Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Zabezpečení komunikace prvků IoT v prostředí smart home

Diplomová práce

Autor: Martin Šustr
Studijní obor: ai2-p

Vedoucí práce: Mgr. Josef Horálek, Ph.D.

Hradec Králové Duben 2021

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 16.4.2021 Martin Šustr

Poděkování:

Děkuji vedoucímu diplomové práce Mgr. Josefu Horálkovi, Ph.D. za metodické vedení práce, cenné rady a odborný dohled.

Anotace

Diplomová práce se zabývá bezpečností komunikace IoT prvků v prostředí smart home a jeho vylepšení. Práce zkoumá aktuální pohled na zabezpečení v prostředí smart home. Následně se podrobně zaměřuje na popis technologií Bluetooth a ZigBee s důrazem na jejich zabezpečení a zranitelnosti. Získané informace posloužily jako podklad pro navržení bezpečné komunikace s důrazem na nízkou cenu a dostupnost řešení. Bylo navrženo zabezpečení s využitím RADIUS protokolu, který byl doplněn dalšími bezpečnostními mechanismy. Návrh zabezpečení a bezpečnostních mechanismů byl následně prakticky implementován za pomoci malých jednodeskových počítačů Raspberry Pi, platformy Arduino a řídící aplikace napsané v jazyce Java. Vytvořená smart home síť sloužila pro ověření funkcionality navrženého zabezpečení. Výsledkem práce je úspěšně otestovaný koncept zabezpečení určený pro smart home sítě nehledě na zvolenou technologii bezdrátového přenosu.

Annotation

Title: Security of IoT communication for SMART home

The diploma thesis deals with the security of communication of IoT elements in the smart home environment and its improvement. The thesis examines the current view of security in the smart home environment. Subsequently, it focuses in detail on the description of Bluetooth and ZigBee technologies with emphasis on their security and vulnerabilities. The information obtained served as a basis for designing secure communication with an emphasis on low cost and availability of the solution. Security was designed using the RADIUS protocol, which was supplemented by other security mechanisms. The design of security and safety mechanisms was then practically implemented using small single-board computers Raspberry Pi, the Arduino platform and application written in Java. The created smart home network was used to verify the functionality of the proposed security. The result of the work is a successfully tested security concept designed for smart home networks, regardless of the selected wireless transmission technology.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Dnes již není žádnou novinkou, že téměř každý má doma osobní počítač, či smartphone. Většina těchto zařízení je připojených k celosvětové internetové síti, která umožňuje vzájemnou interakci i na vzdálenost několik tisíc kilometrů téměř okamžitě. Počítače a mobilní telefony jsou poměrně komplexní zařízení, ale stále častěji se využívají i jiné technologické vychytávky, které mají za úkol zjednodušit a zpříjemnit lidem každodenní život.

Když člověk dorazí domů nemusí použít fyzický klíč, ale může odemknout prostřednictvím otisku prstu skrze svůj mobilní telefon a zároveň se automaticky zapne topení, když je detekována jeho přítomnost v obývacím pokoji. Když kurýr přinese balíček, ale nikdo není doma, lze s ním mluvit na dálku prostřednictvím elektronického zvonku. Chytrý budík ráno zazvoní dle aktuální fáze spánku a zároveň se připraví oblíbená káva. Samozřejmě existuje celá řada dalších možností.

Všechny výše uvedené příklady jsou výsledkem možné spolupráce a komunikace mnoha elektronických zařízení a senzorů. V současné době může být téměř cokoliv v domácnosti digitální a připojené přímo či nepřímo k internetu nebo smartphonu, tato zařízení používají ke komunikaci různé protokoly. Množina jednoduchých digitálních zařízení tvoří chytrou domácnost neboli smart home a pokud jsou zařízení zároveň schopná komunikace vzdáleně prostřednictvím internetu, pak lze říct, že se jedná o prvky internetu věcí neboli Internet of Things zkráceně IoT.

Zařízení chytré domácnosti mohou značně ulehčit život, ale také zkomplikovat v případě narušení bezpečnosti. Přenášená data často obsahují citlivé informace, které mohou případní útočníci zneužít dle své libosti. Dobré zabezpečení se tedy stává základním prvkem smart home.

Diplomová práce se bude zabývat analýzou, návrhem a praktickým ověřením zabezpečení zařízení smart home. Nejprve budou představeny principy komunikačních technologií Bluetooth a ZigBee a jejich zabezpečení. V praktické části bude navrženo řešení bezpečné komunikace pro potřeby smart home s ohledem na dostupnost takového řešení, které bude následně otestováno.

1 Aktuální pohled na smart home a jeho zabezpečení

Internet věcí neboli IoT se stává stále častěji skloňovaným výrazem. Dnes již téměř každý člověk vlastní mobilní telefon či nějaká jiná zařízení, která jsou schopná navzájem komunikovat prostřednictvím internetu. V roce 2017 bylo aktivních přes 8,4 miliard zařízení IoT [1] a očekává se, že v roce 2020 bude těchto zařízení mezi 20–40 miliardami [2]. Mezi prvky IoT lze zařadit i menší a méně komplexní zařízení, než je již zmíněný smartphone, jde například o chytré náramky, osvětlení, termostat či zámek dveří. Jednou z oblastí využití prvků IoT je smart home neboli chytrá domácnost.

Marikyan a spol. [3] definuje smart home jako rezidenci vybavenou smart technologiemi zaměřenými na poskytování na míru šitých služeb uživatelům. Tyto smart technologie umožňují monitorovat, kontrolovat a podporovat obyvatele, což může zlepšit kvalitu života a podpořit nezávislý život.

Podobně i Balta-Ozkan a spol. [4] říká, že smart home je rezidence vybavená komunikační sítí propojující senzory, domácí spotřebiče a zařízení. Tyto prvky lze vzdáleně monitorovat, přistupovat k nim nebo je ovládat, a které poskytují služby, jež odpovídají potřebám uživatelů.

Oproti tomu De Groote a spol. [5] zobecňuje definici na budovu jako takovou. Inteligentní budovy jsou flexibilně propojeny a interagují s energetickým systémem a jsou schopny efektivně vyrábět, ukládat a / nebo spotřebovávat energii.

Definicemi [4] a [5] se inspiroval Darby [6], který našel společný prvek ve významu komunikačních sítí pro vzájemné propojení zařízení nebo subsystémů, které umožňují vzdálený přístup a kontrolu spolu s poskytováním služeb.

Mezi prvky smart home, lze tedy zařadit jakékoliv zařízení v domácnostech, jež je ovládané vzdáleně například přes mobilní telefon. Tyto prvky ke komunikaci používají různé komunikační technologie, mezi které lze zařadit Wi-Fi [7], která zahrnuje rodinu standardů IEEE 802.11 [8], Bluetooth [9] či ZigBee [10]. Vzdálené ovládání přináší větší komfort, ale také otevírá nové možnosti, jak se dostat k citlivým a osobním datům uživatelů a možnost zneužití chytrého zařízení např. pro útoky typu DDoS. Zajištění bezpečné komunikace je tedy velmi důležité a podstatné pro provozování chytrých zařízení, a to nejen v domácnosti.

Dle Aliho a spol. [11] existuje pět základních cílů zabezpečení ve smart home.

Autentifikace: ověření komunikujících stran nebo uživatele, kdo je uživatel a co tvrdí, jaká data odesílá autor žádosti.

Autorizace: zajištění definice všech přístupových práv uživatelů za účelem utilizace systémových zdrojů.

Důvěrnost: zajišťuje, že k soukromým údajům v systému mají přístup pouze oprávnění uživatelé.

Integrita: zajišťuje, že data jsou udržována konzistentně a správným způsobem. Změny a ztráty dat budou oznámeny.

Dostupnost: zajišťuje, aby pro každého oprávněného uživatele byly vždy k dispozici všechny služby a aby tyto zdroje byly chráněny před jakoukoliv hrozbou.

Shouran a spol. [12] k těmto cílům přidává ještě neodmítání.

Neodmítání: ujištění, že budou existovat nepopiratelné důkazy k ověření pravdivosti jakéhokoli nároku entity.

Dále Ali [11] i Shouran [12] zmiňují útoky, jež mohou nastat, a na který z výše zmíněných cílů útočí. Jedná se například o odposlouchávání, analýza síťového provozu, pozměnenní zprávy nebo imitace uživatele.

Obecné modely a přístupy, které byly zmíněny výše jsou komplikované a často složitě implementovatelné. Proto Lee a spol. [13] rozděluje smart home síť do třech vrstev (koncová zařízení, domácí gateway a cloud) a zjednodušuje požadavky na domácí gateway na integritu, autentizaci a důvěrnost. Lee navrhuje konkrétní architekturu domácí gateway založenou na blockchain technologii, která řeší problémy s důvěrností, integritou a autentizací. Dále použil SHA2 šifrování k zabezpečení komunikace na mezi koncovými zařízeními a gateway. Navrhované řešení je ovšem velmi náročné na výpočetní výkon, který je potřeba k provozu blockchain.

Cloudová řešení, jaké předpokládá i Lee, jsou dnes stále více využívána a je tedy nutné se zabývat i jejich bezpečností. Národní úřad pro kybernetickou a informační bezpečnosti (NÚKIB) proto připravuje „cloudovou vyhlášku“, která definuje

bezpečnostní pravidla a úrovně informačních systémů pro veřejnou správu či pro orgány veřejné moci, jež budou využívat cloudových služeb [14].

Štěpán a spol. navrhli zcela nový protokol [15], který je uzpůsoben zařízením v typickém světě IoT, tedy takovým zařízením, která jsou velmi limitována hardwarem a výpočetním výkonem. Tento protokol využili ke komunikaci ve frameworku HAuSy [16], který je navržen pro implementaci smart home. Jejich framework používá třívrstvou architektura, která obsahuje server, subsystémy a uzly. Server je obecnou náhradou řídicí jednotky, poskytuje API pro uživatelské aplikace, zajišťuje zabezpečení a pravidla uživatelů. Jeden sever komunikuje s několika subsystémy, které jsou zpravidla umístěny v každé hlavní místnosti v domě a komunikují s koncovými uzly. Uzly pak obsahují mikrokontroler, který pracuje s periferiemi [17].

Dey a Hossain [18] navázali na práci Kumara a spol. [19], který navrhl mechanizmus založení klíče relace s využitím protokolu třetí strany, který je založený na kryptografii s využitím symetrických klíčů. V jeho schématu důvěryhodný poskytovatel služeb ukládal určité bezpečnostní parametry, včetně symetrického klíče, offline do domácí gateway a každého IoT zařizení. Před vytvořením zabezpečené relace mezi domácí gateway a chytrým zařízením se nejprve vytvoří dva klíče relace a navzájem se autentizují na základě parametrů, které dříve přijali. Day s Hossainem [18] se snažili vyřešit nutnost offline registrace každého nového zařízení. Jejich řešení spočívá v implementaci Diffie-Hellmanovi výměny klíčů [20] a vstupu uživatele, který do domácí gateway zadává jednotlivá ID zařízení.

Wazid a spol. [21] uvažuje hierarchickou IoT síť, která obsahuje typy uzlů, jako je gateway, klastrový uzel a senzorový uzel, jež jsou uspořádány hierarchicky, a pro niž navrhl protokol založený na autentizaci uživatelů pomocí výměny klíčů. Protokol používá třífaktorovou autentizaci pomocí karty uživatele, hesla a biometrických údajů a je postaven na šesti fázích: 1) off-line registrace senzorového uzlu, 2) registrace uživatele, 3) přihlášení uživatele, 4) autentizace a výměna klíčů, 5) změna hesla a biometrie a 6) nasazení nového senzorového uzlu.

Chofor a spol. [22] navrhl schéma autorizace zařízení pro smart home zařízení, která jsou připojena k nedůvěryhodnému cloudovému systému. K ověření používá

protokol FIDO [23], který slouží uživatelům k autentizaci jejich zařízení. Prezentované řešení zachovává anonymitu uživatele, výrobci totiž nemohou vytvořit propojení mezi různými uživatelskými účty, jelikož jedinou informací související s uživatelem je veřejný klíč FIDO a pseudonym.

Pecorella, Pierucci a Nizzi [24] navrhli použití frameworku SHIELD, který tvoří další vrstvu ochrany mezi firewallem poskytovatelem internetových služeb (ISP) a domácí gateway. SHIELD je schopný detekovat útoky a na základě „sentimentu" se rozhoduje jaký postoj k němu zaujme. Navíc může informovat okolní domácí gateways o tom, že se takový útok odehrál.

Meng a spol. [25] se zaměřil na zabezpečení hlasem ovládaných uživatelských rozhraních v prostředí smart home jako je například Amazon Alexa. Využil vlastního frameworku WiVo [26], který pomocí předem nahraných a zanalyzovaných příkazů uživatele rozpoznává, zda je požadavek oprávněný (proveden registrovaným uživatelem) nebo se jedná o spoofing útok.

Na osobní asistenty v prostředí smart home, jako je Amazon Alexa nebo Microsoft Cortana a jejich zabezpečení se zaměřil i Edu a spol. [27]. Edu vidí největší zranitelnost v komunikaci mezi uživatelem a osobním asistentem, ovšem rozsah možných útoků je velký a neexistuje jedno komplexní řešení. Edu doporučuje zdokonalit autentizační a autorizační metody prováděním systematického hodnocení bezpečnosti s využitím umělé inteligence a zároveň doporučuje zlepšit povědomí uživatelů o možných hrozbách a obraně.

Ramapatruni a spol. [28] navrhl model, který je schopný učit se pomocí skrytého Markovova modelu typické používaní smart home sítě a detekovat v něm anomálie. Jejich model rozpoznal útok s 97 % přesností. Jejich model ovšem předpokládá pouze jednoho uživatele. Do budoucna plánují zobecnit model pro více uživatelů s více daty a s využitím i jiných nástrojů machine learningu a big data.

Machine learning využil také Makkar a spol. [28] při návrhu frameworku detekujícího spam pro zařízení IoT. Jejich framework vyhodnocuje pět machine learningových modelů, které využívají různých metrik a velké kolekce dat. Každý model počítá skóre spamu s ohledem na vylepšené vstupní funkce. Toto skóre přestavuje důvěryhodnost zařízení IoT podle různých parametrů.

Oblastí, které mohou být ohroženy různými typy útoků, je hodně. Podobně i přístupy, které se snaží zabezpečit komunikaci v prostředí smart home, jsou různé. Některé přístupy se zaměřují na zvýšení zabezpečení komunikace v prostředí internetu, jiné přímo na hrozby na úrovni domácí sítě. Navržena byla komplexní řešení využívající nejmodernějších technologií, jako je blockchain nebo machine learning. Dále byla navržena specifická řešení, jako je rozpoznávání hlasu v případě použití hlasově ovládaných asistentů. Ovšem základem stále zůstává vyřešení bezpečné autentizace a autorizace, jakožto základním stavebním prvkem bezpečnosti v smart home. Cílem práce je navrhnout takové řešení, které nabídne dostatečnou ochranu smart home s ohledem na náklady takového zabezpečení.

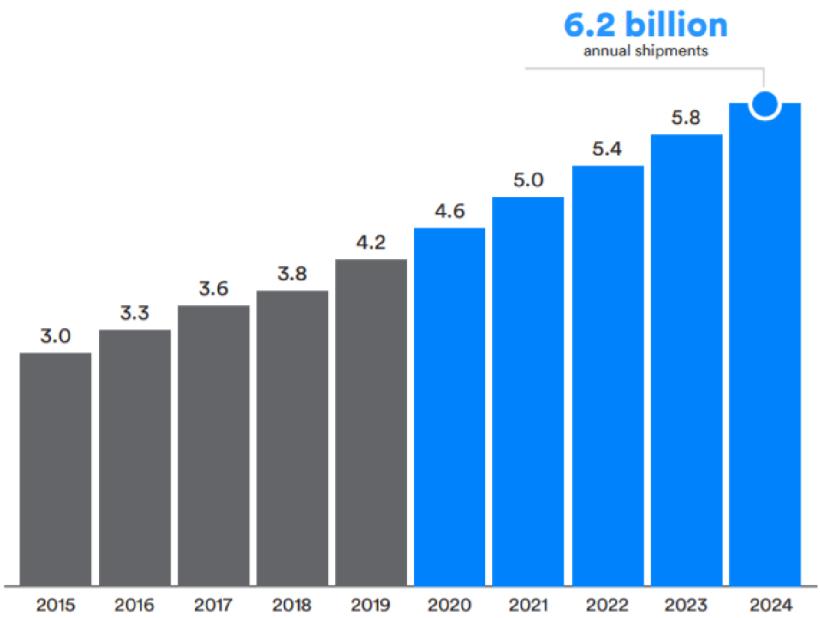
2 Bluetooth

Bezdrátová technologie Bluetooth je komunikační systém krátkého dosahu určený k nahrazení kabelů propojujících přenosná nebo pevná elektronická zařízení. Klíčovými vlastnostmi bezdrátové technologie Bluetooth jsou robustnost, nízká spotřeba energie a nízké náklady. [30]

2.1 Historie

V roce 1993 dostal Jaap Haartsen, inženýr bezdrátové komunikace, který pracoval ve společnosti Ericsson, úkol vyvinout připojení krátkého dosahu pro mobilní telefony. V té době byly dráty a kabely velkou překážkou komunikačních systémů. To platilo zejména pro nástroje komunikující na krátké vzdálenosti, avšak vyžadující hodně nastavení. [31]

Proto v roce 1995 Haartsen požádal o pomoc kolegu Svena Mattisona a společně se jim podařilo vyvinout takzvané multi-komunikátorové odkazy nebo zkráceně MC odkazy. Haartsen a Mattison jsou často označováni za tvůrce Bluetooth. Byl to ovšem technický ředitel společnosti Ericsson, kdo jako první zahájil vývoj této technologie, čtyři roky před povoláním Haartsena. Na projektu samozřejmě pracovalo mnoho různých lidí, každý s vlastním příspěvkem. [31]



Obrázek 1 - Roční dodávky Bluetooth zařízení [32]

2.1.1 Název a logo

Tato kapitola byla zpracována dle [33].

Bluetooth neboli modrý zub bylo přízvisko krále Harralda Gormssona, který dostal, jelikož měl jeden mrtvý zub modré barvy. Král Harrald byl, ale také znám spojením Norka a Dánska roku 958.

V roce 1996 se sešli zástupci společností Intel, Ericsson a Nokia, aby naplánovali standardizaci této rádiové technologie krátkého dosahu na podporu konektivity a spolupráce mezi různými produkty a průmyslovými odvětvími. Během této schůze Jim Kardach ze společnosti Intel navrhl název Bluetooth jako dočasné krycí označení, jelikož podobně jako král Herrald spojil Skandinávii, tak i tyto tři společnosti právě chtěly spojit svět počítačů a telekomunikačního odvětví s bezdrátovou technologii krátkého dosahu.

Později, když přišel čas vybrat vhodné pojmenování nové technologie, mělo být Bluetooth nahrazeno buď pojmem RadioWire nebo PAN (Personal Area Network). Bohužel obě uvažované varianty jednoznačně neidentifikovali novou technologii, a tak se označení Bluetooth stalo i trvalým názvem.

Bílý symbol v modrém poli, jež je součástí loga, vznikl spojením runových iniciál krále Harralda (Hagall) (ᚼ) Bluetooth (Bjarkan) (ᛒ).

2.2 Verze

Tato kapitola popisuje historii verzí technologie Bluetooth a byla zpracována z těchto zdrojů: [30], [34], [35].

2.2.1 Verze 1.x

Bluetooth verze 1.0 a 1.0b byly vydány v červenci roku 1999 a obsahovaly mnoho problémů, které výrobcům znesnadňovaly výměnu informací produktů mezi sebou. Součástí obou verzí byla také povinná výměna hardwarové adresy Bluetooth zařízení.

V únoru roku 2001 byla vydána verze 1.1, jež obsahovala opravy na některé problémy z předchozích verzí. Hlavním vylepšením této verze oproti předchozí byla autentizace. Součástí autentizace je proces generování klíče, který zajišťuje master zařízení. Ovšem pokud slave komunikuje rychleji než master, tak si mohl myslet, že

on je mastrem, což způsobovalo zmatky a znemožnění platné komunikace. Verze 1.1 tento problém vyřešila podrobnějším definováním kroků během autentizace.

Bluetooth 1.2 byl vydán v listopadu 2003. Tato verze obsahovala některá další vylepšení, ale zároveň byla zpětně kompatibilní s verzí 1.1. Bluetooth 1.2 byl certifikován jako standart IEEE 801.15.1-2005 [36].

2.2.2 Verze 2.x

Verze 2.0 byla vydána v listopadu 2004. Hlavním rozdílem oproti starší verzi bylo zavedení EDR (Enhanced Data Rate), což pomohlo zvýšit přenosovou rychlost na 2,1 Mb/s. Dále se snížila spotřeba energie snížením pracovního cyklu.

Po necelých třech letech, v červenci 2007, byla vydána verze 2.1 a byly přidány některé funkce, které zahrnují:

- Snížení spotřeby energie v režimu vyhledávání.
- Kooperace s NFC.
- Obnovení pozastaveného šifrování.

2.2.3 Verze 3.x

Bluetooth ve verzi 3.0 byl vydán v dubnu 2009. Hlavními funkcemi, které byly přidány do této verze, bylo zjednodušení zjišťování a nastavování zařízení Bluetooth. Dále umožnila zobrazení všech typů služeb, které zařízení poskytuje.

Bluetooth 3.0 je také známý jako vysokorychlostní (High Speed) kvůli výraznému zvýšení rychlosti oproti starším verzím. Bluetooth verze 3.0 poskytovalo přenosovou rychlost až 20 Mb/s.

2.2.4 Verze 4.x

Verze 4.0 byla vydána v prosinci 2009 a vylepšila některé funkce pro lepší dosah a připojení. Přelomovým vylepšením této verze ovšem bylo zavedení protokolu Bluetooth Low Energy (BLE), který obsahuje uvedení zařízení do režimu spánku nebo hibernace v případě, že se zařízení delší dobu nepoužívá.

Bluetooth 4.1 byl vydán v prosince 2013. Přinesl vylepšení v oblasti připojení díky flexibilitě a variabilitě časového intervalu opětovného připojení. Zařízení se mohou automaticky znovu připojit, například v případě, kdy nejsou dočasně v

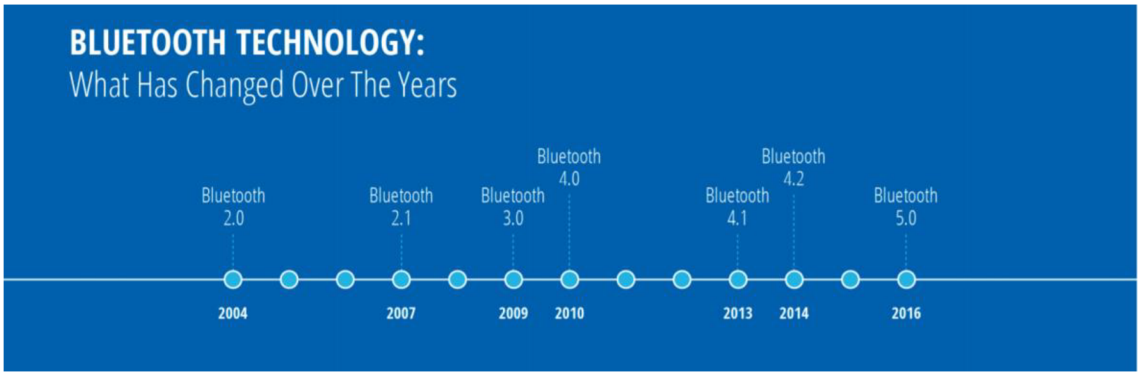
dosahu. Uživatel již nemusí zařízení znovu připojovat ručně. Dále k vylepšení přenosu dat zavedením technologie známé jako „Bluetooth smart technology“, která umožňuje hromadný přenos dat.

V prosinci 2014 bylo vydáno Bluetooth verze 4.2. Tato verze zvyšovala přenosovou rychlost 2,6x oproti starší verzi, což znamenalo rychlejší stahování.

2.2.5 Verze 5.x

Bluetooth verze 5.0 byla představena v prosinci 2016. Opět došlo ke zlepšení efektivity, zvýšení rychlosti a dosahu. Zcela novou funkcí je dual audio, která umožňuje přehrávat zvuk na dvou připojených zařízeních současně.

V lednu roku 2019 byla představena verze 5.1 a krátce na to v prosinci roku 2019 i nejaktuálnější verze 5.2. V těchto verzích opět došlo k vylepšení ve spotřebě elektrické energie různých typů zařízení.



Obrázek 2 - Vývoj verzí Bluetooth [35]

2.2.6 Souhrn

Od roku 1999, kdy byla vydána první verze Bluetooth, docházelo k postupnému vylepšovaní technologie, hlavní změny v jednotlivých verzích shrnuje Tabulka 1.

Tabulka 1 - Vylepšení ve verzích Bluetooth [35]

Verze	Rok vydání	Důležitá vylepšení
1.0	1999	-
1.2	2003	Změna frekvence, RSSI
2.0	2004	2,1 Mb/s
2.1	2007	3,0 Mb/s
3.0	2009	24 Mb/s
4.0	2010	Nižší spotřeba energie, nižší latence připojení
4.1	2013	Vylepšená správa napájení zařízení spárováním, jež umožňuje automatické zapnutí a vypnutí
4.2	2014	Vylepšené zabezpečení, prodloužení délky datového paketu Low Energy
5.0	2016	Vyšší přenosová rychlost (48 Mb/s), lepší energetická účinnost, větší dosah a silné point-to-point spojení a spolehlivost

2.3 Technologie Bluetooth

Dnes existují tři hlavní typy technologie Bluetooth:

a) Bluetooth Basic Rate (BR)/Enhanced Data Rate (EDR),
b) Bluetooth Low Energy (BLE),
c) Bluetooth Mesh.

Tyto technologie budou podrobněji popsány v jednotlivých podkapitolách, jež byly zpracovány zejména ze zdrojů [35], [37], [38], [39], [40].

2.3.1 Bluetooth BR/EDR

Bluetooth pracuje v nelicencovaném industriálním, vědeckém a lékařském frekvenčním pásmu 2,4000 GHz až 2,4835 GHz. V tomto pásmu funguje řada technologií, včetně standardu IEEE 802.11 b/g/n/ac pro bezdrátové lokální sítě (WLAN), což má za následek přehlcení tohoto pásma z hlediska objemu bezdrátových přenosů. Bluetooth využívá pro přenosy technologii Frequency

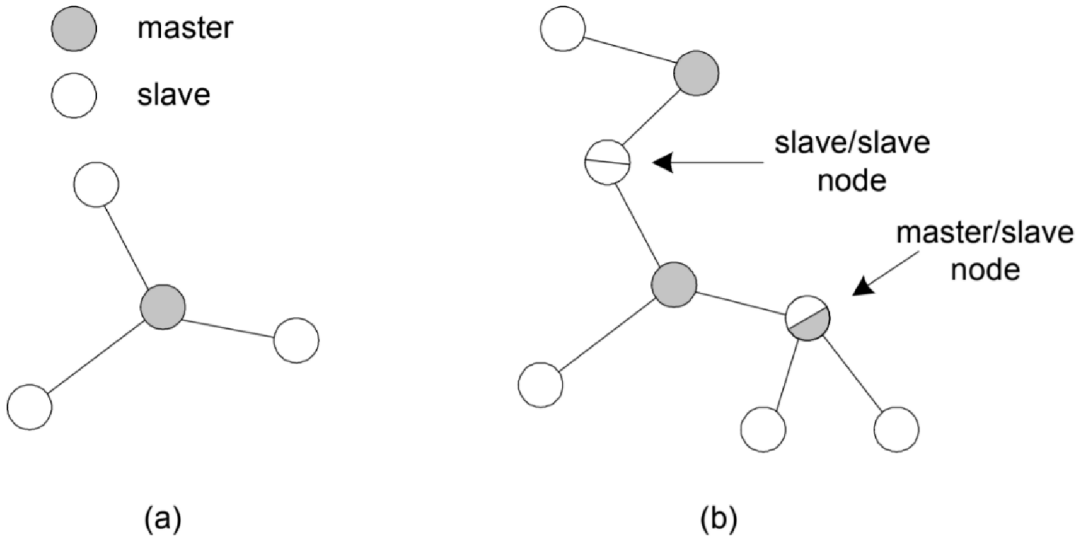
Hopping Spread Spectre (FHSS). FHSS snižuje rušení a chyby přenosu, ale poskytuje minimální zabezpečení přenosu.

Jsou definovány dva režimy přenosu dat: Basic Rate (BR), která používá binární frekvenční modulaci (FM) k minimalizaci složitosti transceiveru a Enhanced Data Rate (EDR), která využívá modulaci Phase Shift Keying (PSK). Režim Basic Rate (BR) je povinnou součástí specifikace Bluetooth. BR používá modulaci Gaussian Frequency Shift Keying (GFSK). V režimu EDR je schéma, použité pro modulaci, změněno uvnitř paketu.

Komunikace mezi Bluetooth Basic Rate (BR)/EDR zařízeními využívá 79 různých rádiových kanálů o šířce 1 megahertz (MHz). Díky použití FHSS dochází ke skoku (změně) frekvence přibližně 1600krát za sekundu v případě datových či hlasových připojení a 3200krát za sekundu v režimu vyhledávání. Kanál se používá po velmi krátkou dobu, po které následuje skok do jiného kanálu označeného předem stanovenou pseudonáhodnou sekvencí. [37]

Kombinace zařízení připojených přes Bluetooth v režimu ad-hoc se nazývá piconet. V piconetu funguje jedno zařízení Bluetooth jako master a ostatní zařízení jako slave. Zařízení, které zahájí komunikaci, je master a ostatní zařízení jsou slave. Piconet se skládá z minimálně dvou a maximálně osmi připojených zařízení (jeden hlavní a sedm podřízených). Každé zařízení Bluetooth může fungovat jako master nebo slave. Pokud je k dispozici jedno zařízení typu slave, tak se jedná o jednoduché spojením typu point-to-point. Uspořádání point-to-multipoint může mít maximálně sedm aktivních slave, jež řídí jedno master zařízení. Slave zařízení si vždy vyměňují data prostřednictvím master uzlu. Komunikace napříč piconety vytváří tzv. scatternet. Scatternet je vytvořen tehdy, když existuje jedno Bluetooth zařízení, které zastává roli slave v jednom piconetu a současně zastává roli master nebo slave v jiném piconetu.

Všechna zařízení sdílejí hodinový takt master Bluetooth. Základní doba taktu je 312,5 mikrosekund. Slot 625 mikrosekund se skládá ze dvou hodinových cyklů. Master odesílá sudé sloty a přijímá liché sloty. V případě slave zařízení dochází k přenosu v lichých slotech a příjem probíhá v sudých slotech. Slave zařízení čeká na dotazy od master zařízení a poskytuje odpovědi.

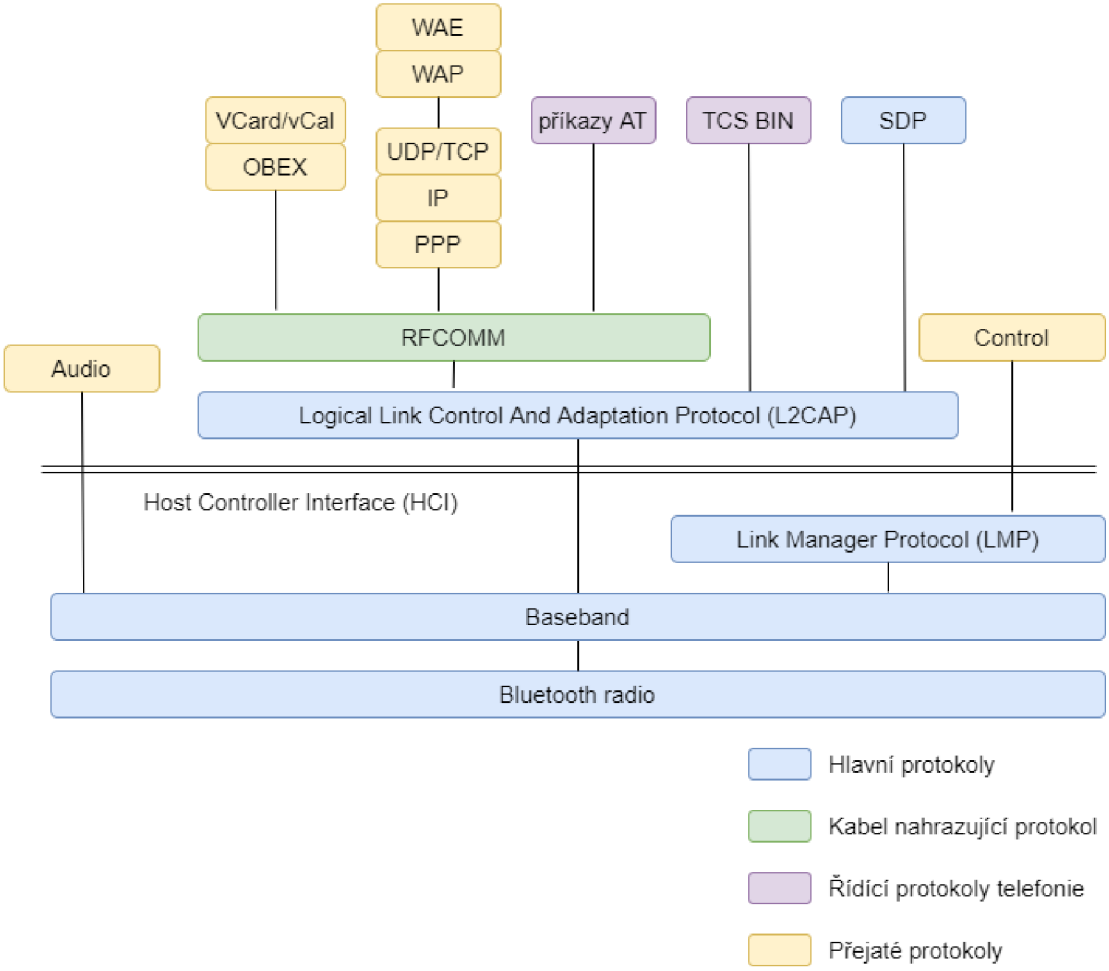


Obrázek 3 - Síťová konfigurace Bluetooth piconet (a) a scatternet (b) [41]

2.3.2 Protokol Bluetooth BR/EDR

Architektura Bluetooth BR/EDR využívá sedm různých protokolů:

- Radio protokol,
- Protokol základního pásma (Baseband),
- Radio Frequency Communication (RFCOMM),
- Service Discovery Protocol (SDP),
- Link Management Protocol (LMP),
- Logical Link Control and Adaptation Layer Protocol (L2CAP),
- Protokol Host Controller Interface (HCI).



Obrázek 4 - Bluetooth protokoly [35], [40]

Radio protokol

Radio protokol definuje rádiové vlastnosti bezdrátové technologie Bluetooth a leží na fyzické vrstvě modelu ISO/OSI. Jak již bylo zmíněno výše, frekvenční pásmo Bluetooth je rozděleno na 79 kanálů po 1 MHz. Každý kanál je dále rozdělen na 625 mikrosekundových časových slotů, což má za následek 1600 slotů za sekundu. Data jsou přenášena prostřednictvím těchto slotů a kanálů.

Baseband

Protokol Baseband zpracovává odeslané a přijaté signály. Dále se stará o řešení oprav chyb, paketů a řízení toku. Každé zařízení Bluetooth má dva parametry, které tvoří základ komunikace Bluetooth. První z nich je jedinečná 48bitová adresa typu IEEE 802 přiřazená každému zařízení Bluetooth již při výrobě, podobně jako MAC adresa zařízení Ethernet. Druhým parametrem jsou volně běžící 28bitové hodiny, které tikají jednou za 312,5 ms.

Navázání spojení

Bluetooth zařízení mohou navázat synchronní Synchronous Connection-Oriented (SCO) nebo asynchronní Asynchronous Connectionless (ACL) spojení. Datové pakety používají primárně ACL, kdežto hlasové pakety používají SCO. ACL spojení je primárně určeno pro datové přenosy. Spojení skrze ACL mohou dosáhnout maximální rychlosti přenosu dat 57,6 Kb/s v jednom směru a 721 Kb/s ve druhém směru. Zařízení typu master kontroluje ACL spojení, aby alokoval šířku pásma pro zařízení slave. ACL spojení podporuje broadcast pro posílání zpráv z master zařízení všem slave zařízením v piconetu.

Spojení SCO podporují symetrická spojení typu point-to-point a používají se hlavně pro hlasové přenosy. Pro spojení SCO je každý šestý slot vyhrazen pro vysílací kanál a rezervace následujícího slotu je provedena pro přijímací kanál. Po navázání spojení odesílají pakety SCO master i slave zařízení dle potřeby. Datové a hlasové přenosy jsou povoleny v jednom typu paketu SCO s opakovaným přenosem datové části v případě poškození paketu.

Adresace

Adresa zařízení Bluetooth je pevně dána a nelze ji změnit. Je rozdělena do tří částí:

- LAP: Dolní část adresy skládající se z 24 bitů (adresa výrobce)
- UAP: Horní část adresy skládající se z 8 bitů
- NAP: Nevýznamná část adresy skládající se z 16 bitů

Celkový adresní prostor je 2^{32} a odpovídá polím LAP a UAP.

Formát paketu

Pořadí bitů při definování paketů a zpráv přes rozhraní Bluetooth se řídí formátem Little-Endian. Každý paket se skládá ze tří entit: přístupového kódu, hlavičky a samotných dat (payload). Přístupový kód i hlavička má pevně danou délku 72 respektive 54 bitů. Payload může být velký 0 až 2753 bitů.

Přístupový kód identifikuje všechny pakety vyměněné v rámci piconetu. Přístupový kód se také používá pro procedury paging a inquiring, v takovém případě

není ve zprávě vyžadována hlavička ani data. Podle toho jsou definovány tři různé druhy přístupových kódů.

- Přístupový kód kanálu (CAC) – identifikuje piconet
- Přístupový kód zařízení (DAC) – pro stránkovací požadavky a odpovědi
- Přístupový kód Inquring (IAC) – k vyhledání zařízení Bluetooth v dosahu

Hlavička paketu obsahuje informace sloužící k řízení spojení. Jde o šest částí, které zahrnují adresu člena, kód typu, řízení toku, indikaci potvrzení, pořadové číslo a kontrolu chyby hlavičky.

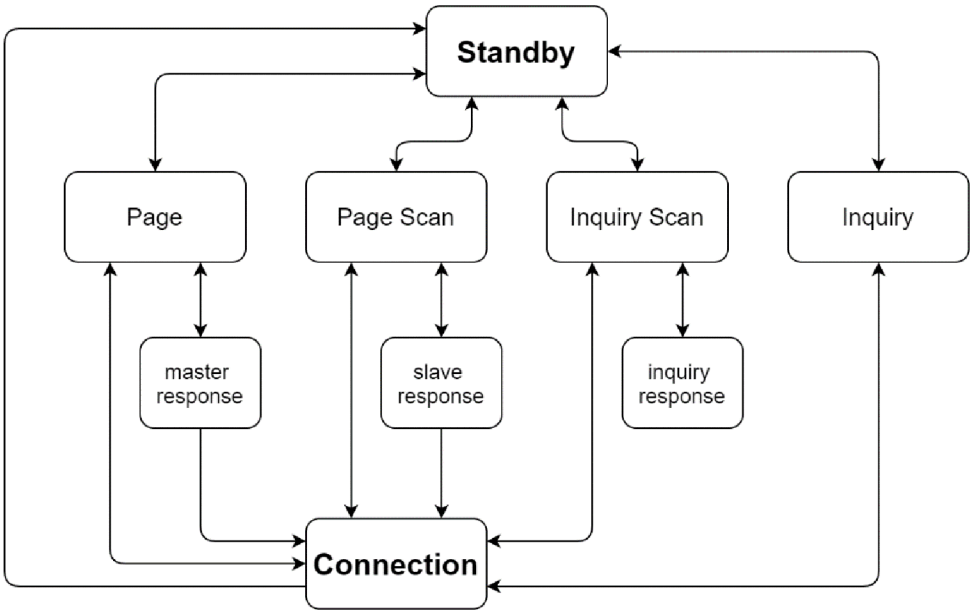
Typy paketů

Celkem dvanáct různých typů paketů, označených 4bitovým kódem TYPE, je definováno pro SCO a ACL. Jsou rozděleny do čtyř segmentů. První segment je vyhrazen pro čtyři řídicí pakety, které jsou stejné pro SCO i ACL spojení. Druhý segment je vyhrazen pro pakety zabírající jeden časový slot. Třetí segment je vyhrazen pro pakety zabírající tři časové sloty a čtvrtý segment představuje pakety, které zabírají pět časových slotů.

Řízení stavů spojení

Obrázek 5 zobrazuje diagram řízení stavů spojení. Existují dva hlavní stavy STANDBY a CONNECTION. Ostatní stavy jsou dílčí stavy, které definují stav propojení zařízení, kdykoli je do piconetu přidáno nové zařízení typu slave.

Stav STANDBY je výchozí stav, ve kterém zařízení spotřebovává pouze malé množství energie. Zařízení iniciuje nebo reaguje na dotazy typu page nebo inquiring od jiných zařízení. Pokud je rozpoznáno vhodné spojení, může zařízení přejít do stavu CONNECTION. Zařízení funguje jako master, pokud iniciuje požadavek page scan na jiná zařízení.



Obrázek 5 - Bluetooth stavy spojení a jejich přechody [42]

Link Manager Protocol

Zprávy protokolu Link Manager Protocol (LMP) se vyměňují na úrovni správce spojení a slouží k nastavení spojení mezi zařízeními Bluetooth. Poskytují kontrolu a vyjednávání o velikosti paketu při přenosu dat. LMP také poskytuje řízení a správu napájení, adresování, manipulaci s režimem připojení a přepínání typu master-slave konfigurace spojení a stavů spojení v piconetu. Nabízí také funkce zabezpečení, včetně výměny šifrovacích klíčů mezi zařízeními pro autentizaci a šifrování. Zprávy LMP mají vyšší prioritu než uživatelská data.

Host Controller Interface

Host Controller Interface (HCI) poskytuje příkazové rozhraní pro Baseband a správce spojení (Link Manager). Poskytuje také přístup k řídicím registrům a stavu hardwaru. HCI je přístupný na hostiteli, hostitelském řadiči a transportní vrstvě.

Logical Link Control and Adaptation Protocol

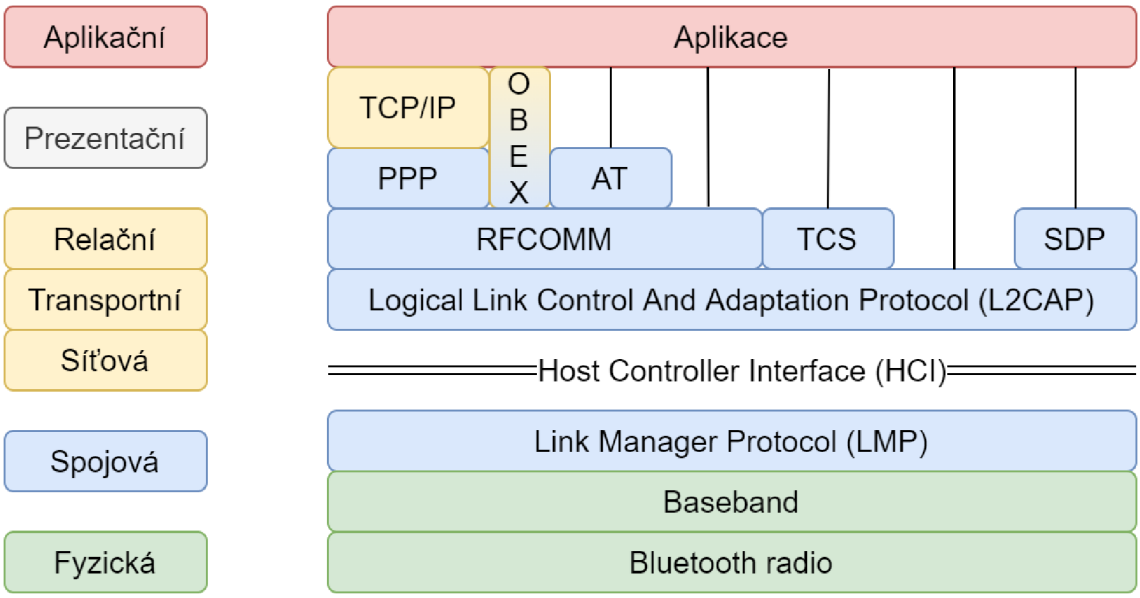
L2CAP je strukturován přes protokol Baseband a je přítomen v linkové vrstvě. L2CAP segmentuje pakety pro přenos Bluetooth a když paket projde skrze L2CAP jiného zařízení Bluetooth, jeho původní podoba se obnoví opětovným sestavením. L2CAP podporuje pouze ACL nikoliv SCO spojení.

Service Discovery Protocol

Protokol Service Discovery Protocol (SDP) používá model zprávy request-response, takže každá transakce má jednotku PDU (Protocol Data Unit) odpovědi a jednu PDU požadavku.

RFCOMM

RFCOMM je transportní protokol, který umožňuje emulaci sériového portu RS232 přes protokol L2CAP.



Obrázek 6 - Zařazení protokolů Bluetooth do modelu ISO/OSI [42]

2.3.3 Bluetooth Low Energy (BLE)

Fyzická vrstva v Bluetooth Low Energy (BLE) je redukovaná a optimalizovaná oproti fyzické vrstvě u Bluetooth BR. Zatímco u technologie BR je frekvenční rozsah rozdělen do 79 kanálů, BLE používá 40 kanálů, z nich tři slouží k vyhledávání zařízení. Díky tomu, že BLE při procesu vyhledávání prochází méně kanálů, je dosaženo efektivního a rychlého navázání spojení ve srovnání s Bluetooth BR / EDR. Šíře každého kanálu je u BLE 2 MHz na rozdíl od 1 MHz u BR, což snižuje nároky na rádio-frekvenční filtrování.

BLE umožňuje dodavatelům integrovaných obvodů provádět optimalizace, které jsou náročné pro BR / EDR Bluetooth. Tyto optimalizace umožňují, aby čipy v single-mode režimu byly energeticky účinnější než čipy v dual-mode nebo klasické

čipy. Srovnatelně jsou profily BLE vrstveny přes Generic Attribute Profile (GATT) pomocí protokolu GATT / ATT. Naproti tomu profily Bluetooth BR / EDR obecně určují své protokoly, což poskytuje lepší přizpůsobivost, ale činí implementaci složitější.

Mezi klíčové technologické cíle technologie Bluetooth Low Energy (ve srovnání s technologií Bluetooth BR / EDR) patří nižší spotřeba energie, nižší nároky na paměť, efektivní postupy zjišťování a připojení, krátké délky paketů a jednoduché protokoly a služby.

Tabulka 2 - Klíčové rozdíly mezi Bluetooth BR/EDR a Low Energy [37]

Charakteristika	Bluetooth BR/EDR	Bluetooth Low Energy
Kanály fyzické vrstvy	79 kanálů po 1MHz	40 kanálů po 2 MHz
Vyhledávání / Připojení	Inquiry/Paging	Adverising
Počet zařízení slave v piconetu	7 (aktivních) / 255 (celkem)	Neomezeno
Typický dosah	30 m	50 m
Maximální výstupní výkon	100 mW	10 mW

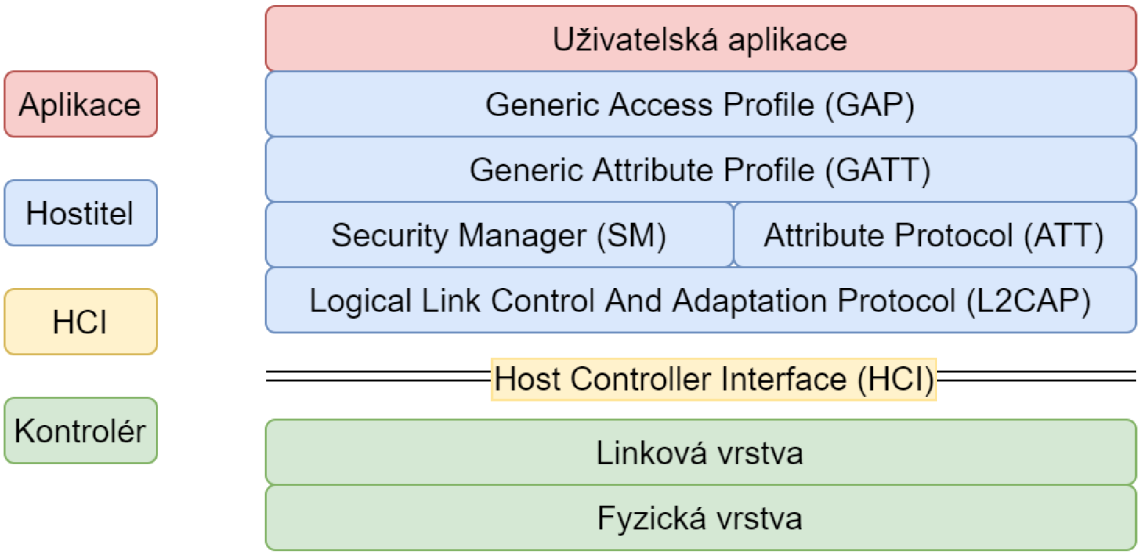
2.3.4 Protokoly BLE

Bluetooth Low Energy rozděluje protokoly do třech vrstev: řídící (controller), hostitelská (host) a aplikační (application).

Fyzická a Linková vrstva jsou součástí řídící vrstvy. Funkce vyšší vrstvy, protokol Logical Link Control and Adaptation Protocol (L2CAP), Attribute Protocol (ATT), Generic Attribute Profile (GATT), Security Manager (SM) a Generic Access Profile (GAP) jsou umístěny v hostitelské vrstvě. Host Controller Interface (HCI) umožňuje komunikaci mezi hostitelskou a řídící vrstvou. Aplikační vrstva se nachází nad hostitelskou vrstvou. L2CAP provádí segmentaci a opětovné sestavení paketů. Komunikace zařízení je definována skrze ATT a SM implementuje autentizaci, párování a distribuci klíčů.

Přenášená data jsou uchovávána v databázi GATT serveru. Tato data lze přemístit do GATT klienta. Aplikace určuje, zda je server/klient GATT udržován na hlavním nebo klientském uzlu. GATT specifikuje rámec pro komunikaci mezi GATT serverem a databázemi klientů.

Server GATT organizuje informace na základě hierarchie ({profil-služba, vlastnosti-atributy}). Atributy lze definovat jako malé adresovatelné datové entity GATT. Služba zahrnuje funkce a je charakterizována specifickými vlastnostmi. Skupiny služeb jsou popsány pomocí profilu. GAP identifikuje způsoby vyhledávání dalších zařízení BLE a definuje čtyři typy: advertiser, scanner/initiator, master a slave.



Obrázek 7 - Bluetooth Low Energy protokoly a vrstvy [35]

2.3.5 Bluetooth Mesh

Bluetooth Mesh standart [43] je založený na modelu publikace / subskripce, ve kterém se mohou vydavatelé (publisher) a odběratelů (subscriber) přihlásit k odběru jakéhokoli tématu. Například elektrické spínače mohou publikovat a elektrické lampy se mohou přihlásit k odběru tématu. Uzel sítě se může přihlásit k odběru několika adres, ale publikovat na jednu adresu. Adresy jsou obvykle uloženy v seznamu odběratelů.

Mesh standard obsahuje různé adresy: unicast a skupinové (group) adresy. Každý uzel získá jedinečnou unicast adresu, jakmile se připojí do sítě mesh. Skupina uzlů je reprezentována skupinovou adresou. Aby se mohl uzel připojit ke skupině, je nejdříve potřeba do seznamu odběratelů přidat novou adresu skupiny. Jakmile uzel získá skupinovou adresu, může mu kterýkoli jiný uzel odeslat zprávu prostřednictvím unicast adresy nebo skupinové adresy dané skupiny. Mesh topologie se používá k propojení vydavatelů a odběratelů.

Standard Bluetooth Mesh umožňuje komunikaci mezi uzly prostřednictvím skenování (scanning), inzerce (advertising) a zaplavovacích (flood) mechanismů. Flood mechanismus umožňuje uzlům v síti opakovaně předávat zprávy dalším uzlům, dokud nedosáhnou správného cíle. Uzly Bluetooth Mesh posílají pakety v náhodném časovém intervalu, nikoli v pevném intervalu, který používá BLE. Inzertní kanály jsou skenovány celou dobu pracovního cyklu pro příjem příchozích paketů. Uzly sítě proto nepřetržitě skenují příchozí pakety, s výjimkou časových intervalů, kdy dochází k přenosu.

Mesh standard dále definuje funkci Friendship, která pomáhá spořit energii. Tato funkce umožňuje zařízením připojeným ke stálému zdroji napájení (friend uzel) pomáhat low-power uzlům skenovat a připojit se k mesh síti. Friend uzel ukládá zprávy určené pro low-power uzel a předává zprávy, které dostává z low-power uzlu do mesh sítě.

2.4 Zabezpečení

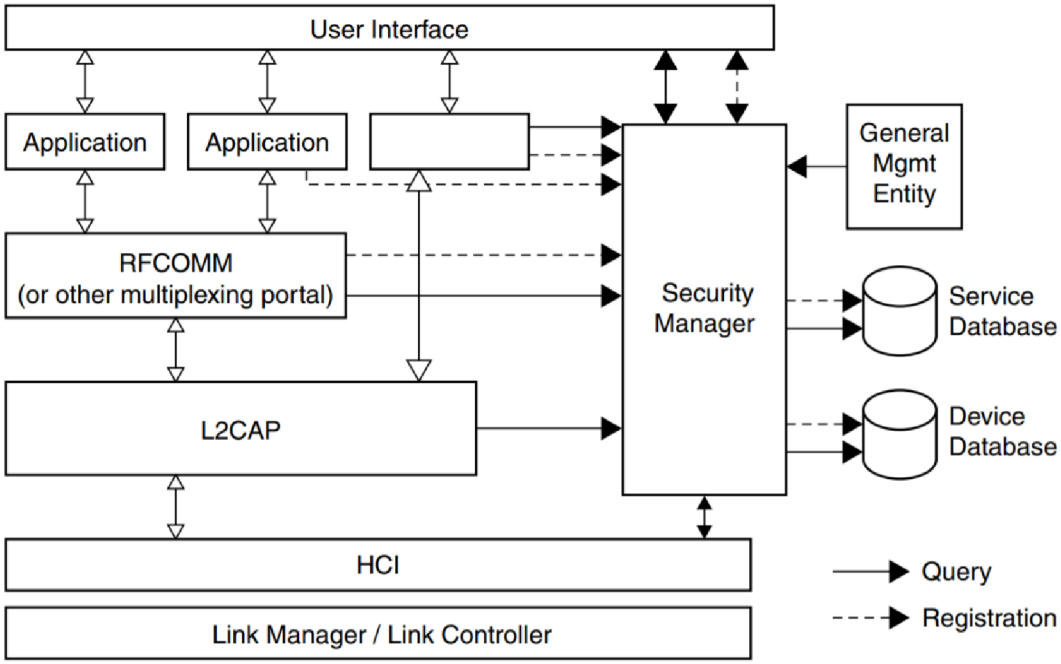
Zabezpečení Bluetooth podporuje autentizaci a šifrování. Tyto funkce jsou založeny na tajném klíči, který sdílí dvojice zařízení. Postup párování se používá, když dvě zařízení komunikují poprvé, aby vygenerovaly tento klíč. Tato kapitola byla zpracována z těchto zdrojů: [35], [37], [38], [42], [44].

Existují čtyři režimy zabezpečení zařízení:

- Režim zabezpečení 1: Nezabezpečený
- Režim zabezpečení 2: Vynucené zabezpečení na úrovni služby.
- Režim zabezpečení 3: Vynucené zabezpečení na linkové úrovni.
- Režim zabezpečení 4: Vynucené zabezpečení na úrovni služby využívající zabezpečené spojení

V nezabezpečeném režimu zařízení nebude iniciovat žádný bezpečnostní postup. Režim zabezpečení 2 zpracovávají vyšší vrstvy na základě zásad a postupů přístupu definovaných na úrovních služeb. Před vytvořením kanálu na úrovni L2CAP nejsou zahájeny žádné bezpečnostní postupy. Režim zabezpečení 3 zahrnuje bezpečnostní postupy na úrovni linkové vrstvy před vytvořením spojení a před vytvořením kanálů. Režim zabezpečení 4 je podobný režimu 2, tedy postupy jsou

definovány na úrovni služby, ale navíc využívá Secure Simple Pairing (SSP), ve kterém je použita vylepšená Diffie-Hellman šifra pro generování klíče spojení.



Obrázek 8 - Architektura zabezpečení Bluetooth [44]

Architekturu zabezpečení technologie Bluetooth ukazuje Obrázek 8. Správce zabezpečení (Security Manager) je klíčová funkce, která spravuje bezpečnostní klíče a provádí bezpečnostní postupy pro různé vrstvy protokolu.

K udržení zabezpečení v linkové vrstvě se používají čtyři parametry: veřejná adresa, která je jedinečná pro každé zařízení Bluetooth, dva tajné klíče (autentizační a šifrovací) a náhodné číslo, které se liší pro každou novou transakci. Tajné klíče jsou odvozeny během inicializace a poté už nejsou nikdy zveřejněny.

Autentizační algoritmus využívá 128bitový klíč. Pro šifrovací algoritmus je velikost klíče variabilní od 8 do 128 bitů, dle použitého algoritmu. Délka šifrovacího klíče je domluvena mezi zařízením master a slave na přijatelnou velikost, jak je určeno každou aplikací. Náhodné číslo je generované pseudonáhodným generátorem v jednotce Bluetooth a jeho délka je 128bitů.

2.4.1 Úrovně zabezpečení

Existují zde dva druhy úrovně zabezpečení: autentizace a autorizace.

Autentizace

Autentizace ověří, kdo je na druhém konci spojení. Bluetooth ji implementuje pomocí autentizační procedury založené na uloženém klíči spojení nebo párovací procedurou. Za účelem splnění různých požadavků na dostupnost služeb bez zásahu uživatele se ověřování provádí po zjištění požadované úrovně zabezpečení služby. Ověřování tedy nelze provést, když je navázáno spojení ACL.

Autentizace se provádí při odeslání požadavku na připojení ke službě. Používá se následující postup:

1. Žádost o připojení je odeslána do L2CAP.
2. L2CAP žádá o přístup správce zabezpečení.
3. Správce zabezpečení se dotáže databáze služeb.
4. Správce zabezpečení se dotáže databáze zařízení.
5. Je-li to nutné, správce zabezpečení vynutí autentizační a šifrovací proceduru.
6. Správce zabezpečení uděluje přístup a L2CAP pokračuje v nastavování připojení.

Autentizaci lze provést oběma směry: klient autentizuje server a naopak.

Autorizace

Když je jednomu zařízení povolen přístup k druhému, vzniká koncept důvěry. Důvěryhodná zařízení mají povolený přístup ke službám. Oproti tomu nedůvěryhodná zařízení mohou vyžadovat autorizaci na základě interakce uživatele, než je udělen přístupu ke službám. Existují dva druhy úrovní důvěryhodnosti zařízení:

1. *Důvěryhodné zařízení*: Zařízení s pevným vztahem (spárované), které má důvěryhodný a neomezený přístup ke všem službám.
2. *Nedůvěryhodné zařízení*: Toto zařízení bylo dříve ověřeno, je uložen klíč spojení, ale zařízení není v databázi zařízení označeno jako důvěryhodné.

3. *Neznámé zařízení* je také nedůvěryhodné zařízení. Pro toto zařízení nejsou k dispozici žádné bezpečnostní informace.

U služeb je požadavek na autorizaci, autentizaci a šifrování nastaven nezávisle. Požadavky na přístup definují tři úrovně zabezpečení:

- Služby vyžadující autorizaci a autentizaci: Automatický přístup je poskytován pouze důvěryhodným zařízením. Ostatní zařízení vyžadují ruční autorizaci.
- Služby, které vyžadují pouze autentizaci: Autorizace není nutná.
- Služby otevřené pro všechna zařízení: Autentizace se nevyžaduje, žádné schválení přístupu není vyžadován před udělením přístupu ke službě.

2.4.2 Limitace

Ověřeno je pouze zařízení, nikoli jeho uživatel. Neexistuje žádný mechanismus pro přednastavení autorizace pro každou službu. Pružnější bezpečnostní politiku je však možné implementovat se současnou architekturou bez nutnosti měnit architekturu protokolu Bluetooth. Rovněž není možné vynutit jednosměrný provoz.

2.5 Výhody a nevýhody

V této kapitole budou zmíněny čtyři výhody a čtyři nevýhody technologie Bluetooth dle [45].

Výhody

- **Bezdrátová technologie:** Jednou z hlavních výhod Bluetooth je, že pro přenos dat nevyžaduje žádnou formu vodičů. Díky tomu lze pohodlně odesílat a přijímat soubory. Mnoho dalších aplikací také využívá bezdrátovou technologii Bluetooth. Mezi takové aplikace patří osobní bezpečnostní systémy, lokalizační zařízení a zařízení pro kontrolu zdravotního stavu.
- **Dostupnost:** Bluetooth je technologie dostupná ve většině zařízení, jako jsou chytré telefony a tablety. Díky velkému rozšíření této technologie v rozličných zařízeních lze Bluetooth považovat za dostupnou technologii.

- **Uživatelská přívětivost:** K běžnému používání Bluetooth není nutná odborná znalost technologie. Proces párování je z uživatelského hlediska snadný, jelikož není nutná žádná instalace softwaru či ovladačů.
- **Energetická účinnost:** Další výhodou Bluetooth je jeho energetická účinnost, která vede k relativně nízké spotřebě energie. Efektivita byla navíc ještě zvýšena se standardem Bluetooth Low Energy (BLE). Z tohoto důvodu je technologie Bluetooth vhodná pro elektronická zařízení napájená z baterií.

Nevýhody

- **Přenosová rychlost:** Všechny bezdrátové technologie mají relativně pomalý přenos dat, což platí zejména v případě technologie Bluetooth. Bluetooth 3.0 a Bluetooth 4.0 má obecně přenosovou rychlost 25 Mb/s. Bluetooth proto není ideální pro přenos velkých souborů, jako jsou audio a videa.
- **Dosah:** Maximální dosah, jež Bluetooth nabízí, je 100 metrů. Bluetooth má obecně malý dosah komunikace (obvykle nižší než připojení Wifi). V závislosti na verzi, povaze zařízení a prostředí se dosah připojení Bluetooth značně liší.
- **Zabezpečení:** Přestože Bluetooth implementuje různé bezpečnostní mechanismy je úroveň zabezpečení Bluetooth podstatně nižší, zejména kvůli využívání rádiových frekvencí. Útočníci mohou signál snadno odchytit a dostat se tak k osobním informacím.
- **Kompatibilita:** Většina implementací Bluetooth je založena na standardu BR/EDR, ale stále mohou existovat problémy s kompatibilitou. Je to způsobeno mnoha důvody, jako je používaní odlišných profilů, ovladačů a verzí. Zejména technologie Bluetooth Low Energy není kompatibilní s předchozími verzemi.

3 ZigBee

Technologie ZigBee byla vyvinuta, aby poskytovala bezdrátové připojení s nízkou spotřebou pro širokou škálu síťových aplikací zabývajících se monitorováním a řízením. ZigBee je celosvětový otevřený standard spravovaný sdružením ZigBee Alliance. ZigBee PRO byl poté vyvinut jako vylepšení původního standardu ZigBee a poskytuje řadu dalších funkcí, které jsou zvláště užitečné pro velmi velké sítě (mohou zahrnovat stovky nebo dokonce tisíce uzlů). [46]

3.1 Historie

Tato kapitola shrnuje vývoj a nejzásadnější změny mezi jednotlivými specifikacemi standardu ZigBee. Ke zpracování byly využity zejména tyto zdroje: [8], [10], [47], [48].

3.1.1 IEEE 802.15.4

Standard 802.15.4 byl definován v roce 2003 a o tři roky později tedy v roce 2006 byl nahrazen novou verzí [49]. Jedná se o bezdrátovou osobní síť poskytující nízkou datovou rychlost s nízkými náklady, která je definována pro fyzickou a linkovou vrstvu referenčního modelu ISO/OSI. Fyzická vrstva pracuje v jednom z nelicencovaných rádio-frekvenčních pásmech 2,4GHz, 915MHz a 868MHz. Pásma 915MHz a 868MHz není možné v některých zemích využívat. V České republice je pásmo 915MHz vyhrazeno pro mobilní sítě [50]. Standard z roku 2003 poskytuje rychlosti 20 a 40 kbit/s v pásmu 868/915 MHz a 250 kbit/s v případě 2,4GHz pásma. Revize z roku 2006 zvýšila rychlost i v pásmech 868/915 až na 250 kbit/s.

3.1.2 ZigBee 2004/2006

Standard ZigBee je založen na standardu IEEE 802.15.4. První verze specifikace ZigBee byla vydána v roce 2004 a zrevidována byla v roce 2006. Je navržen pro menší digitální rádiová zařízení s nízkou spotřebou, která vytvářejí osobní síť (PAN). ZigBee 2006 pokrývá vzdálenosti od 10 do 100 metrů.

3.1.3 ZigBee PRO (2007)

Vylepšená verze ZigBee specifikace byla dokončena v roce 2007. ZigBee PRO oproti předchozí verzi používá jiný způsob routování, proto je limitována kompatibilita při použití v síti starší specifikace. Ačkoli ZigBee a ZigBee PRO jsou podobné technologie, existují mezi nimi rozdíly, které popisuje Tabulka 3.

Tabulka 3 - Porovnání ZigBee a ZigBee PRO [51]

Specifikace	ZigBee 2006	ZigBee PRO 2007
Rok vydání	2006	2007
Adresace	Založená na stromech	Stochastická
Routování	Tree	Mesh
Route Aggregation	Není podporováno	Ano
Řešení konfliktních ID sítě	Není podporováno	Ano
High security mód	Není podporováno	Ano

3.1.4 ZigBee PRO (2015/2017) neboli ZigBee 3.0

Jako ZigBee 3.0 jsou označovány zařízení, které plně podporují specifikaci ZigBee PRO z roku 2015, jež byla zrevidována v roce 2017. Tato nová specifikace přinesla spousty zásadních změn, díky nimž není kompatibilní se staršími verzemi.

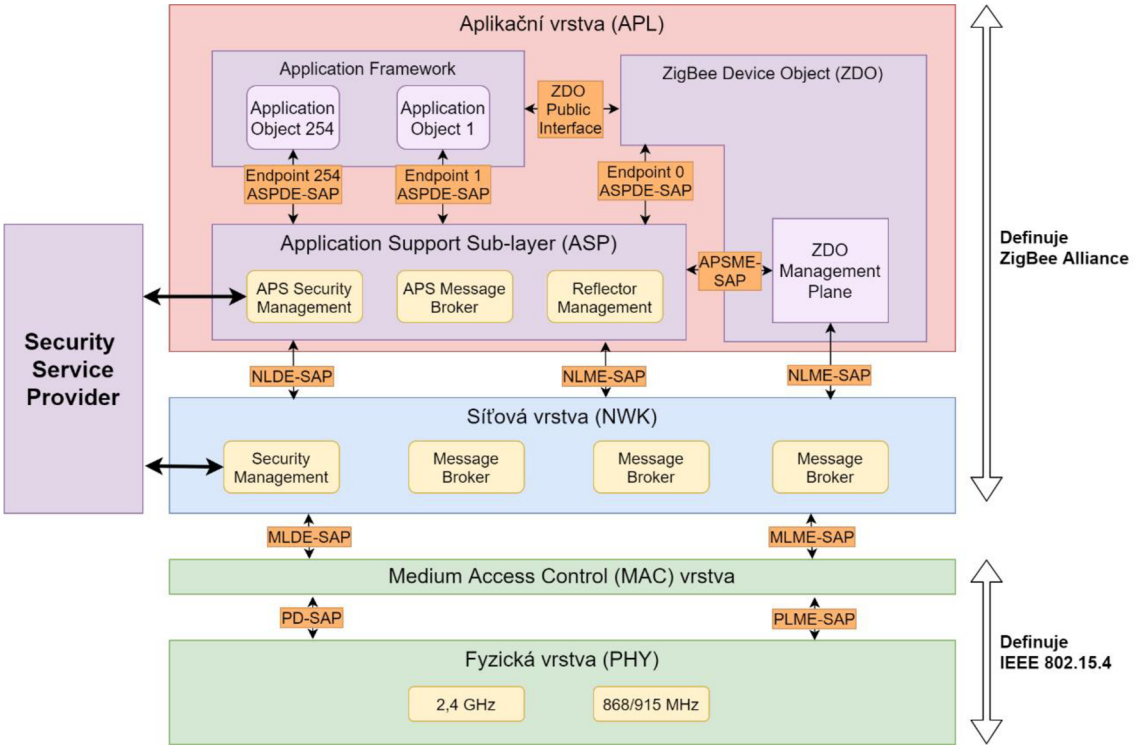
Specifikace ZigBee 3.0 přidává do sítí ZigBee správu dětských zařízení, vylepšené funkce zabezpečení a nové možnosti topologie sítě. Přidání zařízení do sítě bylo také vylepšeno a stalo se více konzistentní díky Base Device Behavior (BDB). Největší změnou je přidání Green Power technologie, díky níž mohou být nyní některá zařízení pro domácí automatizaci napájena řešeními pro získávání energie, což snižuje dopad baterií na životní prostředí. [52]

3.2 Architektura

Architektura protokolu ZigBee (viz. Obrázek 9) se skládá ze čtyř vrstev:

- Fyzická vrstva (PHY),
- Medium Access Control (MAC) vrstva,
- Síťová vrstva (NWK),
- Aplikační vrstva (APL).

Každá vrstva poskytuje sadu služeb vystavených pro vyšší vrstvy přes Service Access Point (SAP). Fyzická vrstva a vrstva MAC jsou definovány standardem IEEE 802.15.4, zatímco síťové a aplikační vrstvy jsou definovány standardem ZigBee. Tato kapitola byla zpracována dle [48], [53].



Obrázek 9 - Architektura ZigBee protokolu [48]

3.2.1 Fyzická vrstva

Fyzická vrstva operuje na dvou oddělených frekvenčních rozsazích: 868/915 MHz a 2.4 GHz. Fyzická vrstva je zodpovědná za vytváření paketů, příjem paketů, transparentnost údajů a řízení spotřeby.

3.2.2 MAC vrstva

Mezi povinnosti vrstvy MAC patří kontrola přístupu k rádiovému kanálu pomocí mechanismu CSMA-CA, přenos rámců beacon, synchronizace a zajištění spolehlivého mechanismu přenosu. Existují čtyři typy rámců MAC: datové rámce, rámce beacon, potvrzovací rámce a rámce příkazů MAC. Zabezpečení této vrstvy je založeno na standardu IEEE 802.15.4 rozšířeném o CCM (Cryptographic block Ciphers Mode), který poskytuje pouze funkce šifrování a integrity. CCM je vylepšené počítadlo se schématem šifrování provozu v režimu CBC-MAC. Horní vrstvy

nastavují výchozí klíč vrstvy MAC na klíč aktivního síťového klíče a klíč spojení vrstvy MAC na libovolný klíč spojení z horní vrstvy.

3.2.3 Síťová vrstva (NWK)

Síťová vrstva zajišťuje správný provoz podvrstvy MAC IEEE 802.15.4 a poskytuje vhodné servisní rozhraní aplikační vrstvě. Síťová vrstva je propojena s aplikační vrstvou prostřednictvím datové entity (NLDE) a management entity (NLME). NLDE generuje PDU na úrovni sítě, poskytuje směrování a zabezpečení specifické pro topologii. NLME konfiguruje nové zařízení, spouští síť, provádí připojení, opětovné připojení a opuštění síťové funkce, poskytuje možnosti adresování, zjišťování okolních zařízení, zjišťování tras, řízení příjmu a směrování.

Vrstva NWK je zodpovědná za kroky zpracování potřebné pro bezpečný přenos odchozích rámců a bezpečný příjem příchozích rámců. Mechanismus ochrany rámce vrstvy NWK používá pro autentizaci a důvěrnost Advanced Encryption Standard (AES) a CCM.

3.2.4 Aplikační vrstva (APL)

Aplikační vrstva se skládá ze ZigBee Device Objects (ZDO), Application Support sub-layer (APS) a aplikačního frameworku (Application Framework).

ZigBee Device Objects (ZDO)

ZDO jsou aplikace, které využívají primitiva vrstvy síťové a vrstvy APS k implementaci koncových zařízení ZigBee, routerů ZigBee a koordinátorů ZigBee. Poskytuje rozhraní mezi aplikačními objekty, profilem zařízení a APS. ZDO je odpovědný za inicializaci APS, NWK a poskytovatele bezpečnostních služeb (SSP).

Shromažďuje informace o konfiguraci z koncových aplikací, aby určil a implementoval zjišťování zařízení a služeb, správu zabezpečení (načítání klíčů, založení klíčů, přenos a ověřování klíčů), správu sítě (zjišťování sítě, připojení k síti, odpojení ze sítě, resetování síťového připojení a vytváření sítí), správa uzlů a skupin.

ZigBee Device Objects (ZDO) spravuje zásady zabezpečení a konfiguraci zabezpečení zařízení. Objekty uvedené jako povinné v příslušné specifikacemi jsou obsaženy ve všech zařízeních ZigBee.

Application Support sub-layer (APS)

APS poskytuje rozhraní mezi NWK a APL. Poskytuje služby pro navazování a udržování bezpečnostních vztahů. Služby jsou poskytovány prostřednictvím datové entity APS (APSDE) a APS management entity (APSME). APSDE poskytuje služby přenosu dat mezi aplikačními entitami. APMSE poskytuje bezpečnostní služby, párování zařízení a správu skupin.

Vrstva APS umožňuje, aby zabezpečení rámce bylo založeno na odkazových klíčích nebo síťovém klíči. Vrstva APS je zodpovědná za kroky zpracování potřebné k bezpečnému přenosu odchozích rámců, bezpečnému příjmu příchozích rámců a bezpečnému založení a správě kryptografických klíčů. Horní vrstvy řídí správu kryptografických klíčů vydáváním primitiv do vrstvy APS.

Aplikační framework

Aplikační framework je prostředí, ve kterém jsou hostovány aplikační objekty (lze jich definovat až 254). Obvykle se jedná o objekty aplikace definované výrobcem. Definuje aplikační profily (dohody pro zprávy, formáty zpráv a akce zpracování, které vývojářům umožňují vytvářet interoperabilní distribuovanou aplikaci využívající aplikační entity, které se nacházejí na samostatných zařízeních) a klastry.

3.3 Typy zařízení

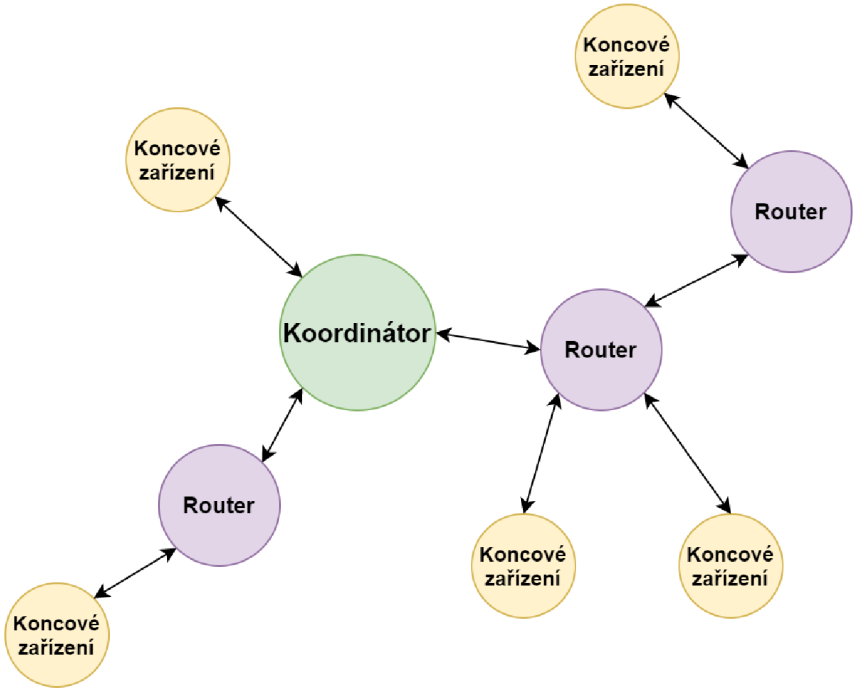
ZigBee zařízení jsou kombinací aplikačních, fyzických a logických zařízení. Aplikační zařízení jsou různé senzory, které shromažďují informace z okolního prostředí a dodávají informace koordinátorovi sítě.

Fyzická zařízení definovaná normou IEEE 802.15.4 jsou dvou typů, FFD (zařízení s plnou funkcí) a RFD (zařízení s omezenou funkcí). FFD je schopen vykonávat všechny funkce související se sítí, a proto jej lze použít jako koordinátora, router i koncové zařízení. Je zásobován nepřetržitým zdrojem energie, aby byl vždy vzhůru. RFD má omezené funkce a je obvykle napájen z baterie. RFD jsou v zásadě senzorové, které snímají různé parametry, nebo může být koncovým zařízení sloužícím ke shromažďování informací o síti od routeru či koordinátora.

Logická zařízení ZigBee rozděluje na tři typy (viz. Obrázek 10), které jsou hardwarově stejné, ale v rámci sítě mají specifickou roli. Těmito typy jsou:

- Koordinátor
- Router
- Koncové zařízení

Tato kapitola byla zpracována zejména dle [48], [52], [53], [54].



Obrázek 10 - Typy zařízení ZigBee PRO 2007 [53]

Koordinátor

Koordinátor je zařízení odpovědné za vytváření, provádění a celkovou správu sítě ZigBee. Je zodpovědný za konfiguraci úrovně zabezpečení sítě a konfiguraci adresy trust centra (centrum důvěry). Obvykle je trust centrem samotný koordinátor sítě. Koordinátor ovšem může určit alternativní trust centrum (v síti smí být pouze jedno trust centrum). Koordinátor také udržuje seznam aktuálně asociovaných zařízení a usnadňuje podporu osiřelého skenování (orphan scanning) a opětovného zpracování, aby umožnil dříve asociovaným zařízením znovu se připojit k síti. V každé síti je pouze jeden koordinátor, a proto jej nelze nikdy uvést do režimu spánku. Koordinátor také může podle potřeby vykonávat funkce routeru.

Router

Router je prostřední uzlové zařízení odpovědné za směrování paketů mezi koncovými zařízeními nebo mezi koncovým zařízením a koordinátorem. Pokud je v síti povoleno zabezpečení, router potřebuje povolení z trust centra, aby se mohl připojit síti. Router také může vykonávat funkce koncového zařízení. Při určitých příležitostech mohou routery ostatním routerům a koncovým zařízením umožnit připojení k síti a budou udržovat seznam aktuálně přidružených zařízení a usnadňovat podporu osiřelého skenování a opětovného připojení, aby bylo dříve připojeným zařízením umožněno znovu se připojit k síti. Vzhledem k tomu, že routery propojují více částí sítě, nelze je uvést do režimu spánku.

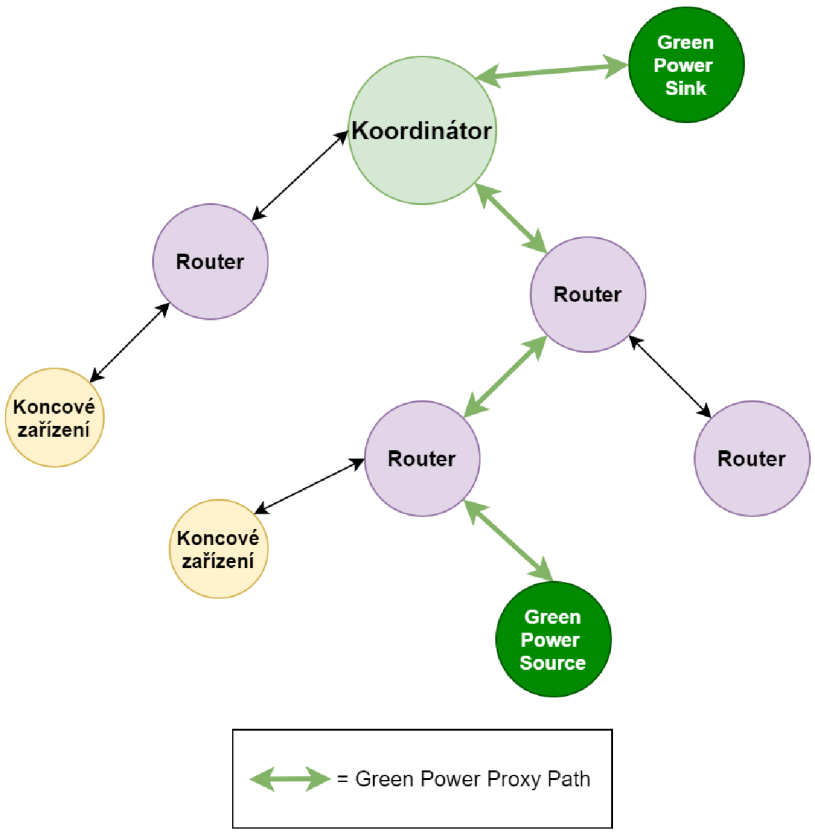
Koncové zařízení

Koncové zařízení je obvykle zařízení senzorového uzlu, které monitoruje a shromažďuje data prostředí. Koncová zařízení jsou na rozdíl od routerů nebo koordinátorů napájena z baterie a potřebují méně energie. Proto je lze po určitou dobu uvést do režimu spánku, aby se šetřila energie, pokud není monitorována žádná aktivita. Koncová zařízení nemohou směrovat provoz ani povolit ostatním uzlům připojit se k síti. Poslední dvě zmíněné vlastnosti ovšem platí za předpokladu, že router nebo koordinátor nevykonává také funkci koncového zařízení.

ZigBee 3.0 a Green Power

ZigBee 3.0 dále rozšiřuje typy zařízení o zařízení typu Green Power. Green Power je koncové zařízení, které snižuje spotřebu energie snížením počtu bitů, které protokol Green Power vyžaduje k výměně vzduchem. Toho je dosaženo optimalizací velikostí paketů, které musí uzel Green Power odeslat a snížením počtu paketů, které odesílá pro danou úlohu.

Ve specifikaci ZigBee PRO 2017 je dále uvedeno, že každé zařízení ZigBee 3.0 se směrovacími schopnostmi (router nebo koordinátor) musí pro zajištění dopředné kompatibility implementovat funkci Green Power Basic Proxy (GPBP) v1.1.1. GPBP umožňuje směrovacím zařízením tunelovat rámce Green Power Device Frames (GPDF) ze zařízení Green Power Source do zařízení Green Power Sink, což umožňuje funkci Green Power (viz. Obrázek 11) v jakékoli síti ZigBee 3.0, bez ohledu na vlastní aplikaci konkrétního zařízení.



Obrázek 11 - Zigbee 3.0 Green Power zařízení [52]

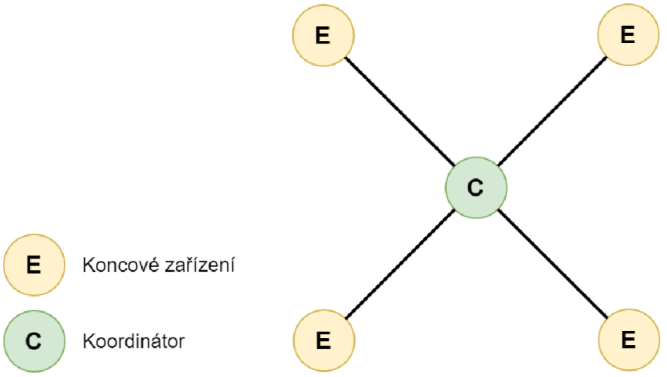
3.4 Topologie

IEEE 802.15.4 podporuje čtyři typy topologie sítě: hvězda (star), strom (tree), klastrový strom (cluster tree) a mesh. ZigBee používá pouze hvězdicovou, stromovou a mesh topologii. Kapitola byla zpracována dle [53], [54].

Hvězdicová topologie

Hvězdicová topologie má jednoho koordinátora a několik koncových zařízení. Koncová zařízení komunikují s koordinátorem. V této topologii není žádný router. Hlavní nevýhodou této sítě je, že pokud selže koordinátor, celá síť bude

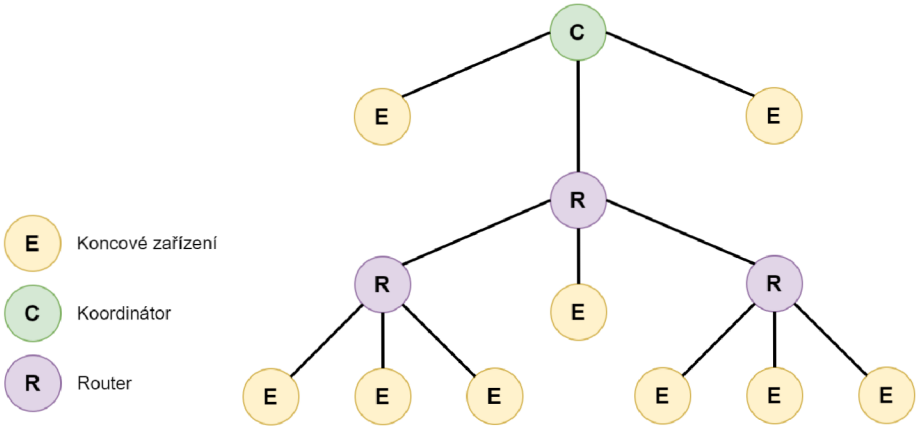
nefunkční, protože neexistuje žádná alternativní cesta od zdroje k cíli. Koordinátor se stává úzkým hrdlem této topologie.



Obrázek 12 - Hvězdicová topologie [54]

Stromová topologie

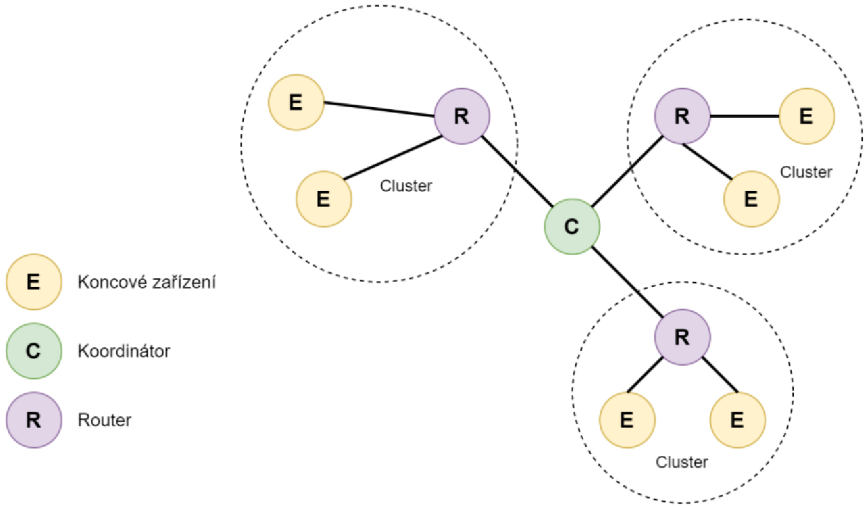
Topologie typu strom má koordinátora, router a koncová zařízení. Koncová zařízení se nazývají potomci. Koordinátor a směrovač se nazývají rodiče.



Obrázek 13 - Stromová topologie [54]

Klastrová stromová topologie

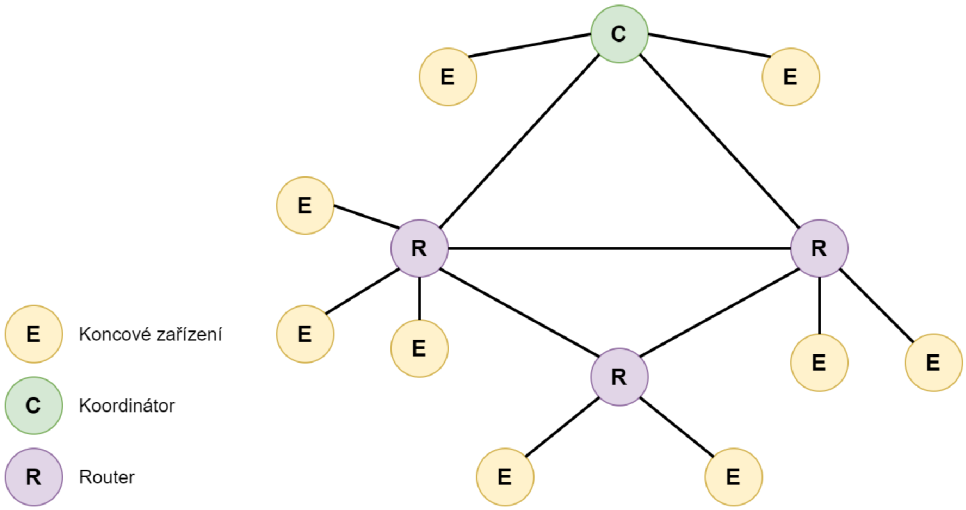
Rodič se stromem se nazývá klastr a celý klastr v této topologii je propojen pomocí ID klastru. Pouze IEEE 802.15.4 podporuje topologii klastrových stromů, ZigBee ji nepodporuje.



Obrázek 14 - Klastrová stromová topologie [54]

Mesh topologie

Mesh topologie umožňuje plnou peer-to-peer komunikaci. Mesh má jednoho koordinátora, více směrovačů pro rozšíření sítě a volitelná koncová zařízení. Koordinátor je odpovědný za vytvoření sítě a výběr určitých klíčových parametrů sítě. Tato topologie má funkci samo-léčení, díky němuž je nejméně zranitelné vůči selhání spojení. Vlastnosti topologie mesh jsou: jakékoliv zařízení může komunikovat s jakýmkoli zařízením, přidání nebo odebrání zařízení je snadné, odstranění mrtvé zóny, zvýšený dosah atd.



Obrázek 15 - Mesh topologie [54]

3.5 Zabezpečení

Pro zabezpečení ZigBee využívá 128bitový šifrovací algoritmus AES, který zahrnuje bezpečnostní služby jako vytváření klíčů, přenos klíčů, ochrana rámců a správa zařízení. Zabezpečení v ZigBee je charakterizováno jednoduchostí, přímostí a end-to-end zabezpečením. Každá vrstva je zodpovědná za zabezpečení rámce, který z ní pochází (jednoduchost), každý zdrojový a cílový uzel je odpovědný za přímou výměnu klíče (přímost) a data jsou přenášena bez nutnosti dešifrace a šifrace při každém skoku (end-to-end). V této části budou popsány bezpečnostní funkce, které nabízejí standardy ZigBee ve formátu zabezpečených rámců, a také bezpečnostní klíče. Kapitola byla zpracována dle: [47], [55].

ZigBee standart nabízí několik vestavěných bezpečnostní služeb pro zabezpečení komunikace prostřednictvím rámců mezi jednotlivými uzly:

1. Využívá symetrické šifrování k ochraně dat před zneužitím jinými stranami, které nedrží kryptografický klíč. Výměna a správa klíčů bude podrobněji představena v následující podkapitole.
2. ZigBee využívá algoritmus AES se šifrováním CCM (Cryptographic block Ciphers Mode). Při přenosu dat nabízí autentizaci i důvěrnost. ZigBee však zjednodušuje proces šifrování tím, že umožňuje opětovné použití stejného klíče na každé úrovni architektury ZigBee protokolu.
3. ZigBee zavádí kontrolu integrity zpráv (MIC), aby udržel integritu dat před modifikací zpráv jinými stranami. Tato služba může také zajistit, aby data pocházela z uzlu, který má kryptografický klíč.
4. ZigBee může předcházet forwarding útoku poskytnutím počítadla sekvenční aktuálnosti (sequential freshness counter) z pořadí rámců (vstupní nebo výstupní rámec). Pokaždé, když je odeslán nebo přijat nový rámec, se toto počítadlo vynuluje. V důsledku toho, pokud jakýkoli škodlivý uzel v síti přeruší komunikaci, aby mohl přeposlat některé z dříve přijatých nebo odeslaných rámců, lze jej snadno detekovat z počítadla. Tímto brání útočníkovi v tom, aby nutil síť hledat jinou cestu a odesílat pakety škodlivému uzlu.

5. Autentizaci lze řídit ve vrstvách NWK a APS pomocí aktivního síťového klíče (network key) a klíče spojení (link key). Díky tomu lze informace synchronizovat mezi zařízeními a současně poskytovat autentičnost prostřednictvím sdílených klíčů
6. Jednou z hodnotných služeb zabezpečení, které ZigBee využívá, je centrum důvěry (trust centrum). Spravuje nová zařízení integrovaná do sítě a pravidelně také aktualizuje sdílený klíč v síti. Koordinátor obvykle plní také úlohu trust centra a musí být známý pro všechny ostatní uzly v síti. Hlavními funkcemi trust centra je správa autentičnosti sdílených klíčů nových zařízení (distribuce) a povolení end-to-end zabezpečení mezi zařízeními nebo uzly.

Existují tři typy klíčů zabezpečení, které standard ZigBee využívá: hlavní (master key), síťový (network key) a klíč spojení trust centra (trust center link key). Každý klíč má různé funkce zabezpečení a lze jej sdílet mezi zařízeními různými technikami. Master klíč je předinstalovaný z výroby nebo nainstalován trust centrem. Tento klíč není určen k použití v šifrování, ale spíše se používá jako sdílené tajemství, když proces založení klíče provádějí dvě zařízení. Na druhou stranu je síťový klíč získán přenosem klíčů nebo před-instalací k zabezpečení vrstvy NWK. Unicast (jednosměrové) zprávy jsou prováděny klíčem spojení na aplikačních úrovních (APF, APS), kde jsou získávány buď prostřednictvím založení klíče, přenosem nebo před-instalací.

3.5.1 Výměna klíčů

Pro snadnější analýzu standardního bezpečnostního módu ZigBee protokolu bude proces výměny klíčů popsán po jednotlivých krocích.

Počáteční generování klíče

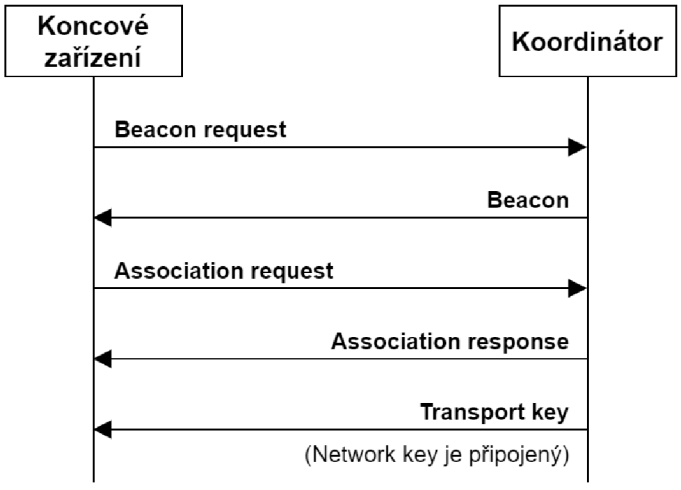
Každá síť ZigBee musí mít jednoho koordinátora a koordinátor musí vygenerovat síťový klíč při inicializaci sítě. Tento síťový klíč by měl být stejný pro celou síť a koordinátor by jej měl odeslat všem zařízením v síti. První fází je tedy generování síťového klíče.

Poté, pokud se koncové zařízení připojilo k síti, mu koordinátor odešle síťový klíč. Klíč však nebude odeslán jako prostý text. Koordinátor jej zašifruje pomocí před-konfigurovaného klíče spojení. Tento klíč má zařízení a je třeba jej sdílet s koordinátorem. Nyní může nastat několik situací. V ZigBee se před-konfigurovaný globální klíč používá jako výchozí klíč spojení. Je to jednoduché, protože v tomto procesu není nutné žádné sdílení klíčů. Koordinátor i zařízení ho znají, protože je globální. Ale jistě je zranitelný, jelikož ho zná i útočník.

Pro zajištění ochrany před touto zranitelností, je v ZigBee 3.0 použit ke sdílení jedinečného klíče spojení takzvaný instalační kód. Instalační kódy jsou 128bitová náhodná data a 16bitové kódy kontroly cyklické redundance (CRC), které jsou předávány prostřednictvím hash funkce Matyas-Meyer-Oseas (MMO) za účelem generování klíče spojení. Tento odvozený klíč je použit místo známého globálního klíče spojení. Instalační kódy jsou obecně pevně zakódovány do zařízení během výrobního procesu. Odpovídající instalační kód je poté součástí zařízení a je odeslán koordinátorovi prostřednictvím mimopásmové metody, jako je uživatelské rozhraní.

Připojení do zabezpečené sítě

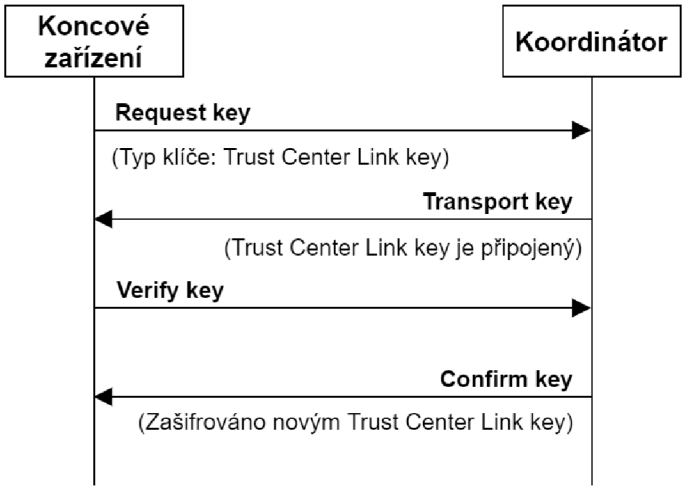
V síti ZigBee by všechna zařízení měla vlastnit síťový klíč, jakmile je dokončeno úplné připojení k síti. Za distribuci klíče je zodpovědný koordinátor. Síťový klíč se obvykle sdílí v okamžiku, kdy bylo koncovému zařízení povoleno připojit se k síti. Probíhající komunikaci pro připojení do zabezpečené sítě popisuje Obrázek 16.



Obrázek 16 - Připojení do zabezpečené sítě [47]

Spojované zařízení posílá broadcast beacon žádost (Beacon request) pro zjišťování sítě. Koordinátor poslouchá beacon žádosti a po jejich přijetí odešle rámec beacon spojovanému zařízení, který obsahuje informace o koordinátorovi jako je PAN ID, verze protokolu atd. Koordinátor použije zejména parametr Association Permit, který indikuje, zda se jiná zařízení mohou přidat do sítě. Pokud je tento parametr nastavený na *true*, pak spojované zařízení odešle žádost o připojení (Association request) koordinátorovi. Koordinátor pošle odpověď (Association response) s informací, zda byl pokus o připojení úspěšný. Nyní je zařízení připojeno, ale neautorizováno. Poté co se koordinátor rozhodne připojit zařízení do sítě, odešle koordinátor síťový klíč zařízení pomocí příkazu transportního klíče (Transport key). Jakmile zařízení obdrží síťový klíč, stává se autorizovaným zařízením uvnitř sítě.

V ZigBee je přidána další vrstva zabezpečení pomocí aktualizace klíče spojení trust centra. Když se zařízení úspěšně připojí k síti, požádá o nový klíč spojení trust centra, který je jedinečný pro koordinátora a konkrétní zařízení. Koordinátor používá tento klíč k šifrování všech zpráv odesílaných do zařízení v aplikační vrstvě a naopak. Proces komunikace popisuje Obrázek 17.



Obrázek 17 - Žádost nového klíče spojení Trust centra [47]

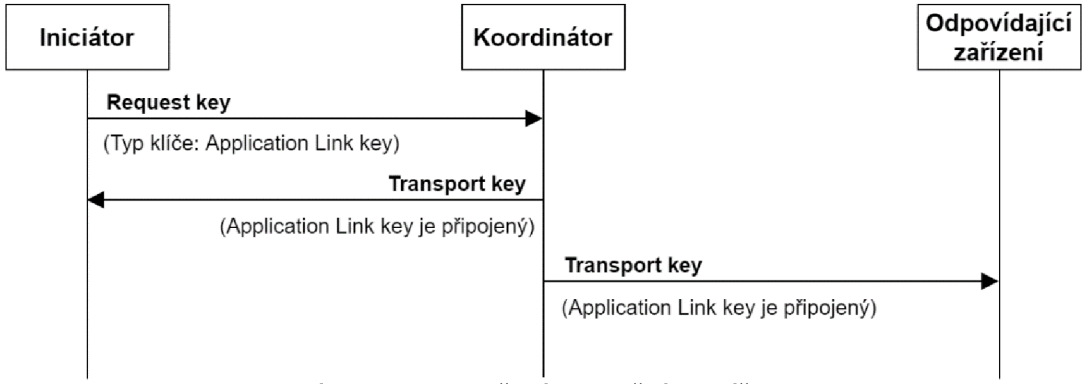
Poté, co se zařízení připojí k síti, požádá o nový klíč spojení trust centra zasláním příkazu klíče žádosti (Request key) koordinátorovi s odpovídajícím typem klíče. Koordinátor vygeneruje klíč spojení trust centra (Trust Center Link key) pro toto zařízení a odešle jej do zařízení pomocí příkazu transportního klíče (Transport key), který je zašifrován předchozím klíčem spojení (před-konfigurovaný klíč

spojení). Zařízení poté odešle ověřovací klíč APS (Verify key) koordinátorovi pro ověření nového klíče spojení trust centra. Koordinátor potvrdí klíč zašifrováním příkazu potvrzení klíče (Confirm key) novým klíčem spojení trust centra a jeho odesláním do zařízení.

Po všech uvedených výměnách zpráv začne zařízení používat nový klíč spojení trust centra a nahradí před-konfigurovaný klíč spojení pro veškerou následující komunikaci.

Vytvoření aplikačního klíče

ZigBee používá takzvaný aplikační klíč spojení (Application Link key) k podpoře zabezpečení aplikací typu end-to-end. K vytvoření klíče spojení mezi iniciačním zařízením a odpovídajícím zařízení je třeba zapojit trust centrum (koordinátor). Pořadí zpráv a probíhající komunikaci znázorňuje Obrázek 18.



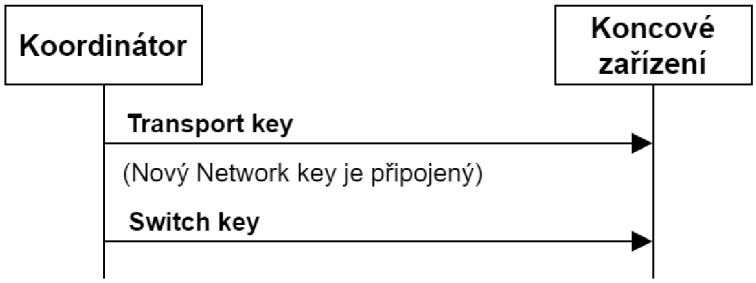
Obrázek 18 - Vytvoření aplikačního klíče [47]

Iniciační zařízení nejprve odešle příkaz klíče žádosti (Request key) koordinátorovi. V této zprávě musí iniciátor nastavit typ klíče (aplikační klíč spojení) a adresu partnera. Koordinátor vygeneruje klíč spojení pro tato zařízení a odešle jim je pomocí příkazu transportního klíče (Transpor key). Od té doby mohou tato dvě zařízení používat nově vygenerovaný aplikační klíč spojení k dosažení zabezpečené komunikace.

Aktualizace síťového klíče

ZigBee aktualizuje svůj síťový klíč pravidelně nebo podle potřeby. Aby se zabránilo replay útoku, síť ZigBee obsahuje 32bitové počítadlo rámců, které se zvyšuje při každém přenosu paketů. Pokud je dosaženo maximálního počtu tedy

0xFFFFFFFF, již nelze přenést žádný paket a musí dojít k aktualizaci síťového klíče, která resetuje počítadlo. Postup popisuje Obrázek 19.



Obrázek 19 - Aktualizace síťového klíče [47]

Koordinátor vygeneruje nový síťový klíč a pomocí příkazu transportního klíče (Transport key) jej vysílá do všech zařízení. Když všechna zařízení obdrží nový síťový klíč, koordinátor vyšle příkaz přepnutí klíče (Switch key). Po obdržení příkazu všechna zařízení začnou používat nový síťový klíč namísto předchozího.

3.6 Výhody a nevýhody

Výhody a nevýhody technologie ZigBee vycházejí zejména z jeho určení pro rozsáhlé senzorové sítě a jsou následující [56], [57]:

Výhody

- Je méně složitý než Bluetooth.
- Nastavení sítě je velmi jednoduché a snadné.
- Nemá centrální ovladač a zátěž je distribuována rovnoměrně po síti.
- Lze snadno bezdrátově monitorovat a ovládat domácí spotřebiče.
- Patent protokolu je bezplatný, tudíž nákladově efektivní.
- Síť je dobře škálovatelná a je snadné do ní přidat další zařízení ZigBee.

Nevýhody

- Jeho použití pro přenos osobních informací je velmi riskantní.
- ZigBee má nízkou přenosovou rychlost.
- Výměna za zařízení kompatibilní se ZigBee může být nákladná.
- Zatím není k dispozici mnoho koncových zařízení.
- Nelze jej použít jako venkovní bezdrátový komunikační systém, zejména díky omezenému dosahu.
- Není bezpečný jako systém založený na Wi-Fi.

4 Bluetooth vs ZigBee

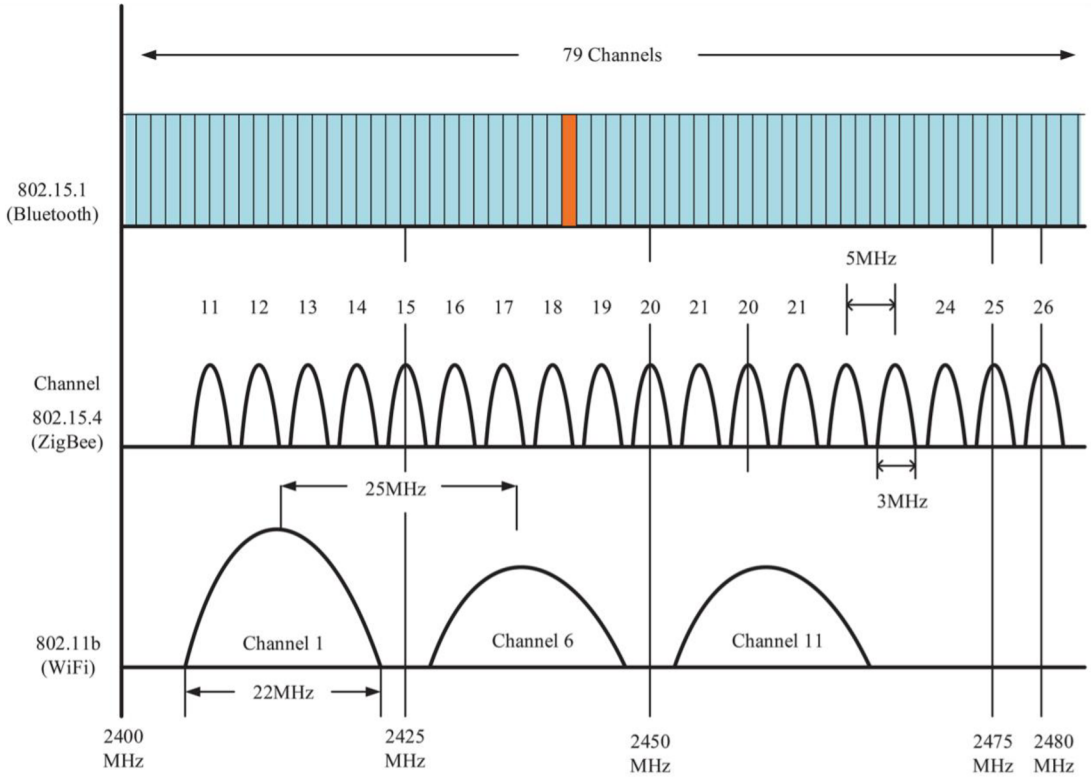
V předchozích kapitolách byly podrobně přestaveny bezdrátové technologie Bluetooth a ZigBee. V této kapitole budou tyto dvě technologie přímo porovnány. Základní parametry a jejich rozdíly popisuje Tabulka 4. Ke zpracování byly použity tyto zdroje: [58], [59], [60], [61], [62], [63].

Tabulka 4 - Porovnání základních parametrů Bluetooth a ZigBee [59], [60]

Standard	Bluetooth Classic	ZigBee
IEEE specifikace	802.15.1	802.15.4
Frekvenční pásmo	2.4GHz	868/915 MHz; 2.4 GHz
Maximální rychlost signálu	1 Mb/s	250 kb/s
Běžný dosah	10 m	10-100 m
Vysílací výkon	0-10 dBm	(-25) - 0 dBm
Počet RF kanálů	79	1/10; 16
Šířka kanálu	1 MHz	0.3/0.6 MHz; 2 MHz
Typ modulace	GFSK	BPSK (+ ASK), O-QPSK
Základní topologie	Piconet	Star
Rozšířená topologie	Scatternet	Mesh
Maximální počet uzlů	8	> 65000
Ochrana data	16-bit CRC	16-bit CRC
Kompatibilita zařízení	Pouze od stejného výrobce	Mezi všemi zařízeními podporující Bluetooth

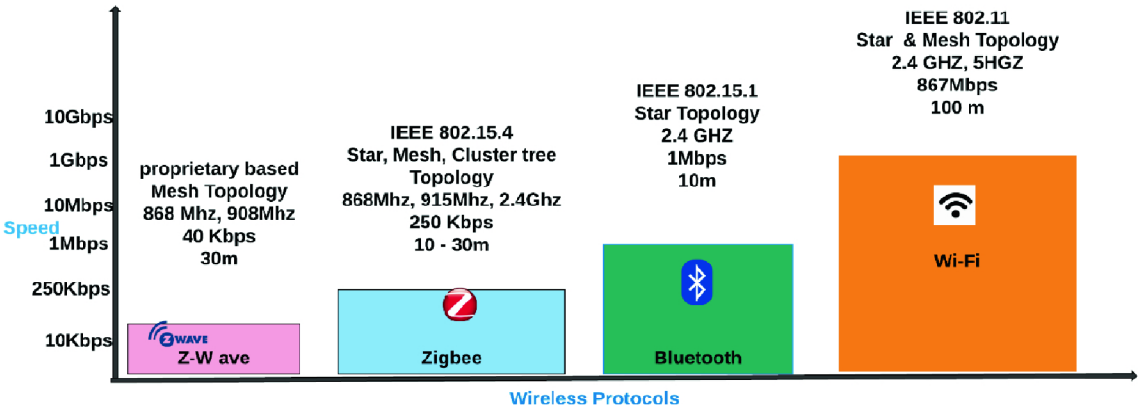
Většina bezdrátových technologií, jež se dnes běžně používají v oblasti smart home, využívá pro komunikaci bezlicenční pásmo 2,4GHz. Jak poznamenal Chen [58], důsledkem toho vzniká mezi-technologická interference (CTI), což významně ovlivňuje kvalitu a spolehlivost komunikace v tomto pásmu. V této oblasti má největší výhodu ZigBee, které nabízí komunikaci i v pásmech 868/915 MHz.

Škálovatelnost hraje významnou roli v budoucím rozšiřování sítě. ZigBee dovoluje vytvořit síť o více jak 65000 uzlech, což je výrazně více než 8 zařízení v piconetu poskytovaném Bluetooth do verze 4.0. S příchodem technologie Bluetooth Low Energy došlo k vyrovnání těchto technologií v oblasti velikosti sítě.



Obrázek 20 - Interference mezi Bluetooth, Zigbee a Wi-Fi [58]

Obě technologie byly vytvořeny zejména pro použití v přenosných zařízeních s limitovaným množstvím energie. Z toho důvodů je spotřeba energie bezdrátové technologie velmi důležitý faktor. ZigBee vždy platila za energeticky nenáročnou technologii oproti Bluetooth, s čímž souvisí i její nižší přenosová rychlost. Opět ovšem záleží na výběru mezi klasickým Bluetooth a BLE. BLE umí být podobně hospodárná jako ZigBee, daní je snížení přenosové rychlosti.



Obrázek 21 - Přenosová rychlost bezdrátových technologií [63]

Hlavní výhodou Bluetooth je jeho rozšíření, snadné používání a kompatibilita zařízení od různých výrobců. Bluetooth dnes má téměř každý mobilní telefon či

chytrý náramek. ZigBee s postupem času nabírá na popularitě, ale jeho používání je limitováno uzavřeným ekosystémem výrobců zařízení podporujících komunikaci skrze ZigBee.

S tím, jak se obě technologie vyvíjejí, nelze říct, která by byla lepší v případě použití v oblasti smart home. Výběr technologie velmi záleží zejména na potřebách uživatele. ZigBee je vhodná, pokud je cílem vytvořit rozsáhlou senzorovou síť s relativně nízkou spotřebou a cenou postavené na zařízeních od stejného výrobce. Bluetooth oproti ZigBee zaujme výrazně nízkou cenou, kompatibilitou zařízení od různých výrobců a snadným použitím.

5 Analýza zranitelností

V této kapitole budou přestaveny základní bezpečností principy používané v informačních systémech, dále budou přestaveny hrozby ve vztahu k technologiím Bluetooth a ZigBee.

5.1 Model CIA

Základem bezpečnosti informačních systémů je model CIA, který popisuje tři hlavní cíle informační bezpečnosti: důvěrnost (Confidentiality), integrita (Integrity) a dostupnost (Availability). Kapitola o modelu CIA byla zpracována dle [37], [55], [64], [65].

Důvěrnost

V dnešním světě je zásadní, aby lidé chránili své citlivé soukromé informace před neoprávněným přístupem. Určití lidé, zařízení nebo procesy by měli mít povoleno čí omezeno prohlížení dat, souborů a položek, jako jsou uživatelské jméno, kombinace hesel, lékařské záznamy atd. Důvěrnost se týká prohlížení dat nebo informací, protože pokud mají neoprávnění lidé přistup k těmto datům nebo informacím, mohou je zneužít dle libosti.

Ochrana důvěrnosti závisí na schopnosti definovat a vynutit určité úrovně přístupu k informacím. V některých případech je nutné rozdělení informací do různých kolekcí, které jsou organizovány podle toho, kdo potřebuje přístup k informacím a jak citlivé tyto informace jsou. Citlivost se obvykle vyjadřuje výší utrpěné škody, pokud by došlo k porušení důvěrnosti.

Bluetooth zajišťuje důvěrnost pomocí šifrování datových přenosů proudovou šifrou E0. Použitý tok klíčů je generován pomocí algoritmu, který jako vstup přijímá: adresu zařízení, náhodné číslo, číslo slotu a šifrovací klíč. Šifrovací klíč je generován interním generátorem klíčů. Proud klíčů používaný k šifrování každého datového paketu se mění na základě jednotlivých paketů, protože číslo slotu je pokaždé jiné, všechny ostatní proměnné zůstávají statické.

Také ZigBee využívá šifrování dat k zajištění důvěrnosti. K šifrování se používá algoritmus AES. ZigBee však zjednodušuje proces šifrování tím, že umožňuje opětovné použití stejného klíče na každé vrstvě architektury ZigBee.

Integrita

Kybernetická bezpečnost vyžaduje, aby data přenášená, zpracovávaná a uložená nebyla změněna z původní podoby, ať už náhodou nebo cíleně. Například pokud se změní jeden bit zprávy, může se změnit celá zpráva. Čímž může být celá zpráva poškozená nebo nečitelná.

Integrita je základní součást modelu CIA a zodpovídá za ochranu data před odstraněním nebo pozměněním od kteréhokoliv neoprávněného subjektu. Zároveň zajišťuje možnou obnovu dat v případě, že autorizovaná osoba provedla změnu, která neměla být provedena.

Základní zajištění integrity implementuje jak ZigBee, tak Bluetooth pomocí kontroly cyklické redundance (CRC) na konci každé zprávy. Obvykle se používá 16bitová varianta, ale například Bluetooth Low Energy využívá 24bitovou [66].

Dostupnost

Dostupnost garantuje, i přes všechny zavedené bezpečnostní opatření pro práci s hardwarem, softwarem, lidmi, procesy atd., oprávněným uživatelům vykonávat jejich práci, tak jak to potřebují. Je vyžadováno, aby oprávnění uživatelé mohli snadno přistupovat ke zdrojům, které potřebují k vykonávání své práce, a zároveň je nutné zajistit, aby byl systém schopný vyvážit zátěž v případě bezpečnostního incidentu či útoku.

Klasické Bluetooth poskytuje topologii typu hvězda, kdy centrální prvek (master) nesmí nikdy selhat, jinak selže celá síť. Řešením může být použití technologie Bluetooth Mesh, kde není pouze jedno zařízení, které je schopné fungovat jako master.

Koordinátor je nejdůležitějším uzlem sítě ZigBee. ZigBee standart podporuje více topologii včetně mesh, zároveň ovšem definuje, že koordinátor v síti může být právě jeden. Selháni koordinátoru tedy povede k selhání celé sítě.

5.2 Kategorizace hrozeb

Vyhláška č. 82/2018 Sb. [67] kategorizuje kybernetické bezpečnostní hrozby do čtyřech úrovní: nízká, střední, vysoká a kritická. Jednotlivé úrovně popisuje Tabulka 5.

Tabulka 5 - Kategorizace hrozeb [67]

Úroveň	Popis
Nízká	Zranitelnost neexistuje nebo je zneužití zranitelnosti málo pravděpodobné. Jsou zavedena bezpečnostní opatření, která jsou schopna včas detekovat možné zranitelnosti nebo případné pokusy o jejich zneužití.
Střední	Zneužití zranitelnosti je málo pravděpodobné až pravděpodobné. Jsou zavedena bezpečnostní opatření, jejichž účinnost je pravidelně kontrolována. Schopnost bezpečnostních opatření včas detekovat možné zranitelnosti nebo případné pokusy o překonání opatření je omezena. Nejsou známé žádné úspěšné pokusy o překonání bezpečnostních opatření.
Vysoká	Zneužití zranitelnosti je pravděpodobné až velmi pravděpodobné. Bezpečnostní opatření jsou zavedena, ale jejich účinnost nepokrývá všechny potřebné aspekty a není pravidelně kontrolována. Jsou známé dílčí úspěšné pokusy o překonání bezpečnostních opatření.
Kritická	Zneužití zranitelnosti je velmi pravděpodobné až víceméně jisté. Bezpečnostní opatření nejsou realizována nebo je jejich účinnost značně omezena. Neprobíhá kontrola účinnosti bezpečnostních opatření. Jsou známé úspěšné pokusy překonání bezpečnostních opatření.

Vyhláška také definuje sedmnáct hrozeb, z nichž jsou pro oblast smart home důležité zejména tyto:

- zneužití identity,
- škodlivý kód (například viry, spyware, trojské koně),
- zneužití nebo neoprávněná modifikace údajů,
- ztráta, odcizení nebo poškození aktiva,
- zneužití vnitřních prostředků, sabotáž,
- napadení elektronické komunikace (odposlech, modifikace).

5.3 Známé zranitelnosti

Tato kapitola představuje známé zranitelností technologií Bluetooth a ZigBee.

5.3.1 Bluetooth

Použitá verze Bluetooth je důležitá pro určení bezpečnostních hrozeb a zranitelnosti. Bezpečnost komunikace mezi zařízeními je tak silná jako nejslabší článek (tj. zařízení s nejstarší verzí). Vzhledem k tomu, že se dnes stále používá mnoho starších zařízení, zranitelnosti ve starších verzích Bluetooth existují i v současnosti. V této kapitole budou popsány zranitelnosti technologie Bluetooth po jednotlivých verzích dle [68].

Zařízení s verzí před Bluetooth 1.2

Klíče spojení, které jsou založeny na statických jednotkových klíčích, se používají pro párování a mohou být znovupoužity. Je-li klíč načten, mohou škodlivá zařízení odposlouchávat původní zařízení a také se za něj vydávat.

Zařízení s verzí před Bluetooth 2.1 + EDR

Je povoleno použití krátkých kódů PIN. Tyto PIN kódy mohou útočníci snadno uhodnout, díky jejich krátké délce. Tyto verze poskytují nedostatečnou správu PIN kódů, což je klíčová schopnost zabezpečení na podnikové úrovni. Další zranitelností je opakování pseudonáhodného proudu znaků po 23,3 hodinách. Tím se zvyšuje schopnost protivníka dešifrovat zprávy.

Verze 2.1 a 3.0

Pokud se zařízení v režimu zabezpečení 4 připojují k zařízením, která nepodporují režim zabezpečení 4, použijí se pro připojení dřívější režimy zabezpečení. Je například možné, že bude použit režim zabezpečení 1, který nenabízí žádné zabezpečení. Další zranitelností těchto verzí je použití statického klíče SSP, což zvyšuje zranitelnost zařízení vůči útokům typu Man-in-the-Middle.

Zařízení s verzí před Bluetooth 4.0

Existuje neomezený počet požadavků na výzvu k ověření, což umožňuje protivníkům získat přehled o tajných klíčích spojení. Kromě toho je funkce proudové šifry E0, která se používá v raných verzích, považována za slabou.

Všechny verze Bluetooth

Protivníci mohou zobrazit a případně upravit klíče spojení, pokud jsou uloženy nesprávně. Minimální délka šifrovacích klíčů není definována, takže klíče mohou být dlouhé i 1 bajt. Neexistuje žádné ověřování uživatelů. Standard Bluetooth zahrnuje pouze ověřování zařízení.

5.3.2 ZigBee

V této kapitole bude definováno několik zranitelností, které vyplývají z různých útoků a jejich následků dle [55].

Útoky se mohou zaměřit na proniknutí do sítě, aby ukradly legitimní přístupové údaje a odhalily tak důvěrnosti a autentizace. Tento typ útoku může být pasivní (protivník se snaží poškodit síť bez ohledu na to, jaký prostředek nasadí) nebo aktivní (protivník se zaměřuje na konkrétní privátní informace).

Dalším typem útoku, které ovlivňují protokoly řízení přístupu v ZigBee, je útok Man-in-the-Middle, kdy do sítě útočník zavádí falešné informace o směrování během zjišťování uzlů nebo tras. V důsledku toho může být ovlivněna integrita rámce i jeho autenticita.

Za určitých okolností se útočníci mohou pokusit odeslat falešný datový rámec, aby uvedli příjemce v omyl nebo vytvořili jiný typ útoků jako v DoS. Proto může být ovlivněna dostupnost, integrita a autenticita sítě ZigBee.

Další formou útoků je cílení na veřejné aplikační profily ZigBee. Očekává se, že ZigBee bude nasazen v mnoha službách, což vyžaduje překrývání v síťových klastrech (provozovaných jiným poskytovatelem služeb), aby se zachovalo pohodlí zákazníků. To by vyžadovalo zabezpečený protokol, který by umožňoval přístup dalším klastrům služeb. Příkladem útoku by bylo poškození síťového klastru služeb bez značek pro vystopování k zajištění odpovědnosti (neodmítnutí).

6 Praktická část

6.1 Stanovení cílů

Cílem praktické části diplomové práce bylo navrhnout bezpečnou komunikaci vhodnou pro smart home sítě s přihlédnutím k řešení bezpečnosti, dostupnosti řešení a jeho ceně. Navržené řešení bylo následně otestováno za pomoci malých jednodeskových počítačů platformy Arduino a Raspberry Pi.

6.2 Návrh zabezpečení

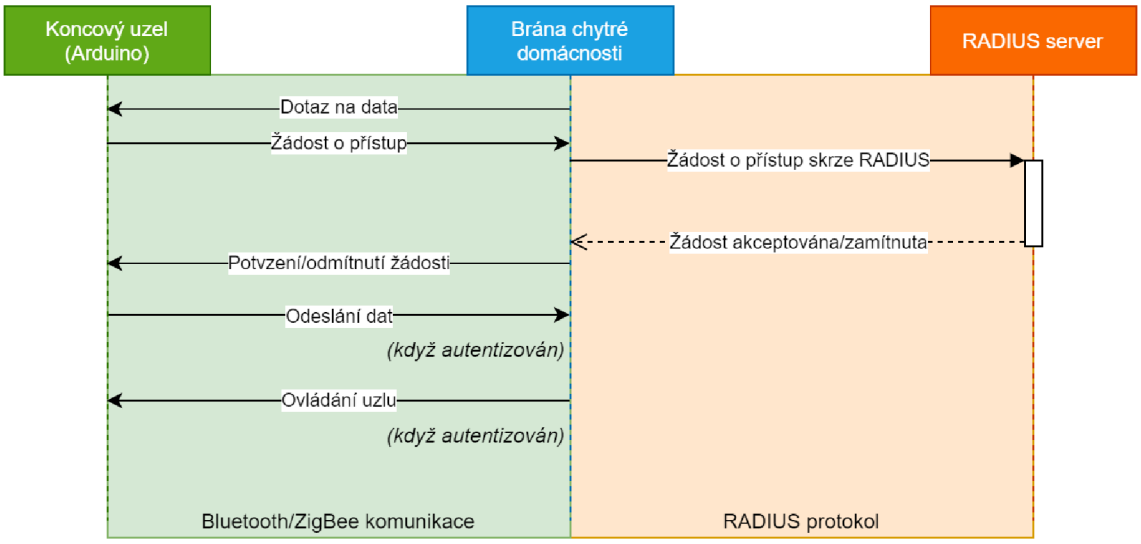
Bezdrátové komunikační technologie Bluetooth i ZigBee obsahují různé zranitelnosti, které mohou vést k porušení modelu CIA. Různí autoři nabízejí mnoho řešení, jak zvýšit bezpečnost ve smart home, základem je ovšem bezpečná autentizace jednotlivých prvků v síti.

Navrženo tedy bylo využít protokoly AAA (autentizace, autorizace a účtování) pro zvýšení bezpečnosti smart home sítě. Mezi protokoly AAA patří RADIUS, DIAMETER či TACACS.

Přestože DIAMETER je následníkem protokolu RADIUS, nebyl nikdy příliš rozšířen. Z tohoto důvodu byl pro potřeby diplomové práce zvolen protokol RADIUS. Protokol RADIUS poskytuje vysokou úroveň zabezpečení síťové komunikace. Funguje na principu klient/server, kdy klient slouží jako prostředník k autentizaci konkrétních uživatelů. Klíčovým bezpečnostním prvkem je sdílené tajemství, tedy informace, pomocí které je klient autentizován na serveru. Tato informace není nikdy přenášena po síti. Po síti jsou přenášena uživatelská jména a zašifrovaná hesla. [69]

6.2.1 Architektura a návrh komunikace

Uvažována byla architektura, kdy koncové uzly komunikují skrze bránu chytré domácnosti (brána smart home), která je zároveň klientem RADIUS serveru. Koncový uzel může přestavovat jakýkoliv prvek smart home jako je chytrá žárovka, či senzorový uzel poskytující informace o okolním prostředí. Brána umožňuje komunikaci s koncovými uzly přes domácí síť, případně může být dostupná i skrze internet. Navržený způsob komunikace popisuje Obrázek 22.



Obrázek 22 - Návrh komunikace smart home

Brána nejprve odešle požadavek na data koncovému uzlu. V případě neautentizovaného uzlu dojde k odeslání žádosti o přístup, který obsahuje přihlašovací údaje pro RADIUS server. Tyto údaje brána použije k vytvoření žádosti na RADIUS server. RADIUS server odešle výsledek žádosti (potvrzení či zamítnutí). Brána přepošle výsledek konkrétnímu uzlu.

V případě úspěšné autentizace koncový uzel může odesílat data a naopak brána může koncový uzel ovládat. Navržený způsob komunikace není závislý na použité bezdrátové technologii, je tedy aplikovatelný jak na Bluetooth tak na ZigBee.

6.2.2 Bezpečnostní mechanismy

Základním prvkem zvyšující bezpečnost je využití RADIUS serveru. Navrženy byly také bezpečnostní mechanismy, jejichž úkolem je bezpečnost dále podpořit.

Prvním mechanismem je využití časových limitů pro autentizaci, kdy po určitém čase dojde k zneplatnění žádosti o přístup odeslaný koncovým uzlem. S tímto souvisí také využití jednorázových bezpečnostních tokenů generovaných během odesílání žádosti o autentizaci a kontrolovaných po přijetí odpovědi. Cílem je znemožnit autentizace v případě odlišných tokenů. Dále také nutnost autentizace po vypršení nastaveného časového intervalu.

Předpokladem bezpečné komunikace je i odesílaní šifrovaných dat, zejména hesel, a proto je navrženo využití hashovacích funkcí pro komunikaci mezi koncovými uzly a bránou smart home.

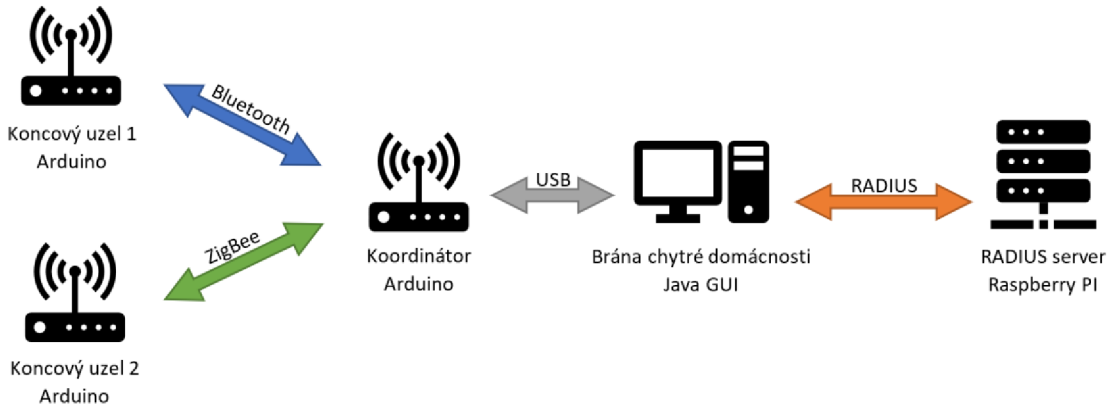
Posledním mechanismem je přidaní funkce, která na úrovni brány pozmění výsledné heslo, které bude porovnáno s databází hesel na straně RADIUS serveru.

6.3 Návrh prostředí a implementace

Tato kapitola popisuje navržené prostředí a konkrétní implementaci, jejímž cílem je ověření principu zabezpečení, který byl popsán v předchozí kapitole.

6.3.1 Schéma smart home sítě

Pro otestování navrženého konceptu byla připravena smart home síť dle schématu zobrazeném na Obrázek 23.



Obrázek 23 - Schéma navržené smart home sítě

Síť je tvořena koncovými uzly, které představují prvky chytré domácnosti, umožňující odesílat data o svém prostředí a reagovat na pokyny. Komunikace koncových uzlů je zajištěna pomocí bezdrátových technologií Bluetooth a ZigBee.

Pro propojení mezi bránou chytré domácnosti a koncovými uzly je použit koordinátor. Koordinátor komunikuje s koncovými uzly skrze bezdrátové technologie a slouží tak jako prostředník, umožňující bráně komunikaci s jednotlivými uzly. Koordinátor i koncové uzly jsou postaveny na platformě Arduino.

Koordinátor je propojen pomocí USB rozhraní s bránou chytré domácnosti, která je řídícím prvkem celé smart home sítě. Brána je hostována na stolním počítači a logiku obstarává Java aplikace poskytující grafické uživatelské rozhraní.

Poslední částí sítě je RADIUS server, který je hostovaný na počítači Raspberry Pi. RADIUS server komunikuje skrze protokol RADIUS s bránou chytré domácnosti a slouží jako autorizační autorita sítě.

6.3.2 Použité technologie

Arduino a Arduino IDE

Arduino je open-source elektronická platforma založená na snadno použitelném hardwaru a softwaru. Hardware je tvořen jednočipovými deskami, které pomocí vstupů, výstupů a dalších elektronických součástek ovlivňují a snímají své okolí. Arduino IDE je vývojové prostředí založené na jazyce Java pro psaní softwaru, který je možné pomocí tohoto prostředí nahrát na desku Arduino. [70]

Technologie Arduino byla využita pro vytvoření IoT prvků domácí sítě.

Programovací jazyk Java a framework Spring

Aplikace pro smart home bránu byla vytvořena v programovacím jazyku Java ve verzi 8, který již podporuje funkcionální programovaní pomocí lambda funkcí. Programovací jazyk Java je jedním z nejpoužívanějších objektově orientovaných jazyků, který vyvinula firma Sun Microsystems v roce 1995. [71]

Pro snadnější a rychlejší programování v jazyce Java byl využit framework Spring. Spring je nejpopulárnější Java framework. Základními vlastnostmi Spring frameworku spočívají v implementaci vzorů Inversion of Control a Dependency Injection, jež umožňují ponechat vytváření a provázání objektů na samotném frameworku. [72]

Vývojové prostředí IntelliJ IDEA a Maven

Pro tvorbu aplikace smart home brány bylo použito vývojové prostředí IntelliJ IDEA od společnosti JetBrains. Jedná se o komerční produkt vytvořený v jazyku Java, který je zaměřený na vytváření Java aplikací, i když podporuje i další programovací jazyky. [73]

Apache Maven je nástroj pro řízení softwarových projektů. Na základě konceptu projektově objektového modelu (POM) Maven spravuje sestavení aplikace z jednoho konfiguračního souboru. [74]

Raspberry Pi

Raspberry Pi je jednodeskový počítač o velikosti mobilního telefonu. Vznikl s cílem zpřístupnit informace o fungování počítačů zejména studentům informačních technologií. První verze byla představena nadací Raspberry Pi Foundantion v roce 2012. [75]

FreeRADIUS server

FreeRADIUS založili v červnu roku 1999 Miquel van Smoorenburg a Alan DeKok. První „alfa" verze kódu byla zveřejněna v srpnu 1999. Jedná se o velmi oblíbenou implementaci RADIUS serveru, která je poskytována pod licencí open-source. [76]

Wireshark

Wireshark je široce používaný analyzátor síťových protokolů. Umožňuje sledovat probíhající komunikaci uvnitř sítě na mikroskopické úrovni. Je pokračováním projektu zahájeného Geraldem Combsem v roce 1998. [77]

6.3.3 Použité komponenty

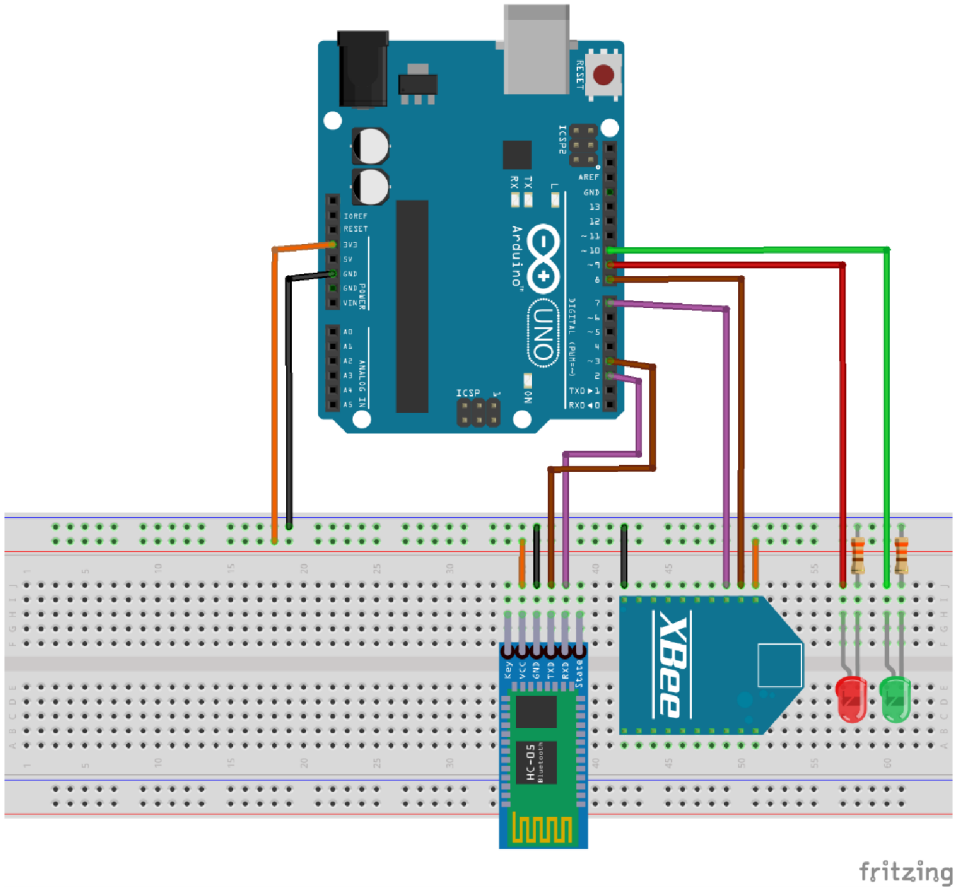
Pro stavbu smart home prvků byly použity následující komponenty:

- 1x Arduino Leonardo
- 1x Arduino UNO R3
- 1x NHduino UNO
- 1x Bluetooth modul HC-05
- 1x Bluetooth modul HC-06
- 2x XBee Series 1, 2.4 GHz
- 2x XBee adaptér do nepájivého pole
- 3x Nepájivé pole
- 4x LED dioda (2x zelená, 2x červená)
- 4x 330Ω rezistor
- 2x MCP 9700 (teplotní senzor)
- Propojovací/napájecí kabely

6.3.4 Zapojení HW komponent

Koordinátor

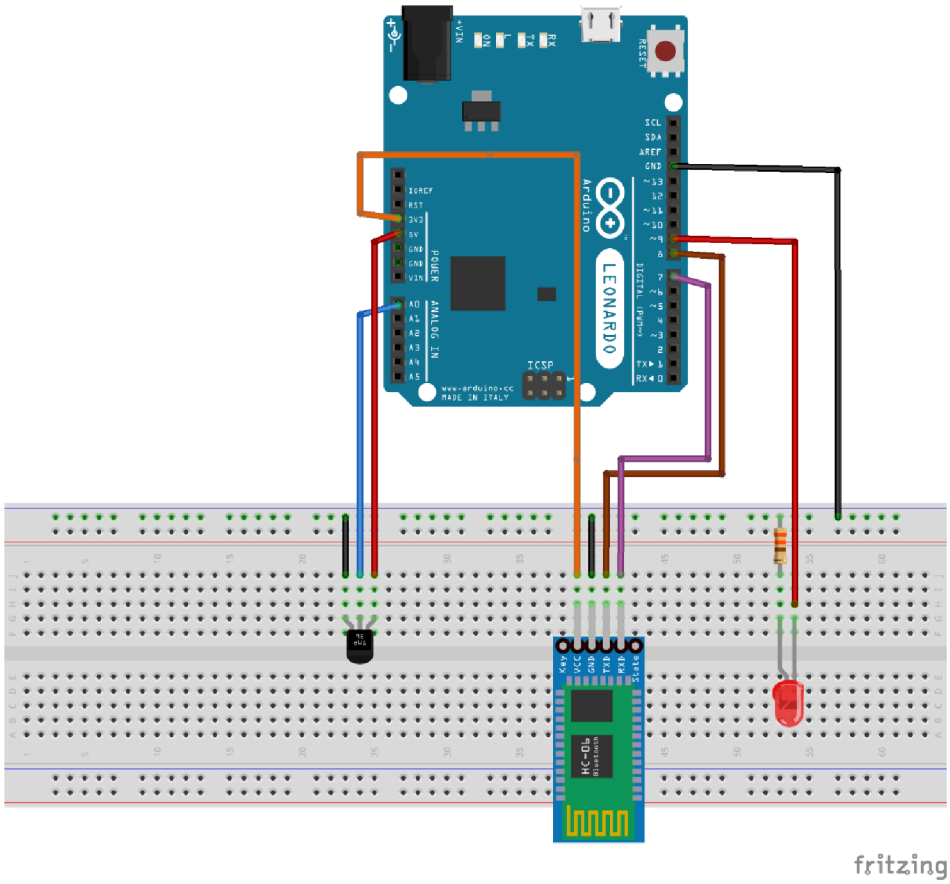
Řídícím prvkem koordinátoru je deska Arduino Uno R3. K desce je připojen modul XBee skrze digitální porty 7 a 8. Dále je k Arduinu připojen Bluetooth modul HC-05 přes porty 2 a 3, které jsou také digitální. Pro indikaci přenosu dat slouží dvě LED diody připojené k portům 9 a 10. Zelená dioda indikuje komunikace skrze XBee a červená skrze Bluetooth.



Obrázek 24 - Schéma zapojení koordinátoru

Koncový uzel Bluetooth

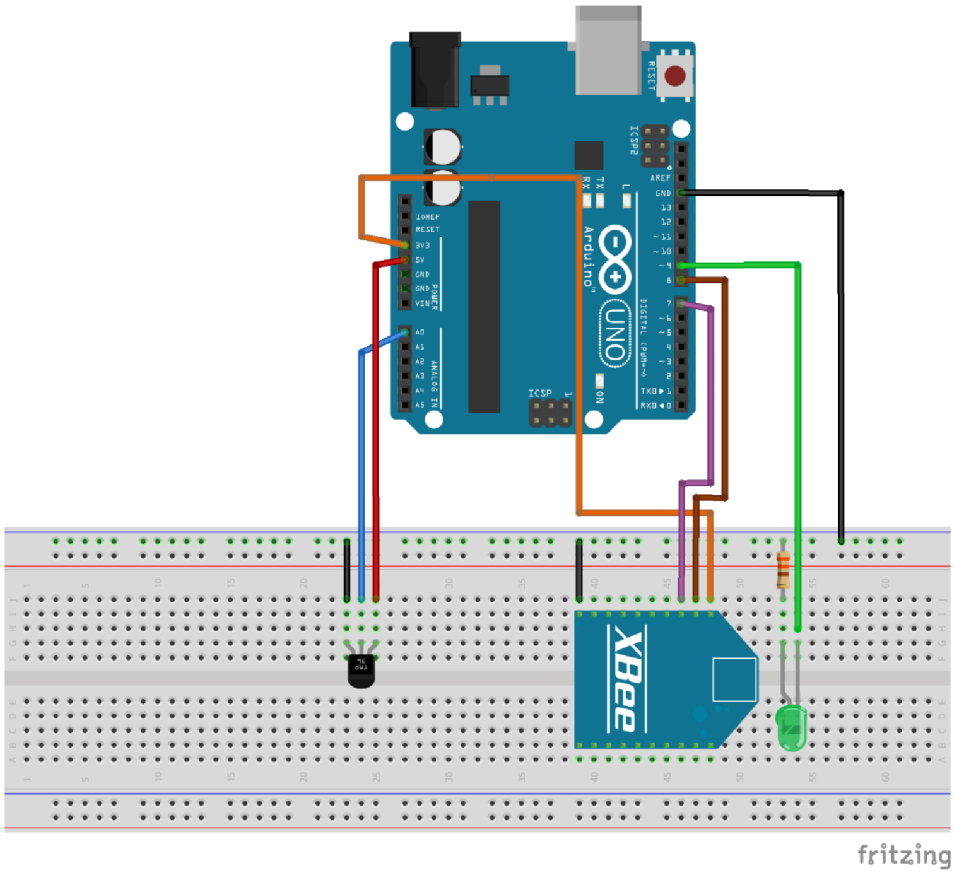
První koncový uzel je postaven na desce Arduino Leonardo a k bezdrátové komunikaci s koordinátorem využívá technologie Bluetooth. Tuto komunikaci zajišťuje Bluetooth modul HC-06, který je s deskou propojen digitálními porty 7 a 8. Pro demonstraci funkčnosti byl ke koncovému uzlu připojen teplotní senzor skrze analogový port A0 a také LED dioda, která je připojena k digitálnímu portu 9.



Obrázek 25 - Schéma zapojení koncového uzlu s Bluetooth

Koncový uzel ZigBee

Druhý koncový uzel využívá desky NHduino UNO, který poskytuje stejné vlastnosti jako originální Arduino UNO R3, avšak využívá čip CH340G, pro jehož správné fungování je vyžadována instalace správného ovladače [78]. Bezdrátovou komunikaci zde obstarává modul XBee pomocí technologie ZigBee a je připojen k digitálním portům 7 a 8. Opět je připojen teplotní senzor (port A0) a LED dioda (port 9) pro demonstraci přijímání a odesílání užitečných dat.



Obrázek 26 - Schéma zapojení koncového uzlu se ZigBee

6.3.5 Natavení modulů XBee a Bluetooth

V této části je popsáno nastavení jednotlivých bezdrátových modulů před použitím v navržené síti chytré domácnosti.

XBee

Moduly XBee byly nastaveny skrze USB adaptér v programu X-CTU [79]. Tabulka 6 zobrazuje upravené parametry oproti výchozímu nastavení.

Tabulka 6 - Moduly XBee a jejich nastavení

Parametry	Koordinátor	ZigBee uzel
ID	2244	2244
MY	1234	0002
DL	0	1234
CE	Coordinator	End device
EE	1 (Enabled)	1 (Enabled)
KY	1122334455	1122334455

Parametr ID je unikátní označení sítě, do které přísluší daný modul. Pro umožnění komunikace mezi moduly je nutné, aby byly součástí stejné sítě (stejné ID). Parametr MY přestavuje adresu zařízení o délce 16-bitů. Adresa zařízení je také zapsána v parametru DL, ovšem tato hodnota představuje adresu cílového zařízení pro komunikaci, obvykle se jedná o adresu koordinátora. Koordinátor může komunikovat s jakýmkoliv zařízením uvnitř sítě, a proto má tento parametr nastaven na výchozí hodnotu 0. Parametr CE určuje, zda modul pracuje jako koncový uzel nebo koordinátor, který může být v síti pouze jeden. Parametry EE a KY slouží k povolení šifrování (parametr EE) pomocí šifrovacího algoritmu AES a nastavení šifrovacího klíče (parametr KY). [80]

Bluetooth

Moduly pro Bluetooth komunikaci byly nejprve, před použitím v koncových uzlech, nastaveny pomocí AT příkazů.

Modul HC-06, který zastává roli komunikačního prvku na koncovém uzlu, může být pouze v roli slave, změněn tedy byl pouze PIN příkazem AT+PSWD4488.

Modul HC-05, který je součástí koordinátoru, byl nastaven jako master pomocí příkazu AT+ROLE=1. Následně byl nastaven inquiring mód, tak aby modul HC-5 mohl vyhledat ostatní Bluetooth zařízení, tedy modul HC-06. Pro spárování je nutné, aby zařízení měli stejný PIN, proto byl nastaven stejný i pro modul HC-05. Následně již došlo ke spárování a bindování se zařízením HC-06 příkazem AT+PAIR a AT+BIND. Dále byl modul HC-05 nastaven, tak aby se připojoval pouze ke spárovaným zařízením, příkazem AT+CMODE=1. Příkazem AT+LINK došlo k propojení těchto zařízení. Nastavení modulu HC-05 zajistilo, aby se tento modul spojil pouze se zařízením HC-06 ihned jakmile bude dostupné.

Tabulka 7 - Moduly Bluetooth a jejich nastavení

Parametry	HC-05	HC-06
ROLE	1 (Master)	0 (Slave)
PIN	4488	4488
BAUD_RATE	9600	9600

6.3.6 RADIUS server a jeho nastavení

RADIUS server je autorizační autoritou navržené sítě. Existují různé implementace s různými druhy licencí. S ohledem na snahu o snížení nákladů byla vybrána implementace FreeRADIUS ve verzi 3.0.21, která je licencována jako open-source a zároveň poskytuje všechny potřebné funkce pro zabezpečení smart home sítě.

FreeRADIUS byl nainstalován pomocí nástroje yum na Raspberry Pi 4B s operačním systémem Raspbian, který je jednou z linuxových distribucí. FreeRADIUS obsahuje dva základní konfigurační soubory sloužící pro autentizaci, jedná se o soubory „authorize“ a „clients.conf“.

Soubor „clients.conf“ obsahuje informace o RADIUS klientech. Roli klienta v navržené síti zastává brána chytré domácnosti. O každém klientovi musí být zaznamenána IP adresa a secret. Secret je sdílené heslo, které zná jak server, tak klient.

Soubor „authorize“ obsahuje informace o uživatelích, kteří mají být autentizováni, v tomto případě se jedná o koncové uzly. O každém uživateli musí být zaznamenáno přihlašovací jméno a heslo.

Pro ověření funkčnosti autentizačního procesu byl také upraven hlavní konfigurační soubor RADIUS serveru „radiusd.conf“. V tomto souboru bylo změněno nastavení logování tak, aby byl zapsán jakýkoliv pokus o autentizaci.

Všechny konfigurační soubory RADIUS serveru jsou součástí archivu v příloze 1.

6.3.7 Funkce koncových uzlů

V této kapitole bude podrobně popsán řídící program koncových uzlů a jeho funkcí. Řídící program koncových uzlů byl napsán v prostředí Arduino IDE a záměrně byl vytvořen tak, aby mohl být použit na obou koncových uzlech, tedy jak na uzlu komunikující skrze ZigBee, tak i komunikující skrze Bluetooth. Programy se od sebe liší pouze různým výchozím pojmenováním uzlu.

Program využívá knihoven SoftwareSerial, MD5 a SHA256. Knihovna SoftwareSerial umožňuje sériovou komunikaci skrze jiné digitální porty, než jsou k tomu určeny výrobcem a rozšiřuje možnosti desek, které neposkytují více

sériových linek. Knihovny MD5 a SHA256 poskytují hashovací funkce MD5, respektive SHA256.

Funkce setup() a loop()

Každý program pro Arduino obsahuje funkce setup() a loop(). Funkce setup() slouží k prvotní inicializaci Arduina. V případě koncových uzlů dojde ke spuštění sériové linky s rychlostí 9600 baudů, inicializaci pinu pro LED diodu a zahashování výchozího hesla. Dále je nastaven seed pro generátor pseudonáhodných čísel.

```
void setup() {
  remote.begin(9600);
  remote.println("Arduino zapnuto.");
  pinMode(LED_PIN, OUTPUT);
  hashPass = hashMD5(hashSHA256(password));
  randomSeed(analogRead(0));
}
```

Obrázek 27 - Funkce setup() koncového uzlu

Po skončení funkce setup() program pokračuje funkcí loop(), ve které je kód vykonáván v nekonečné smyčce. Funkce loop() koncových uzlů obsahuje funkci handleRemoteData(), jež zajišťuje bezdrátovou komunikaci a bude detailněji popsána v další části této kapitoly. Dále pak obsahuje časové validace. První je validace bezpečnostního odhlášení zařízení po uplynutí více jak jedné hodiny od přihlášení. Následuje aktualizace teploty z teplotního senzoru, která probíhá pouze jednou za deset vteřin, pomocí metody rightTemp(). Poslední je validace časového limitu pro přihlášení, který je dvacet vteřin.

```
void loop() {
  handleRemoteData();
  if (authorized && getTime() > timeAuthStart + authPeriod) {
    authorized = false;
  }

  if (getTime() > timeTempStart + tempPeriod) {
    timeTempStart = getTime();
    temp = rightTemp();
  }

  if (getTime() > timeAuthRequestStart + authRequestPeriod) {
    isAuthorizing = false;
  }
}
```

Obrázek 28 - Funkce loop() koncového uzlu

Hashovací funkce hashMD5() a hashSHA256()

Obě hashovací funkce přijímají jako vstupní parametr vstupní řetězec znaků typu String, který se následně zahashuje provoláním příslušné knihovny (MD5 nebo SHA 256). Výsledek se poté převede opět do objektu String a je vrácen na výstupu.

```
String hashMD5(String input) {
  char charPass[input.length() + 1];
  input.toCharArray(charPass, input.length() + 1);
  unsigned char* hash = MD5::make_hash(charPass);
  char* md5str = MD5::make_digest(hash, 16);
  return md5str;
}
```

Obrázek 29 - Funkce hashMD5() koncového uzlu

Funkce rightTemp()

Měření teploty zajišťuje funkce rightTemp(). Funkce získá data z teplotního senzoru, kterou následně převede na teplotu ve stupních celsia. Z důvodu získaní přesnějších výsledků a snížení odchylky senzoru je vypočtena průměrná teplota z pěti měření v intervalech po 50ms.

Funkce handleRemoteData()

Funkce handleRemoteData() zajišťuje příjem a interpretaci dat z bezdrátové komunikace. Pokud je uzel přihlášen, je úkolem zpřístupnit data z teplotního senzoru na vyžádání a také umožnit ovládání LED diody, či změnu hesla, kterou zajišťuje funkce changePassword(). V případě, že uzel ještě nebyl autentizován, pak je úkolem odesílání autentizační žádosti pomocí funkce sendAuthRequest() a ověření případné odpovědi, kdy se musí shodovat očekávaný ověřovací token s přijatým. Pokud jsou přijata neočekávaná data a uzel nebyl autentizován, pak jsou tato data ignorována a buffer je vyčištěn.

Funkce sendAuthRequest()

Úkolem funkce sendAuthRequest() je vytvoření a odeslání autentizační žádosti. Funkce nejprve získá náhodné čtyřmístné číslo pomocí funkce getRandomNumber(). Toto číslo je společně s názvem uzlu zahashováno pomocí MD5 a slouží jako ověřovací token. Samotná žádost pak obsahuje název uzlu, zahashované heslo a vygenerované číslo. Poté se již jen nastaví dvaceti vteřinový interval, po který bude platný ověřovací token žádosti.

```
void sendAuthRequest() {
  int randNum = getRandomNumber();
  authCRC = hashMD5(username + randNum);
  String authToken = String("auth" + DELIMITER +
  username + DELIMITER + hashPass + DELIMITER + randNum);
  remote.println(authToken);
  isAuthorizing = true;
  timeAuthRequestStart = getTime();
}
```

Obrázek 30 - Funkce sendAuthRequest() koncového uzlu

Funkce changePassword()

Změnu hesla obstarává funkce changePassword(). Vstupem je řetězec znaků obsahující staré a nové heslo. Po rozdělení vstupního řetězce je porovnáno staré heslo s heslem uloženém v koncovém uzlu. Pokud je heslo shodné, nové heslo je nastaveno a zahashováno. Vzhledem ke změně hesla je koncový uzel okamžitě odhlášen (nastavení příznaku „authorized“ na false).

```
boolean changePassword(String passChangeString) {
  int splitIndex = passChangeString.indexOf(DELIMITER);
  int endIndex = passChangeString.indexOf('\r');
  String oldPass = passChangeString.substring(0, splitIndex);
  String newPass = passChangeString.substring(splitIndex + 1, endIndex);
  if (oldPass.equals(password)) {
    password = newPass;
    hashPass = hashMD5(hashSHA256(password));
    authorized = false;
    return true;
  }
  return false;
}
```

Obrázek 31 - Funkce changePassword() koncového uzlu

6.3.8 Funkce koordinátora

Roli koordinátora zajišťuje deska Arduino s moduly XBee a Bluetooth. Hlavním úkolem řídícího programu Arduina je řízení komunikace mezi programem domácí brány a jednotlivými uzly.

Použitá deska Arduino Uno disponuje pouze jednou sériovou linkou, a proto byla opět využita knihovna SoftwareSerial. Byly vytvořeny dvě instance, jedna pro XBee modul a druhá pro Bluetooth modul. Výchozí sériová linka byla využita pro propojení skrze USB a komunikaci s programem domácí brány.

Funkce setup() a loop()

Funkce setup() inicializuje všechny tři sériové linky a notifikační diody. Knihovna SoftwareSerial neumožňuje souběžnou komunikaci na obou instancích, a tak funkce loop() postupně provolává funkce bluetoothLoop(), readDataFromGW() a xbeeLoop(). Tyto funkce přidělují čas jednotlivým linkám a zajišťují komunikaci s uzly a programem domácí brány, podrobněji budou popsány v dalších částech této kapitoly.

Funkce bluetoothLoop() a xbeeLoop()

Funkce bluetoothLoop() a xbeeLoop() obstarávají stejnou logiku, pouze provolávají různé sériové linky. Příslušná funkce nejdříve přepne komunikaci na danou linku a odešle znak ‚?', který slouží jako žádost o data z koncového uzlu. Poté se vykonávání programu pozastaví na jednu vteřinu, kdy se čeká na naplnění bufferu daty z uzlu, které jsou poté interpretovány provoláním funkce checkAndReadXbeeData() respektive checkAndReadBluetoothData()

```
void xbeeLoop() {
  xbee.listen();
  xbee.print('?');
  delay(1000);
  checkAndReadXbeeData();
}
```

Obrázek 32 - Funkce xbeeLoop() koordinátora

Funkce checkAndReadXbeeData() a checkAndReadBluetoothData()

Funkce checkAndReadXbeeData() a checkAndReadBluetoothData() nejprve kontrolují, zda jsou k dispozici nějaká data z uzlu. Pokud jsou data dostupná dojde k probliknutí příslušné notifikační diody a následně jsou data přečtena. K těmto datům je připojena informace o zdroji dat a data jsou přeposlána ke zpracování do programu domovské brány.

```
void checkAndReadXbeeData() {
  while (xbee.available() > 0) {
    digitalWrite(GREEN_LED_PIN, HIGH);
    delay(10);
    digitalWrite(GREEN_LED_PIN, LOW);

    while (xbee.available() == 0);
    String response = xbee.readString();
    Serial.print("XB_");
    Serial.print(response);
  }
}
```

Obrázek 33 - Funkce checkAndReadXbeeData() koordinátora

Funkce readDataFromGW()

Příchozí data z domácí brány jsou dále distribuována pomocí funkce readDataFromGW(). Funkce dle příznaku rozpozná, na který uzel má data přeposlat, poté přepne komunikaci na příslušnou linku, pro kterou pošle data. Funkce počká vteřinu na odpověď a následně provolá funkci pro kontrolu a čtení dat.

```
void readDataFromGW() {
  if (Serial.available()) {
    c =  Serial.read();
    if ('b' == c) {
      while (Serial.available() == 0);
      String text = Serial.readString();
      bluetooth.listen();
      bluetooth.println(text);
      delay(1000);
      checkAndReadBluetoothData();
    } else if ('x' == c) {
      while (Serial.available() == 0);
      String text = Serial.readString();
      xbee.listen();
      xbee.println(text);
      delay(1000);
      checkAndReadXbeeData();
    }
  }
}
```

Obrázek 34 - Funkce readDataFromGW() koordinátora

6.3.9 Implementace aplikace brány

Struktura projektu

Třídy byly z důvodu čitelnosti rozděleny do následujících částí:

- **config** – Balíček obsahující třídu s konfigurací pro Spring
- **gui** – Všechny třídy poskytující funkce pro uživatelské rozhraní jsou v tomto balíčku.
- **model** – Obsahuje modelové třídy.
- **service** – Třídy poskytující funkce pro komunikaci skrze sériovou linku, komunikaci s RADIUS serverem a bezpečnostní funkce jsou součástí balíčku service.

Konfigurační soubory pak lze nalézt v adresáři „resource“.

Použité knihovny

V této kapitole jsou popsány knihovny použité v aplikaci brány chytré domácnosti s využitím nástroje Maven.

- **TinyRadius** – TinyRadius je jednoduchá, malá a rychlá Radius knihovna psaná v jazyce Java schopná odesílat a přijímat RADIUS pakety všech typů. [81]
- **jSerialComm** – jSerialComm je knihovna poskytující přístup ke standardním sériovým portům bez nutnosti použití externích knihoven. [82]
- **logback-classic** – Knihovna logback-classic umožňuje snadno zaznamenávat události aplikace do souborů či konzole, dle použitých úrovní závažnosti (error, debug, info a trace). Další výhodou je možnost automatické rotace logů dle velikosti a času. [83]
- **spring-context** – Jak již bylo zmíněno v kapitole 6.3.2, aplikace brány používá Spring framework pro vývoj aplikací Java Enterprise, dnes označované jako Jakarta EE. [84]
- **guava** – guava je obsáhlá knihovna od společnosti Google poskytující mnoho různých utilit. V aplikaci byla využita, jelikož poskytuje implementace hashovacích algoritmů. [85]

- **junit a spring-test** – Knihovny junit a spring-test slouží ke snadné implementaci testů pro aplikace psané v jazyce Java.

Služby

Aplikace brány obsahuje tři hlavní třídy implementující logiku aplikace:

- **RadiusService** – Třída využívající knihovny TinyRadius pro implementaci RADIUS klienta komunikující s RADIUS serverem za účelem autentizace.

```
    private boolean sendAccessRequest(RadiusClient rc, String name, String password) {
        try {
            AccessRequest request = new AccessRequest(name, password);
            request.setAuthProtocol(AccessRequest.AUTH_CHAP);

            LOG.debug("Sending:\n" + request.toString());

            RadiusPacket reply = rc.authenticate(request);
            if (reply.getPacketType() == RadiusPacket.ACCESS_ACCEPT) {
                LOG.info("User '{}' was successfully authenticated", name);
                return true;
            }
        } catch (Exception e) {
            LOG.error("RADIUS authentication failed", e);
        }
        return false;
    }
}
```

Obrázek 35 - Ukázka kódu třídy RadiusService aplikace brány smart home

- **SecurityService** – Třída poskytující metody pro zabezpečení jako je vygenerování hesla koncového uzlu pro RADIUS server či vytvoření potvrzovacího tokenu po úspěšné autorizaci. Využívá knihovny guava pro použití hashovacích funkcí.

```
public String generateRadiusPassFromHash(String password) {
    int sum = 0;
    for (int i = 0; i < password.length(); i++) {
        int num = password.charAt(i);
        sum += num;
    }
    String hexSum = Integer.toHexString(sum);
    StringBuilder sb = new StringBuilder();
    sb.append(hexSum).append(password).append(hexSum).append(seed);
    return hashSHA256(sb.toString());
}
```

Obrázek 36 - Ukázka kódu třídy SecurityService aplikace brány smart home

- **SerialCommunicationService** – Veškerou komunikaci skrze sériovou linku obstarává třída SerialCommunicationService. Poskytuje funkce pro zjištění dostupných portů, následně umožňuje jejich připojení či odpojení. Poskytuje také veškeré funkce pro zápis a čtení dat.

Grafické rozhraní

Pro vytvoření základního grafické rozhraní byla využita knihovna Swing, která je součástí Java SE. Hlavní třída MainFrame je potomkem třídy JFrame a pomocí ní je vytvořeno základní okno grafického rozhraní. Ostatní třídy grafického rozhraní jsou rozděleny do jednotlivých balíčků, dle účelu následovně:

- **event** – Obsahuje třídy, které implementují listenery pro reakce na vzniklé události, jako je odeslání formuláře pro změnu hesla.
- **panel** – Balíček panel obsahuje jednotlivé panely, které tvoří celé okno aplikace. Poskytují formuláře, tlačítka a různé prvky pro zobrazení informací.
- **modal** – Obsahuje třídy pro vytvoření modálních oken.

Konfigurační soubory

Kromě konfiguračních souborů, které využívá vývojové prostředí či nástroj Maven, používá aplikace další dva konfigurační soubory:

- **gw.properties** – Obsahuje základní parametry, které mohou být specifické pro každou instanci brány smart home. Mezi tyto parametry patří IP adresa RADIUS serveru, secret (sdílené tajemství mezi aplikací a RADIUS serverem) a seed (náhodný řetězec znaků, pro zvýšení bezpečnosti hesel koncových uzlů).

```
gw.ip=192.168.0.189
gw.secret=deWVr4uJbh+LLw+u3jE-URwm$p3zAYFf
gw.seed=jdka2ewes
```

Obrázek 37 - Ukázka obsahu souboru gw.properties

- **logback.xml** – Jedná se konfigurační soubor, který používá knihovna logback-classic, umožňující nastavení logovací politiky, která je požadována.

6.3.10 Výměna a struktura dat

V této kapitole bude popsána struktura a formát posílaných dat mezi koncovými uzly, koordinátorem a bránou uvnitř sítě.

Žádost o data

O data z koncových uzlů žádá periodicky koordinátor odesláním znaku „?“ na příslušný uzel.

Autentizace

Pokud není koncový uzel autentizován, odešle po přijetí žádosti o data autorizační žádost. Autorizační žádost se skládá z parametrů dle Tabulka 8, které jsou součástí jednoho řetězce odděleny podtržítkem.

Tabulka 8 - Parametry autorizační žádosti odesílané koncovým uzlem

Parametr	Délka	Popis
auth	4 znaky	Jedná se o pevně stanové označená autentizační žádosti.
Název	až 18 znaků	Parametr obsahující název koncového uzlu, tak jak je zaregistrován v RADIUS serveru.
Heslo	32 znaků	Heslo uzlu, které bylo nejprve zahashováno pomocí SHA256, ovšem kvůli knihovně SoftwareSerial, která limituje velikost bufferu na 64 znaků, je heslo ještě zahashováno pomocí MD5.
Náhodné číslo	4 znaky	Parametr obsahující náhodné číslo mezi 1000 a 9999. Používá se k následnému ověření autorizačního tokenu.

V případě, že autorizace proběhne v pořádku, smart home brána vygeneruje potvrzovací token. První znak je vždy „A“ a označuje potvrzení autorizace. Následuje 32 znaků, jedná se o MD5 hash spojení názvu uzlu s náhodným číslem, které bylo odesláno v autentizační žádosti. Stejný hash si vygeneroval i koncový uzel během odesílání žádosti. Přijatý a poznamenaný hash jsou porovnány a pokud jsou stejné, uzel je autentizován a může začít odesílat data a přijímat příkazy.

Změna hesla

Změna hesla může být provedena pouze pokud je uzel autentizován. První znak je vždy „p“, který značí, že je očekávána změna hesla. Následuje původní heslo a nové heslo. Obě hesla jsou rozdělena podtržítkem. O výsledku změny hesla je odeslána informace zpět do brány.

Ovládání LED diody uzlu

Vypnutí diody je provedeno po přijetí znaku „0“. K zapnutí diody dojde po přijetí znaku „1“. Tyto funkce jsou zpřístupněny pouze autentizovaným uzlům.

Odeslání teploty z uzlu

Teplota je odesílána pouze po přijetí žádosti o data a pouze pokud se uzel nachází v autentizovaném stavu. Teplota je odesílána ve formátu „temp_“ + teplota ve stupních celsia s přesností na dvě desetinná místa.

6.4 Ověření funkcionality

Mechanismus zabezpečení a základní funkcionalita navrženého prostředí, které byly popsány v kapitolách 6.2 a 526.3, byly otestovány pomocí aplikace brány smart home a jejich logů. Pro kontrolu RADIUS paketů byl využit program Wireshark.

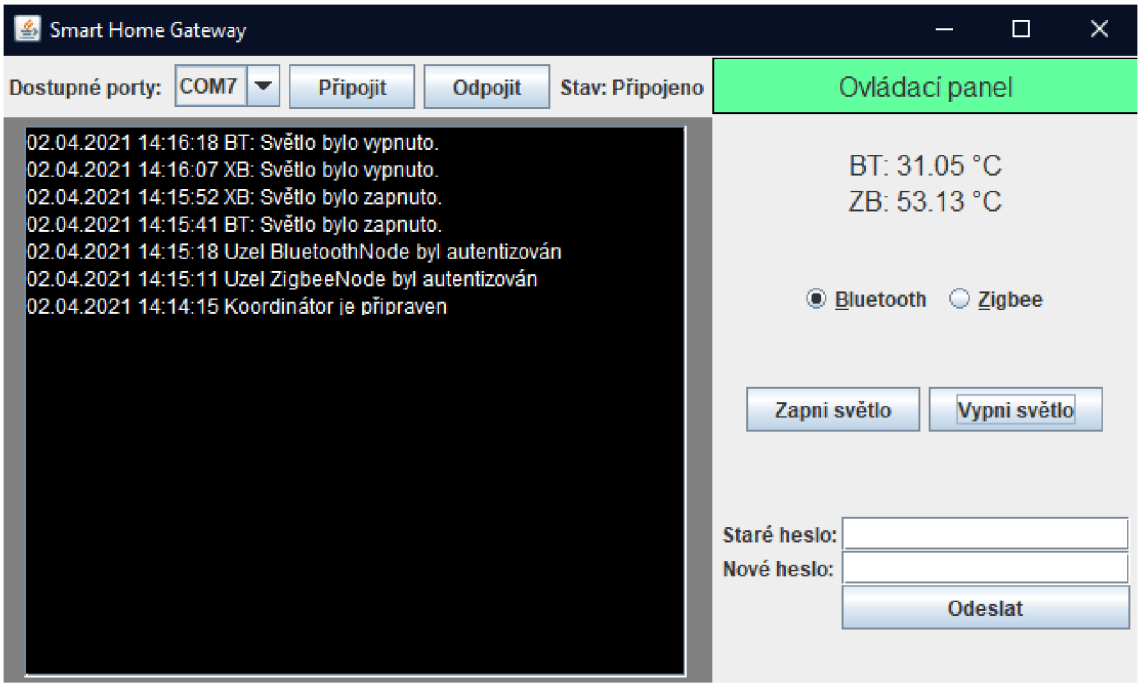
6.4.1 Popis aplikace brány chytré domácnosti

Grafické rozhraní je rozděleno na tři sekce.

První sekce obsahuje prvky pro připojení koordinátora skrze USB port, jedná se o seznam dostupných portů, tlačítka pro připojení a odpojení vybraného portu a informaci o stavu připojení.

Druhá sekce zobrazuje důležité informace o tom, co se děje v síti. Tyto informace mají časové razítko a jsou řazeny od nejaktuálnější po nejstarší.

Poslední sekcí je panel pro správu jednotlivých koncových uzlů. Tento panel obsahuje informace o přijatých datech (teplotě), dále pak tlačítka pro zapnutí a vypnutí LED diody. Posledním prvkem je formulář pro změnu hesla. Tlačítka typu radio určují, se kterým uzlem mají být provedené akce pomocí tlačítek.



Obrázek 38 - Ukázka grafického rozhraní aplikace

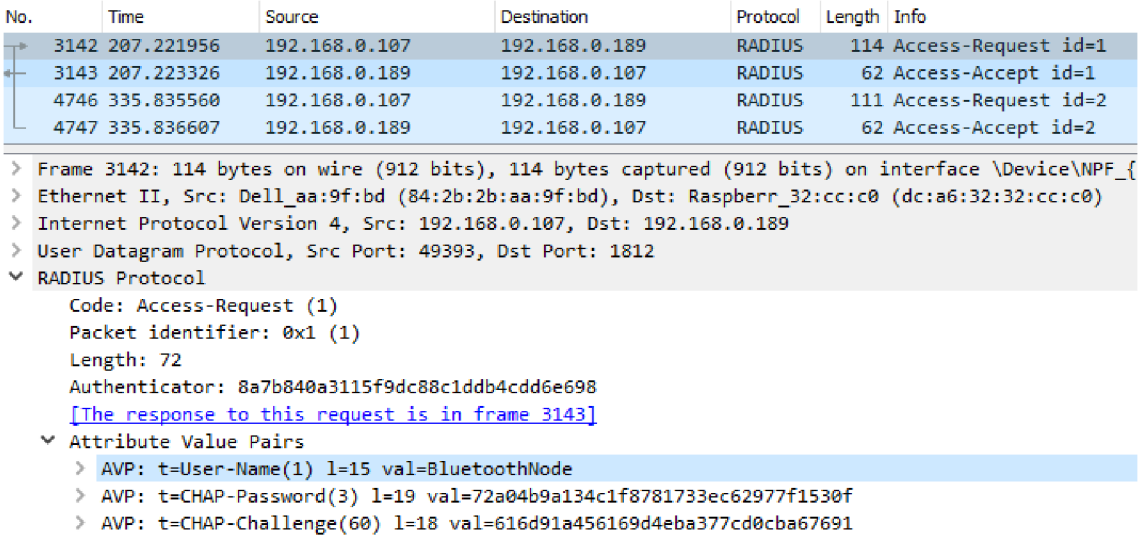
6.4.2 Test autentizace

Základním testovacím scénářem je automatická autentizace uzlů po připojení. Po dotazu od koordinátora, uzel odešle autentizační žádost, která je přeposlána na bránu chytré domácnosti. Zde dojde k úpravě a finálnímu odeslání autentizační žádosti na RADIUS server. V případě úspěšné autentizace dojde k odeslání autentizačního tokenu zpět koncovému uzlu.

```
12:15:08 [Thread-2] DEBUG cz.uhk.iot.service.SerialCommunicationService Koordinátor je připraven
12:15:11 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService
XB_auth_ZigbeeNode_e0b5e3679b0681493f77d56cd7429a72_4599
12:15:11 [Thread-3] DEBUG cz.uhk.iot.service.RadiusService Sending:
Access-Request, ID 1
User-Name: ZigbeeNode
12:15:11 [Thread-3] INFO  org.tinyradius.util.RadiusClient send Access-Request packet: Access-Request, ID 1
User-Name: ZigbeeNode
12:15:11 [Thread-3] INFO  org.tinyradius.util.RadiusClient received packet: Access-Accept, ID 1
12:15:11 [Thread-3] INFO  cz.uhk.iot.service.RadiusService User 'ZigbeeNode' was successfully authenticated
12:15:11 [Thread-3] INFO  cz.uhk.iot.service.SerialCommunicationService Writing data
'xAa44092d6bf6c3069ed27745b65aef0b6'
12:15:18 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService
BT_auth_BluetoothNode_e0b5e3679b0681493f77d56cd7429a72_3692
12:15:18 [Thread-3] DEBUG cz.uhk.iot.service.RadiusService Sending:
Access-Request, ID 2
User-Name: BluetoothNode
12:15:18 [Thread-3] INFO  org.tinyradius.util.RadiusClient send Access-Request packet: Access-Request, ID 2
User-Name: BluetoothNode
12:15:18 [Thread-3] INFO  org.tinyradius.util.RadiusClient received packet: Access-Accept, ID 2
12:15:18 [Thread-3] INFO  cz.uhk.iot.service.RadiusService User 'BluetoothNode' was successfully
authenticated
12:15:18 [Thread-3] INFO  cz.uhk.iot.service.SerialCommunicationService Writing data
'bA8a4395f7c207add52b16764925b65fc6'
12:15:23 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_temp_55.96
12:15:25 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_temp_23.34
```

Obrázek 39 - Log aplikace při autentizaci

Obrázek 39 znázorňuje autentizaci dvou uzlů. Nejprve odeslal svou žádost uzel ZigbeeNode. Po kontaktování RADIUS serveru byl přijat paket s pozitivním výsledkem autentizace (Access-Accept). Na základě úspěšného ověření na straně RADIUS serveru aplikace vygenerovala potvrzovací token a odeslala jej na koncový uzel. Dalším, kdo žádal o autentizaci, byl uzel BluetoothNode. Také tato žádost byla úspěšná a uzel byl autentizován. Tato skutečnost byla dokázána tím, že uzly následně začaly odesílat informace o teplotě. Jednotlivé RADIUS pakety zachytil také nástroj Wireshark viz. Obrázek 40.



Obrázek 40 - Úspěšná autentizace z pohledu Wiresharku

V případě neúspěšné autentizace na straně RADIUS serveru dojde k odeslání paketu Access-Reject. O této skutečnosti je informována pouze aplikace brány, která již dále s koncovým uzlem nekomunikuje a čeká na další žádosti.

```
Fri Apr  2 14:15:11 2021 : Auth: (2) Login OK: [ZigbeeNode] (from client pc port 0)
Fri Apr  2 14:15:18 2021 : Auth: (3) Login OK: [BluetoothNode] (from client pc port 0)
Fri Apr  2 14:19:55 2021 : Auth: (4) Login incorrect (chap: Password comparison failed:
password is incorrect): [BluetoothNode/<via Auth-Type = CHAP>] (from client pc port 0)
```

Obrázek 41 - Autentizace z pohledu RADIUS serveru

6.4.3 Test komunikace aplikace s uzly

Dalším testem bylo ověření komunikace již autentizovaných uzlů a aplikací brány. Pro demonstraci uzly poskytují informaci o teplotě a aplikace může ovládat LED diodu na konkrétním uzlu.

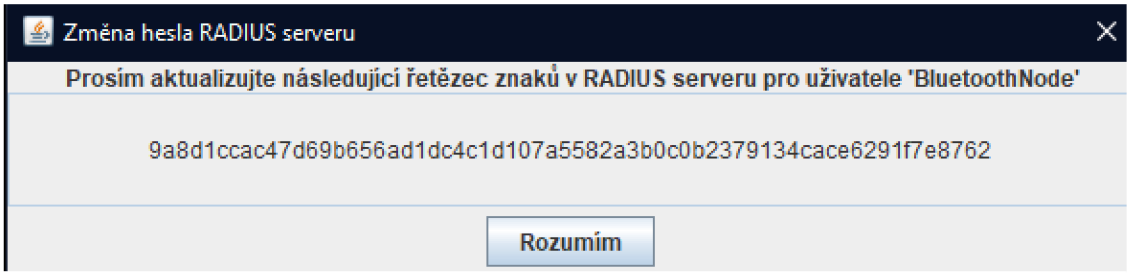
```
12:15:35 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_temp_23.14
12:15:36 [AWT-EventQueue-0] INFO  cz.uhk.iot.service.SerialCommunicationService Writing data 'b1'
12:15:38 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_temp_68.07
12:15:41 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_Světlo bylo zapnuto.
12:15:43 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_temp_23.14
12:15:46 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_temp_58.40
12:15:47 [AWT-EventQueue-0] INFO  cz.uhk.iot.service.SerialCommunicationService Writing data 'x1'
12:15:48 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_temp_30.66
12:15:52 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_Světlo bylo zapnuto.
12:15:54 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_temp_60.35
12:15:56 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_temp_30.86
12:15:59 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_temp_60.35
12:16:01 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_temp_30.86
12:16:04 [AWT-EventQueue-0] INFO  cz.uhk.iot.service.SerialCommunicationService Writing data 'x0'
12:16:04 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_temp_57.71
12:16:07 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_Světlo bylo vypnuto.
12:16:09 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_temp_30.66
12:16:12 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_temp_57.71
12:16:13 [AWT-EventQueue-0] INFO  cz.uhk.iot.service.SerialCommunicationService Writing data 'b0'
12:16:14 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_temp_30.66
12:16:18 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService BT_Světlo bylo vypnuto.
12:16:20 [Thread-3] DEBUG cz.uhk.iot.service.SerialCommunicationService XB_temp_53.03
```

Obrázek 42 - Log aplikace během komunikace s uzly

Obrázek 42 znázorňuje konkrétní komunikaci mezi aplikací brány a uzly BluetoothNode (BT) a ZigbeeNode (XB). Nejprve bylo zapnuto světlo na uzlu BluetoothNode a následně na ZigbeeNode odesláním znaku „1“, poté byla světla vypnuta odesláním znaku „0“. Oba příkazy obsahovaly příznak pro odlišení uzlu na úrovni koordinátora. Teplota byla přijímána periodicky, dotazovaní na teplotu řídí koordinátor a data předává aplikaci brány.

6.4.4 Test změny hesla

Posledním důležitým testem byla změna hesla konkrétního uzlu. Předpokladem pro změnu je již autentizovaný uzel. Heslo bylo změněno pomocí grafického rozhraní. V případě úspěšné změny hesla dojde vygenerování nového řetězce, pro autentizaci na straně RADIUS serveru. Dále je okamžitě odhlášen koncový uzel, který následně začne posílat novou žádost o autentizaci. Vygenerovaný řetězec (viz. Obrázek 43) je zobrazen pouze v okamžiku změny hesla a je nutné jej zaregistrovat do RADIUS serveru, do té doby jsou žádosti o autentizaci odmítány. Po registraci nového hesla může být uzel opět úspěšně autentizován.



Obrázek 43 - Ukázka nově vygenerovaného řetězce pro RADIUS server

6.5 Zhodnocení řešení

Navržené zabezpečení pomocí protokolů AAA, konkrétně tedy skrze protokol RADIUS, splnilo stanovené cíle v podobě zvýšení bezpečnosti smart home sítě. Výhodou použitého zabezpečení je jeho cenová dostupnost, jelikož existují na trhu i produkty, které jsou poskytovány zadarmo pod licencí open-source jako je FreeRADIUS server a knihovny, které ho podporují. Pro instalaci RADIUS serveru není nutný speciální hardware, dokonce jsou jeho distribuce součástí oficiálních balíčků pro linuxové systémy.

Další výhodou použitého řešení je nezávislost na použité komunikační technologii mezi koncovými uzly a bránou sítě. Tato skutečnost byla prakticky ověřena, když byl v navržené síti záměrně použit jeden uzel komunikující skrze Bluetooth a druhý pomocí ZigBee technologie.

Základní mechanismus zabezpečení pomocí RADIUS protokolu, který staví na několika vrstvé architektuře zabezpečení díky sdílenému heslu mezi klientem a serverem, byl podpořen několika dalšími navrženými mechanismy pro zvýšení bezpečnosti komunikace mezi prvky smart home sítě, jako je využití autentizačních tokenů a časových limitů.

Navržená smart home síť byla prakticky implementována s využitím dostupných jednodeskových počítačů Raspberry Pi a platformy Arduino. Základním kontrolním prvkem byla aplikace brány smart home, která je implementována v programovacím jazyku Java s využitím Spring frameworku. Aplikace brány také poskytla grafické rozhraní pro ověření funkčnosti zabezpečení a komunikace.

Zabezpečení i bezpečnostní mechanismy byly následně úspěšně ověřeny v prostředí navržené sítě. Testována byla zejména autentizace prvků sítě a jejich komunikace s aplikací brány. Pro demonstraci ovládání uzlů byly uzly vybaveny LED diodou, která byla rozsvěcena a zhasínána skrze aplikaci brány. Čtení dat z uzlů bylo ověřeno odesíláním teploty z teplotních senzorů a zobrazováno v grafickém prostředí aplikace brány.

Ověřena byla také možnost změny hesla a následná registrace nového hesla na straně RADIUS serveru. Změna hesla je nutná zejména po prvním zapojení uzlů do sítě, kdy je nastaveno výchozí heslo, jelikož nemají off-line paměť pro udržení hesla.

6.6 Možnosti rozšíření

Navržená architektura zabezpečení byla prakticky otestována na zjednodušené smart home síti, jejímž cílem bylo ověřit funkčnost navrženého konceptu zabezpečení a jeho mechanismů. S ohledem na použité komponenty existuje několik omezení, které by bylo vhodné vylepšit před reálnou aplikací.

V oblasti zabezpečení není příliš vhodné použití zastaralého algoritmu MD5, který byl použit zejména z důvodu omezení knihovny SoftwareSerial, jež poskytuje pouze 64bitový buffer a přepínání mezi jejími instancemi knihovny na koordinátoru. Vhodné by tedy bylo nahrazení této knihovny jinou, či jiným typem Arduino desky podporující více sériových linek hardwarově, což by umožnilo využití bezpečnějších algoritmů jako je SHA256 napřímo. Předpokladem je také výběr pouze jedné technologie bezdrátového přenosu, což by odstranilo nutnost přepínat mezi technologiemi (sériovými linkami) na úrovni koordinátora.

Dalším vylepšením by mohla být automatická registrace nového hesla na RADIUS serveru. Možností, kterou nabízí i FreeRADIUS server je využití SQL databáze, ve které by byly hesla uloženy. Do této databáze by mohla mít přístup také aplikace, která by běžela na stejném počítači jako RADIUS server a starala by se o aktualizaci hesel.

Použité desky Arduino UNO a Leonardo neposkytují žádný úložný prostor, kde by bylo možné uložit nastavení jako je heslo či jméno uzlu. Z tohoto důvodu je nutné po každém restartu desek změnit heslo. Možností by bylo použít moduly s SD kartou, ve které by tyto informace mohly být uchovány i pokud dojde k přerušení napájení.

Nabízí se také možnost připojení k bráně smart home sítě prostřednictvím internetu, což může být námět na další rozšíření, stejně jako použití více koncových uzlů, které představují jednotlivé prvky smart home sítě.

7 Shrnutí výsledků

V rámci diplomové práce bylo navrženo zabezpečení IoT prvků v prostředí smart home. Byla navržena vícevrstvá architektura, která zahrnuje koncové uzly, bránu smart home a RADIUS server.

Základem navrženého mechanismu je ověření pomocí RADIUS serveru, který slouží jako autentizační autorita smart home sítě. RADIUS server komunikuje pouze s RADIUS klienty, jehož roli zastává brána smart home. Brána zajišťuje příjem autentizačních žádostí od koncových uzlů, přidává k žádosti další údaje, které jsou následně odeslané skrze RADIUS paket na RADIUS server.

Základní princip autentizace pomocí RADIUS serveru je sdílené tajemství, které se neodesílá skrze síť. Tento princip byl ještě dále vylepšen několika navrženými bezpečnostními mechanismy s cílem co nejvíce snížit riziko neoprávněného přístupu.

Využití protokolů AAA, mezi než patří i RADIUS, bylo zvoleno zejména z důvodu existujících open-source distribucí a knihoven, což snižuje náklady na použití navrženého zabezpečení. Snaha o dostupnost a nízkou cenu řešení byla jedním z cílů diplomové práce.

Navržené zabezpečení a další bezpečnostní mechanismy byly následně úspěšně implementovány s využitím malých a levných jednodeskových počítačů Raspberry Pi a platformy Arduino. Raspberry Pi sloužil jako hostovací počítač RADIUS serveru, koncové uzly byly implementovány na platformě Arduino. Aplikace brány smart home byla implementovaná v jazyce Java s využitím frameworku Spring a poskytla grafické rozhraní implementované pomocí knihovny Swing.

Cílem vytvořeného prostředí bylo ověřit funkčnost navrženého konceptu zabezpečení. Pro komunikaci mezi uzly a bránou byly záměrně použity bezdrátové technologie Bluetooth i ZigBee, což umožnilo ověřit nezávislost navrženého zabezpečení na vybrané technologii bezdrátového přenosu. Funkčnost samotného zabezpečení byla následně ověřena sadou testů, kdy byla ověřena autentizace, ovládání již autentizovaných uzlů a změna hesla. Odeslané RADIUS pakety byly ověřeny s využitím programu Wireshark, který slouží k zachytávání síťové komunikace.

8 Závěry a doporučení

S tím, jak roste počet IoT zařízení vystává i potřeba bezpečné komunikace těchto zařízení. Diplomová práce se zabývala zejména prvky IoT používané v prostředí smart home, které používají ke komunikaci bezdrátové technologie Bluetooth a ZigBee.

V první části byl zmapován aktuální pohled na smart home a jeho zabezpečení z pohledu odborné literatury. Autoři nabízejí mnoho řešení na několika úrovních méně či více komplexních, ale základem pro bezpečnou komunikaci stále zůstává zajištění správné autentizace a autorizace.

Diplomová práce se následně zaměřila podrobně na technologie Bluetooth a ZigBee s důrazem na jejich zabezpečení a zranitelnosti. Bylo zjištěno, že zabezpečení těchto technologií roste s každou novou specifikací. Ovšem otázkou bylo, jak zvýšit zabezpečení již existujících a používaných prvků smart home, které podporují pouze starší specifikace.

Výsledkem byl tedy návrh zabezpečení založený na protokolu RADIUS, jehož principem je využití sdíleného tajemství mezi RADIUS klientem a serverem, které není přenášeno po síti. Navržena byla třívrstvá architektura sítě smart home a vzájemná komunikace mezi jednotlivými vrstvami. Základní vrstvy jsou tvořeny koncovými uzly, bránou chytré domácnosti a RADIUS serverem.

Navržený systém zabezpečení byl následně prakticky implementován. Základním cílem bylo ověřit funkcionalitu navrženého zabezpečení a bezpečnostních mechanismů stejně jako nezávislost na použité technologii bezdrátové komunikace. Autentizace i následné ovládání prvků bylo funkční a prokázalo funkčnost navrženého řešení. Jedním z důležitých aspektů návrhu byla také dostupnost a nákladnost řešení, která byla zajištěna použitím platformy Arduino a počítačů Raspberry Pi společně s open-source distribucí RADIUS serveru a potřebných knihoven.

Navržené a implementované prostředí však není vhodné k běžnému používání uživateli, jeho úkolem bylo pouze ověřit možnosti, které navržené zabezpečení poskytuje. Obsahuje také několik omezení z důvodu použitého hardwaru a softwaru. Bylo by vhodné vyřešit automatickou registraci nového uživatele na

straně RADIUS serveru. Dále by určitě bylo vhodné nahrazení zastaralého kryptografického algoritmu MD5, který byl použit zejména kvůli délce výsledného řetězce, jež byla limitována na straně koncových uzlů.

9 Seznam použité literatury

[1] ORTIZ, Jorge; CRAWFORD, Catherine; LE, Franck. DeviceMien: network device behavior modeling for identifying unknown IoT devices. In: *Proceedings of the International Conference on Internet of Things Design and Implementation*. 2019. p. 106-117.

[2] GUPTA, Vini; TRIPATHI, Sharda; DE, Swades. Green sensing and communication: A step towards sustainable IoT systems. *Journal of the Indian Institute of Science*, 2020, 1-16.

[3] MARIKYAN, Davit; PAPAGIANNIDIS, Savvas; ALAMANOS, Eleftherios. A systematic review of the smart home literature: A user perspective. *Technological Forecasting and Social Change*, 2019, 138: 139-154.

[4] BALTA-OZKAN, Nazmiye; BOTELER, Benjamin; AMERIGHI, Oscar. European smart home market development: Public views on technical and economic aspects across the United Kingdom, Germany and Italy. *Energy Research & Social Science*, 2014, 3: 65-77.

[5] DE GROOTE, Maarten; VOLT, Jonathan; BEAN, Frances. Is Europe ready for the smart buildings revolution. *Building Performance Institute Europe (BPIE)*, 2017. ISBN: 9789491143182

[6] DARBY, Sarah J. Smart technology in the home: time for more clarity. *Building Research & Information*, 2018, 46.1: 140-147.

[7] Wi-Fi Alliance, *Wi-Fi Generations* [online]. [cit. 28.10.2020] Dostupné z: https://www.wi-fi.org/discover-wi-fi

[8] IEEE COMPUTER SOCIETY LAN/MAN STANDARDS COMMITTEE, et al. IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11*, 2018.

[9] WOOLLEY, Martin. Bluetooth Core Specification Version 5.2. Feature Overview [online]. Bluetooth, 2020 [cit 28.10.2020] Dostupné z: https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf

[10] The ZigBee Alliance, *ZigBee Specification* [online]. The ZigBee Alliance, 2015. [cit. 28.10.2020] Dostupné z: https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf

[11] ALI, Waqar, et al. IoT based smart home: Security challenges, security requirements and solutions. In: *2017 23rd International Conference on Automation and Computing (ICAC)*. IEEE, 2017. p. 1-6.

[12] SHOURAN, Zaied; ASHARI, Ahmad; PRIYAMBODO, Tri. Internet of things (IoT) of smart home: privacy and security. *International Journal of Computer Applications*, 2019, 182.39: 3-8.

[13] LEE, Younghun, et al. A blockchain-based smart home gateway architecture for preventing data forgery. *Human-centric Computing and Information Sciences*, 2020, 10.1: 1-14.

[14] NÚKIB, Věcný záměr cloudové vyhlášky. *Podklad pro připomínky odborné veřejnosti* [online]. 2020, NÚKIB. Dostupné z: https://nukib.cz/download/publikace/legislativa/navrhy_legislativy/2020-07-31_vecny_zamer_cloudove_vyhlasky_v1.0.pdf

[15] STEPAN, Jan, et al. Low level communication protocol and hardware for wired sensor networks. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 2017, 9.2-4: 53-57.

[16] HAUSY, *About* [online]. HAuSy, 2020. [cit. 31.10.2020]. Dostupné z: https://hausy.org/

[17] STEPAN, Jan, et al. Lightweight protocol for M2M communication. In: *International Conference on Computational Collective Intelligence*. Springer, Cham, 2017. p. 335-344.

[18] DEY, Shreya; HOSSAIN, Ashraf. Session-key establishment and authentication in a smart home network using public key cryptography. *IEEE Sensors Letters*, 2019, 3.4: 1-4.

[19] KUMAR, Pardeep, et al. Lightweight and secure session-key establishment scheme in smart home environments. *IEEE Sensors Journal*, 2015, 16.1: 254-264.

[20] BRESSON, Emmanuel; CHEVASSUT, Olivier; POINTCHEVAL, David. Provably secure authenticated group Diffie-Hellman key exchange. *ACM Transactions on Information and System Security (TISSEC)*, 2007, 10.3: 10-es.

[21] WAZID, Mohammad, et al. Design of secure user authenticated key management protocol for generic IoT networks. *IEEE Internet of Things Journal*, 2017, 5.1: 269-282.

[22] CHIFOR, Bogdan-Cosmin, et al. A security authorization scheme for smart home Internet of Things devices. *Future Generation Computer Systems*, 2018, 86: 740-749.

[23] FIDO Alliance, *Specifications Overview* [online]. [cit. 28.10.2020] Dostupné z: https://fidoalliance.org/specifications/

[24] PECORELLA, Tommaso; PIERUCCI, Laura; NIZZI, Francesca. "Network Sentiment" Framework to Improve Security and Privacy for smart home. *Future Internet*, 2018, 10.12: 125.

[25] MENG, Yan, et al. Securing consumer IoT in the smart home: Architecture, challenges, and countermeasures. *IEEE Wireless Communications*, 2018, 25.6: 53-59.

[26] MENG, Yan, et al. Wivo: Enhancing the security of voice control system via wireless signal in iot environment. In: *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 2018. p. 81-90.

[27] EDU, Jide S.; SUCH, Jose M.; SUAREZ-TANGIL, Guillermo. Smart home personal assistants: a security and privacy review. *arXiv preprint arXiv:1903.05593*, 2019.

[28] RAMAPATRUNI, Sowmya, et al. Anomaly detection models for smart home security. In: *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE, 2019. p. 19-24.

[29] MAKKAR, Aaisha, et al. An Efficient Spam Detection Technique for IoT Devices using Machine Learning. *IEEE Transactions on Industrial Informatics*, 2020.

[30] Bluetooth, *Bluetooth Core Specification*, Version 5.2 [online]. Bluetooth 2020, [cit. 17.11.2020] Dostupné z: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=478726

[31] POTHITOS, Adam. *The History of Bluetooth*, 2 August 2017 [online]. [cit. 17.11.2020] Dostupné z: http://www.mobileindustryreview.com/2017/08/the-history-of-bluetooth.html

[32] Bluetooth, *Market Update 2020*, [online]. Bluetooth, 2020. [cit 22.11.2020] Dostupné z: https://www.bluetooth.com/wp-content/uploads/2020/03/2020_Market_Update-EN.pdf

[33] Bluetooth, *Origin of the Bluetooth Name*, [online]. Bluetooth, 2020. [cit 17.11.2020] Dostupné z: https://www.bluetooth.com/about-us/bluetooth-origin/

[34] Bluetooth, *Specifications*, [online]. Bluetooth, 2007, [cit 22.11.2020]. Dostupné z: http://blue-tooth.50webs.com/bluetooth.html

[35] ZEADALLY, Sherali; SIDDIQUI, Farhan; BAIG, Zubair. 25 years of bluetooth technology. *Future Internet*, 2019, 11.9: 194.

[36] "IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (WPAN)," in *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)* , vol., no., pp.1-700, 14 June 2005.

[37] PADGETTE, John. Guide to Bluetooth Security. *NIST Special Publication*, 2017, 800: 121.

[38] KHAN, Umar F.; HAMEED, Shafqat; MACINTYRE, Tim. TCP/IP over Bluetooth. In: *Advances in Computer and Information Sciences and Engineering*. Springer, Dordrecht, 2008. p. 479-484.

[39] Elprocus, *How does Bluetooth work?* [online]. Elprocus, 2020. [cit. 23.11.2020]. Dostupné z: https://www.elprocus.com/how-does-bluetooth-work/

[40] CHADHA, Simranjit Singh; SINGH, Mandeep; PARDESHI, Suraj Kumar. Bluetooth technology: principle, applications and current Status. *IJCSC*, 2013, 4.2: 16-30.

[41] JOO, Yang-Ick, et al. Power-efficient and QoS-aware scheduling in Bluetooth scatternet for wireless PANs. *IEEE Transactions on Consumer Electronics*, 2003, 49.4: 1067-1072.

[42] PATIL, Basavaraj, et al. *IP in wireless networks*. Prentice Hall Professional, 2003.

[43] BAERT, Mathias, et al. The Bluetooth mesh standard: An overview and experimental evaluation. *Sensors*, 2018, 18.8: 2409.

[44] GARG, V. K. Wireless Personal Area Network–Bluetooth. *Wireless Communications & Networking: Series Editor, David Clark, MIT (Amsterdam, the Morgan Kaufmann series in networking, 2010) pp*, 2007, 653-674.

[45] ROOMI, Mishal. *4 Advantages and Disadvantages of Bluetooth* [online]. Hitechwhizz, 2020 [cit. 2020-12-12]. Dostupné z: https://www.hitechwhizz.com/2020/03/4-advantages-and-disadvantages-drawbacks-benefits-of-bluetooth.html

[46] ZigBee Alliance, ZigBee 3.0 Stack User Guide [online]. ZigBee Alliance, 2018 [cit 2021-01-08]. Dostupné z: https://www.nxp.com/docs/en/user-guide/JN-UG-3113.pdf

[47] LI, Li; PODDER, Proyash; HOQUE, Endadul. A formal security analysis of ZigBee (1.0 and 3.0). In: *Proceedings of the 7th Symposium on Hot Topics in the Science of Security*. 2020. p. 1-11.

[48] FARAHANI, Shahin. *ZigBee wireless networks and transceivers*. Newnes, 2011. ISBN 0750683937

[49] IEEE 802.15 WPAN™ Task Group 4 (TG4) [online]. IEEE, 2003 [cit 2021-01-08]. Dostupné z: http://www.ieee802.org/15/pub/TG4.html

[50] Využívání vymezených rádiových kmitočtů [online]. Český telekomunikační úřad, 2020 [cit. 2021-01-08]. Dostupné z: https://www.ctu.cz/vyuzivani-vymezenych-radiovych-kmitoctu

[51] BLUM, Brien. ZigBee and ZigBee PRO: Which feature set is right for you? [online]. Texas Instruments, 2008 [cit. 2021-01-08]. Dostupné z: https://www.eetimes.com/zigbee-and-zigbee-pro-which-feature-set-is-right-for-you/

[52] Texas Instruments Incorporated, *What's New in ZigBee 3.0*, [online]. Texas Instruments Incorporated, 2019 [cit. 2021-01-08]. Dostupné z: https://www.ti.com/lit/an/swra615a/swra615a.pdf?ts=1610062731782

[53] RUDRESH, Vishruta. ZigBee Security: Basics (Part 1) [online]. Kudelski Security Research, 2017 [cit. 2021-01-08]. Dostupné z:

https://research.kudelskisecurity.com/2017/11/01/zigbee-security-basics-part-1/

[54] KUMAR, Tinku; MANE, P. B. ZigBee topology: A survey. In: *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. IEEE, 2016. p. 164-166.

[55] KHANJI, Salam; IQBAL, Farkhund; HUNG, Patrick. ZigBee Security Vulnerabilities: Exploration and Evaluating. In: *2019 10th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2019. p. 52-57.

[56] ASHRIT, Laxmi. What is ZIGBEE Technology in IoT – Architecture, Network Topologies, Applications [online]. Electricalfundablog, 2020 [cit. 2021-01-14]. Dostupné z: https://electricalfundablog.com/zigbee-technology-architecture/#Advantages_of_ZigBee_Technology

[57] Polytechnic Hub, *Advantages and disadvantages of zigbee*, [online]. Polytechnic Hub, 2017 [cit. 2021-01-14]. Dostupné z: https://www.polytechnichub.com/advantages-disadvantages-zigbee/

[58] CHEN, Ying, et al. Survey of cross-technology communication for IoT heterogeneous devices. *IET Communications*, 2019, 13.12: 1709-1720.

[59] KUMAR, NV Rajeesh; BHUVANA, C.; ANUSHYA, S. Comparison of ZigBee and Bluetooth wireless technologies-survey. In: *2017 International Conference on Information Communication and Embedded Systems (ICICES)*. IEEE, 2017. p. 1-4.

[60] VAIDYA, Vishakha D.; VISHWAKARMA, Pinki. A comparative analysis on smart home system to control, monitor and secure home, based on technologies like GSM, IOT, Bluetooth and PIC Microcontroller with ZigBee Modulation. In: *2018 International Conference on Smart City and Emerging Technology (ICSCET)*. IEEE, 2018. p. 1-4.

[61] POTHUGANTI, Karunakar; CHITNENI, Anusha. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. *Advance in Electronic and Electric Engineering*, 2014, 4.6: 655-662.

[62] DANBATTA, Salim Jibrin; VAROL, Asaf. Comparison of ZigBee, Z-Wave, Wi-Fi, and Bluetooth wireless technologies used in home automation. In: *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. IEEE, 2019. p. 1-5.

[63] NAIDU, Gollu Appala; KUMAR, Jayendra. Wireless Protocols: Wi-Fi SON, Bluetooth, ZigBee, Z-Wave, and Wi-Fi. In: *Innovations in Electronics and Communication Engineering*. Springer, Singapore, 2019. p. 229-239.

[64] CONKLIN, Wm Arthur. IT vs. OT security: A time to consider a change in CIA to include resilienc. In: *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016. p. 2642-2647.

[65] NWEKE, Livinus Obiora. Using the CIA and AAA Models to explain Cybersecurity Activities. *PM World Journal*, 2017, 6.

[66] TSIMBALO, Evgeny; FAFOUTIS, Xenofon; PIECHOCKI, Robert. Fix it, don't bin it!-CRC error correction in Bluetooth Low Energy. In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015. p. 286-290.

[67] ČESKO. Vyhláška č. 82/2018 Sb., o bezpečnostních opatřeních, kybernetických bezpečnostních incidentech, reaktivních opatřeních, náležitostech podání v oblasti kybernetické bezpečnosti a likvidaci dat (vyhláška o kybernetické bezpečnosti). In: *Zákony pro lidi.cz* [online]. © AION CS 2010-2021 [cit. 2021-01-31]. Dostupné z: https://www.zakonyprolidi.cz/cs/2018-82

[68] LONZETTA, Angela M., et al. Security vulnerabilities in Bluetooth technology as used in IoT. *Journal of Sensor and Actuator Networks*, 2018, 7.3: 28.

[69] Radius – The Wireshark Wiki. *FrontPage – The Wireshark Wiki* [online]. Wireshark, 2021 [cit. 2021-03-31]. Dostupné z: https://wiki.wireshark.org/Radius

[70] Arduino – Home. *Arduino – Home* [online]. Arduino, 2021 [cit. 2021-03-31] Dostupné z: https://www.arduino.cc/

[71] What is Java? Definition, Meaning & Features of Java Platforms. *Meet Guru99 – Free Training Tutorials & Video for IT Courses* [online]. Guru99, 2021 [cit. 2021-03-31]. Dostupné z: https://www.guru99.com/java-platform.html

[72] Spring | Why Spring?. *Spring | Home* [online]. VMware, 2021 [cit. 2021-03-31]. Dostupné z: https://spring.io/why-spring

[73] IntelliJ IDEA: The Capable & Ergonomic Java IDE by JetBrains. *JetBrains: Essential tools for software developers and teams* [online]. JetBrains, 2021 [cit. 2021-03-31]. Dostupné z: https://www.jetbrains.com/idea/

[74] Maven – Welcome to Apache Maven. *Maven – Welcome to Apache Maven* [online]. The Apache Software Foundation, 2021 [cit. 2021-03-31]. Dostupné z: https://maven.apache.org/

[75] Raspberry Pi Foundation – About Us. *Teach, Learn, and Make with Raspberry Pi* [online]. Raspberry Pi Foundation, 2021 [cit. 2021-03-31]. Dostupné z: https://www.raspberrypi.org/about/

[76] FreeRADIUS. *FreeRADIUS* [online]. Copyright © 2018 The FreeRADIUS Server Project and Contributors [cit. 2021-03-31]. Dostupné z: https://freeradius.org/

[77] Wireshark – Go Deep. *Wireshark – Go Deep* [online]. Wireshark, 2021 [cit. 2021-03-31]. Dostupné z: https://www.wireshark.org/

[78] Chinese Arduino | Dancetale Electronics. *Dancetale Electronics | Arduino Microcontroller DIY Electronics* [online]. Dancetale Electronics, 2014 [cit. 2021-04-01].
Dostupné z: https://dancetale.wordpress.com/2014/12/30/chinese-arduino-nhduino/

[79] XCTU – Download and Install the Configuration Platform for XBee/RF Solutions | Digi International. *IoT Solutions, Software, Products, Services for the Industrial IoT | Digi International* [online]. Digi International, 2021 [cit.

2021-04-01]. Dostupné z: https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu

[80] XCTU User Guide. *IoT Solutions, Software, Products, Services for the Industrial IoT | Digi International* [online]. Digi International, 2021 [cit. 2021-04-01].
Dostupné z: https://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm

[81] TinyRadius: Java Radius library. *TinyRadius: Java Radius library* [online]. Dostupné z: http://tinyradius.sourceforge.net/

[82] jSerialComm. *GitHub Pages* [online]. Fazecast, 2021 [cit. 2021-04-02]. Dostupné z: https://fazecast.github.io/jSerialComm/

[83] Logback Home. *Logback Home* [online]. QOS.ch, 2019 [cit. 2021-04-02]. Dostupné z: http://logback.qos.ch/index.html

[84] Spring Framework. *Spring | Home* [online]. VMware, 2021 [cit. 2021-04-02]. Dostupné z: https://spring.io/projects/spring-framework

[85] GitHub – google/guava: Google core libraries for Java. *GitHub: Where the world builds software · GitHub* [online]. GitHub Inc., 2021. [cit. 2021-04-02]. Dostupné z: https://github.com/google/guava

Příloha 1: Struktura archivu se zdrojovými soubory

Níže je uvedena struktura adresářů, souborů a jejich popisu v archivu zrojove_soubory.zip, který je přiložen k elektronické podobě diplomové práce.

- **zdrojove_soubory** – hlavní adresář
  - **arduino_source** – adresář obsahující jednotlivé projekty pro Arduino desky.
    - **Coordinator**
      - **Coordinator.ino** – soubor aplikace Arduino IDE obsahující zdrojový kód koordinátora psaný pro desku Arduino Uno.
    - **EndNodeBT**
      - **EndNodeBT.ino** – soubor aplikace Arduino IDE obsahující zdrojový kód Bluetooth uzlu psaný pro desku Arduino Leonardo.
    - **EndNodeXB**
      - **EndNodeXB.ino** – soubor aplikace Arduino IDE obsahující zdrojový kód ZigBee uzlu psaný pro desku Arduino Uno.
  - **smart_home_gw.zip** – archiv obsahující zdrojové soubory aplikace brány chytré domácnosti napsané v jazyce Java, v prostředí IntelliJ IDEA. Struktura aplikace je popsána v kapitole 6.3.9.
  - **radius_config** – adresář obsahující konfigurační soubory pro RADIUS server.
    - **authorize** – Obsahuje uživatele, kteří mají být autentizováni.
    - **clients.conf** – Obsahuje RADIUS klienty a jejich parametry.
    - **radius.conf** – Základní konfigurační soubor RADIUS serveru.

Příloha 2: Zadání práce

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání diplomové práce

Autor:	**Bc. Martin Šustr**
Studium:	I1800121
Studijní program:	N1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název diplomové práce:	**Zabezpečení komunikace prvků IoT v prostředí smart home**
Název diplomové práce AJ:	Security of IoT communication for SMART home

Cíl, metody, literatura, předpoklady:

Cílem práce na provést analýzu, návrh a praktické ověření zabezpečení komunikace IoT prvků v prostředí smart home za využití komunikační technologie bluetooth nebo zigbee. V teoretické části autor představí principy komunikace za využití technologie bluetooth a zigbee s důrazem na jejich bezpečnost. Dále se autor zaměří na známé zranitelnosti těchto technologií, které ověří a provede jejich analýzu, na základě které doporučí nejvhodnější technologii pro návrh a testování v prostředí smart home.

V praktické části autor navrhne řešení bezpečné komunikace vhodné pro smart home s přihlédnutím k řešení bezpečnosti, dostupnosti řešení a jeho ceně. Navržené řešení autor prakticky otestuje za využití malých jednodeskových počítačů platformy Arduino nebo Raspberry pi.

IBN MINAR, Nateq Be-Nazir. *Bluetooth Networking and Its Security Architecture. Germany: LAP Lambert Academic Publishing, 2012. ISBN 9783848439782.*

RAY, Niranjan K. a Ashok Kumar TURUK. Handbook of research on advanced wireless sensor network applications, protocols, and architecture. Hershey, PA: IGI Global, [2017]. ISBN 978-152-2504-863.

Garantující pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	Mgr. Josef Horálek, Ph.D.
Datum zadání závěrečné práce:	21.10.2019