informací. Tyto informace nejsou na zařízení nijak šifrována a jsou čitelná.

- **DevEUI:** je globální identifikátor zařízení podle IEEE EUI-64 [13]. DevEUI musí být jedinečné číslo sloužící síťovému serveru k rozlišení jednotlivých zařízení. Není důležité, která aktivační procedura byla provedena, je nutností jak u ABP tak OTAA. Pro OTAA aktivaci musí mít zařízení tuto hodnotu uloženou v paměti před začátkem připojovací procedury. Při ABP aktivaci nemusí mít zařízení uložené DevEUI přímo v paměti, ale specifikace doporučuje uložení přímo v paměti.
- **AppEUI:** je globální aplikační identifikátor stejně jako předchozí dle IEEE EUI-64 [13]. Hodnota je poté přenášena v rámci Join-request kvůli identifikaci aplikace, ke které zařízení patří.
- **AppKey:** je 128 bitový aplikační klíč, délku 128 bitů má z toho důvodu, že je používán v rámci šifrování AES. Je to kořenový klíč, který je uložen na zařízení v nešifrovaném tvaru. Používá se vždy při aktivační metodě OTAA pro odvození relačních klíčů (AppSKey a NwkSKey). Tyto relační klíče jsou specifické pro každé koncové zařízení. Následně je těmito relačními klíči šifrována komunikace.

Uložené informace po aktivaci:

- **DevAddr:** Adresa koncového zařízení, která se skládá ze 32 bitů a slouží k identifikaci zařízení v rámci aktuální sítě. Tato adresa je koncovému zařízení přidělena síťovým serverem.
- **NwkSKey** je síťový relační klíč, specifický pro každé koncové zařízení. Tento relační klíč používá síťový server i koncové zařízení k výpočtu a ověření MIC (message integrity code) všech datových zpráv pro zajištění integrity dat. Používá se také pro dešifrování nebo šifrování FRMPayloadu (užitečných dat), ale pouze pokud jsou jako data přenášeny MAC příkazy.

- **AppSKey:** je aplikační relační klíč specifický pro každé koncové zařízení. Používá jej koncové zařízení a aplikační server k šifrování a dešifrování FRM-Payloadu ve kterém jsou přenášena aplikační data.

2.2.2 Aktivace koncových zařízení

V rámci protokolu LoRaWAN jsou dvě možnosti aktivace koncového zařízení. První možností je ABP (Activation By Personalization), tato možnost však neposkytuje dostatečné zabezpečení, dále v textu je popsáno, proč tomu tak je. Druhou možností, která je používána v rámci celé práce, je OTAA (Over The Air Activation), jde o aktivaci tzv. vzduchem a je bezpečnější, než první zmíněná aktivace.

Activation By Personalization

Znamená aktivaci člověkem a to tak, že zadá všechny podstatné informace ručně na zařízení a server. Tím odpadá připojovací procedura pomocí zpráv join-request a join-accept při připojení k síti. Aktivace probíhá tak, že jsou informace jako DevAddr a relační klíče (NwkSKey a AppSkey) ručně přidány do paměti koncového zařízení místo informací DevEUI, AppEUI a AppKey.

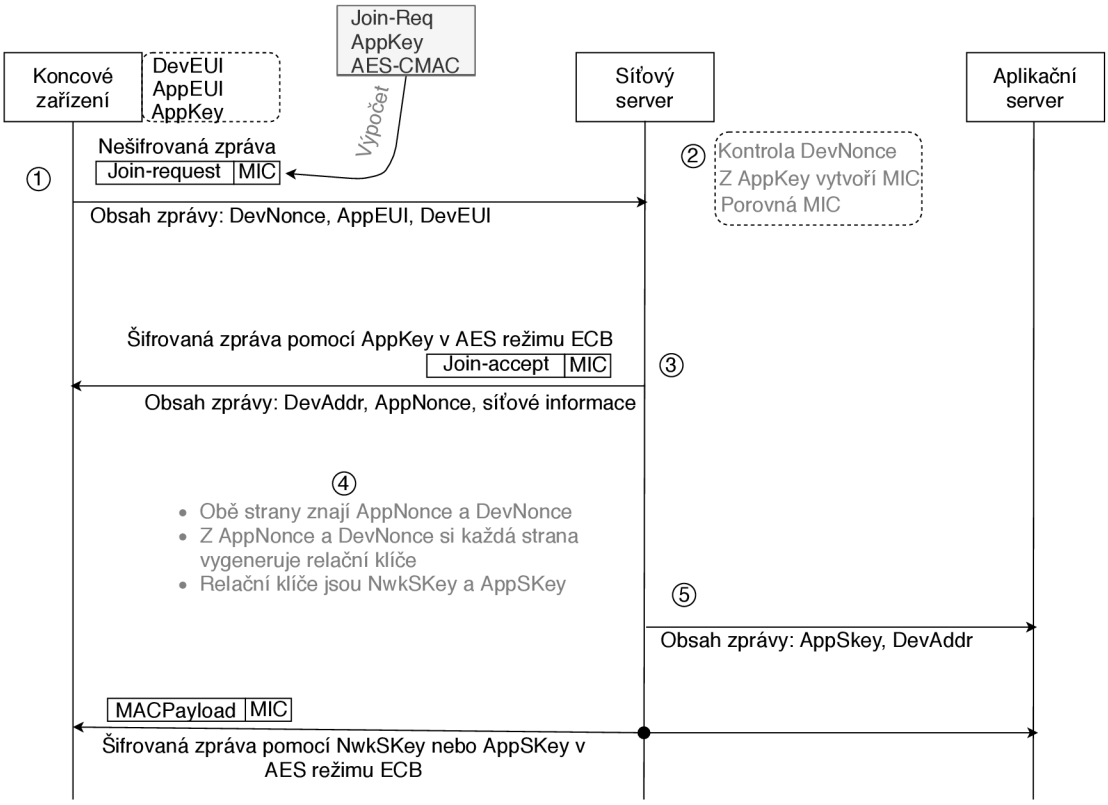
Postup aktivace ve stručnosti je následující. Na koncovém zařízení, síťovém serveru a aplikačním serveru jsou nastaveny potřebné informace, a tak koncové zařízení vysílá paket po té, co je zapnuto. Pokud má koncové zařízení payload k odeslání, vysílá jej v rámci paketu bez jakéhokoli nastavení signalizace v síti. Paket je šifrován pomocí AppSKey a relačních klíčů integrity. Poté co síťový server obdrží paket, provede vyhledání relačních klíčů integrity na základě DevAddr přijatého paketu. Síťový server ověří MIC pomocí nalezených klíčů. Síťový server pak posílá šifrovaný payload přijatého paketu na aplikační server.

Řešení aktivace ABP se nedoporučuje používat, skrz bezpečnostní rizika jakož to viditelné informace relačních klíčů a žádná připojovací procedura. Z toho plyne, že zařízení jsou více náchylnější na útok. Tuto aktivační metodu je doporučeno používat pouze v izolované síti a to jen k testování sítě.

Over The Air Activation

Je aktivační proces, při kterém musí koncové zařízení projít připojovací procedurou. Koncové zařízení musí touto procedurou projít vždy, když je u něj ukončena relace a započne novou relaci. Pro připojovací proceduru je nutné, aby koncové zařízení mělo uloženy informace jak DevEUI, AppEUI a AppKey. V připojovací proceduře dochází k výměně dvou zpráv se serverem a zařízením, jsou to zprávy join-request a join-accept.

Na obr. 2.3 lze vidět průběh registrace zařízení pomocí aktivace OTAA. V textu bude následně tedy popsána detailní aktivace, jak lze vidět na tomto obrázku.



Obr. 2.3: Přenos zpráv v rámci aktivace OTAA.

1. Nejdříve je tedy nutné, jak již bylo popsáno, aby byly na koncovém zařízení uloženy údaje, jako DevEUI, AppEUI a AppKey. Pokud je tedy tato podmínka splněna, tak koncové zařízení odesílá zprávu typu join-req. Tato zpráva obsahuje DevNonce, AppEUI a DevEUI. Nad celou touto zprávou je vygenerován MIC pomocí šifrování AES v režimu CMAC. Zpráva typu join-request není šifrována a tedy všechny údaje v ní obsaženy jsou přenášeny v čitelném formátu.
2. Jakmile síťový server obdrží zprávu typu join-request, zkontroluje zdali nebyla hodnota DevNonce již použita. Poté si vygeneruje MIC nad join-requstem sám za použití hodnot, jež jsou definovány na něm. Poté porovná vlastní MIC s MIC příchozím a pokud se shodují pokračuje registrační procedura k dalšímu kroku.
3. Pokud byl tedy join-request přijat síťovým serverm, tak síťový server vygeneruje DevAddr, AppNonce (náhodně generované číslo) a NetID. Poté síťový server vytvoří zprávu join-accept, která obsahuje tedy DevAddr, AppNonce, NetworkID a další hodnoty síťového nastavení. Na síťovém serveru je vygenerován MIC pro tuto zprávu a celá tato zpráva je poté zašifrována pomocí

AppKey a to šifrováním AES v režimu ECB a odeslána na koncové zařízení.

4. Nyní jak síťový server tak koncové zařízení mají stejné informace a oba znají hodnoty AppNonce a DevNonce. Z těchto hodnot si každý vygeneruje zvlášť relační klíče NwkSKey a AppSKey.
5. Nakonec síťový server posílá relační klíč AppSKey a také DevAddr na aplikační server. Zde už probíhá šifrování například přes HTTPS.

Poté co úspěšně proběhne aktivace, může být již odeslán MACPayload, ve kterém je přenášen FRMPayload, jenž obsahuje aplikační payload nebo MAC příkazy. Zpráva se skládá z hlavičky MHDR poté z hodnoty FHDR, FPort, FRMPayload a nakonec MIC. Z celé zprávy je šifrován pouze FRMPayload a je šifrován jedním z relačních klíčů. Pokud jsou přenášena aplikační data, tak je šifrován aplikačním relačním klíčem. Když jsou přenášeny pouze MAC příkazy, tak je obsah šifrován síťovým relačním klíčem. Šifrování zde probíhá pomocí AES-128 v režimu ECB. Přesný popis šifrování dat je popsán v dalších částech této práce.

2.3 LoRaWAN verze 1.1.X

Verze tohoto protokolu byla specifikována v roce 2017 [4]. Nejdůležitější novinky v rámci protokolu LoRaWAN 1.1.X se týkají bezpečnosti, zařízení třídy B, nových MAC příkazů, zdokonaleného připojování a také podpory pro roaming [14].

2.3.1 Novinky ve verzi 1.1.X

Vylepšení zabezpečení:

V LoRaWAN jsou tři typy problémových útoků. Útok při připojení zařízení k serveru, útok při obdržení zprávy a útok při odeslání zprávy. Pro zmírnění a eliminaci těchto útoků se v LoRaWAN 1.1.X nachází nově čítač rámců, který nesmí být znovu použit během stejné relace. Například u aktivace metodou ABP nebude již možné použít reset a tím vynulovat hodnotu rámce. Z toho vyplývá, že hodnota čítače rámců musí být ukládána v trvalé napěťově nezávislé paměti (NVRAM).

Další velká změna proběhla v hodnotách, které jsou důležité při připojovací proceduře, pro aktivaci pomocí metody OTAA. Jsou to hodnoty DevNonce a AppNonce, hodnota AppNonce byla přejmenována na JoinNonce. Tyto hodnoty nejsou již náhodně generovány, jako tomu bylo u verze 1.0.x. Nyní jsou to hodnoty, které se po připojení zařízení postupně načítají od nuly a jsou uloženy v trvalé paměti zařízení.

Z toho vyplývá další novinka, a to nutnost napěťově nezávislé paměti na koncovém zařízení.

Důležité pro zmírnění útoku při přenosu zprávy je, že součástí zprávy je čítač odeslaných zpráv. To znamená, že v LoRaWAN 1.0 se dá jakákoli odeslaná zpráva

na bránu potvrdit (ACK). V LoRaWAN 1.1 je čítač rámců odeslaných zpráv a ví se, který rámec odeslané zprávy byl aktuálně potvrzen.

Klíče

V nové verzi protokolu LoRaWAN jsou také velké změny v distribuci klíčů. Vedle již známého aplikačního klíče (AppKey) z protokolu LoRaWAN 1.0 je zde nový tajný klíč a to síťový klíč NwkKey. Tento klíč je obdobou síťového relačního klíče (NwkSKey), ale s tím rozdílem že tento klíč je namísto jednoho relačního klíče použit ke generování tří nových relačních klíčů.

Prvním z těchto relačních klíčů vygenerovaných pomocí NwkKey je Network session encryption key (NwkSEncKey). Tento klíč slouží slouží k šifrování MAC příkazů. Další dva klíče Forwarding Network session integrity key (FnwkSIntKey) a Serving Network session integrity key (SnwkSIntKey) k výpočtu a ověření integrity zprávy (MIC). Aplikační relační klíč je zde stále jako u protokolu LoRaWAN 1.0.2. Tento klíč je odvozen z aplikačního klíče (AppKey). Odvození klíčů je ve verzi LoRaWAN 1.1.x více komplexní, než u verze 1.0.2.

Specifikace zařízení třídy B

Další důležitou novinkou v protokolu LoRaWAN 1.1.x je specifikace zařízení třídy B. Třída A je zaměřena na co nejnižší spotřebu a třída C oproti tomu naslouchá stále, a tak má velkou spotřebu. Třída B je něco mezi, jde hlavně o synchronizaci jednotlivých oken mezi zařízením a sítí, a to zaručuje menší energetickou náročnost, než třída C, protože zařízení může být uspáno díky synchronizaci, a tak nenaslouchá stále. Zařízení třídy B pracuje tak, že brána posílá pomocí broadcastu posílá beacon rámec na všechny koncová zařízení z důvodu synchronizace času těchto zařízení. Pokud je tedy vytvořeno komunikační okno s koncovým zařízením, může síťový server nakonfigurovat vlastní přenosovou rychlost a také frekvenci. Koncové zařízení třídy B může také informovat síťový server pomocí rámce o výběru správné brány pro třídu B. Pokud je koncové zařízení v pohybu, informuje síť jednou za čas, která brána je v jeho dosahu a tedy dobrá síla signálu.

Nové MAC příkazy

Ve verzi 1.1.x přibylo několik nových MAC příkazů. MAC příkazy jsou příkazy, které může síťový server použít pro konfiguraci zařízení vzdáleně, ale tyto příkazy slouží také pro koncová zařízení, aby zjistili informace ze sítě jako například ADR (Adaptive Data Rate) limity a prodlevy, které lze nyní konfigurovat pro ADR. Zařízení také může zažádat síťový server o absolutní čas kvůli časové známce (timestamp), a to když je potřeba pro data ze senzorů určit přesný čas. Další MAC příkazy ke

znovupřipojení k síťovému serveru nebo také k obdržení nových relačních klíčů, zařízení také může informovat síťový server, které relační klíče byly použity. Nové MAC příkazy jsou také důležité pro roaming. Nakonec přibyly ještě nové MAC příkazy pro přenosové parametry. Celkově přidávají nové MAC příkazy spousty funkcionalit.

Připojovací postup:

V protokolu LoRaWAN 1.0.2 bylo pro připojení zařízení do sítě potřeba, aby zařízení mělo nakonfigurovány informace jako DevEUI, AppEUI a AppKey. V LoRaWAN 1.1.x je potřeba pro aktivační proceduru konfigurace JoinEUI, DevEUI, NetworkKey a AppKey. Hodnota JoinEUI nahradila v nové verzi hodnotu AppEUI. Pokud síťový server uvidí request message, vyhledá připojovací server podle názvu hostitele, poté je připojovací server zodpovědný za odvození relačních klíčů a posílá tyto klíče na aplikační a síťový server. Připojovací server může být provozován důvěryhodnou třetí stranou, to znamená, že může být provozován společnostmi, které jsou zaměřené na bezpečné ukládání tajných klíčů bez ohledu na operátora síťového serveru, ale samozřejmě také výrobci zařízení můžou hostovat připojovací server, to stejné může i operátor síťového serveru. Připojovací postup umožňuje aktivaci koncového zařízení na hostující síti nebo na jiných sítích. Koncové zařízení může kontaktovat připojovací server a díky tomu měnit síťové servery. Pokud bude změněna síť, tak lze nakonfigurovat v připojovacím serveru tuto změnu a zařízení již může začít používat jinou síť, v podstatě jde o jednodušší přepojování mezi sítěmi.

Novinkou pro LoRaWAN 1.1 je také roaming. Podporuje dva druhy a to pasivní roaming a handover roaming. Pasivní roaming může být podporován i v LoRaWAN 1.0, a to protože zde nezáleží tolik na koncových zařízeních. Síťový server obdrží pakety a přeposílá je na jiný síťový server, pro který jsou tyto pakety určeny. Tento přenos je založen na síťovém ID, které je součástí veřejné adresy zařízení. Handover roaming již vyžaduje zařízení LoRaWAN 1.1, protože zařízení ví o tom, že může jít o komunikaci se sloužícím síťovým serverem a ne přímo o komunikaci s domácím síťovým serverem. Takže domácí síťový server je ten, kde je uložen profil koncového zařízení a to je hlavní server pro toto zařízení. Sloužící síťový server je síťový server, kde je zařízení aktivováno. Je zde také směrovací síťový server pro směrování paketů mezi domácím serverem a sloužícím serverem.

2.3.2 Uložené informace na koncovém zařízení

Uložené informace před aktivací

- **JoinEUI:** je globální aplikační identifikátor podle IEEE EUI-64 [13]. Tento identifikátor slouží k identifikaci připojovacího serveru, který při připojovací

proceduře slouží k odvození relačních klíčů. Pro koncové zařízení aktivované pomocí OTAA, musí být JoinEUI uloženo v paměti zařízení a to ještě před tím, než začne připojovací procedura. U aktivace pomocí ABP není JoinEUI potřeba.

- **DevEUI:**jedná se taktéž o identifikátor EUI-64. DevEUI musí být jedinečné číslo sloužící síťovému serveru k rozlišení jednotlivých zařízení. Zde není důležité, která aktivační procedura byla provedena, je nutností jak u ABP tak OTAA. Pro OTAA aktivaci musí mít zařízení tuto hodnotu uloženou v paměti před začátkem připojovací procedury. Při ABP aktivaci nemusí mít zařízení uložené DevEUI přímo v paměti, ale je to doporučeno.
- **Kořenové klíče zařízení (AppKey a NwkKey):** tyto klíče jsou šifrovány pomocí šifrovacího algoritmu AES-128 [15]. Tyto klíče jsou koncovému zařízení specifikovány již při výrobě. Pokud se koncové zařízení připojí k síti pomocí OTAA, tak NwkKey slouží k odvození relačních klíčů FNwkSIntKey, SNwkSIntKey a NwkSEncKey a AppKey pro odvození relačního klíče AppSKey. Při aktivaci pomocí OTAA musí být NwkKey i AppKey uloženy na koncovém zařízení. U ABP nemusí být tyto klíče uloženy na koncovém zařízení.
- **Klíče JSIntKey a JSEncKey:** jsou klíče, které se odvozují pouze pro aktivaci OTAA mají určitou životnost a jsou odvozeny z kořenového klíče NwkKey. JSIntKey slouží k výpočtu MIC při odeslání zprávy Rejoin-Request a také k odpovědi Join-Accept. JSEncKey slouží k šifrování zprávy Join-Aceept, která byla spuštěna zprávou Rejoin-Request.

Uložené informace po aktivaci

- **Adresa koncového zařízení (DevAddr):**skládá se ze 32 bitů a slouží k identifikaci zařízení v rámci aktuální sítě. DevAddr je koncovému zařízení přidělena síťovým serverem zařízení.
- **Forwarding Network session integrity key (FNwkSIntKey):** je síťový relační klíč specifický pro konkrétní koncové zařízení. Koncové zařízení jej používá pro výpočet MIC (message integrity code) nebo části MIC z důvodu zajištění integrity.
- **Serving Network session integrity key (SNwkSIntKey):** je síťový relační klíč specifický pro konkrétní koncové zařízení. Koncové zařízení jej používá k ověření MIC z důvodu zajištění integrity a také k výpočtu poloviny MIC.
- **Network session encryption key (NwkSEncKey):**je síťový relační klíč specifický pro konkrétní koncové zařízení. Slouží k šifrování a dešifrování MAC příkazů odeslaných jako payload. V LoRaWAN verzi 1.0 slouží pro MAC

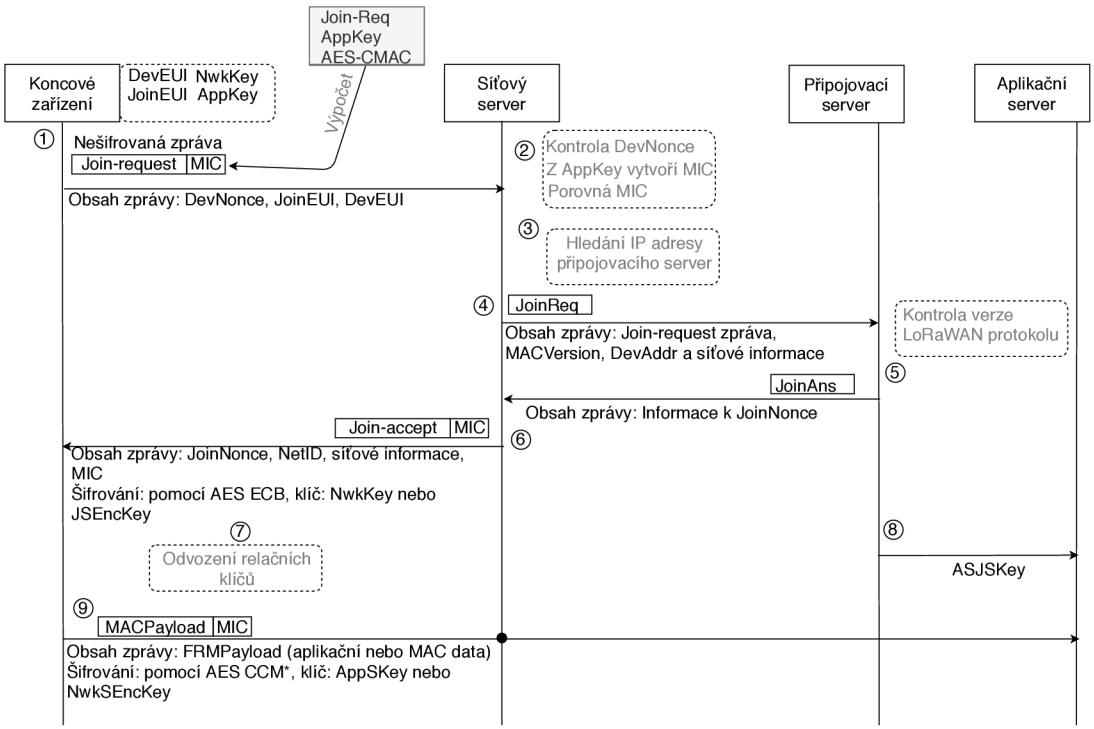
zprávy a pro výpočet MIC pouze jeden stejný klíč.

- **Aplikační relační klíč (AppSKey):**je aplikační relační klíč specifický pro konkrétní koncové zařízení. Klíč používá aplikační server i koncové zařízení a to k šifrování a dešifrování payloadu. Klíč je zde právě pro to, aby síťový server neměl možnost změnit přenášená data a nebo odvodit z těchto dat nějaké informace.

Aktivace koncového zařízení

Jsou dvě možnosti jak aktivovat koncová zařízení. První možností je over the air activation (OTAA) a druhou activation by personalization (ABP).

- **Activation By Personalization (ABP):** Vzhledem k tomu, že aktivace pomocí metody ABP je obdobná jako u starší verze 1.0.2, tak zde již není popsána.
- **Over The Air Activation (OTAA):** Tento typ připojení koncového zařízení je zabezpečenější než ABP. Pro tuto aktivaci využívá koncové zařízení připojovacího serveru a musí projít připojovací procedurou. Před začátkem připojovací procedury musí být na zařízení nastaveny tyto informace: DevEUI, JoinEUI, NwkKey a AppKey. Aktivační postup OTAA se používá pro vzájemné ověření koncového zařízení se sítí, to znamená, že je zajištěna autorizace zpráv. Na obr. 2.4 lze vidět průběh registrace zařízení pomocí OTAA. V textu je popsána aktivace, kterou lze vidět na tomto obrázku.
  1. Jak bylo již dříve v textu napsáno, na zařízení musí být nejdříve nakonfigurovány informace jako DevEUI, JoinEUI, NwkKey a AppKey. Pokud je podmínka splněna, tak koncové zařízení odesílá zprávu typu join-request na síťový server. Tato zpráva obsahuje informace: JoinEUI, DevEUI a DevNonce. Nad celou touto zprávou je vygenerován MIC pomocí šifrování AES v režimu CMAC. Tato zpráva není nijak šifrována.
  2. Poté co síťový server obdrží zprávu typu join-request, zkontroluje hodnotu DevNonce a to tak, že síťový server udržuje hodnotu DevNonce a pokud se hodnota nezvýšila o jednu od předchozí, je zpráva zahozena. Následně si vygeneruje vlastní MIC a porovná ho s příchozím, pokud jsou si rovny, je zpráva přijata.
  3. Síťový server na základě JoinEUI v právě přijaté zprávě join-request použije DNS (Domain Name System) k vyhledání připojovacího serveru, pokud tedy není tato adresa připojovacího serveru síťovém serveru již předem nastavena. Pokud IP adresa připojovacího serveru není nalezena je zpráva zahozena.
  4. Síťový server posílá zprávu typu JoinReq na připojovací server. Obsah



Obr. 2.4: Přenos zpráv v rámci aktivace OTAA LoRaWAN 1.1.x.

této zprávy je celá zpráva join-request, která přišla z koncového zařízení, MACVersion(verze protokolu LoRaWAN), DevAddr a síťové informace.

5. Připojovací server zpracuje JoinReq zprávu na základě verze protokolu LoRaWAN a posílá zpět na síťový server zprávu typu JoinAns. Tato zpráva obsahuje připravený PhyPayload pro zprávu typo join-accept a jsou zde přenášeny informace pro vytvoření relačních klíčů SNwkSIntKey, FNwkSIntKey, NwkSEncKey tedy hodnota JoinNonce.
6. Pokud síťový server úspěšně obdrží zprávu typu JoinAns, může odeslat zprávu typu join-accept na koncové zařízení. Zpráva obsahuje JoinNonce, NetID, DevAddr a další síťové informace a hodnotu MIC. Celá zpráva je šifrována pomocí AES-128 dešifrování v režimu ECB a klíč je použit na základě typu join-requestu. Pokud jde o klasický join-request, je použit jako klíč pro dešifrování ECB klíč NwkKey, pokud se jedná o typ zprávy rejoin-request, pak je použit klíč JSEncKey.
7. Poté již můžou být odvozeny relační klíče, AppSKey, FNwkSIntKey, SNwkSIntKey a NwkSEncKey, odvození každého klíče probíhá pomocí AES-128 šifrování v režimu ECB.
8. Připojovací server posílá také aplikačnímu serveru klíč ASJSKey, tento klíč slouží k dešifrování AppSKey, který přichází se zprávou nesoucí aplikační payload na aplikační server. Je zde také možnost, že aplikační server

bude dostávat přímo AppSKey od připojovacího serveru, to již závisí na nastavení.

9. Následně tedy koncové zařízení odesílá MACPayload, v této zprávě je již obsažen FRMPayload, který již nese aplikační data. Tato zpráva je nejdříve odeslána a síťový server, který ke zprávě přidává šifrovaný AppSKey. Síťový server přeposílá zprávu pak na aplikační server. FRMPayload je šifrován buď pomocí AppSKey to pokud nese aplikační data, a nebo pomocí NwkSEncKey a to pokud nese MAC příkazy. Šifrování probíhá pomocí metody AES-128 v režimu CCM*.

2.4 Bezpečnost

V této části bude rozebrána celková bezpečnost protokolu LoRaWAN a to s ohledem na nejnovější verzi 1.1 [4] tak i na starší verzi 1.0.2 [3]. Jsou zde také popsány problémy v rámci zpětné kompatibility obou verzí. LoRaWAN 1.1 řeší většinu bezpečnostních mezer ze starších verzí, ale v rámci zpětné kompatibility,vzhledem k tomu, že nelze všechny zařízení přenést na verzi 1.1, je v protokolu stále velké množství bezpečnostních rizik. V této části jsou popsány základy bezpečnosti, typy útoků, problémy v rámci zpětné kompatibility a bezpečnostní prvky. Pro popis bezpečnosti ve zpětné kompatibilitě bylo čerpáno z článku věnujícímu se tomuto tématu [12].

2.4.1 Bezpečnostní základy:

Autenticita je výraz, který popisuje zda-li jde v případě LoRaWAN o správné zařízení (chtěné) v síti. V LoRaWAN je autenticita poskytována síťovou vrstvou. Každé zařízení má 4 bajtovou adresu, tato adresa ovšem není unikátní, ale zajišťuje autenticitu. Dále je zde integrita zprávy, integrita v LoRaWAN znamená, zdali nebyla zpráva nějak upravena. K zajištění integrity slouží MIC (Message Integrity Chceck). Integrita zprávy je také zajišťována síťovou vrstvou. Síťový server může ověřit, že jde o správné zařízení pouze pokud má správné relační klíče. Důvěryhodnost zpráv je poskytována síťovou vrstvou i aplikační vrstvou a to pomocí šifrování AES-128.

Bezpečnost na fyzické vrstvě

Fyzická vrstva LoRa protokolu LoRaWAN neposkytuje žádnou úroveň zabezpečení, ale nachází se na ní jeden kontrolní prvek. Jedná se o CRC (cyklický redundantní součet), což je v podstatě indikace toho, které bity byly při přenosu změněny a jestli je zpráva shodná s původní zprávou. Většina prolomení bezpečnosti se děje právě na fyzické vrstvě LoRa.

Link check

Tato funkce umožňuje koncovému zařízení zjistit zda je linka dostupná, ale také předává informace o tom, jaká je kvalita signálu a kolik bran pokrývá zařízení. To je velmi důležitá informace pro koncové zařízení, aby vědělo zda je vůbec v místě, které pokrývá brána. Pokud pokrytí není, mohou použít jiný prostředek pro komunikaci nebo zkusit komunikaci později.

Potvrzení datové zprávy:

Jde o potvrzení přijatých a odeslaných zpráv. To je pro koncové zařízení ujištění, že odeslaná datová zpráva byla přijata síťovým serverem nebo také naopak že zpráva odeslaná od serveru na zařízení byla přijata koncovým zařízením. Zařízení čeká na odpověď síťového serveru pomocí ack. Zařízení může opakovat odeslání zprávy do té doby než přijde ack.

Optimalizace využití kanálu:

Pracuje pomocí ADR (adaptive data rate). Tento mechanizmus redukuje paketovou ztrátovost a také přispívá do bezpečnostního řešení, vzhledem k tomu že velká ztrátovost paketů činí řešení méně bezpečným.

Detekce útoků opakováním

Opakování zpráv v LoRaWAN může být velmi nebezpečné. Například odeslané zprávy z koncových zařízení mohou spouštět různé akce, po provedení útoku man in the middle může být opakovaně zasílána stejná zpráva, která spouští určitou akci a bylo by tak možné vyřadit zařízení.

Veřejné informace v LoRaWAN

V LoRaWAN se nachází některé informace, které jsou veřejné. S ohledem na bezpečnost je důležité si uvědomit, že některé informace, které jsou posílány vzduchem, může zachytit kdokoli.

V podstatě se jedná o všechny informace ve zprávě typu join-request (JoinEUI, DevEUI, DevNonce, AppEUI). Například JoinEUI odkazuje na připojovací server, ale také říká útočníkům, kde jsou uloženy kořenové klíče. U DevEUI je nebezpečí to, že jde o identifikátor koncového zařízení a může také označovat LoRaWAN modul a verzi nebo také model a cenu koncového zařízení, které bylo použito. Další veřejnou informací je adresa koncového zařízení, což je 4 bajtová adresa, kterou může používat více zařízení, tato adresa může označovat LoRaWAN síť, jelikož prvních pár bitů

označuje použitou síť LoRaWAN. Také je veřejně viditelná délka aplikačního payloadu a také příkazy MAC jsou viditelné. Proto je dobré maskovat aktivitu. Důležité je používat unikátní DevEUI, jehož číslo se nevztahuje k předchozím informacím. Použitím fixní délky payloadu nelze odvodit žádnou informaci.

Použité šifrování

LoRaWAN používá šifrování AES se 128 bitovými klíči. V rámci AES LoRaWAN pracuje v režimu CCM(čítač s cbc-mac),CMAC a ECB(electronic codebook). CCM režim poskytuje důvěryhodnost a autentizaci. CCM režim je v LoRaWAN použit pouze pro šifrování MAC příkazů a aplikačních dat. CMAC režim je použit pro autentizaci. ECB režim je použit pro odvození klíčů z kořenových klíčů a také pro šifrování potvrzovacích zpráv. Kontrola integrity se provádí pouze na koncovém zařízení.

Klíče

Protokol LoRaWAN používá v závislosti na verzi různé klíče pro šifrování. Tyto klíče zaručují integritu a autenticitu zpráv. Také zabezpečují zprávy proti přečtení. Klíče v protokolu LoRaWAN jsou pro každé zařízení specifické. V protokolu LoRaWAN 1.1.0 se o distribuci klíčů stará připojovací server. Klíče v protokolu LoRaWAN jsou kořenové a relační. Kořenové slouží především pro odvození relačních klíčů.

2.4.2 Zpětná kompatibilita protokolu LoRaWAN

V rámci zpětné kompatibility mezi protokoly verze 1.0.2 a 1.1.x může dojít k prolomení bezpečnosti. Ve verzi 1.0.2 je několik bezpečnostních mezer například ve správě relací, v připojovací proceduře, v potvrzovacím mechanizmu a v rámci integrity. V protokolu 1.1 je většina těchto bezpečnostních mezer vyřešena. V této části je popsáno řešení bezpečnostních mezer z verze 1.0.2.

Opakované použití hodnoty čítače rámců

Verze 1.0.2 není nijak zabezpečena proti opakovanému použití hodnoty čítače rámce. Tato bezpečnostní mezera se týká jak aktivace pomocí ABP tak pomocí OTAA. K opakovanému použití hodnoty rámce dojde, když je koncové zařízení resetováno nebo když je čítač zaplněn v rámci ABP. V rámci OTAA k opakovanému použití dojde, když je čítač zaplněn během relace. LoRaWAN 1.1 tuto chybu zabezpečení řeší tím, že zavádí mechanizmus pro novou tvorbu klíčů u koncového zařízení (rekeying). Tento mechanizmus je proveditelný díky novému typu zprávy Rejoin-request a novému MAC příkazu ForceRejoinReq. Před zaplněním čítače musí být vytvořen

nový relační kontext. U ABP musí být čítač rámců uložen v napěťově nezávislé paměti a tato paměť nesmí být resetována navzdory ztrátě napájení nebo restartu samotného zařízení.

Opakované použití hodnoty nonce

Podle definice znamená hodnota nonce hodnotu, která může být použita pouze jednou. Hodnoty nonce ve verzi 1.0.2 (DevNonce a AppNonce) jsou náhodně vygenerovány a tak se může stát, že může být znovu vygenerována a použita hodnota, která již byla jednou použita, jelikož ve verzi 1.0.2 nejsou sledovány předešlé hodnoty. Nemůže být tedy zcela zabráněno opětovnému použití této hodnoty. Ve verzi 1.1 je tato bezpečnostní mezera řešena pomocí čítače nonce hodnot a opětovnému použití je zabráněno ukládáním a sledováním těchto hodnot.

Nedostatečný mechanizmus ochrany proti přehrávání potvrzovací zprávy

Tento ochranný mechanizmus je nedostatečný v rámci verze 1.0.2 z toho důvodu, že koncová zařízení, pracující na této verzi, nejsou schopna sledovat dostatečný počet hodnot. Koncová zařízení, pracující na verzi 1.1, sledují poslední hodnotu JoinNonce pojmenovanou jako JoinNonce_last a udržují tuto hodnotu. Koncové zařízení příjme JoinNonce pouze, když je tato hodnota navýšena oproti poslední sledované hodnotě (JoinNonce_last).

Nedostatečný mechanizmus ochrany proti přehrávání zprávy žádosti

Síťový server 1.0.2 nesleduje všechny hodnoty DevNonce, a tak útočník může být schopen znovu přehrát zprávu Join-request. Síťový server, pracující na verzi 1.1, sleduje poslední hodnotu DevNonce a udržuje tuto hodnotu v paměti jakožto DevNonce_last a tak musí být stejně jako u předchozího případu vždy hodnota vyšší než poslední hodnota.

Potvrzovací zpráva nesouvisející s žádostí

V rámci verze 1.0.2 není žádná asociace mezi potvrzovací zprávou a žádostí, která tuto zprávu spustila. Ve verzi 1.1 je tato bezpečnostní mezera řešena zahrnutím DevNonce ve výpočtu MIC v rámci potvrzovací zprávy. Koncové zařízení očekává potvrzovací zprávu při odpovědi na zprávu žádosti, kterou odeslalo na základě hodnoty DevNonce. MIC kontrola selže pokud byla potvrzovací zpráva spočítána jinou hodnotou DevNonce než zařízení očekává.

Selhání přepínání relačního bezpečnostního kontextu

V rámci verze 1.0.2 koncové zařízení neověřuje relační kontext. Může vzniknout situace, kde koncové zařízení a síťový server končí s různými bezpečnostními kontexty a neprobíhá mezi nimi komunikace. Verze 1.1 řeší tento problém zavedením nových MAC příkazů RekeyInd a RekeyConf. RekeyInd je síťovým serverem interpretován jako potvrzení nové relace a je poslán na koncová zařízení. Po potvrzení od koncových zařízení, síťový server zahazuje všechny staré bezpečnostní kontexty, které udržuje. RekeyConf je odpovědí a umožňuje koncovému zařízení ověřit bezpečnostní kontext na síťovém serveru.

Chybějící end-to-end ochrana

Výpočet kontroly MIC je prováděn na síťovém serveru. To znamená, že je integrita aplikačních dat nechráněná, když jsou data přenášena ze síťového serveru na aplikační server. Navíc může dojít i k úpravám aplikačních dat. Integrita těchto dat je ponechána na koncové aplikaci. Pokud není zavedena end-to-end integrita, bezpečnost aplikačních dat závisí pouze na poctivosti síťového serveru a jeho zabezpečení a zabezpečení kanálů mezi síťovým serverem a aplikačním serverem.

2.4.3 Útoky na LoRaWAN verze 1.0.2

Přehrávání zpráv a jejich odposlech

Tento útok lze aplikovat na koncová zařízení aktivována jak pomocí ABP tak OTAA. Opětovné použití hodnoty čítače rámce umožní útočníkovi přehrát a nebo odposlechnout zprávu. Díky tomu je útočník schopen zachytit a sledovat více zpráv, které jsou vytvořeny pomocí stejných relačních klíčů, jež mají stejnou hodnotu čítače rámců. V rámci úspěšného útoku přehrání zprávy, síťový server při komunikaci s aplikačním serverem zahazuje legitimní data z koncového zařízení, a tak aplikační server zpracovává sice platná data ale stará. Díky opětovnému použití čítače rámců může útočník odposlechnout zprávu, jelikož může obnovit text z šifrované zprávy.

Přehrávání zpráv a odposlech v rámci falešné relace

Jde o podobný útok jako v předchozím případě, ale zde je falešná relace vytvořena na síťovém serveru. Zde je využita zranitelnost v rámci zprávy žádosti o spojení. Když je zpozorována určitá hodnota AppNonce, útočník přehraje zprávu Join-request z již zaznamenané relace na síťovém serveru a tím vyvolá novou falešnou relaci. Potvrzení relace na síťovém serveru nezabrání útoku, vzhledem k tomu, že útočník má platné rámce, které jsou vytvořeny pro použití nové relace a jejich přehrání způsobí, že

síťový server potvrdí falešnou relaci. I v nové verzi 1.1 je útočník schopen přesvědčit síťový server, aby použil novou relaci.

Ack spoofing

Tento útok zneužívá nedostatku asociací mezi uznáním a potvrzením dat. Útočník může zachytit potvrzení (ack) zprávy odeslané ze serveru na koncové zařízení a později použít toto potvrzení k potvrzení odeslané zprávy z koncového zařízení na síťový server. Útok může být proveden i v opačném směru, tedy když koncové zařízení odesílá zprávu na síťový server, ale v tomto případě musí útočník předcházet příjmu rámců odeslaných z koncového zařízení na síťový server pomocí brány, pracující na stejném rozsahu.

Bit flipping

Tento útok využívá chybějící ochrany integrity aplikačních dat v rámci end-to-end. Při tomto útoku se předpokládá, že transportní vrstva mezi síťovým a aplikačním serverem není zabezpečena a že útočník muže pracovat na kanálu mezi síťovým a aplikačním serverem. Díky tomu je útočník schopen provést úpravu přenesených dat aplikací. Díky tomuto útoku může být ohrožena důvěrnost aplikačních dat.

Odepření služby na koncovém zařízení pomocí přehrání potvrzovací zprávy

Tento útok využívá nedostatečného mechanizmu ochrany proti přehrávání potvrzovací zprávy. Útočník odpoví na žádost o připojení od koncového zařízení ještě před síťovým serverem. Koncové zařízení odvodí útočníkův relační klíč. Koncové zařízení a síťový server skončí každý s jinými odvozenými klíči a ztratí schopnost komunikovat, což vede k odepření služeb.

Odepření služby na koncovém zařízení pomocí přehrání zprávy žádosti

Tento útok využívá nedostatečného mechanizmu ochrany proti přehrávání zprávy žádosti. Když je zpráva žádosti přehrána útočníkem v libovolném čase, tak se na síťovém serveru vytvoří nová relace. Útočník sice není schopen zajistit, aby síťový server potvrdil novou relaci a zrušil tu starou, nicméně tím, že čeká na to až koncové zařízení odešle žádost o připojení a přehraje sám zprávu s žádostí o připojení před příchodem potvrzení relace z koncového zařízení, útočník je schopen rozhodit komunikaci síťovému serveru a koncovému zařízení. Pokud síťový server udržuje více nepotvrzených relací na koncové zařízení, přehráním více žádostí může být síťový server přinucen zahodit relační kontext koncového zařízení bez čekání na jakékoli potvrzení.

2.4.4 Zpětná kompatibilita verze 1.1

Síťový server je zodpovědný za rozhodnutí, jaká verze protokolu bude používána a vybírá nejvyšší běžnou verzi mezi ním a koncovým zařízením. Toto rozhodnutí je směřováno na připojovací server v rámci pole MACVersion zprávy JoinReq nebo RejoinReq, lze vidět na obr.2.4. Připojovací server zpracovává JoinReq nebo RejoinReq odeslaný koncovým zařízením podle hodnoty pole MACVersion. Připojovací server nastaví OptNer flag v potvrzovací zprávě podle obdržené hodnoty pole MACVersion. Koncové zařízení se tedy dozví rozhodnutí síťového serveru a podle toho pracuje. Například koncové zařízení vybírá strategii odvození relačních klíčů a podle potřeby vyjednává se síťovým serverem pomocí MAC příkazů RekeyIND a RekeyConf.

Koncové zařízení pracující na verzi 1.1 a síť na verzi 1.0.2

V tomto případě je tedy síťový server starší verze a koncové zařízení pracuje na verzi 1.1, ale chová se jako ve verzi 1.0.2 pro odvození relačních klíčů a odvozuje všechny relační klíče zahrnující AppSKey pomocí kořenového klíče NwkKey. Kotrolu MIC, výpočet MIC a šifrování provádí koncové zařízení podle specifikace 1.0.2. Koncové zařízení 1.1 musí pracovat jako 1.0.2 i v jiných případech. Koncové zařízení nemůže použít ochranu proti přehrávání zpráv, která je postavena na sledování hodnoty JoinNonce_last, protože síťový server používá náhodné AppNonce. Mechanizmus potvrzování relačního kontextu taktéž nelze použít, jelikož MAC příkazy RekeyInd a RekeyConf nejsou známy na koncové straně síťového serveru.

Koncové zařízení pracující na verzi 1.0.2 a síť na verzi 1.1

I v tomto případě musí být upravena zpětná kompatibilita, když síťové prvky chtějí komunikovat s koncovým zařízením. Připojovací server se pro odvození relačních klíčů musí chovat jako ve verzi 1.0 a odvodit všechny relační klíče použitím jednoho kořenového klíče. Kontrolu MIC, výpočet MIC a šifrování provádí síť jako ve specifikaci pro verzi 1.0.2. Ze stejného důvodu jako v předchozím případě nemůže použít ochranu proti přehrávání zpráv. Nemůže být také použit mechanizmus potvrzení relace, síť nemá možnost změnit klíče na koncovém zařízení pomocí MAC příkazu ForceRejoinReq.

3 Návrh řešení LoRaWAN SDR

V rámci této části je nejdříve vybrán hardware a software potřebný ke konstrukci LoRaWAN sítě a k softwarovému radiu. Postupně jsou vybrány všechny komponenty potřebné pro sestrojení sítě a také pro následné zachytávání komunikace probíhající v této síti. Je také zvolen vhodný software pro SDR. Po té je popsána konstrukce a zprovoznění LoRaWAN sítě. Nakonec je zde popsán návrh dekódování založen na již hotových řešeních v této oblasti.

3.1 Výběr HW a SW pro LoRaWAN síť

3.1.1 Výběr Hardwaru

Brána

V rámci tvorby vlastní LoRaWAN sítě byla vytvořena vlastí brána z jednotlivých součástí. Jednotlivé součásti brány byly zvoleny na základě již velké řady hotových a správně funkčních řešeních postavených na jednotlivých součástech [16, 17]. Pro tvorbu brány byly tedy zvoleny následující součásti.

Raspberry Pi 3: Zařízení, které je použito jako základová deska celého řešení brány. Toto zařízení umožní konfiguraci koncentrátoru. Zařízení bylo zvoleno z toho důvodu, že většina bran je právě postavena na tomto zařízení, a tak je ke zprovoznění brány dostupné velké množství literatury. Další výhodou je dostupnost tohoto zařízení.

Koncentrátor ic880a: LoRaWAN koncentrátor pracující na frekvenci 868 MHz. Zařízení dokáže přijímat pakety od různých koncových zařízení, odeslaných s různou hodnotou činitele rozprostření(spreading factor) až na osmi různých kanálech současně. Základem koncentrátoru je čip sx1301. Zařízení bylo zvoleno z důvodu velkého množství dostupné dokumentace ke zprovoznění a také z toho důvodu, že v době výběru zařízení nebyly konkurenční koncentrátory používány. Konkurencí tomuto koncentrátoru je v současnosti koncentrátor RAK831, ale ic880a má výhodu lepší ceny.

Ostatní součásti Pro snadnější propojení Raspberry Pi a koncentrátoru ic880a byla zvolena redukční deska iC880A LoRaWAN Gateway Backplane v2.0 [18]. V době výběru hardwaru se tato deska jevila jako nejvhodnější řešení hlavně díky ceně. Dále byla použita anténa na frekvenci 868 MHz.

Koncové zařízení

Koncové zařízení bylo postaveno na desce pro univerzální použití s názvem Things UNO, která je postavena na Arduinu Leonardo, jediné v čem se liší od Arduina je přidaný chip RN2483 od firmy Microchip [19]. Díky tomu umožňuje snadné a rychlé připojení do sítě a univerzální použití pro různé senzory. Díky použití Arduina lze používat jazyk wiring k nakonfigurování senzorů. Jako senzor byl zvolen DHT11.

Hardware pro softwarově definované radio

Při výběru hardwaru byly vzaty v potaz tato zařízení RTL-SDR USB [20], USRP1 WBX [21], LimeSDR mini [22], YARD Stick One [23] a HackRF One [24]. Jednotlivá zařízení jsou porovnány v rámci Tab. 3.1. Údaje v tabulce byly převzaty ze stránek a oficiálních dostupných zdrojů, jež jsou uvedeny u jednotlivých zařízení. Ke všem zařízením nebyly nalezeny všechny informace. Pro testování softwarově definovaného radia pro LoRaWAN bylo nejprve zvoleno nejlevnější řešení a to RTL-SDR USB, které je dostačující pro příjem signálu, jeho výhodou je velmi nízká cena. RTL-SDR USB pracuje pouze jako přijímač a ne jako vysílač k testování a seznámení se s SDR a GNU Radiem je vhodné. V práci bude třeba i zařízení, které dokáže vysílat. Pro to bylo v rámci výběru zvoleno zařízení LimeSDR mini jakožto nejvhodnější zařízení, vzhledem k jeho ceně a vybavenosti. Levnějším zařízením, které zvládne vysílat bylo pouze Yard Stick One, k tomuto zařízení bohužel není dostatek kvalitní dokumentace a také použitelnost není tak všestranná jako u LimeSDR mini. Zařízení Yard Stick One také pracuje pouze na některých frekvencích. Dalším podobným zařízením je HackRF One, který má větší frekvenční rozsah, ale oproti LimeSDR mini je výrazně dražší.

V rámci práce bude použito pro softwarově definované radio pro LoRaWAN, LimeSDR mini zařízení, se kterým je možno i vysílat. Pokud bude vysílací výkon slabý, je možnost použít zařízení USRP1 WBX a porovnat výsledky těchto dvou zařízení.

3.1.2 Software

Radiová komunikace a protokoly jsou převážně založeny na hardwaru, a tak není zrovna jednoduché přeprogramovat nebo přenastavit jejich nastavení. Tento nedostatek flexibility je problémový hlavně v případě, že dojde k chybě v hardwaru, firmwaru nebo softwaru. Tyto chyby nelze snadno a efektivně vyřešit. Zařízení jsou závislá na funkčnosti hardwaru a nemohou být znovu nakonfigurovány, tak aby dokázaly pracovat s jinými bezdrátovými protokoly nad rámec toho, co poskytuje

Tab. 3.1: Srovnání parametrů zařízení pro SDR.

	LimeSDR mini	**RTL-SDR**	**Yardstick One**	**HackRF One**	**USRP1 WBX**
Cena	139 $	20 $	100$	299$	620$
Frekvenční rozsah	10 MHz– 3,5 GHz	24 MHz– 1766 MHz	max 1 GHz	1 MHz– 6 GHz	50 MHz –2,2 GHz
Použitý chip	LMS7002M	820T2	CC1111	MAX2837, MAX5864, RFFC5072	–
Šířka pásma RX	30,72 MHz	3,2 MHz	–	20 MHz	40 MHz
Kanály RX/TX	1/1	1/0	1/1	1/1	1/1
Vzorkovací rychlost	30,72 MS/s	3,2 MS/s	–	20 MS/s	64 MS/s
Vysílací výkon	10 dBm– 15 dBm	–	18 dBm– 20 dBm	10 dBm– 15 dBm	18 dBm– 20 dBm

samotný hardware. Softwarově definované radio má za cíl poskytnout řešení těchto problémů.

Softwarově definované radio je navrženo jakožto vzorec pro zařízení s bezdrátovou komunikací, pomocí něj lze softwarově přeprogramovat a překonfigurovat různé typy rádiových komunikací. Fyzické komponenty tohoto radia jsou pouze anténa a převodník analogového signálu na digitální na straně přijímače a rovněž tak převodník z digitálního signálu na analogový a anténa na straně vysílače. Zbytek funkcí lze nastavit softwarově [25]. Pro softwarové nastavení a práci s radiem existují různé SDR frameworky, tedy je zapotřebí software, který umožňuje interakci. Nejpoužívanějším softwarem pro práci s SDR radiem je GNU radio a také Matlab. Úspěch GNU radia a Matlabu spočívá hlavně v tom, že poskytují snadno ovladatelné nástroje pro manipulaci se signály. V rámci této práce bylo pro SDR zvoleno GNU radio vzhledem k většímu množství knihoven v rámci LoRa.

GNU Radio

GNU radio je software umožňující uživatelům navrhnout, simulovat a použít různé radiové technologie v reálném světě. Jedná se o graf, který je postaven na velkém množství knihoven bloků, které lze zpracovávat, kombinovat a vytvářet díky těmto blokům aplikace pro zpracování signálu. GNU radio může být použito pro rozsáhlou

oblast reálných rádiových aplikací jako například zpracování zvuku, mobilní komunikace, sledování družic, radary a další. To vše pomocí počítačového softwaru.

Při tvorbě radiokomunikačních zařízení byl vytvořen specifický obvod pro detekci specifické signálové třídy. Navrhnout takový specifický integrovaný obvod, který by zvládl dekódovat a zakódovat určitý přenos a ladit jej je velmi nákladné. SDR zpracovává signály pomocí algoritmů v softwaru v počítači. GNU Radio framework je určený pro psaní aplikací, ve kterých jde zpracovávat signál v počítači. Funkce GNU radia jsou zprostředkovány v snadno použitelných blocích, nabízí vynikající škálovatelnost, poskytuje rozsáhlou knihovnu standardních algoritmů a funkce GNU radia lze velice dobře optimalizovat pro velké množství různých platforem. Právě kvůli dobré optimalizaci bylo GNU radio v této práci zvoleno pro softwarově definované radio pro LoRaWAN.

3.2 Zprovoznění komunikace LoRaWAN

V rámci této části je popsáno, jak byly sestaveny jednotlivé prvky LoRaWAN sítě a jak byla zprovozněna komunikace v rámci tohoto protokolu. Nejdříve je popsán postup konstrukce brány z prvků popsaných v předchozí části. Poté je popsána konstrukce koncového zařízení a nakonec připojení na server a výběr serveru. Při zprovoznění komunikace vzniklo několik problému, které jsou v rámci textu zmíněny a je popsáno jejich řešení.

3.2.1 Konstrukce brány

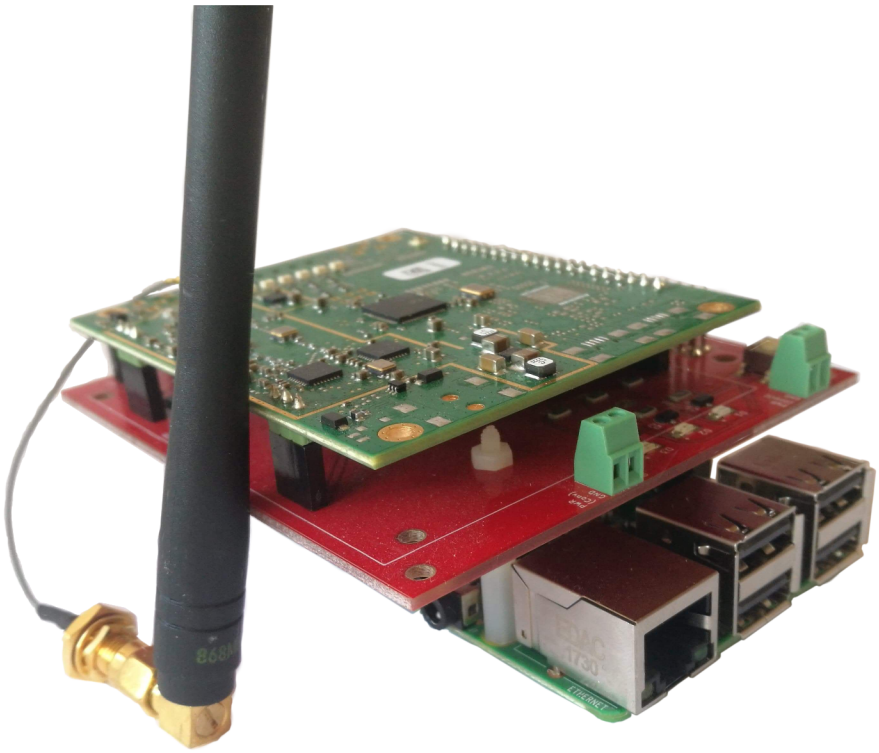
Jak již bylo řečeno, v rámci práce byla sestavena vlastní brána z komponentů zmíněných v předchozím textu. Podrobný výpis součástek potřebných na sestavení brány lze vidět v Tab.3.2. Z jednotlivých součástek byla sestavena brána, jak lze vidět na obr.3.1. Poté bylo nutné nakonfigurovat celou bránu pro připojení na síťový server. V rámci testování byl prozatím zvolen síťový server Loriot [26]. V rámci práce byl zprovozněn i vlastní LoRaWAN server, ale zatím na něj nebyla připojena brána. Popis zprovoznění serveru bude popsán v další části.

Postup nastavení brány

Aby bylo možno bránu nakonfigurovat, byl na Raspberry Pi nainstalován operační systém Raspbian, ve kterém jsou prováděny všechny konfigurace brány. Brána byla nejdříve nakonfigurována podle oficiálních souborů TTN [27]. Poté byla na bráně nastavena konfigurace podle Loriotu [26]. Zde nastal problém, jelikož se bránu nedařilo spojit se serverem. Problém byl v knihovně libssl, na zařízení se musí nacházet

Tab. 3.2: Součástky ke stavbě brány.

Součástka	**Cena**	**Odkaz**
IC880A SPI koncentrátor	3 075 Kč	[28]
Rapberry Pi 3 B+	863 Kč	[29]
Redukční deska IC880A	1 120 Kč	[30]
Napájecí zdroj	264 Kč	[31]
Pigtail pro iC880A-SPI	168 Kč	[32]
Anténa 868 MHz	120 Kč	
MicroSD karta 16 GB	460 Kč	



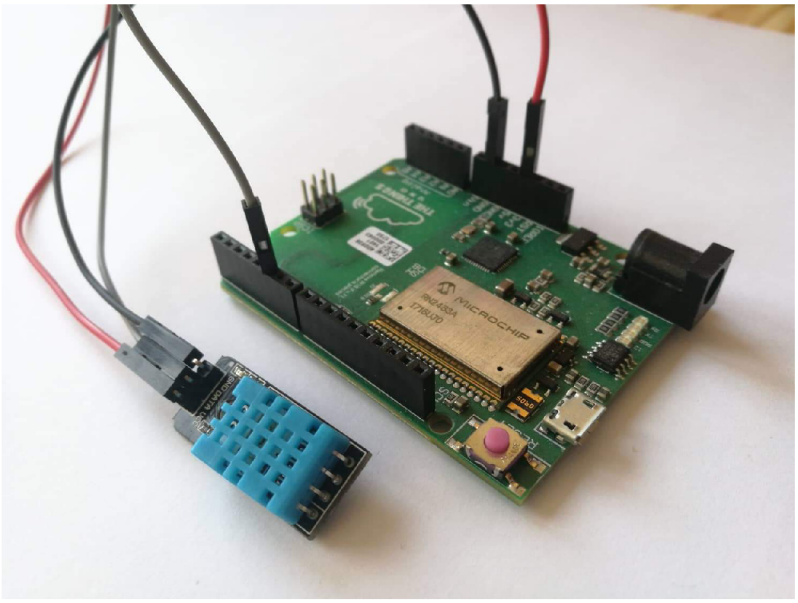
Obr. 3.1: Brána LoRaWAN.

knihovna libssl verze 1.0.0 a ne novější. Po nainstalování knihovny libssl 1.0.0 byl pokus o spojení se serverem Loriotu již úspěšný. Brána byla nakonfigurována a funkční, po sestavení koncového zařízení byla na bránu odeslána úspěšně data a brána pracovala správně.

3.2.2 Konstrukce koncového zařízení

Koncové zařízení bylo zkonstruováno na desce Thing UNO, která je upravenou verzí Arduina s čipem RN2483 od firmy Microchip. K zařízení byl připojen teplotní senzor

DHT 11 pouze pro otestování, zda-li koncové zařízení posílá data na server Loriot. Zařízení lze vidět na Obr.3.2. Senzor byl nastaven v programu Arduino IDE pomocí již hotové knihovny TheThingsNetwork. Tato knihovna obsahuje již připravený kód pro komunikaci tohoto zařízení. Poté byla přidána knihovna pro senzor DHT11 a mírně poupraven kód pro zasílání dat. Zařízení se podařilo zprovoznit a správně komunikovalo se serverem Loriot. Při konstrukci koncového zařízení nedošlo k žádnému problému.



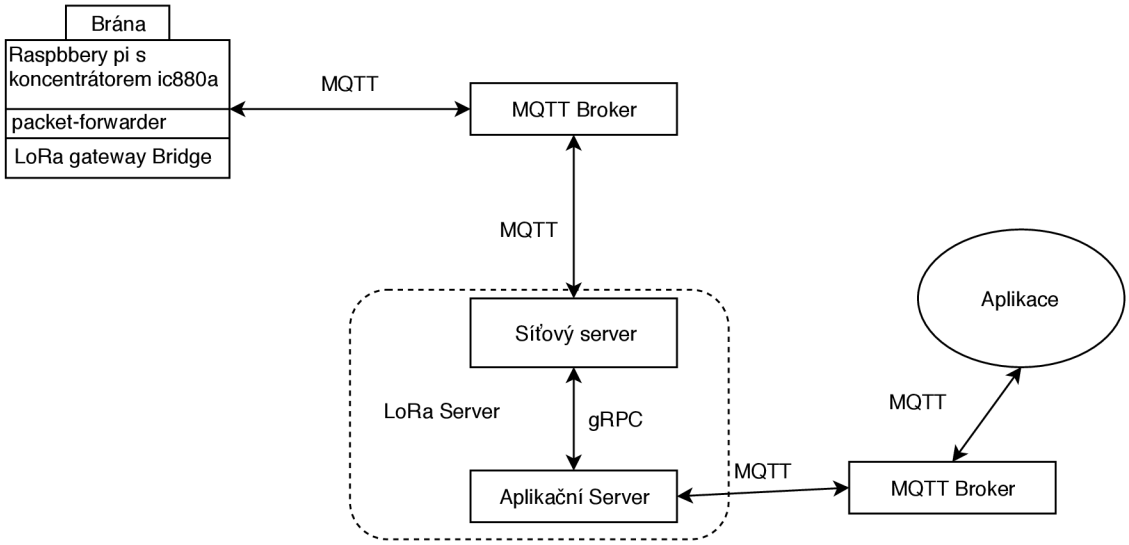
Obr. 3.2: Koncové zařízení LoRaWAN.

3.2.3 LoRaWAN server

Vzhledem k lepší manipulaci a přesnějším dohledem nad serverem, byl zprovozněn vlastní LoRaWAN server, který umožňuje vytvořit celé síťové řešení LoRaWAN. Tento server byl zprovozněn na stejném raspberry pi jako brána. Server byl zprovozněn pomocí open source řešení [33], které autor vytvořil pro jednodušší optimalizaci vlastních serverů.

Pro úspěšný provoz serveru, musí být nejdříve na bráně packet-frowarder [34]. Packet-forwarder je program nutný k běhu brány, směřuje radio frekvenční pakety, které po přijetí od koncového zařízení obdrží koncentrátor, na síťový server pomocí IP/UDP nebo MQTT [35] a naopak odesílá pakety, které přijal od síťového serveru. Pokud je na bráně přímo LoRa gateway bridge, tak je možné posílat pakety rovnou pomocí MQTT. Pokud je LoRa gateway bridge mimo bránu, tak jsou pakety na něj směřovány pomocí IP/UDP a poté z něj směřovány na server pomocí MQTT. Mezi serverem a LoRa gateway bridgem pracuje samozřejmě MQTT broker. Síťový

server pak posílá skrz vnitřní logiku pomocí gRPC [36] paket na aplikační server, z něj je v tomto případě pomocí Node-red [37] a mqtt brokeru směřován na koncovou aplikaci. Na obr.3.3 lze vidět, jak probíhá komunikace brány se serverem v této práci.



Obr. 3.3: Komunikace brány s LoRa serverem.

4 Dešifrování LoRaWAN

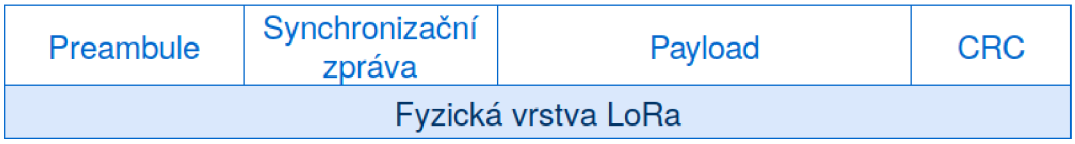
4.1 LoRaPHY

V této části je popsána fyzická vrstva LoRa. Je zde rozebrána její demodulace na základě dostupné literatury a dokumentace. Informace o demodulaci LoRa a o dosažených výsledcích byly převzaty z několika článků a prací, které se věnují tematice LoRa a SDR [38, 9, 39, 40]. V rámci této dokumentace bylo zjištěno, jak pracuje fyzická vrstva LoRa. Fyzická vrstva LoRa je uzavřená a proprietární a není tedy dostupná žádná oficiální specifikace pro použití v rámci open-source.

4.1.1 Fyzická vrstva LoRa

LoRa je standard pro bezdrátové sítě založen na modulaci Chirp Spread Spectrum (CSS). Jde o signál jehož frekvence v závislosti na čase klesá nebo vzrůstá. Signály chirpů mají konstantní amplitudu a zabírají celou šířku pásma. CSS používá celou šířku pásma pro vysílání signálů. Pokud se frekvence mění z vyšší hodnoty na nižší, tento proces se nazývá klesající chirp, pokud se frekvence mění z nižší hodnoty na vyšší, tento proces se nazývá vzrůstající chirp. CSS umožňuje vysílat signály na velké vzdálenosti.

LoRa reprezentuje symboly jako okamžité změny ve frekvenci chirpu. Tedy symboly jsou reprezentovány jako kmitočtově modulované chirpy. LoRa používá tři různé šířky pásma 125 kHz, 250 kHz a 500 kHz a činitele rozprostření od SF 7 až po SF 12 [41]. Na Obr. 4.1 lze vidět formát LoRa vrstvy. Fyzická vrstva LoRa zahrnuje 8 symbolů preambule, 2 synchronizační symboly, fyzický payload a optimálně i CRC. Preambule slouží k detekci LoRa signálů. Synchronizační zpráva se používá k detekci začátku LoRa payloadu. Payload obsahuje MAC příkazy a zprávu. CRC slouží ke kontrole blokových chyb.



Obr. 4.1: Formát LoRa vsrtvy.

4.1.2 Dekódování LoRa

Při kódování je cyklicky posunuta frekvence signálu. Pokud se tedy vynásobí přijatá frekvence frekvencí posunutou s inverzní hodnotou chirpu, vyjde signál kon-

stantní frekvence, který má specifickou frekvenci charakteristickou pro vysílaný signál. Pokud se použije rychlá fourierova transformace (FFT) na celý symbol, pak bod s nejvyšší energií bude představovat symbol, který byl vyslaný[9].

V rámci dokumentace [38, 9, 39, 40] byly o modulaci LoRa zjištěny informace, které pomohli popsat tuto fyzickou vrstvu. Důležité informace k tvorbě SDR lora byly popsány v článku [38]. Jde hlavně o informace v rámci dekódování. V dokumentu je popsáno, jak opravdu pracuje dekódování modulace LoRa oproti oficiální dokumentaci. Bylo zjištěno, že pro kování je použit Greyův kód, metoda whitening, interleaving společně s dopřednou korekcí chyb. Díky těmto zjištěním bylo možné zkonstruovat softwarově definované radio pro modulaci LoRa. K tomuto účelu byla vytvořena open-source implementace gr-lora [42].

4.1.3 gr-LoRa

Jedná se o open-source implementaci fyzické vrstvy LoRa, která je tvořena a upravována komunitou. Slouží k tomu, aby urychlila vývoj v rámci IoT a výzkum v rámci bezpečnosti. Gr-lora definuje bloky pro implementaci přijímačů a vysílačů kompatibilních s LoRa metodami, které jsou popsány v dokumentu [38]. Modulace a kódování a následná demodulace a dekódování jsou zpracovávány jednotlivými bloky v rámci GNU radia. Gr-lora podporuje příjem a vysílání při nastavení činitele rozprostření na hodnotu 8 a při kódování 4/8 na všech šířkách pásma.

4.2 Odposlech LoRaPHY

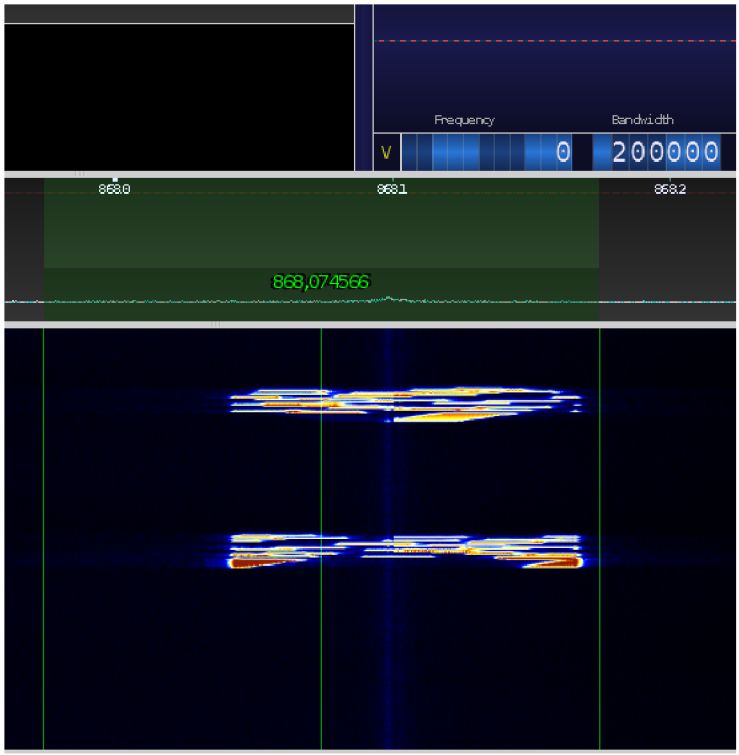
V práci bylo nejdříve nutné odposlechnout pouze fyzickou vrstvu LoRa, aby bylo ověřeno zda přijaté pakety pomocí GNU radia jsou správné a nejsou přijaty nějaké zavádějící data. V této části bude postupně popsáno jak bylo docíleno odposlechu fyzické vrstvy a co k tomu bylo třeba.

4.2.1 Zprovoznění odposlechu

Nejdříve ze všeho bylo testováno zachytávání signálů LoRa na frekvenci 868 MHz. K tomuto účelu bylo nakonec použito zařízení RTL-SDR DVB Tuner. Bylo také testováno zařízení LimeSDR mini, ale zařízení RTL-SDR bylo nakonec dostačující pro tento účel. Toto zařízení lze vidět na obr.4.2. Pro vysílání a testování bylo nejdříve zvoleno zařízení Things Uno s čipem RN2483,které bylo ukázáno v předchozí části, toto zařízení vysílalo na LoRaWAN definovaných frekvencích pro EU v pásmu 868 MHz.

Obr. 4.2: Zařízení RTL-SDR pro zachycení radiového signálu.

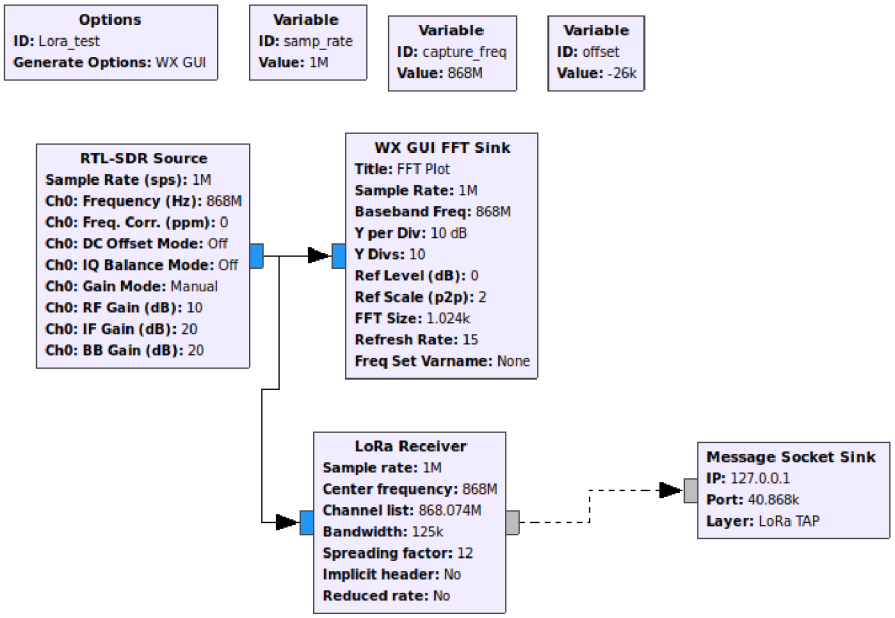
Nejdříve bylo tedy zprovozněno vysílání na koncovém zařízení. Následně byl použit software CubicSDR [43], ve kterém lze vidět vizuálně přenos na rádiových frekvencích. Pro zařízení RTL-SDR museli být nainstalovány ovladače, aby bylo možné pomocí něj odposlouchávat [44]. Po úspěšném zprovoznění zařízení byl proveden odposlech na frekvenci 868 MHz, výsledek je vidět na obr. 4.3. Na tomto obrázku jdou vidět chirpy, typické pro modulaci LoRa, jež jsou roztaženy přes celou šířku pásma 125 kHz. Díky tomuto lze říct, že zařízení RTL-SDR je schopno přijímat modulaci LoRa a lze tedy pokračovat v další části a to zapojení GNU Radia do tohoto procesu.



Obr. 4.3: LoRa signál zobrazen v softwaru CubicSDR.

4.2.2 Nastavení GNU Radia

Pro odchycení LoRaWAN signálu byl použit software GNU Radio. Tento software pracuje s jednotlivými bloky, které simulují fyzický hardware. Na obr. 4.4 lze vidět sestavení jednotlivých bloků pro příjem radiového signálu LoRa. Nejdříve byl použit blok RTL-SDR, který slouží jako zdroj signálu zařízení RTL-SDR, blok umožňuje převést signál ze zařízení do GNU Radia. Tento blok byl propojen se dvěma dalšími bloky WX GUI FFT Sink a LoRa Rceiver. První jmenovaný blok se chová jako spektrální analyzér, který aplikuje krátkodobou Fourrierovu transformaci. V tomto bloku byla nastavena vzorkovací frekvence na 1 MHz a frekvence základního pásma na 868 MHz. Jedná se o blok, který zprostředkovává grafické rozhraní. Druhý blok umožňuje příjem a dekódování LoRa signálu, tento blok pochází z knihovny gr-lora od autora z githubu [42]. Blok byl nastaven pro zachytávání a dekódování na frekvenci 868,100 MHz spreading factor byl zvolen na hodnotě 12. Posledním blokem je Message Socket Sink, tento blok vytvoří zprávu z příchozích dat a posílá je na adresu 127.0.0.1 (loopback) a díky tomu je poté možné zachytit zprávu pomocí Wiresharku.



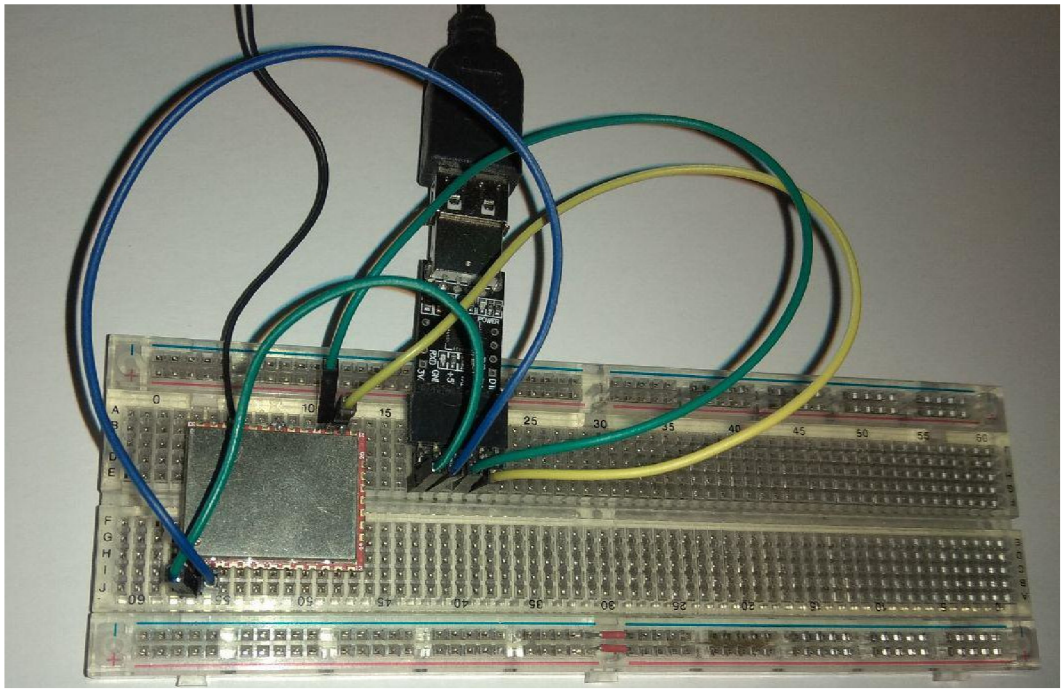
Obr. 4.4: Bloky v GNURadiu pro příjem radiového LoRa signálu.

4.2.3 Zachycení zprávy

Fyzická zpráva LoRa se skládá z Preambule, ta může mít velikost od 0 až po 65536 bajtů, dále pak z PHDR (hlavička), PHDR-CRC a CRC tyto informace zaručují

integritu zprávy a generuje si je sám vysílač. Je zde také samozřejmě samotný Payload.

Pro testování odposlechu byl použit samostatný čip RHF PS01509 [45], který pracuje s modulací LoRa a je připraven pro zařízení pracující s protokolem LoRaWAN. Zařízení lze vidět na obr.4.5. Tento čip byl zvolen hlavně díky kvalitní dokumentaci [45] a díky velkému množství příkazů AT. Komunikace se zařízením probíhá tedy pomocí AT příkazů.



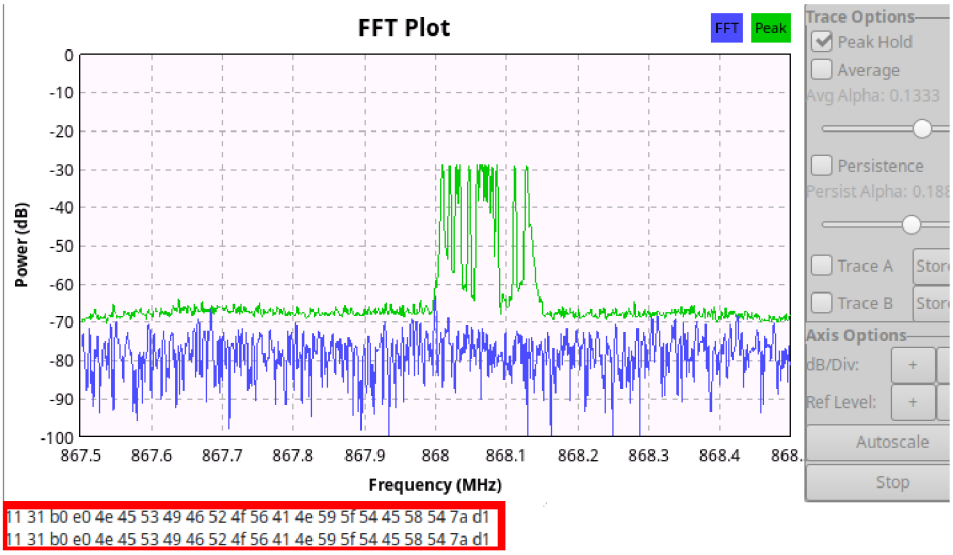
Obr. 4.5: Čip RHF použitý jako koncové zařízení.

Pro jednoduší odposlech a testování bylo zařízení nastaveno pomocí AT příkazů 4.1, nejdříve byly zakázány všechny kanály mimo 868,100 MHz a to kvůli snadnějšímu testování, číslo 0 značí že byl povolen pouze kanál na frekvenci 868,100 MHz ostatní kanály byly zakázané. Následně pro pohodlnější testování byl zrušen duty-cycle, aby se při vysílání každé zprávy nemuselo čekat. Vzhledem k tomu, že bylo nutné otestovat pouze fyzickou vrstvu bez jakéhokoli šifrování, byly zprávy posílány v testovacím režimu, což je proprietární režim pro zasílání hexadecimálních zpráv. Ze zařízení byla vysílána zpráva NESIFROVANY_TEXT. Následně jsou zde také popsány příkazy, které jsou používány při komunikaci se síťovým serverem, tím je myšleno při komunikaci v rámci vyšší vrstvy protokolu LoRaWAN. Nejdříve je zde příkaz pro vyvolání aktivační procedury OTAA. Následně jsou již vysílány zprávy, které jsou šifrovány podle protokolu LoRaWAN.

Výpis 4.1: Nastavení zařízení pomocí AT příkazů.

```
//Zákaz kanálů
AT+CH=NUM, 0
//zrušení duty-cycle
AT+LW=DC, OFF
//Zpráva v proprietráním režimu
AT+PMSG=NESIFROVANY_TEXT
//připojovací procedura
AT+JOIN
//zprávy LoRaWAN
AT+MSG="Ahoj"
```

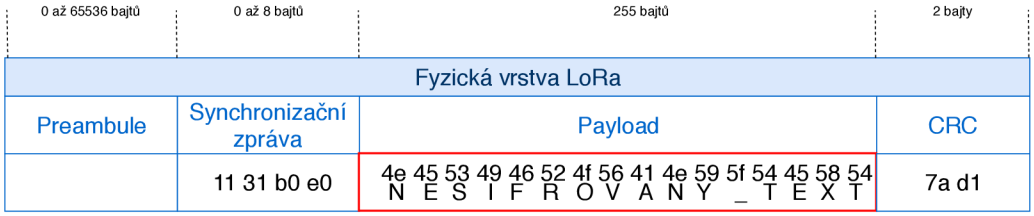
Po nastavení zařízení, z něj byla vysílána zpráva NESIFROVANY_TEXT. Tato zpráva byla zachycena v GNU Radiu pomocí zařízení RTL-SDR. Na obr.4.6 lze vidět zachycení vysílání LoRa pomocí bloku WX GUI FFT Sink, který je grafickým rozhraním pro zachycení signálu v GNU Radiu. Na obrázku je také zobrazen výpis z konzole GNU Radia, který obsahuje dekódovanou zprávu pomocí bloku LoRa Receiver.



Obr. 4.6: Ukázka zachycení v grafickém rozhraní GNU Radia a výpis z konzole GNU Radia.

Následně byla fyzická vrstva dekódována. Pokud je signál vysílán pouze pomocí modulace LoRa a není do vysílání začleněna vyšší MAC vrstva, tak jsou data posílána nešifrovaně, pouze v hexadecimálním tvaru. Na obr. 4.7 lze vidět popis paketu při vysílání pomocí modulace LoRa. Paket byl zachycen pomocí GNU Radia a pomocí bloku LoRa Receiver byl obsah dekódován na hexadecimální posloupnost. Na

obrázku lze vidět rozdělení podle bajtů a ukázku nezašifrovaného přenosu. Zachycena byla tato hexadecimální posloupnost: 11 31 b0 e0 4e 45 53 49 46 52 4f 56 41 4e 59 5f 54 45 58 54 7a d1.



Obr. 4.7: Rozdělení zachycené zprávy v rámci formátu LoRa vrstvy.

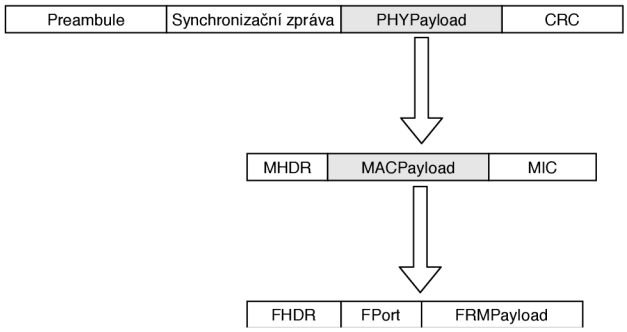
4.3 LoRaWAN MAC vrstva

Po odchycení fyzické vrstvy, bylo možné přejít k zachycení a dešifrování vyšší vrstvy MAC. V této části bude popsáno, jak byly zprávy zachyceny a postupně dešifrovány. Jaké hodnoty jsou nutné k dešifrování zprávy nesoucí aplikační payload. Je zde také detailně popsána aktivační procedura OTAA. Vysílání a zachytávání fyzického payloadu probíhalo stejně jako v předešlém případě s tím rozdílem, že nyní byl již payload šifrován. V této části je již využíván dříve nakonfigurovaný LoRa Server. Díky vlastnímu síťovému serveru a bráně bylo možné ověřovat správnost přenesených zpráv a podle toho dešifrovat jednotlivé zprávy. Kontrola odchycených dat v GNU Radiu probíhala v rámci porovnání s daty zachycenými v MQTT brokeru, tím byla ověřena správnost přenesených zpráv. Celé dešifrování je prováděno na protokolu LoRaWAN verze 1.0.2, na tomto protokolu pracuje jak koncové zařízení tak síťový server.

Nejdříve je zde ukázáno a popsáno z jakých částí se skládá zpráva nesoucí aplikační data. Na obr.4.8 lze vidět strukturu LoRaWAN zprávy. Nejdříve je zde samotná fyzická zpráva, ve které je obsažen PHYPayload hodnoty, okolo jsou data přidaná vysílačem. V poli PHYPayload jsou následně uloženy tyto informace: MHDR, MACPayload a MIC. MHDR je je hlavička protokolu LoRaWAN specifikuje typ zprávy a další informace o protokolu například i verzi. MIC je hodnota, která zajišťuje integritu zprávy (informace o ní byly již popsány dříve v práci).

MACPayload nebo také datový rámec v sobě nese informace jako FHDR, FPORT a FRMPayload. FHDR je hlavička rámce. FPort slouží k určení, zda jde o aplikační payload nebo o přenos MAC příkazů. Nakonec FRMPayload obsahuje aplikační data nebo MAC příkazy. FRMPayload je jediná část celého PHYPayloadu, která je šifrovaná. Šifrována je buď aplikačním relačním klíčem (AppSKey), a to pokud

nese aplikační data a nebo síťovým relačním klíčem (NwkSKey) pokud nese MAC příkazy.

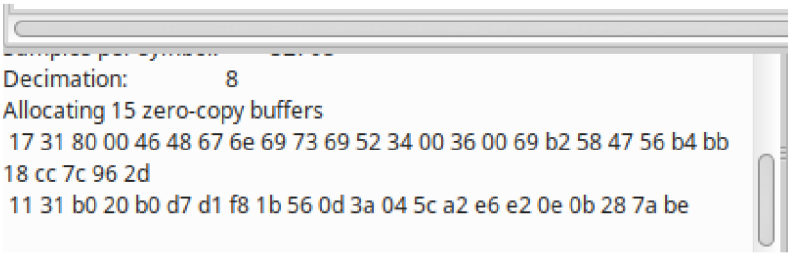


Obr. 4.8: Struktura zprávy LoRaWAN.

Z toho vyplývá, že pro dešifrování aplikačního payloadu je nutné znát aplikační relační klíč. Následně v textu je ukázáno, jak docílit získání všech potřebných hodnot k dešifrování aplikačního payloadu.

4.3.1 Zachycení a dešifrování

Aby bylo možné odvodit relační klíče, je nutné odchytit celou aktivační proceduru OTAA. Nejdříve byly v GNU Radiu zachyceny zprávy join-request a join-accept, to lze vidět na obr. 4.9, kde je ukázán výpis z konzole GNU Radia. Jedná se o zprávy fyzické vrstvy složeny ze synchronizační zprávy, PHYPayloadu a CRC, preambule je v GNU Radiu filtrována a tak zde její hodnota není. Hodnota preambule není ani v tomto případě důležitá.



Obr. 4.9: Zachycené zprávy join-request a join-accept v GNU Radiu.

AppKey

Pro dešifrování a generování relačních klíčů je nutné nejdříve získat AppKey. Tento klíč je uložen v paměti koncového zařízení a je také viditelný na síťovém serveru. AppKey lze získat z koncového zařízení, na kterém není tento klíč nijak šifrován.

V tomto případě je na zařízení RHF nastaven aplikační klíč AppKey s hodnotou 2b7e151628aed2a7abf7158809cf4f3c.

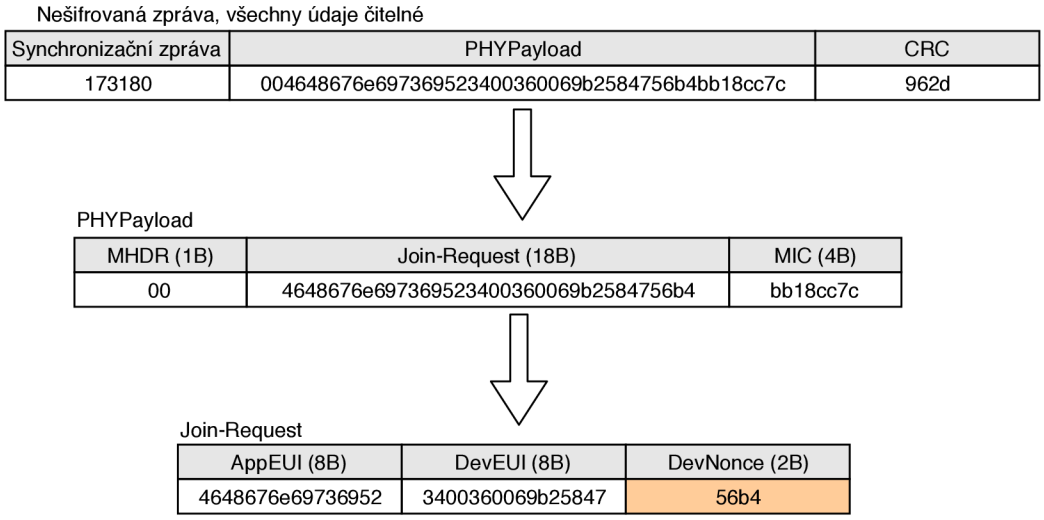
Zpráva typu join-request

Nejdříve zde bude rozebrána zpráva typu join-request. Jak již bylo popsáno dříve v práci, zpráva typu join-request není šifrována. Na obr.4.10 je zpráva rozdělena do jednotlivých částí a je ukázáno, co je ve zprávě přenášeno.

Jako první je ukázána zpráva jako nejzákladnější fyzický přenos, ze které je potřeba vyříznout PHYPayload, okolní bajty jsou přidány vysílačem zařízení.

Následně je ze zprávy PHYPayload odříznuta hlavička MHDR a součet pro zajištění integrit MIC. Poté zbude čistá nešifrovaná zpráva typu join-request.

Následně jsou ze zprávy již čitelné jednotlivé detaily sítě. Nejdříve první 8 bajtů je hodnota AppEUI, tato hodnota je ovšem ve zprávě reverzně otočena. Hodnota AppEUI je po reverzním otočení: 526973696e674846. Stejně tak hodnota DevEUI je reverzně otočena a její hodnota je tedy 4758b26900360034. Nakonec nejdůležitější hodnota DevNonce, tato hodnota bude následně použita na vygenerování relačních klíčů. Hodnota DevNonce je ve správném tvaru (není reverzně otočena).



Obr. 4.10: Struktura zprávy join-request.

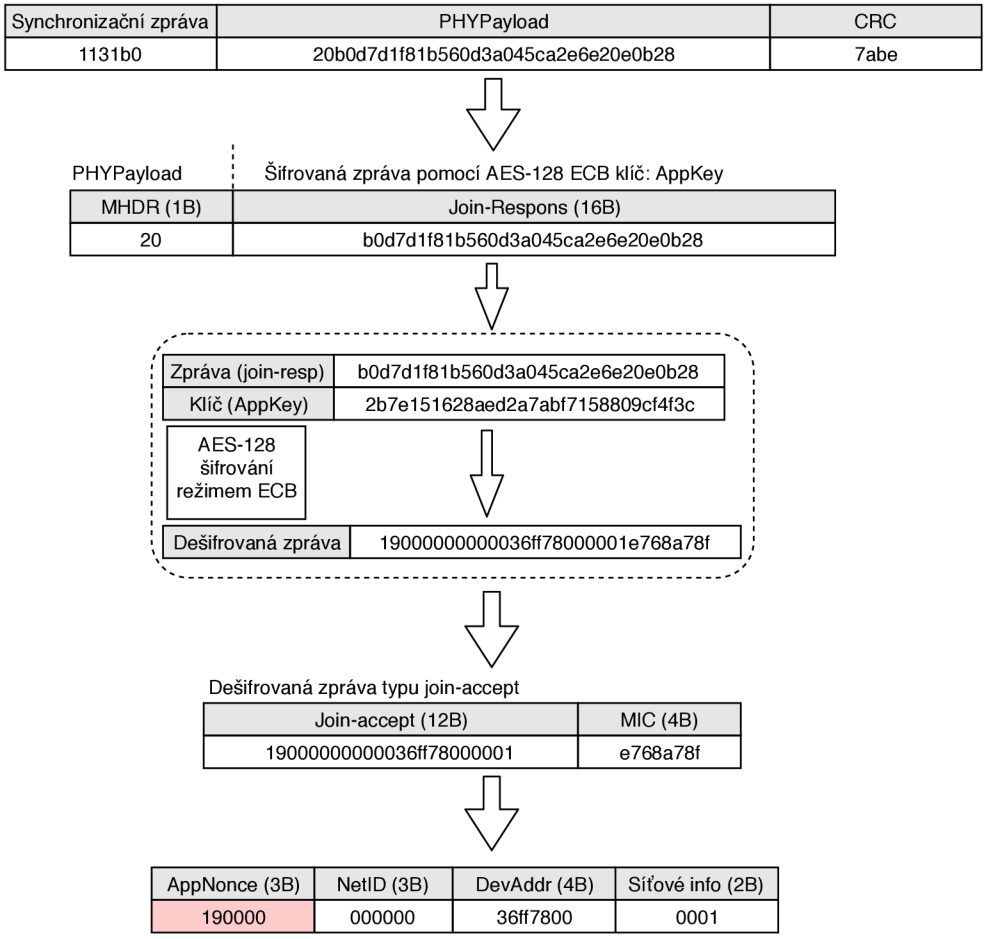
Zpráva typu join-accept

Druhou odchycenou zprávou z aktivační procedury je zpráva typu join-accept. Na obr.4.11 lze vidět strukturu a postup dešifrování. Z této zprávy je nutné získat hodnotu AppNonce nutnou pro generování relačních klíčů NwkSKey a AppSKey. Zpráva typu join-accept je ale šifrována metodou AES-128 v režimu ECB jako klíč, pro tento režim je použit aplikační klíč AppKey.

Nejdříve je opět ukázán fyzický přenos zprávy. Poté je PHYPayloadu odříznuta hlavička MHDR a zbude pouze zašifrovaná posloupnost join-response zprávy, která je šifrována aplikačním klíčem AppKey.

Zpráva join-response je bez hlavičky MHDR, protože je nutné, aby měla zpráva velikost 16 bajtů. Zpráva byla znovu zašifrována pomocí metody AES-128 metodou ECB. Procesem zašifrování této zprávy lze dostat dešifrovanou zprávu. Dešifrování probíhalo v programu, který byl napsaný v rámci této práce a bude prezentován v pozdější části práce. Tento program byl napsán v jazyce Python a bylo využito knihovny PyCrypto a balíčku Crypto.Cipher pro použití šifrování AES-128.

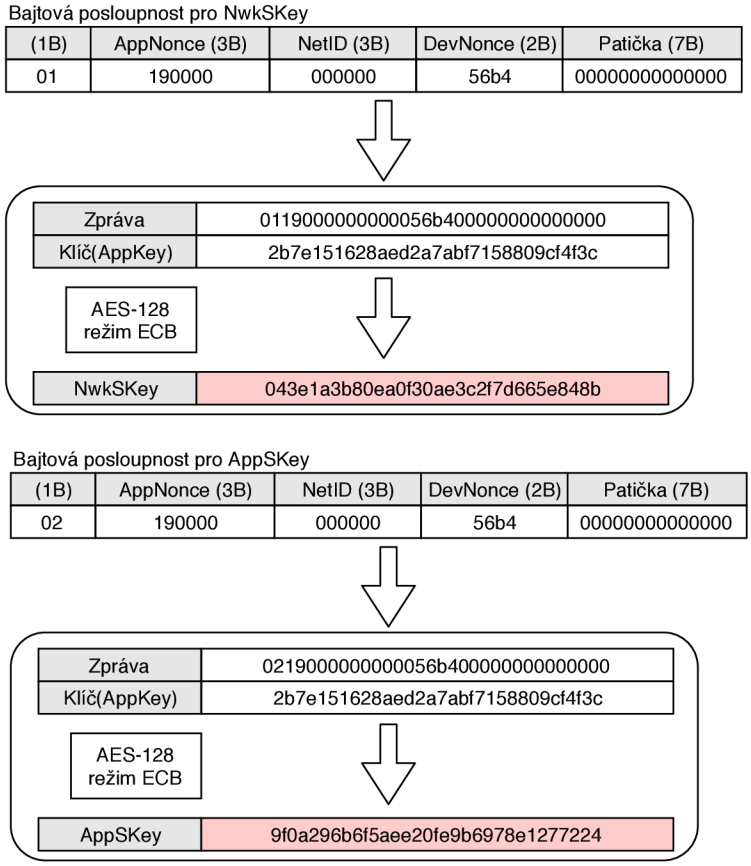
Poté co byla zpráva dešifrována, je ze zprávy odřezána hodnota MIC. Následně již lze vidět strukturu zprávy join-accept. V poslední části je důležitá především hodnota AppNonce. Ta je stejně jako hodnota DevNonce použita ke generování relačních klíčů. V další části bude popsáno, jak se tyto relační klíče generují.



Obr. 4.11: Struktura zprávy join-accept.

Generování relačních klíčů

Po získání hodnot DevNonce a AppNonce je možné vygenerovat relační klíče. Další informací, kterou je třeba znát, je NetID, ta je také již známá díky dešifrování zprávy typu join-accept. Poslední částí bajtové posloupnosti je patička, jedná se o 7 bajtů, které slouží k doplnění počtu nul do 16 bajtů. Nejdříve musí být poskládána hodnota, která bude zašifrována pomocí aplikačního klíče AppKey. Pro každý relační klíč je složena rozdílná hodnota. Rozdíl je ale pouze v prvním bajtu. Pro aplikační relační klíč je první bajt 02 a pro síťový relační klíč je první bajt hodnoty 01. Na obr.4.12 lze vidět posloupnost bajtů pro oba relační klíče a samotné relační klíče vygenerované na základě této posloupnosti šifrování AES-128 v režimu ECB, jako klíč byl opět použit aplikační klíč AppKey.



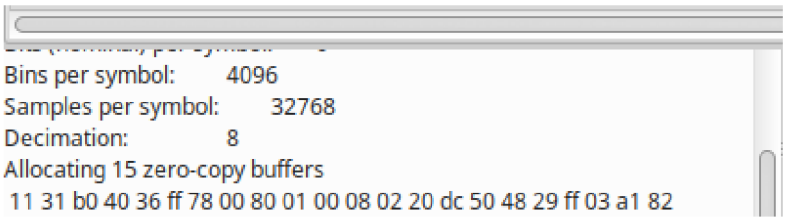
Obr. 4.12: Ukázka generování relačních klíčů.

Zachycení datové zprávy

Poté co jsou vygenerovány relační klíče, je možné dešifrovat i jakoukoli datovou zprávu pro LoRaWAN. Po registraci zařízení pomocí metody OTAA byla zaslána

zpráva s obsahem „Ahoj“. Následně je popsáno, jak bylo dosaženo dešifrování datové zprávy LoRaWAN.

Nejdříve byla ze zařízení RHF odeslána zpráva „Ahoj“. Tato zpráva byla zachycena v GNU radiu pomocí zařízení RTL-SDR. Na obr.4.13 lze vidět výpis z konzole GNU Radia, jde o zachycenou bajtovou posloupnost zprávy odeslané ze zařízení RHF.



Obr. 4.13: Zachycení datové zprávy LoRaWAN v GNU Radiu.

Po odchycení zprávy došlo k jejímu postupnému dekódování, to lze vidět na obr.4.14. Na začátku je znovu fyzický přenos paketu. Celý paket je nešifrován, až na část FRMPayload. Postupně jsou z paketu odřezávány hodnoty až zůstane pouze hodnota FRMPayload, která nese aplikační data a v tomto případě zprávu „Ahoj“. Pro všechny výpočty a dešifrování v rámci datové zprávy, byl pro tuto práci vytvořen program, který bude popsán v pozdější části práce.

Dešifrování hodnoty FRMPayload je více náročné. Nejdříve je nutné zjistit jakým klíčem je zpráva šifrována, jestli jde o aplikační relační klíč (AppSKey), ten se používá pro aplikační payload nebo zdali jde o síťový relační klíč (NwkSKey), který se používá pro MAC příkazy. Tuto skutečnost lze zjistit díky hodnotě FPort, pokud je hodnota rovna nule, je použit NwkSKey, jinak je vždy použit AppSKey. Jak je vidět z obrázku z hodnoty FPort, tak i v tomto případě je použitý klíč AppSKey.

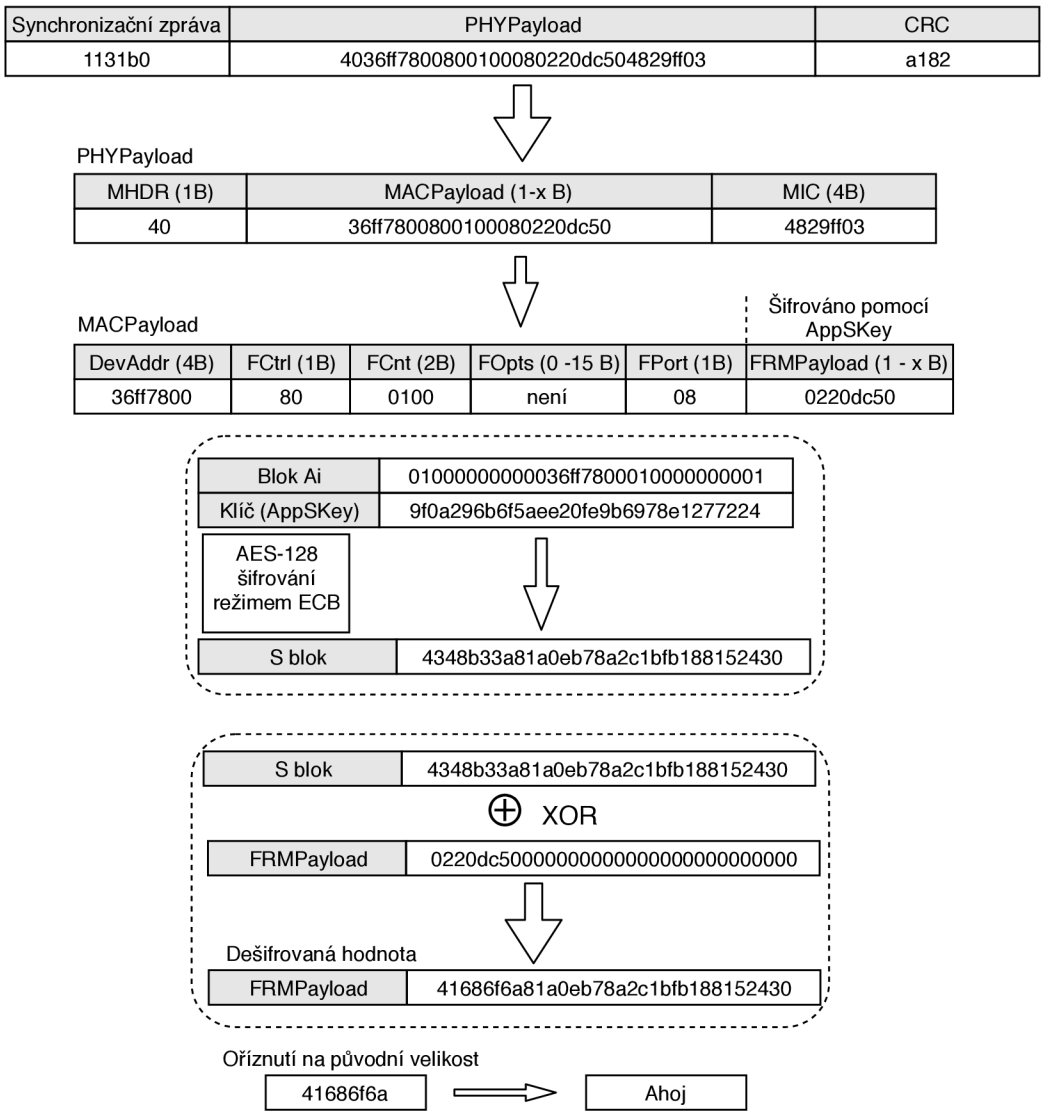
Následně je nutné vygenerovat sekvenci bloků A_i, kde i je počet bloků závislých na velikosti FRMPayloadu. FRMPayload může mít různé velikosti, ale jeden blok A_i může mít pouze 16 bajtů, takže v závislosti na velikosti FRMPayloadu je vytvořen počet Ai bloků. Matematicky A_i (i=1 až k) následující vzorec [3] popisuje tuto skutečnost, hodnota ceil znamená zaokrouhlení čísla vždy na vyšší hodnotu:

$$k = ceil(\frac{delka(FRMPayload)}{16})$$

V tomto případě po dosazení do vzorce vyjde, že stačí pouze jeden A_i blok. Sekvenci bajtů v tomto bloku lze také vidět na obr.4.14. A_i blok je sestaven z šesti nulových bajtů poté následuje adresa zařízení v reverzním tvaru, čtyři bajty hodnoty FCnt, jeden bajt nul a jeden bajt roven počtu všech bloků.

Následně je blok A_i šifrován v tomto případě s klíčem AppSKey pomocí AES-128 ECB. Pokud by bylo více bloků A_i, byly by po jednom šifrovány s klíčem a skládány

za sebe jakožto posloupnost S_i bloků. Poté probíhá operace XOR mezi blokem S a FRMPayloadem, který je doplněn nulami na velikost rovnou bloku S. Nakonec je výsledek oříznut na původní délku FRMPayloadu a tak v tomto případě zbude hexadecimální 4 bajtové číslo, které po převodu na ASCII hodnotu vypíše „Ahoj“.



Obr. 4.14: Dešifrování datové zprávy LoRaWAN.

Program pro dešifrování

Pro usnadnění a zautomatizování výpočtu a dešifrování zpráv LoRaWAN byl v rámci této práce vytvořen program v jazyce Python s grafickým rozhraním využívajícím framewrok kivy [46]. Program byl napsán podle výše popsaného postupu a provádí tedy stejné výpočty a dešifrování. Na obr.4.15 lze vidět grafické rozhraní programu

pro dešifrování zpráv LoRaWAN. Samotný program je k nalezení v příloze této práce. V programu jsou zadány stejné hodnoty jako při předchozím popisu.

Nejdříve je nutné do programu vložit aplikační klíč (AppKey), následně zachycené zprávy pomocí GNU radia join-request (pouze část Join-Request ne CRC a synchronizační část) a zprávu join-accept (zde je nutné odříznout i hlavičku MHDR tedy první bajt zprávy join-accept). Následně po stisku tlačítka „Ok“ jsou vygenerovány informace o zařízení a hlavně relační klíče NwkSKey a AppSKey.

Když je zachycena následně datová zpráva pomocí GNU Radia, vloží se tato datová zpráva (bez synchronizační části a CRC) do pole Datová zpráva. Z výpisu nad tímto polem se zkopírují dříve vygenerované relační klíče do polí NwkSKey a AppSKey. Po následném stisku tlačítka „Ok“ je zpráva dešifrována.

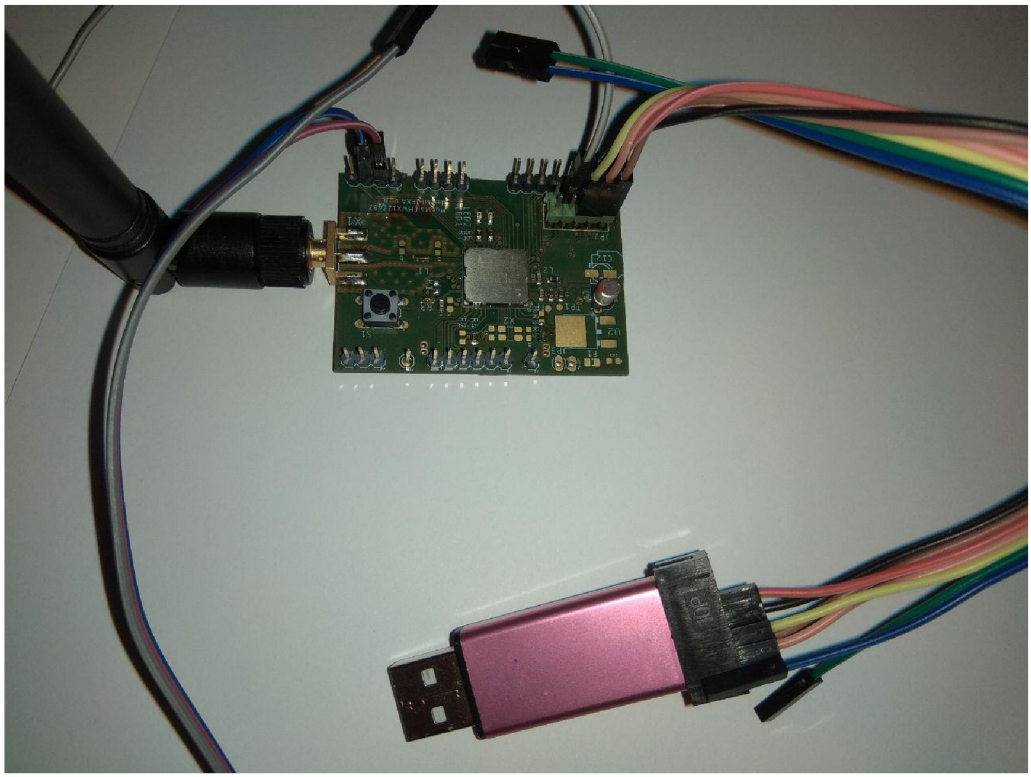
Obr. 4.15: Program pro dešifrování zpráv LoRaWAN.

5 Útok na protokol LoRaWAN

Po úspěšném dešifrování komunikace protokolu LoRaWAN verze 1.0.2, díky které byly získány informace pro prolomení zabezpečení tohoto protokolu byly testovány útoky na síť LoRaWAN. Nejdříve byl proveden útok přehráním zprávy (replay attack) a jako další byla přes falešnou relaci poslána falešná zpráva. Následně v textu bude popsáno jak těchto útoků bylo docíleno.

5.1 Přehrání zprávy

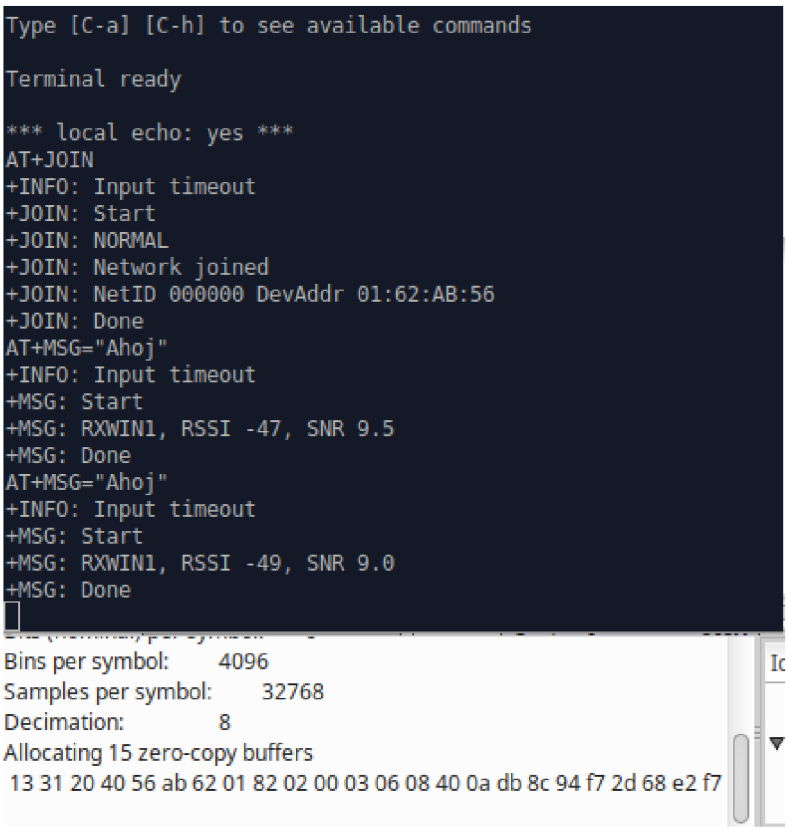
Nejdříve byl proveden útok přehráním zprávy. Vysílání zpráv probíhalo ze zařízení RHF, pro přehrání zprávy bylo využito zařízení, které je postaveno na modulu Murata [47]. Zařízení lze vidět na ob.5.1. Toto zařízení pracuje s rozšiřujícím balíčkem The I-CUBE-LRWAN [48] pro software LoRaWAN. Toto rozšíření bylo zprovozněno na modulu Murata a tím bylo dosaženo přístupu k ovládání vysílání zařízení a donucení vysílání zařízení vlastních zpráv po úpravě kódu. Tento balík rozšíření je dostupný zdarma na stránkách firmy STMicroelectronics [48].



Obr. 5.1: Zařízení pro vysílání zachycené zprávy.

5.1.1 Postup útoku

Nejdříve bylo zařízení RHF pomocí aktivační procedury připojeno na LoRaServer. Následně byla ze zařízení odeslána zpráva „Ahoj“. Tato zpráva byla zachycena pomocí GNU Radia na obr.5.2 lze vidět výpis z konzole GNU Radia, kde je tato zpráva zachycena.



Obr. 5.2: Odeslání zprávy „Ahoj“ a následné zachycení GNU Radiem.

Jsou v podstatě dvě možnosti jak vést útok opakováním. Pokud je na serveru zrušen čítač příchozích zpráv (FCnt) tak není třeba zarušit bránu a může být tento útok využit v podstatě jako DoS útok. Stejná zpráva může být vysílána z koncového zařízení stále dokola a tím vyřadit z provozu původní zařízení. Pokud je čítač zapnut, musí být zarušena komunikace s bránou a může být bez úpravy zprávy odeslána tato zpráva pouze jednou. Zde je ukázán způsob, když není čítač v provozu, ale jde o stejný postup.

Aby bylo možné útok provést bylo nutné z odchycené zprávy odříznout synchronizační zprávu, to jsou první tři bajty, a CRC to jsou poslední dva bajty. Následně byl upraven kód softwaru který je dostupný na oficiálních stránkách The I-CUBE-LRWAN [48]. V podstatě byl upraven pouze soubor main (celý kód souboru main

je v příloze, zbytek kódu je dostupný na stránkách výrobce) a to tak, aby zařízení vysílalo přesně napsanou bajtovou posloupnost bez žádných vlastních výpočtu. Ukázka úpravy kódu v souboru main, která přinutila zařízení vysílat požadovanou posloupnost bajtů viz výpis 5.1.

Výpis 5.1: Nastavení zařízení pomocí AT příkazů.

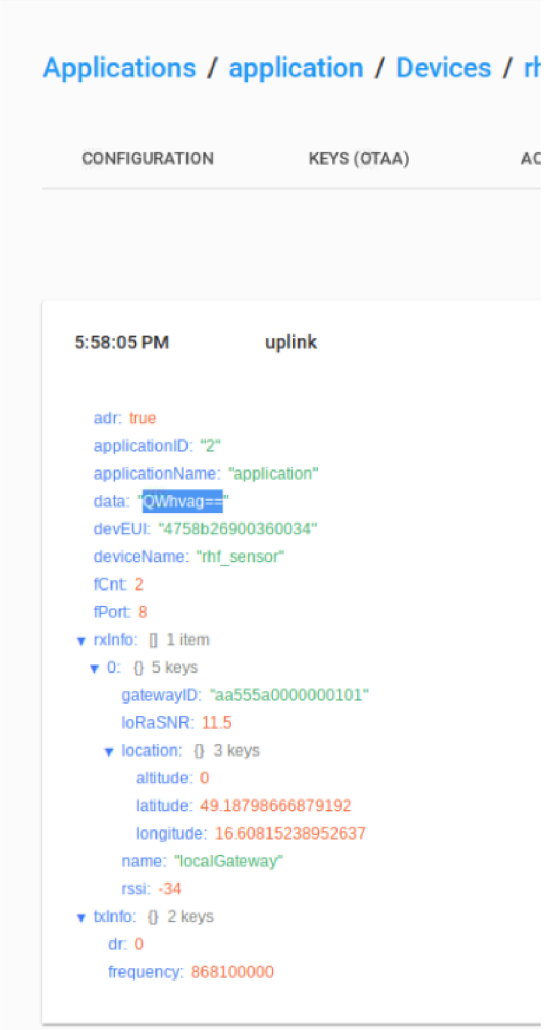
```
Radio.SetTxConfig( MODEM_LORA, 14, 0, 0,
12, 1,
8, false,
true, 0, 0, false, 3000 );
Radio.SetChannel( 868100000 );

uint8_t buf[100];

buf[0] = 0x40;
buf[1] = 0x56;
buf[2] = 0xab;
buf[3] = 0x62;
buf[4] = 0x01;
buf[5] = 0x82;
buf[6] = 0x02;
buf[7] = 0x00;
buf[8] = 0x03;
buf[9] = 0x06;
buf[10] =0x08;
buf[11] =0x40;
buf[12] =0x0a;
buf[13] =0xdb;
buf[14] =0x8c;
buf[15] =0x94;
buf[16] =0xf7;
buf[17] =0x2d;
buf[18] =0x68;


Radio.Send( buf, 19 );
```

Poté byla bajtová posloupnost odeslána ze zařízení Murata. Posloupnost byla úspěšně přijata server, to lze vidět na obr.5.3, kde je úspěšně přijata zpráva „Ahoj“ ve tvaru Base64.



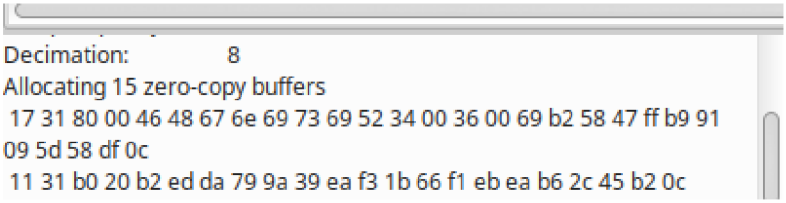
Obr. 5.3: Přijatá přehraná zpráva na LoRa serveru.

5.2 Úprava zprávy

Následně je popsán proces, při kterém byla zachycena zpráva a následně byl změněn obsah zprávy a odeslán na server, tedy zaslání falešné zprávy pomocí falešné relace. Postup byl obdobný jako při předchozím procesu, ale s tím rozdílem že bylo nutné dešifrovat zprávu následně změnit payload a poté znovu zašifrovat a nad touto znovu zašifrovanou zprávou vypočítat MIC a až poté zprávu odeslat.

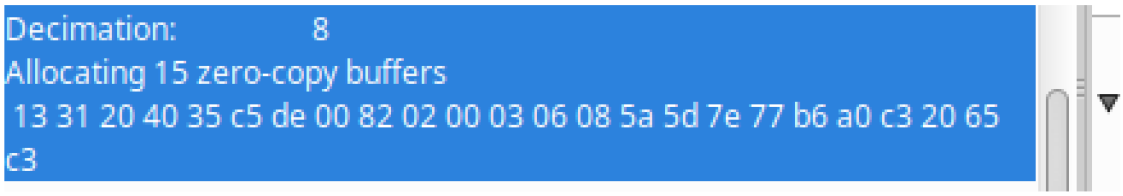
5.2.1 Postup útoku

U tohoto útoku je už nutné zachytit i aktivační proceduru, vzhledem k potřebě odvození relačních klíčů, aby bylo možné zprávu dešifrovat a následně opět zašifrovat. Na obr.5.4 lze vidět tedy zachycení aktivační procedury OTAA.



Obr. 5.4: Zachycení aktivace OTAA.

Následně byla opět vysílána zpráva s obsahem „Ahoj“. Tato zpráva byla zachycena v GNU Radiu to lze vidět na obr. 5.5, a poté byl upraven FRMPayload této zprávy.



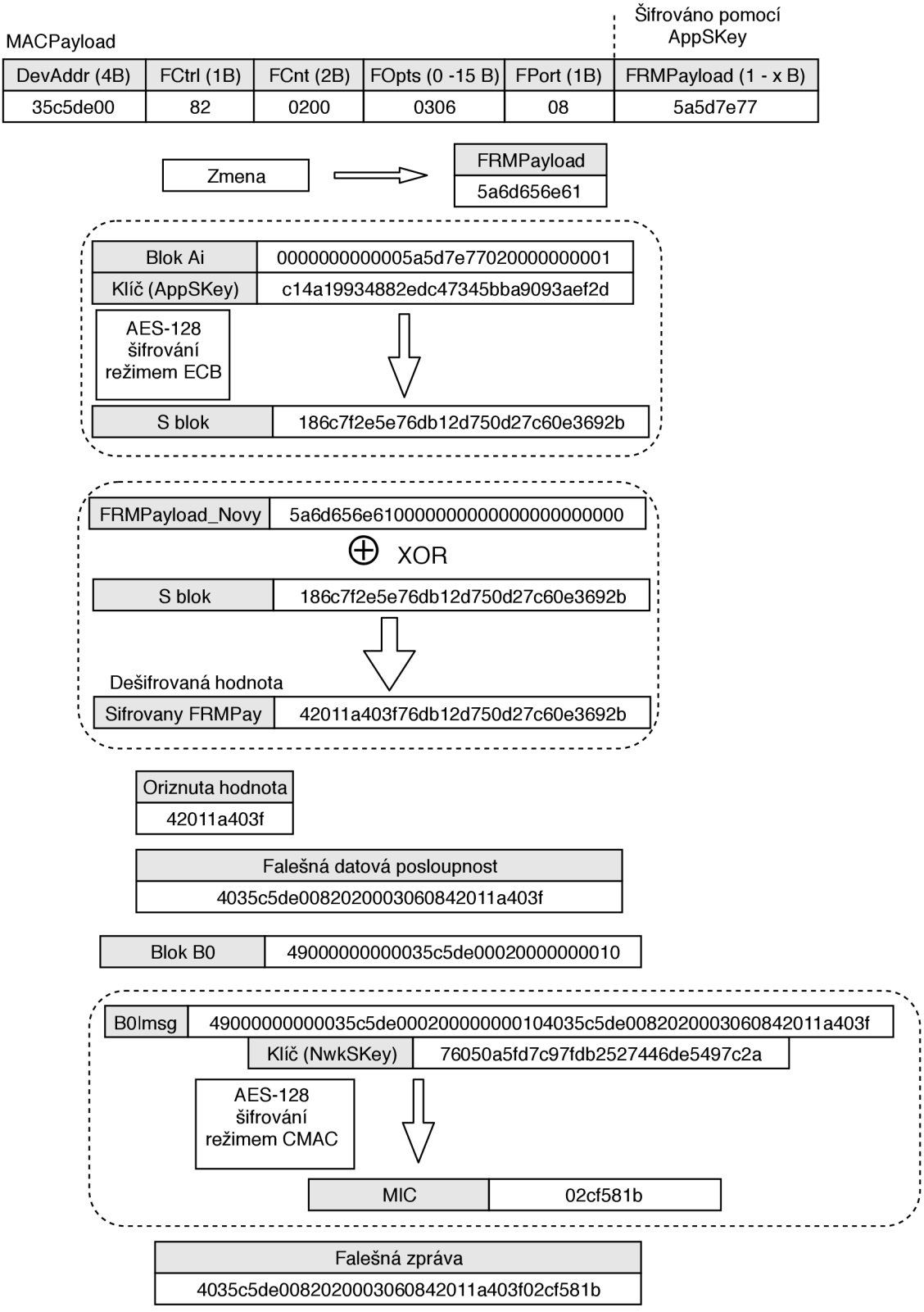
Obr. 5.5: Zachycená zpráva v GNU Radiu.

Změna obsahu zprávy

Nejdříve bylo nutné zachycenou zprávu rozdělit na jednotlivé části, aby bylo možné definovat kde začíná a končí FRMPayload a ten pak upravit. Celý postup je opět na obr.5.6. Následně byl zvolen vlastní payload s obsahem „Zmena“. Poté byl vytvořen nový A_i blok, ten byl následně šifrován pomocí režimu ECB společně s aplikačním relačním klíčem, který byl získán pomocí dekodéru jak bylo popsáno v předchozích částech práce a byl vytvořen blok S.

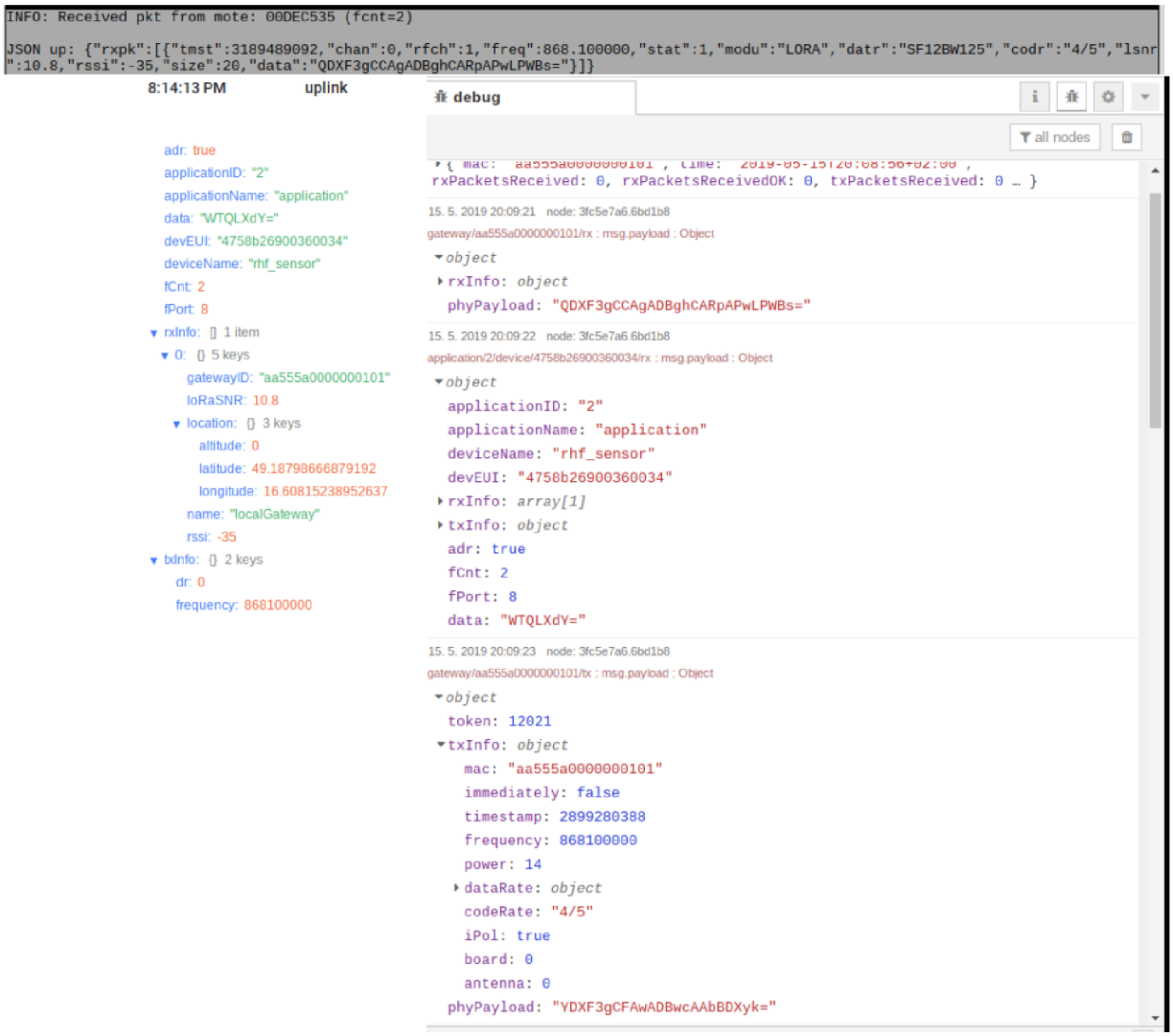
Následně proběhlo zašifrování nové zprávy. Nejdříve bylo nutné provést opačný postup jako u dešifrování. Byla vzata nová bajtová posloupnost FRMPayloadu a za tuto posloupnost byli přidány nuly aby celková hodnota byla 16 bajtů. Poté proběhla operace XOR nového payloadu společně s blokem S. Z výsledku byla odřezána přebývající posloupnost a bylo ponecháno jen tolik bajtů, kolik měl payload před přidáním nul. Tak byl vytvořen nový šifrovaný falešný FRMPayload, ten byl následně přidán k původní posloupnosti kde nahradil původní FRMPayload.

Aby server přijal zprávu, musel být vypočten nad touto zprávou MIC. Výpočet MIC je taktéž vidět na obr.5.6. Nejdříve byl podle specifikace [3] vytvořen blok B_0. Následně byla k tomuto bloku připojena původní zpráva. Tento celek byl poté šifrován AES-128 šifrou v režimu CMAC a jako klíš byl použit relační síťový klíč NwkSKey. T9m vznikne bajtová posloupnost ze které se vyberou pouze první 4 bajty. Následně byla hodnota 4 bajtová hodnota MIC připojena ke zprávě.



Obr. 5.6: Tvorba falešné zprávy.

Poté co byla falešná zpráva vytvořena, tak se už postupovalo obdobně jako u předchozí části (útoku přehráním zprávy). Opět byl upraven kód zařízení Murata jako v předchozím případě byla odeslána bajtová posloupnost tentokrát ale falešné zprávy. Zpráva byla úspěšně přijata serverem. Na obr.5.7, lze vidět jak byla zpráva přijata nahoře bránou, vlevo síťový server a vpravo mqtt směr k aplikaci. Správa byla úspěšně podvržena a a přijata serverem. Jediný problém nastal v tom, že pokud je dešifrována hodnota base64, nevychází zpráva, která byla odeslána „Zmena“, ale nesmyslná hodnota. To je zapříčiněno chybou, která vznikla při výpočtu bloku A_i, kde lze vidět, že jako první bajt je použita hodnota 00 namísto hodnoty správné a to 01. Tímto došlo ke změně hodnoty payloadu. Tato změna ovšem nemá vliv na to, že lze podvrhnout přenášenou zprávu a posílat tak na server FRMPayload jakýkoli. Útok lze tedy považovat za úspěšný.



Obr. 5.7: Ukázka falešné zprávy na bráně, serveru a mqtt.

6 Závěr

V práci byla provedena analýza LPWAN technologie LoRaWAN. Byla popsána celá síť LoRaWAN a její prvky a jejich vzájemná komunikace. Analýza byla zaměřena především na bezpečnost v rámci LoRaWAN a tedy na možné útoky a zabezpečení proti těmto útokům. Byly zde také popsány scénáře v rámci zpětné kompatibility LoRaWAN 1.1 a LoRaWAN 1.0.x a také bezpečnostní mezery které vznikají kvůli této kompatibilitě. Byly také popsány rozdíly v rámci nové verze protokolu LoRaWAN.

Dále se práce zabývala zprovoznění komunikace LoRaWAN a konstrukcí jednotlivých součástí této sítě. Na základě vybraného hardwaru byla zkonstruována a zprovozněna vlastní brána a také vlastní koncové zařízení. Poté byl zprovozněn vlastní server na open-source řešení LoRa Server, který byl pro testování bezpečnosti vhodnější, než jiná řešení a to především možností volby verze LoRaWAN protokolu na kterém bude pracovat. Další výhodou je také možnost sledování celé komunikace a díky tomu si ověřit správnost hodnot.

Následně v práci bylo provedeno dešifrování protokolu LoRaWAN. Nejdříve se práce zaměřila na odchycení informací na fyzické vrstvě LoRa, kde byly úspěšně přeneseny a odchyceny data a poté dekódovány. Díky tomu bylo možné označit GNU Radio a zařízení určené k odposlechu, jako správně pracující.

Následně byly odchyceny šifrované zprávy vyšší vrstvou LoRaWAN. Tyto zprávy byly postupně dešifrovány. V práci bylo ukázáno jaké klíče jsou nutné k dešifrování a jakým způsobem jsou zprávy šifrovány. V práci je popsáno šifrování od aktivačního proce zařízení až po dešifrování payloadu datové zprávy pomocí relačních klíčů.

Nakonec je v práci popsán útok přehráním zprávy. Tento útok může mít za následek znemožnění činnosti pravého zařízení, na úkor zařízení útočného. V této části je popsáno jak tento útok provést. Následně je zde v návaznosti na tento útok popsáno jak změnit obsah zprávy a vysílat tak falešnou zprávu. Bylo zde ukázáno jak bylo docíleno změny zprávy a jak lze posílat falešné zprávy na server. Byly zde ukázány především mezery protokolu verze 1.0.2, ale vzhledem k tomu že spoustu koncových zařízení stále pracuje na této verzi, je toto téma stále aktuální.

Literatura

[1] Mekki, K.; Bajic, E.; Chaxel, F.; aj.: A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express,*,2018

[2] Raza, U.; Kulkarni, P.; Sooriyabandara, M.: Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials,*,ročník 19, č. 2, 2017: s. 855–873.

[3] *LoRaWAN™ Specification v1.0.2* [online]. San Ramon, CA 94583, USA: LoRa Alliance, 2016 [cit. 2019-05-08]. Dostupné z URL: <https://lora-alliance.org/sites/default/files/2018-05/lorawan1_0_2-20161012_1398_1.pdf/>.

[4] *LoRaWAN™ Specification v1.1* [online]. Beaverton, OR 97003, USA: LoRa Alliance, 2017 [cit. 2019-03-14]. Dostupné z URL: <https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf/>.

[5] Ray, P. P.: A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences,*,ročník 30, č. 3, 2018: s. 291–319.

[6] Raza, U.; Kulkarni, P.; Sooriyabandara, M.: Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials,*,ročník 30, č. 3, 2018: s. 291– 319.

[7] *Ingenu* [online]. San Diego , United States: Ingenu, 2008 [cit. 2019-05-14]. Dostupné z URL: <https://www.ingenu.com//>.

[8] *Weightless* [online]. Cambridge, UK: Weightless SIG, 2015 [cit. 2019-05-14]. Dostupné z URL: <www.weightless.org/>.

[9] LoRa Decoding. *All About LoRa and LoRaWAN* [online]. [cit. 2018-12-14]. Dostupné z URL: <http://www.sghoslya.com/p/lora_9.html/>.

[10] Sinha, R. S.; Wei, Y.; Hwang, S.-H.: A survey on LPWA technology: LoRa and NB-IoT. *Ict Express,*,3, č. 1, 2017: s. 14–21.

[11] Committee, L. A. T.; aj.: LoRaWAN 1.1 Specification. *LoRa Alliance, Standard,*,ročník 1, 2017: str. 1.

[12] Dönmez, T. C.; Nigussie, E.: Security of LoRaWAN v1. 1 in Backward Compatibility Scenarios. *Procedia computer science,*,ročník 134, 2018: s. 51–58.

[13] Hinden, R.; Deering, S.: IP version 6 addressing architecture. Technická zpráva, 2006.

[14] *What is new in LoRaWAN 1.1?* [online]. Webinar: he Things Industries, 2017 [cit. 2019-05-11]. Dostupné z URL: <https://www.thethingsnetwork.org/article/what-is-new-in-lorawan-11-live-webinar-by-johan-stokking/>.

[15] Pub, N. F.: 197: Advanced encryption standard (AES). *Federal information processing standards publication,*,ročník 197, č. 441, 2001: str. 0311.

[16] Khutsoane, O.; Isong, B.; Abu-Mahfouz, A. M.: IoT devices and applications based on LoRa/LoRaWAN. *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society,*,IEEE, 2017, s. 6107–6112.

[17] Kim, J.; Song, J.: A secure device-to-device link establishment scheme for LoRaWAN. *IEEE Sensors Journal,*,ročník 18, č. 5, 2018: s. 2153–2160.

[18] IC880A LoRaWAN Gateway Backplane v2.0. *Coredump Shop Electronics* [online]. Coredump Rapperswil, 2015 [cit. 2019-05-15]. Dostupné z URL: <https://shop.coredump.ch/product/ic880a-lorawan-gateway-backplane//>.

[19] RN2483. *Microchip* [online]. Arizona, USA: Microchip Technology, 2015 [cit. 2019-05-15]. Dostupné z URL: <https://www.microchip.com/wwwproducts/en/RN2483/>.

[20] RTL-SDR USB stick with R820T2. *Passion Radio* [online]. [cit. 2018-12-12]. Dostupné z URL: <https://www.passion-radio.com/sdr-receivers/rtl-sdr-r820t2-248.html/>.

[21] WBX. *Ettus Research* [online]. [cit. 2018-12-13]. Dostupné z URL: <https://www.ettus.com/product/details/WBX/>.

[22] LimeSDR-Mini. *MYRIAD RF* [online]. [cit. 2018-12-12]. Dostupné z URL: <https://wiki.myriadrf.org/LimeSDR-Mini/>.

[23] YARD Stick One. *Great Scott Gadgets* [online]. [cit. 2018-12-12]. Dostupné z URL: <https://greatscottgadgets.com/yardstickone//>.

[24] HackRF One. *Great Scott Gadgets* [online]. [cit. 2018-12-12]. Dostupné z URL: <https://greatscottgadgets.com/hackrf//>.

[25] Machado-Fernández, J. R.: Software Defined Radio: Basic Principles and Applications. *Facultad de Ingeniería,*,ročník 24, č. 38, 2015: s. 79–96.

[26] *LORIOT* [online]. [cit. 2018-12-12]. Dostupné z URL: <https://www.loriot.io//>.

[27] Ic880a-gateway. *GitHub* [online]. GitHub [cit. 2018-12-12]. Dostupné z URL: <https://github.com/ttn-zh/ic880a-gateway/>.

[28] IC880A-SPI - LoRaWAN Concentrator 868 MHz. *IMST* [online]. GitHub [cit. 2018-12-12]. Dostupné z URL: <ttps://shop.imst.de/wireless-modules/lora-products/8/ic880a-spi-lorawan-concentrator-868-mhz/>.

[29] Raspberry Pi 3 B. *RPIshop* [online]. GitHub [cit. 2018-12-12]. Dostupné z URL: <http://rpishop.cz/raspberry-pi-3b/283-raspberry-pi-3-model-b-64-bit.html>.

[30] IC880A LoRaWAN Gateway Backplane v2.0. *Coredump Shop Electronics* [online].Coredump Rapperswil, 2015 [cit. 2019-05-15]. Dostupné z URL: <https://shop.coredump.ch/product/ic880a-lorawan-gateway-backplane/>.

[31] Napájecí zdroj. *RPishop* [online].Coredump Rapperswil, 2015 [cit. 2018-12-12]. Dostupné z URL: <http://rpishop.cz/zdroje/192-25a-oficialni-microusb-napajeci-zdroj-cerny.html/>.

[32] U.fl to SMA - Pigtail cable for iC880A-SPI. *IMST* [online].Coredump Rapperswil, 2015 [cit. 2018-12-12]. Dostupné z URL: <https://shop.imst.de/wireless-modules/accessories/20/u.fl-to-sma-pigtail-cable-for-ic880a-spi/>.

[33] *LoRaServer* [online].GitHub, 2016, 2015 [cit. 2018-12-12]. Dostupné z URL: <https://www.loraserver.io//>.

[34] *Lora network packet forwarder project* [online].GitHub: Semtech, 2017 [cit. 2018-12-12]. Dostupné z URL: <https://github.com/Lora-net/packet_forwarder/>.

[35] *MQTT version 3.1.1* [online].OASIS Open: OASIS, 2014, [cit. 2018-12-12]. Dostupné z URL:

<http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html/>.

[36] *GRPC* [online].[cit. 2019-05-12]. Dostupné z URL: <https://grpc.io//>.

[37] *Node-RED* IBM: IBM Emerging Technology, 2016 [cit. 2019-05-12]. Dostupné z URL: <https://nodered.org//>.

[38] Knight, M.; Seeber, B.: Decoding LoRa: Realizing a modern LPWAN with SDR. In *roceedings of the GNU Radio Conference,*,ročník 1, 2016.

[39] DecodingLora. *RevSpace* [cit. 2018-12-14]. Dostupné z URL: <https://revspace.nl/DecodingLora/>.

[40] Mroue, H.; Nasser, A.; Parrein, B.; aj.: Analytical and Simulation study for LoRa Modulation. In *2018 25th International Conference on Telecommunications (ICT),*IEEE, 2018, s. 655–659.

[41] Sornin, N.; Luis, M.; Eirich, T.; aj.: Lorawan specification. *LoRa alliance,*2015.

[42] Gr-lora. *GitHub* rpp0, 2016 [cit. 2019-03-29]. Dostupné z URL: <https://github.com/rpp0/gr-lora>.

[43] Cubic-SDR *GitHub* GitHub, 2015 [cit. 2019-03-29]. Dostupné z URL: <https://cubicsdr.com/>.

[44] *QUICK START GUIDE* RTL-SDR, 2018 [cit. 2019-05-12]. Dostupné z URL: <https://www.rtl-sdr.com/rtl-sdr-quick-start-guide//>.

[45] *RHF-PS01509* RisingHF, 2017 [cit. 2019-05-12]. Dostupné z URL: <https://wiki.ai-thinker.com/_media/rhf-ps01509_lorawan_class_ac_at_command_specification_-_v4.4.pdf/>.

[46] *Kivy (framework)* MIT: Kivy organization, 2011 [cit. 2019-05-14] Dostupné z URL: <https://kivy.org//>.

[47] Murata LoRa (LoRaWAN) Module. *Murata* Murata Manufacturing [cit. 2019-05-15]. Dostupné z URL: <https://www.murata.com/en-eu/products/lpwa/lora/>.

[48] I-CUBE-LRWAN. *ST* STMicroelectronics [cit. 2019-05-15]. Dostupné z URL: <https://www.st.com/en/embedded-software/i-cube-lrwan.html>.

Seznam symbolů, veličin a zkratek

ABP	Aktivace osobou – Activation By Personalization
ACK	Potvrzení – Acknowledgement
ADR	Adaptive Data Rate
AES	Advanced Encryption Standard
AppKey	Application Key
B	Bajt - Byte
b	bite
CCM	Counter with CBC-MAC
CRC	Cyklický redundantní součet – cyclic redundancy check
CSS	Chirp Spread Spectrum
DNS	Domain Name System
DSSS	Direct Sequence Spread Spectrum
ECB	Electronic Codebook
FnwkSIntKey	Forwarding Network session integrity key
GHz	Giga Hertz
IoT	Interner of Things
kilobit	kilo bit
kHz	kilo Hertz
LPWAN	Low Power Wide Area Networks
MHz	Mega Hertz
MIC	Message Integrity Code
MIC	Message Integrity Check
NwkKey	Network Key
NwkKey	Network Key
NwkSEncKey	Network session encryption key
OTAA	Over The Air Activation
SnwkSIntKey	Serving Network session integrity key
TDMA	Time Division Multiple Access

Seznam příloh

A Obsah přiloženého CD

- `/` ..... kořenový adresář přiloženého CD
  - `LoraDecoder` ..... program pro zjednodušení dešifrování LoRaWAN
    - `__pycache__`
    - `venv`
    - `loradecoder.py`
    - `main.py` ..... spouštěcí soubor programu LoraDecoder
  - `Diplomova prace.pdf` ..... Diplomová práce v elektronické podobě