TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Inteligentné relé s WiFi konektivitou do siete eduroam

Bakalárska práca

2019 **Martin Chlebovec**

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
KOŠICE

Inteligentné relé s WiFi konektivitou do siete eduroam
Bakalárska práca

Študijný program:	Počítačové siete
Študijný odbor:	Počítačové inžinierstvo
Školiace pracovisko:	KEMT
Školiteľ:	prof. Ing. Miloš Drutarovský, CSc.

2019 Košice **Martin Chlebovec**

Abstrakt v SJ

Hlavnou témou bakalárskej práce inteligentného relé je bezpečnosť. Dôraz na bezpečnosť sa v informačných technológiách kladie predovšetkým pri komunikačných sieťach a prenášaných informáciách. Zaručuje utajenie, integritu dát po celej trase a ich dôveryhodnosť medzi odosielateľom a príjemcom, pričom sú dáta chránené voči typom rôznym útokov, ktorými sa útočníci snažia odcudziť, poškodiť, alebo zmazať cenné - osobné informácie.

Bakalárska práca poukazuje na spôsoby, akými je možné bezpečnosť implementovať na úrovni lokálnej siete, alebo aj spojenia na vzdialený server do internetu. Práca demonštruje taktiež možnosť využitia open-source hardvéru a jeho prepojenie s podnikovými bezdrôtovými sieťami, ktoré ponúkajú vyšší štandard zabezpečenia. Prenos informácii na vzdialenú lokalitu šifrovaným protokolom s ich následnou reprezentáciou používateľovi.

Pre zvýšenie bezpečnosti sú využité aj implementácie certifikátov pre klienta, server a certifikačnú autoritu na koncových zariadeniach s možnosťou ich vzájomného overenia. V práci sú uvedené elektrotechnické schémy s detailným opisom riešenia problematiky z oblasti bezpečnosti a implementácie jednotlivých prvkov bezpečnosti na úrovni softvéru s dostupnosťou ukážkových zdrojových kódov obsiahnutých v práci.

Kľúčové slová v SJ

eduroam, 802.1X, esp32, Arduino, certifikát, OpenSSL

Abstrakt v AJ

Main topic of the bachelor thesis Smart Relay is security. The emphasis on security in information technology is mainly for communication networks and information transmitted. It guarantees confidentiality, data integrity throughout the route and their trustworthiness between sender and recipient, while data is protected against types of various attacks by which attackers try to steal, damage, or delete valuable - personal information. Thesis points out the ways in which security can be implemented at the local network level, or even the connection to a remote server on the Internet. Bachelor thesis also demonstrates the possibility of using open-source hardware and its interconnection with enterprise wireless networks that offer higher security standards. Transferring information to a remote site with an encrypted protocol and then representing it to the user. In order to increase security, implementation of certificates for the client, server and certification authority on the end devices with the possibility of their mutual verification are also used. The work presents electrotechnical schemes with a detailed description of security issues and implementation of

individual elements of security at the software level with the availability of sample source codes contained in the work.

Kľúčové slová v AJ

eduroam, 802.1X, esp32, Arduino, certificate, OpenSSL

Zadanie práce

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra elektroniky a multimediálnych telekomunikácií

ZADANIE BAKALÁRSKEJ PRÁCE

Študijný odbor: **Počítačové inžinierstvo**
Študijný program: **Počítačové siete**

Názov práce:

Inteligentné relé s WiFi konektivitou do siete eduroam

Smart relay with WiFi conectivity to eduroam network

Študent: **Martin Chlebovec**
Školiteľ: **prof. Ing. Miloš Drutarovský, CSc.**
Školiace pracovisko: **Katedra elektroniky a multimediálnych telekomunikácií**
Konzultant práce:
Pracovisko konzultanta:

Pokyny na vypracovanie bakalárskej práce:

Na báze mikropočítača ESP32 navrhnite a otestujte inteligentné relé umožňujúce jeho bezpečné ovládanie prostredníctvom Internetu. Bezpečné ovládanie realizujte použitím vhodného šifrovania komunikácie s inteligentným relé. Programové riešenie vytvorte v jazyku C s využitím platformy Arduino. Navrhnite vhodné využitie architektúry klient-server a demonštrujte jej využitie na typických príkladoch (scenároch) použitia. V rámci experimentov otestujte aj funkčnosť navrhnutého riešenia v školskej sieti Eduroam.

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 24.05.2019
Dátum zadania bakalárskej práce: 31.10.2018

prof. Ing. Jozef Juhár, CSc.
vedúci/garantujúceho pracoviska

TECHNICKÁ UNIVERZITA V KOŠICIACH FAKULTA ELEKTROTECHNIKY A INFORMATIKY -12-

prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú bakalársku prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 22. mája 2019 .....

vlastnoručný podpis

Poďakovanie

V prvom rade sa chcem srdečne poďakovať vedúcemu práce prof. Ing. Milošovi Drutarovskému, CSc. za rady, vysvetlenie odborných termínov, konzultácie súvisiace s technickou stránkou projektu. Ďakujem odbornému asistentovi Ing. Stanislavovi Slovákovi, PhD. za konzultácie a rady v oblasti návrhu elektrotechnických schém a vyhotovenie hardvérového prototypu s modulom ESP32 spolu s doložením dokumentácie výroby prototypu.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam symbolov a skratiek

802.1x	IEEE štandard zabezpečenia prístupu do siete
AAA	Autentizácia, autorizácia, tarifikácia (Authentication, Authorization, Accounting)
AJAX	Funkcia výmeny dát so serverom pod knižnicou jQuery
Arduino IDE	Integrované vývojové prostredie jazyka Wiring (Arduino)
Arduino core	Kompatibilné knižnice s jazykom Wiring (Arduino) pre ESP čipy, dosky
dBm	Decibel-miliwatt (miera, jednotka výkonu signálu)
Diameter	Zdokonalený RADIUS protokol, škálovateľný

EAP	Rozšírený autentizačný protokol (Extensible Authentication Protocol)
EAPoL	EAP vnorený do Ethernetu - dátová vrstva (EAP over LAN)
EAP-TTLS	Založený na PEAP protokole, overenie servera cez CA certifikát
Enterprise	Typ zabezpečenia WiFi siete - tzv. podnikové, využívajú štandard 802.1X
ESP-IDF	Framework pre programovanie ESP32 v C jazyku (Espressif IoT Development Framework)
GPIO	Vstupno-výstupný vývod (General-purpose input/output)
HTML5	Hypertextový značkový jazyk verzie 5
IoT	Internet vecí (Internet of Things)
IP	Internet protokol/ stupeň ochrany krytím
IPSec	Bezpečnostné rozšírenie IP protokolu (IP Security)
JSON	Formát dát - čitateľný strojovo i človekom, využitie: tabuľky, grafy, databázy
LAN	Lokálna počítačová sieť - fyzické prepojenie počítačov na malej oblasti (Local Area Network)
MAC	Identifikátor sieťového zariadenia, prístup k médiu (Media access control)
MsCHAPv2	Microsoft protokol na preukázanie hesla používateľa (Challenge-Handshake Authentication Protocol)
NPN	Druh tranzistora s dvomi polovodičmi typu N a jedným P
NTLM	Autentizačný protokol systému Windows (NT LAN Manager)
OpenSSL	Kryptografický nástroj pre generovanie certifikátov, knižnica, krypto funkcie
OSI	Vrstvovo založený opis štruktúry protokolov - štandard (Open Systems Interconnection Reference Model)
PEAP	Šifrovaný EAP protokol do TLS vrstvy - preukázanie identity RADIUS servera (Protected EAP)
PHP	Skriptovací jazyk pre tvorbu klient-server aplikácie (Hypertext Preprocessor)
RADIUS	Sieťový protokol na UDP porte 1812, centralizovaný AAA prvok siete

RadSec Transportný protokol pre RADIUS datagrami cez TLS a TCP s overením certifikátmi medzi servermi (RADIUS secure)

Realm Identifikátor organizácie, ku ktorej identita patrí

Revoke Zneplatnenie certifikátu pred koncom platnosti na vyžiadanie

RS232 Komunikačné rozhranie s definovanými napäťovými úrovňami

PSK Označenie pre WiFi sieť so zdieľaným heslom (Pre-shared key)

SANET Slovenská akademická sieť - združenie

SSID Identifikátor bezdrôtovej WiFi siete (Service Set Identifier)

SSL Vrstva bezpečných socketov, použitie pre šifrovanie nad nešifrovanými protokolmi, napr. HTTP→ HTTPS, (Secure Sockets Layer)

SSR Polovodičové relé (Solid-state relay)

TCP Balík protokolov so zárukou doručenia (Transmission Control Protocol)

TLS Protokoly šifrovania dát (Transport layer security)

TTL Logika - štandard v digitálnych obvodoch (Transistor-transistor-logic)

TTLS Rozširuje použitie TLS pre EAP protokol, vytvára šifrovaný tunel po overení servera u klienta

UART Sériová linka, komunikačné rozhranie (Universal asynchronous receiver-transmitter)

UDP Balík protokolov bez záruky doručenia (User Datagram Protocol)

Wiring Zjednodušený jazyk C pre mikrokontroléry (Arduino)

WPA Štandard zabezpečenia WiFi sietí (Wi-Fi Protected Access)

Úvod

V dnešnej dobe sa využívajú komunikačné siete na rôzne účely. Sieťami sa prenášajú dáta rôznych záujmových skupín, predovšetkým osobné dáta používateľov, ale aj podnikové, firemné a štátne dáta rôznej dôležitosti. Postupom času si každá kategória dát vynútila využitie ochrany bezpečnostnými prvkami. V závislosti od dôležitosti dát sú tieto dáta chránené rôznymi typmi bezpečnostných mechanizmov, prípadne ich kombináciou.

Téma mojej bakalárskej práce je koncept návrhu a realizácie sieťového klient-server systému, ktorý predstavuje funkčné inteligentné relé s dôrazom na bezpečnosť, pričom jeho inteligentnosť spočíva v tom, že dokáže fungovať autonómne bez nutnosti zásahu, či neustáleho monitoringu používateľa. Z pohľadu bezpečnosti som chcel dosiahnuť, aby systém používal najmodernejšie zabezpečovacie prostriedky. Využitie školskej bezdrôtovej siete pod štandardom 802.1X (štandard zabezpečenia prístupu do siete) bolo jedným z prvých krokov pre bezpečný systém na úrovni lokálnej siete.

Viac informácii o spomínanom type bezdrôtovej siete spoločne s teóriou jej fungovania je opísané predovšetkým v kapitole 1 a podkapitole 1.1. Bloková schéma vývojovej dosky ESP32 (vývojová doska od firmy Espressif) s perifériami a spôsobom komunikácie je obsiahnutá v kapitole 2. Programová implementácia pripojenia do WiFi siete je opísaná v podkapitole 2.2. Certifikáty boli dôležitým prvkom pre bezpečnú komunikáciu so vzájomným overením, ktoré bolo potrebné vygenerovať a následne implementovať pre klienta aj server. Generovanie certifikátov nástrojom OpenSSL (kryptografický nástroj, knižnica) s ich následnou implementáciou do platforiem Windows, ESP32 ako klienta a server na operačnom systéme Linux je podrobne opísané v práci v podkapitolách 2.3 a 2.4.

Pri návrhu konceptu komunikácie klient-server som využil vývojový kit s čipom ESP32 od firmy Espressif, ktorú som programoval v zjednodušenom jazyku C - Wiringu, pričom som využil softvérovú sadu knižníc Arduino core, keďže doska primárne využíva ESP-IDF (framework postavený na jazyku C pre programovanie ESP32). Zdrojový kód pre ESP32 som písal v editore Arduino IDE. Tento jednoduchý editor obsahuje kompilátor ale aj možnosť nahratia programu do mikrokontroléra. Prvou časťou bakalárskej práce bolo programovanie jednoduchého webového rozhrania pre reprezentáciu dát, ktoré vývojová doska ESP32 odosiela na server. Informácie zaoberajúce sa programovaním čipu ESP32 a webového rozhrania v jazyku PHP (skriptovací jazyk pre klient-server aplikácie) som detailnejšie opísal v kapitole 2 a ďalších podkapitolách. Kapitoly obsahujú aj prehľad jednotlivých modulov a periférii, ktoré v bakalárskej práci využívam s opisom technických parametrov.

Súčasťou práce je aj schématický návrh minimálnej schémy inteligentného relé, ktoré používa výhradne čip ESP32 bez pôvodnej vývojovej dosky a elektroniky okolo čipu. Návrh teda obsahoval kreslenie schémy pre relé v nástroji Autodesk Eagle na technický výkres s napájacou elektronikou, vývodmi a prepojením periférií. Bližšie a detailnejšie informácie o návrhu minimálnej schémy sú obsiahnuté v kapitole 3, kde na návrh nadväzuje aj fotografia prototypu. Pre pochopenie technických a teoretických poznatkov z tém bezpečnosti som v bakalárskej práci využil rôzne pramene dostupnej literatúry na úrovni vedeckých článkov, internetových encyklopédií, katalógových listov, ale aj odborných video-prednášok na rôzne témy z IT konferencií, alebo záujmových činností.

1. Sieť eduroam

Sieť eduroam [1] (education roaming) je celosvetová počítačová infraštruktúra, ktorá je implementovaná vo vedeckej a univerzitnej sfére. Pôvodná myšlienka projektu - siete eduroam vznikla v Holandsku v roku 2002. Sieť je navrhnutá pre celosvetové využívanie študentmi, vedeckými pracovníkmi a pedagógmi.

Je tvorená sieťou RADIUS (označenie protokolu, servera používajúceho tento protokol pre overenie) serverov, ktoré overujú používateľov, ktorí sa chcú do siete pripojiť a využívať služby internetu a LAN (lokálna sieť - Local Area Network) siete. Táto infraštruktúra RADIUS serverov sa často označuje aj ako federácia. RADIUS server je zariadenie poskytujúce služby AAA: autentizácie, autorizácie a tarifikácie (AAA - Authentication, Authorization, and Accounting). Zariadenie je fyzický prvok siete, ktorý overuje prihlasovacie údaje používateľov, ktorí sa do siete pripájajú.

Každý používateľ siete eduroam sa vie pripojiť do siete z akejkoľvek organizácie na svete, ktorá je v projekte eduroam zaradená bez nutnosti meniť konfiguračné nastavenia sieťového profilu. Všetky prístupové body vysielajú SSID (vysielaný identifikátor - názov) - sieť eduroam. Používateľ sa tak pripája, spôsobom, akoby bol v domácej organizácii. Každá organizácia v sieti eduroam sa musí prispôsobiť určitým pravidlám, ktoré sú jednoznačne definované, čo umožňuje ich vzájomnú kompatibilitu.

Hlavné pravidlo a myšlienka projektu je, že ak chce univerzita, alebo iná organizácia byť v sieti eduroam a mať v nej používateľské účty, musí poskytovať službu, teda možnosť pripojiť sa aj používateľom z iných organizácii a poskytnúť im konektivitu do internetu. Používateľ sa pri vstupe do siete musí preukázať svojou identitou a heslom. Projekt siete eduroam rieši iba overenie používateľa infraštruktúrou RADIUS serverov, nie jeho pripojenie do internetu ako také. Konektivitu do internetu zabezpečuje daná organizácia prevádzkujúca túto sieť v projekte eduroam.

Každá organizácia siete eduroam vlastní RADIUS server s určitou definovanou konfiguráciou. Organizácia vytvára a spravuje používateľské účty svojich používateľov. Každý používateľ siete eduroam je jednoznačne priradený k svojej organizácii prostredníctvom realmu. Realm je definovaný v tvare @organizácia.doména. V prípade Technickej univerzity v Košiciach je to realm @tuke.sk [2] pre študentov, pedagógov i vedeckých pracovníkov.

Existujú ešte dva vyžiadané parametre, ktoré sú pri prihlasovaní do WiFi (bezdrôtovej) siete eduroam potrebné. Ak sa používateľ prihlasuje do siete eduroam, musí zadať identitu, ktorá je tvorená jeho menom, alebo identifikátorom a realmom organizácie. Pre Technickú univerzitu je identita definovaná v tvare: identifikátor@tuke.sk. Druhým parametrom, ktorý používateľ pri prihlasovaní do WiFi siete používa je jeho heslo. Heslo je uložené v databáze v čitateľnej podobe, alebo vo formáte NTLM hashu [3] (Hashovacia funkcia hesiel od Microsoftu pre LAN manager). Tieto spôsoby zadania hesla je možné aplikovať aj na strane klienta snažiaceho sa o prístup do siete.

V prípade, že sa používateľ prihlasuje do siete eduroam vo vlastnej organizácii, môže použiť len prihlasovacie meno, respektíve identifikátor bez realmu v prípade, že toto nastavenie siete umožňuje. Tento spôsob je však možné aplikovať iba vo vlastnej organizácii, keďže sa z inej organizácie klient nepripojí, nakoľko neexistuje realm v identite používateľa.

Na Slovensku sieť eduroam prevádzkuje rada univerzít, konektivitu medzi nimi a do internetu zabezpečuje združenie SANET [4] (Slovenská akademická sieť). Každý klient pri pripojení do siete eduroam musí prekonať dve fázy:

1. Autentizácia - preukázanie identity a hesla prostredníctvom prosebníka (supplicant)
2. Autorizácia - odpoveď RADIUS servera Povoľ prístup / Zamietni prístup

Ak používateľ zlyhá pri preukázaní identity a hesla, autorizácia sa vykoná s výsledkom zakáž prístup, klientovi nie je pridelená IP adresa a nemôže využívať služby internetu a LAN siete. Pokus môže opakovať.

V prípade, že klient, ktorý je autentizovaný a autorizovaný vykonáva zakázanú aktivitu, je možné zablokovať iba jeho fyzickú MAC [5] (Riadenie prístupu k médiu) adresu prostredníctvom smerovačov, prípadne realm pre všetkých používateľov danej organizácie prostredníctvom RADIUS servera. Keďže si klient môže zmeniť MAC adresu, v posledných rokoch eduroam implementuje možnosť zablokovať používateľský účet z akejkoľvek organizácie bez nutnosti zablokovať celý realm za predpokladu, že domovská organizácia účtu vysiela istý parameter o používateľovi - tzv. Chargeable user [6], čo je reťazec, ktorý identifikuje používateľa v inej organizácii, ktorá mu poskytuje konektivitu. Tento reťazec sa vráti spoločne s informáciou o autorizácii klienta z domovskej organizácie.

Každý používateľ má tento reťazec unikátny. Nakoľko klient môže využiť anonymnú identitu, organizácia, kde sa klient pripája do siete eduroam nevie meno používateľa, dokáže získať iba tento

identifikačný reťazec z domácej organizácie, avšak vysielanie tohto reťazca nie je nutnosťou. RADIUS server siete eduroam je v organizáciách najčastejšie v softvérovej konfigurácii operačného systému Linux Debian s rozšírením FreeRADIUS.

Okrem nastavenia RADIUS serverov na štandard siete eduroam sa vyžaduje aj nastavenie a overenie dôveryhodných prístupových bodov. Prístupové body obsahujú konfiguráciu s adresou RADIUS servera, prenosovým protokolom s uloženým reťazcom - vzájomne zdieľaným tajomstvom [7]. Tento textový reťazec je pre každý prístupový bod iný, avšak musí sa zhodovať s nastaveným pre daný prístupový bod na strane RADIUS servera.

Sieť eduroam mala pri konfigurácii aj v minulosti isté špecifiká a problémy, ktoré sťažovali jej prvotné nastavenie. Pri starších operačných systémoch bolo nutné manuálne nastavovať profil WiFi siete, kde bolo nutné nastavovať adresu RADIUS servera, inštalovať certifikát certifikačnej autority, manuálne zvoliť metódy overenia v prvom a druhom kroku. Nastavenie bolo pre klientov namáhavé a pre laika bez pomoci takmer nemožné sieťový profil svojho zariadenia nastaviť.

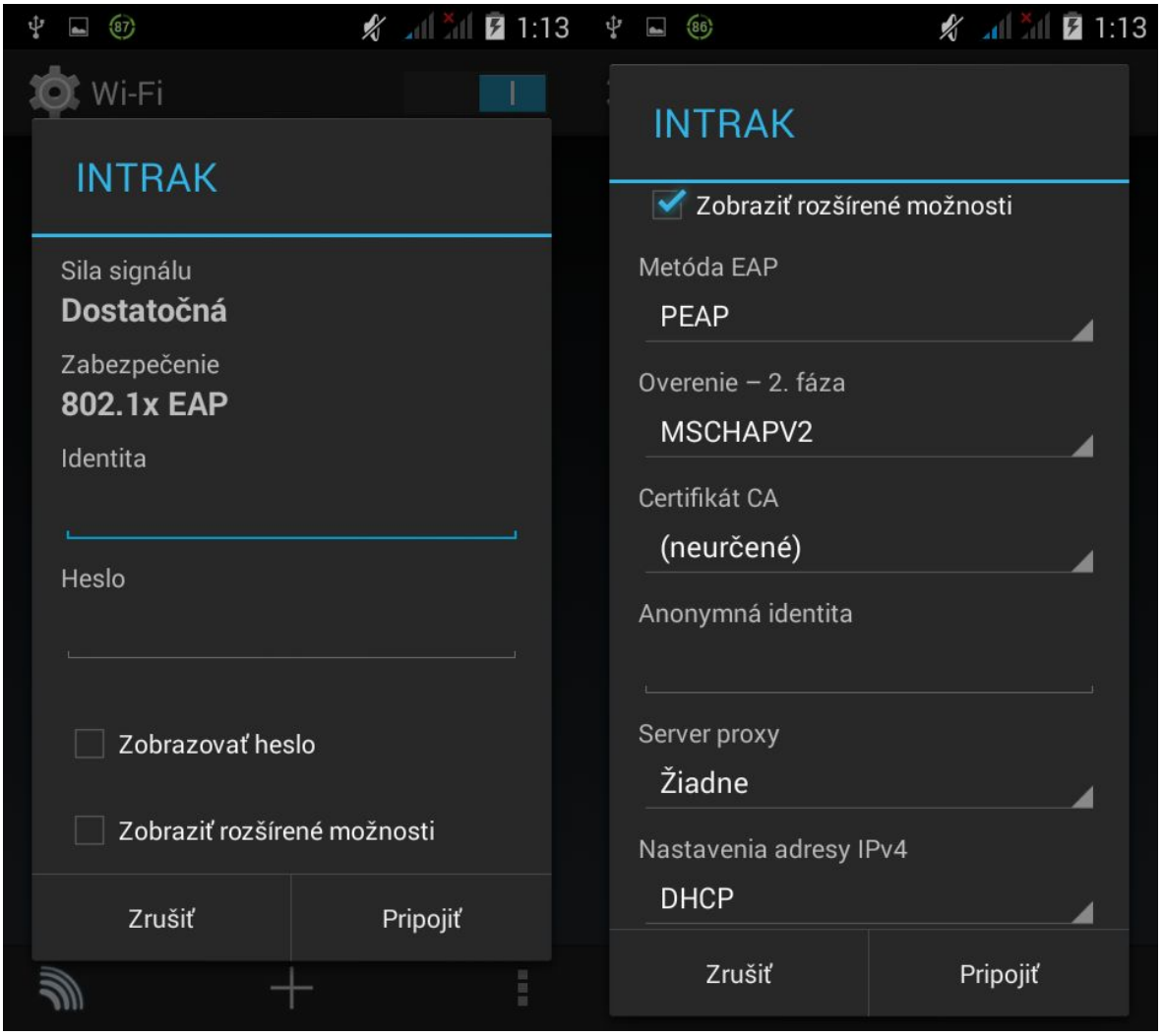
Novšie operačné systém dokázali do istej miery pripojenie automatizovať a v továrenských nastaveniach Windows 8+ a aj Android dôveruje akémukoľvek platnému certifikátu RADIUS servera pri overení pod štandardom 802.1X, ktorý mu RADIUS server v istom kroku predkladá.

Zneplatnený certifikát (revoke certificate) nemá ako rozpoznať, nakoľko nemá prístup k internetu počas autentizácie a autorizácie. Na jednu stranu to predstavuje riziko, na druhú stranu umožňuje prihlásiť sa spôsobom ľahším i pre laikov.

Niektoré systémy, ktoré nemajú plnú podporu Enterprise (podnikových) WiFi sietí pod 802.1X to riešia tým, že dôverujú čomukoľvek - aj neplatnému certifikátu, alebo vyžadujú niektorú časť prosebníka dodatočne nainštalovať. Windows XP podporoval Enterprise WiFi siete pod štandardom 802.1x až od SP3 (Service pack 3 - balík softvérových záplat, opráv, aktualizácii).

Každý používateľ, ktorý sa chce do siete eduroam pripojiť musí mať softvér, ktorý sa nazýva prosebník. Je to softvér, ktorý riadi celý proces autentizácie. Vizualizovaný prosebník operačného systému Windows (Obr. 1) a Android môže vyzerať podobne ako na priloženom obrázku (Obr. 2). V prípade vynechania anonymnej identity, prosebník automaticky doplní identitu používateľa za anonymnú.

Obr. 1 Príklad vyžiadania identity a hesla pod protokolom 802.1X v OS Windows



Obr. 2 Príklad vyžiadania identity a hesla pod protokolom 802.1X v OS Android

1.1 Overenie používateľa v sieti

Prosebník [8] je nástroj podporujúci EAP - rozšírujúci autentizačný protokol. Tento softvér umožňuje komunikáciu s prístupovým bodom, alebo prepínačom v prípade káblovej siete. Pri nadviazaní spojenia s prístupovým bodom, alebo prepínačom je zablokovaná komunikácia používateľa.

Klient je vyzvaný výzvou od prístupového bodu na zadanie identity a hesla na jeho zariadení. Tieto správy sú vymieňané medzi používateľom a prístupovým bodom EAPoL [9] (EAP protokol zapuzdrený do Ethernet rámcov) rámcami na dátovej vrstve OSI [10] (štandard - opis vrstiev skladby protokolov) modelu, teda na úrovni MAC adries, ktoré patria k fyzickému médiu zariadenia.

Správu s používateľskou identitou a heslom získanú od klienta odovzdáva RADIUS serveru prístupový bod. RADIUS server s prístupovým bodom komunikuje pod protokolom RADIUS, alebo Diameter. Ak je používateľ z rovnakej organizácie - zistí sa na základe realmu, RADIUS server overí, či sa identita nachádza v jeho databáze identít spolu s heslom.

Výsledkom autorizácie [11] je odpoveď v tvare ACCESS, alebo REJECT pre prístupový bod. V prípade ACCESS je klientovi pridelená IP adresa z miestnej siete a sú mu otvorené základné TCP (protokoly so zárukou doručenia informácie), UDP (protokoly bez záruky doručenia informácie), IP (komunikačný protokol medzi dvomi zariadeniami) porty, ktoré potrebuje pre bežnú prácu. Číselný zoznam portov a ich služieb, ktoré eduroam vyžaduje otvoriť vrátane smeru komunikácie je obsiahnutý v tabuľke (Tab. 1).

Tab. 1 Číselný zoznam povolených TCP, UDP, IP služieb

Služba	Protokol	Port	Smer komunikácie
Standard IPSec VPN	IP (ESP) IP (AH) UDP(IKE)	50 51 500	Obojsmerne Obojsmerne Von
OpenVPN 2.0	UDP (OpenVPN)	1194	Obojsmerne
IPv6 Tunnel broker	IP	41	Obojsmerne
IPSec NAT-Traversal	UDP (IPSec)	4500	Obojsmerne
Cisco IPSec VPNoTCP	TCP	10000	Von
PPTP VPN	TCP (PPTP)	1723	Obojsmerne

PPTP VPN	IP (GRE)	47	Von
SSH	TCP	22	Von
HTTP	TCP TCP TCP TCP	80 443 3128 8080	Von Von Von Von
Mail služby - odosielanie	TCP (SMTP-SSL) TCP (decbsrv)	465 579	Von Von
Mail služby - prijímanie	TCP (IMAP) TCP (IMAP-SSL) TCP (POP3) TCP (POP3-SSL)	143 993 110 995	Von Von Von Von
FTP	TCP	21	Von

V prípade odpovede REJECT je klient odmietnutý a jeho komunikácia je naďalej blokovaná. Klient ako taký pri autentizácii nikdy nekomunikuje s RADIUS serverom priamo, ale cez prístupový bod, alebo prepínač, ktorý sa označuje termínom autentizátor.

Na UDP port 1812 [12] posiela prístupový bod používateľskú identitu a heslo do RADIUS servera a spätne prístupový bod dostáva odpoveď s výsledkom autorizácie. Klient snažiaci sa o pripojenie do siete nie je o výsledku autorizácie informovaný. V prípade overenia zadaných používateľských informácii klientovi prístupový bod pridelí IP adresu a umožní mu prístup do siete a internetu, alebo v prípade neúspešnej autentizácie mu vyprší timeout, kedy je vyzvaný k opätovnému vloženiu identity a hesla výzvou od prístupového bodu.

Ak klient použije realm iný ako miestnej organizácie, komunikácia sa presmeruje na štátny RADIUS doménový server, ktorý overuje doménu prvej úrovne realmu (.sk, .cz, .at) a ak je doména prvej úrovne iná, odošle sa požiadavka na Root (koreňový) RADIUS server [6] celej eduroam federácie do Holandska.

Tento Root RADIUS server uchováva databázu všetkých realmov a odošle správu na potrebný doménový RADIUS server daného štátu (.sk, .cz, .at), ktorý ďalej správu smeruje už na RADIUS server danej organizácie. Koncový RADIUS server nahliadne do databázy a spätnou požiadavkou informuje o výsledku autorizácie.

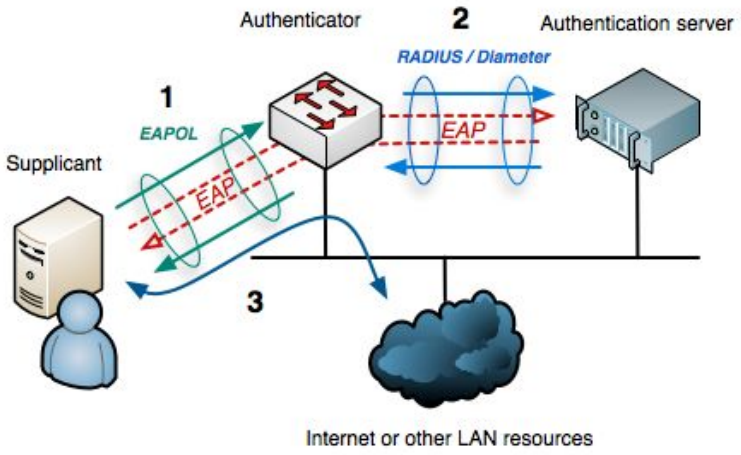
V začiatkoch projektu eduroam fungovalo prepojenie RADIUS serverov pod protokolom IPSec [13], ktorý bol v dôsledku viacerých zlyhaní neskôr nahradený protokolom RadSec [14], ktorý využíva TLS [15] (protokol šifrovania - Transport Layer Security) tunel medzi RADIUS servermi, ktoré sú vzájomne overené certifikátmi. RadSec sa v sieti eduroam využíva na prepojenie jednotlivých RADIUS serverov dodnes.

V prípade, že klient nechce poskytnúť svoju identitu pri pripájaní z inej organizácie, je možné použiť aj tzv. anonymnú identitu [6] v tvare anonymous@organizacia.doména a použiť svoje heslo ku svojmu eduroam účtu vrátane internej identity používateľa. Anonymná identita v prípade Technickej univerzity v Košiciach je anonymous@tuke.sk.

Anonymná identita sa prikladá k identite používateľa. Anonymná identita používateľa sa používa výhradne na nahliadnutie do realmu všetkými RADIUS servermi po ceste overenia, na základe ktorého môže byť požiadavka smerovaná na daný RADIUS server príslušnej organizácie a štátu - domény prvej úrovne. V cieľovej organizácii prebehne overenie používateľa jeho identitou a heslom. Používajú sa dve hlavné metódy overenia, ktoré sú funkčnosťou takmer totožné, rozlišuje sa vonkajší a vnútorný TLS tunel:

- PEAP + MsCHAPv2 - PEAP je metóda overenia serverového certifikátu a jeho mena (hostname) voči klientovi, MsCHAPv2 je metóda overenia klienta voči serveru (identita, heslo)
- EAP-TTLS + MsCHAPv2 - EAP-TTLS je metóda overenia serverového certifikátu a jeho mena (hostname) voči klientovi, MsCHAPv2 je metóda overenia klienta voči serveru (identita, heslo)

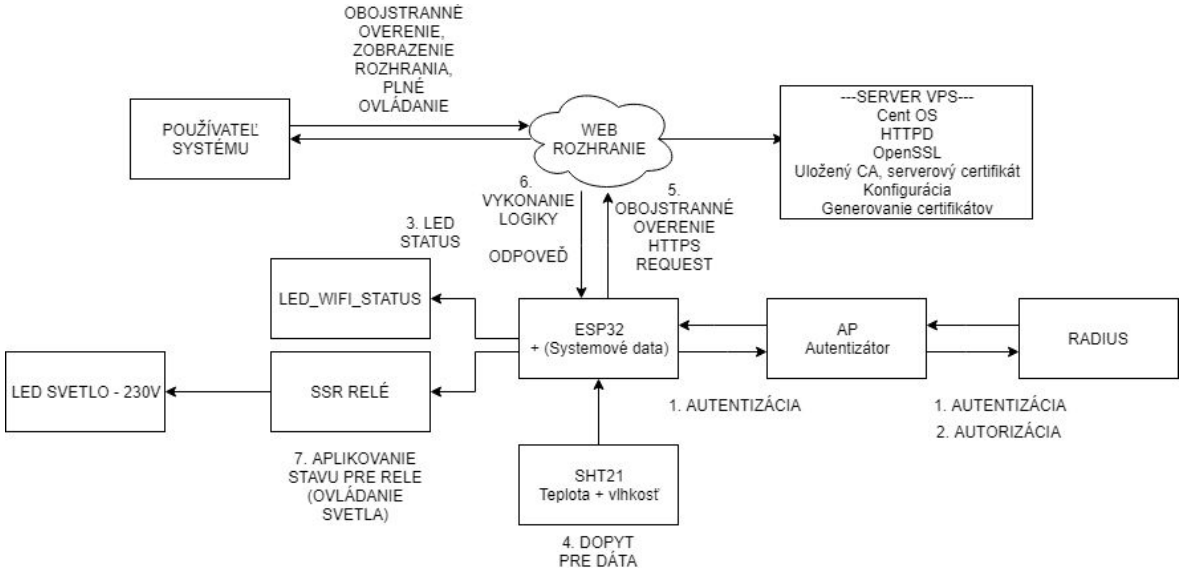
Obrázok (Obr. 3) ilustračne naznačuje komunikáciu medzi používateľom a prístupovým bodom EAPoL rámcami a protokolom RADIUS/Diameter medzi RADIUS serverom a prístupovým bodom, pričom celá komunikácia je zabalená do TLS tunelu. V treťom kroku je používateľ autorizovaný do siete a je mu pridelená IP adresa a povolený prístup do internetu.



Obr. 3 Spôsob komunikácie v sieti pri autentizácii klienta v sieti

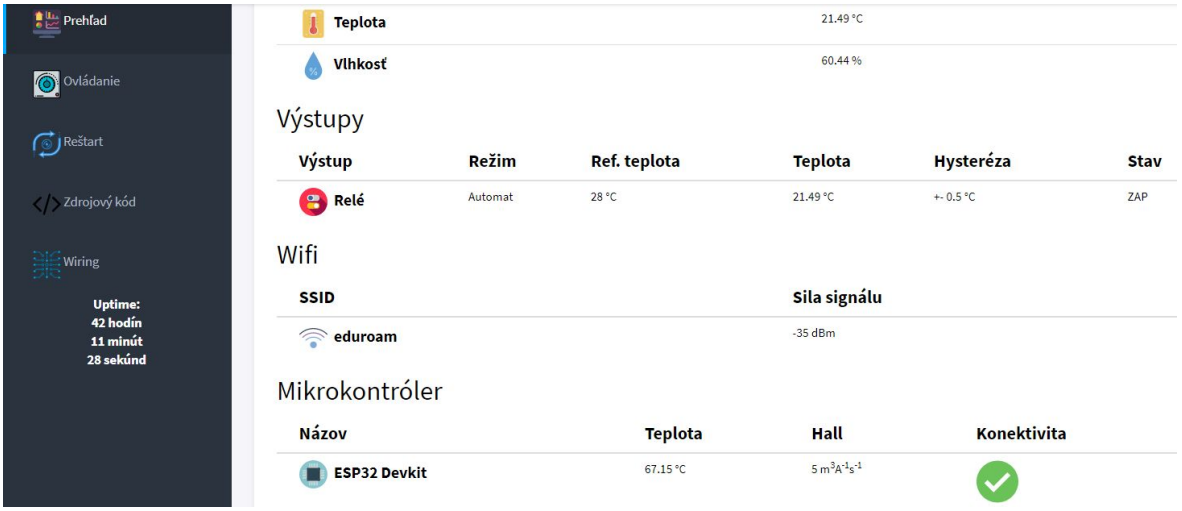
2. Návrh a realizácia inteligentného relé

Návrh a realizácia riešenia témy bakalárskej práce - inteligentného relé pozostávalo zo sledu krokov. Tvorba webového rozhrania bola prvá téma, ktorej som sa venoval, keďže to je práve mozgom celého konceptu inteligentného relé. Na blokovej schéme (Obr. 4) je možné vidieť závislosť celého systému od spojenia s webovým rozhraním. Webové rozhranie (Obr. 5) umožňuje prijímať a reprezentovať dáta, ktoré sú prijaté. Zo začiatku rozhranie umožňovalo prijímať iba teplotu a vlhkosť zo senzora, ktorý bol prepojený s vývojovou doskou.



Obr. 4 Bloková schéma inteligentného relé

Neskôr bolo rozhranie rozšírené aj na záznam systémových dát riadiaceho mikrokontroléra. Ten dokázal na webstránku posielať dáta o teplote procesora, sile signálu na WiFi bod v dBm (decibel-miliwatt) jednotkách, názov SSID WiFi siete, analógovú hodnotu magnetického hall senzora a systémový čas behu systému.



Obr. 5 Webová stránka s prehľadom dát v reálnom čase

Pre webový server som implementoval overenie klienta prostredníctvom klientského certifikátu. Webové rozhranie v sebe ukrýva aj niekoľko praktických funkcionalít. Používateľ, ktorý sa na webstránku dostane po autorizácii na základe predloženého klientského certifikátu môže využiť systém izbového termostatu, ktorý umožňuje ovládať relé výstup automaticky na základe nameranej a cieľovej teploty so zohľadnením hysterézy.

Systém funguje aj autonómne bez prítomnosti používateľa vo webovom rozhraní. Keďže je webové rozhranie umiestnené na verejnej IP adrese, je server dosiahnuteľný a možnosť ovládania je prakticky odkiaľkoľvek. Používateľ v druhom type ovládania relé - manuálnom režime môže na neobmedzený čas zapnúť, alebo vypnúť relé. Na výstup relé je možné pripojiť napájanie pre plynový kotol.

V bakalárskej práci je na výstup relé pripojená LED žiarovka, na ktorej je možné ukázať funkčnosť celého systému a demonštrovať tak autonómnosť systému. Webové rozhranie umožňuje vzdialene reštartovať vývojovú dosku, respektíve samostatný ESP čip. Webstránka slúži aj na uloženie používaných verzií zdrojových kódov pre PSK (pre-shared-key, zdieľané heslo) a Enterprise WiFi siete, pod ktorými je možné prevádzkovať inteligentné relé.

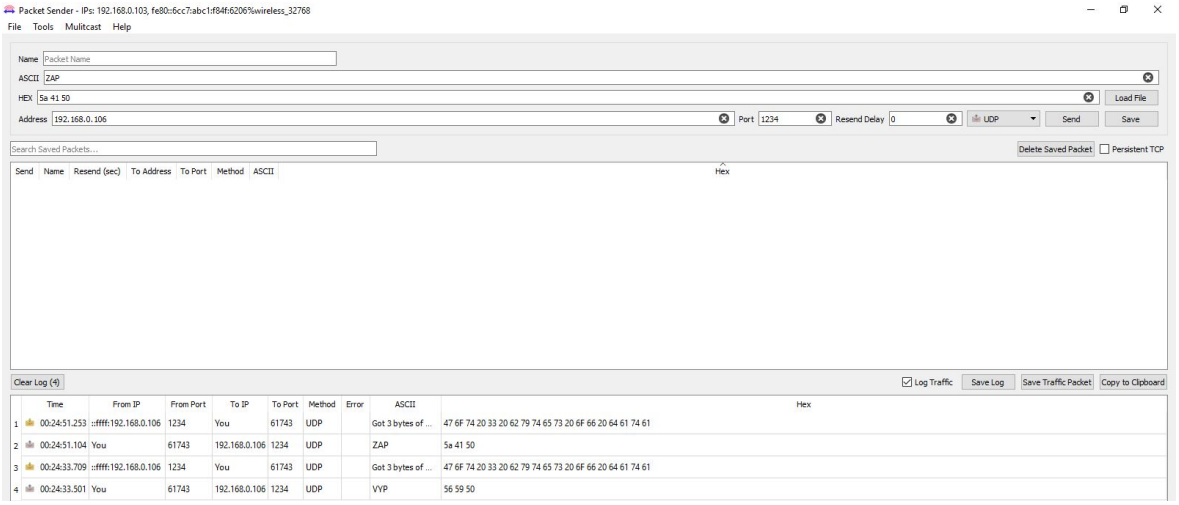
Praktická časť bakalárskej práce sa venovala predovšetkým dôrazu na bezpečnosť s využitím rôznych dostupných nástrojov, ktoré sa bezpečnosťou zaoberajú. Riadiaci mikrokontróler som pripojil na WiFi sieť eduroam, ktorá ponúka najvyšší stupeň bezpečnosti WiFi sietí v súčasnosti zabezpečovacími mechanizmami štandardu WPA/WPA2 Enterprise. Mikrokontróler som overoval voči serveru a server voči klientovi pri nadviazaní spojenia vzájomným predložením certifikátov, v dôsledku čoho som mohol využiť šifrované HTTPS [16] spojenie medzi klientom a serverom.

Spojenie poskytlo bezpečnú prenosovú trasu pre dáta, ktoré sú na web server posielané v čitateľnom formáte. Nakoľko je spojenie šifrované po celej trase, nehrozí odposluch prenášaných informácii. Minimalizuje sa riziko ich odcudzenia, zmeny, či zmazania dát. Prenášané dáta sú odosielané GET [17] metódou, čomu je uspôsobený formát reprezentácie dát, keďže tie sú vložené priamo do požiadavky na cieľový .php súbor. Viacero premenných je oddelených prostredníctvom oddelovača & a dopyt začína znakom ?. Časť programu poukazuje na spôsob reprezentácie dát mikrokontrolérom na cieľový súbor, ktoré sa predávajú serveru a je ich možné rovnakým spôsobom odoslať aj z klientského počítača:
subor.php?teplota=20.35&vlhkost=60.89&ssid=eduroam&signal=-45&hall=4&teplotacpu=68.5

Dáta sa spracujú serverovým jazykom PHP [18] prostredníctvom superglobálnych premenných GET a sú uložené do textových súborov, odkiaľ sa načítavajú a sú reprezentované pre používateľa na stránke. Aby bol systém dostatočne dynamický, využil som knižnicu jQuery [19] vo verzii 2.2.4 a využil som AJAX-ové volania (metóda jQuery knižnice) rôznych .php súborov, ktoré vypisujú obsah z .txt súborov. AJAX umožňuje v pravidelných intervaloch spúšťať cieľové .php súbory, ktoré dynamicky menia obsah stránky bez nutnosti obnovovať celú stránku. Webová stránka je vytvorená v HTML5 (hypertextový značkový jazyk verzie 5) [20] a serverovom jazyku PHP verzie 5.3.3, kompatibilné aj pre verziu 7+.

Okrem webovej verzie bakalárskej práce som vytvoril aj offline verziu pre ovládanie relé prostredníctvom externého programu pre odosielanie datagramov - Packet Sender [21], ktorý datagramy odosielal na mikrokontróler. Pri tejto konfigurácii ESP32 pasívne čaká na vybranom UDP porte, na ktorý prichádzajú požiadavky z programu. Packet Sender je open-source nástroj, aktuálne vo verzii 5.8.5, ktorý umožňuje prijímať, alebo odosielať dáta na vybraný port, pričom zohľadňuje typ portu (UDP, TCP, SSL/TLS). Podporuje plne šifrovanie s možnosťou pridania certifikátov a ich vzájomných overením.

Odosielanie dát na cieľovú adresu je možné vykonať na vyžiadanie - kliknutím, alebo automaticky v pravidelných intervaloch. Obrázok (Obr. 6) zachytáva program a jeho výpis histórie pri odosielaní textových reťazcov ZAP/VYP na čip ESP32 s odpoveďou na prijaté informácie, ktoré čip načítal.



Obr. 6 Odosielanie textových reťazcov po UDP protokole - Packet Sender

Dáta som odosielal na mikrokontróler, ktorý ich spracuje a na základe obsahu aplikuje zmeny výstupu pre relé. Výstup sériovej linky na obrázku (Obr. 7) ukazuje, aké dáta mikrokontróler získal a ako zmenil výstup pre relé na základe prijatej informácie prostredníctvom UDP protokolu. Konfiguráciu Packet Sendera bolo nutné pozmeniť a aplikovať 500 ms pauzu po pripojení pre odoslanie dát tzv. mód pre pomalšie zariadenia. Bez tohto medzikroku sa dáta na strane mikrokontroléra neobjavili a neodpovedal na tieto informácie.

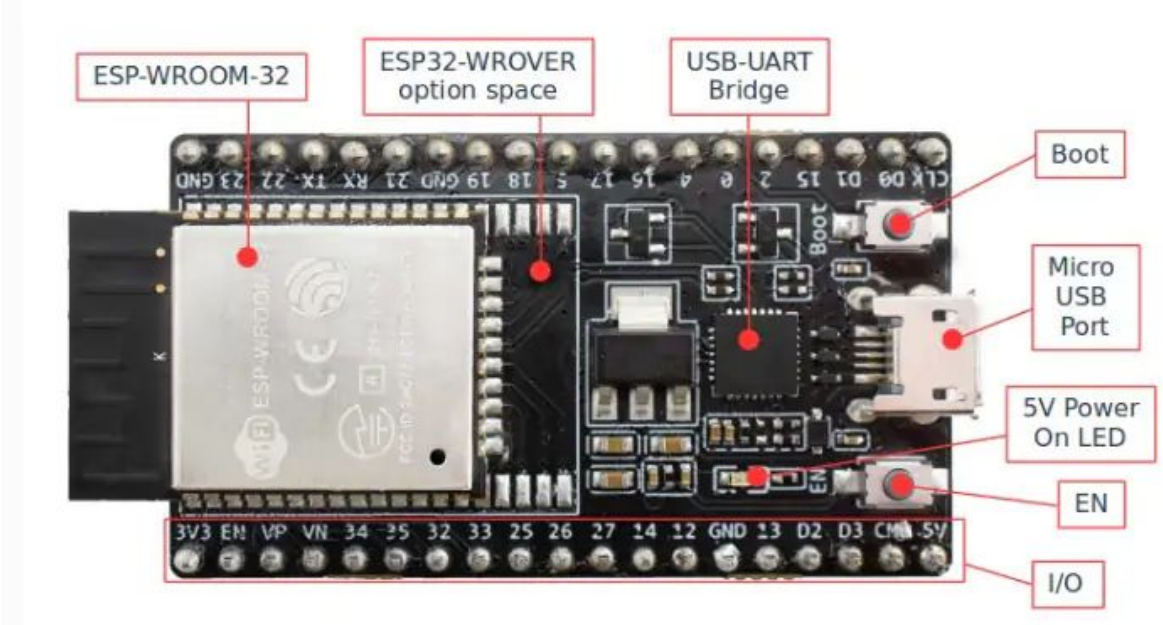
```
UDP Listening on IP: 192.168.0.106
UDP Packet Type: Unicast, From: 192.168.0.103:56063, To: 192.168.0.106:1234, Length: 3, Data: VYP
Vypinam rele
UDP Packet Type: Unicast, From: 192.168.0.103:56063, To: 192.168.0.106:1234, Length: 3, Data: ZAP
Zapinam rele
```

Obr. 7 Prijaté UDP pakety na strane mikrokontroléra - UART výstup

2.1 Použitý hardvér

Hardvérová stránka projektu je zameraná predovšetkým na využitie IoT (Internet vecí) platformy - WiFi čipu ESP32 [22] pre danú aplikáciu inteligentného relé. Čip má dostatok výpočtového výkonu a pamäte s porovnaním s príbuznými platformami Arduino až niekoľko násobne viac. Výpočtový výkon je využitý hlavne pre rôzne krypto operácie, ktoré vykonáva v našej implementácii - pripojenie do siete eduroam, šifrovaná komunikácia po HTTPS protokole, overenie klientským certifikátom voči serveru. ESP32 sa osádza do rôznych vývojových kitov a dosiek, ktoré uľahčujú prácu s čipom, keďže sa na nich okrem čipu nachádza aj prevodník napäťových úrovní spoločne s USB-UART prevodníkom CP2102 [23].

Vďaka tomu je možné dosku programovať priamo cez 5V USB konektor počítača bez nutnosti transformovať napäťové úrovne na požadovaných 3.3V, ktoré ESP32 vyžaduje. V našom prípade som využil vývojový kit ESP32-Devkitc V4 [24], ktorý je bližšie približený na obrázku (Obr. 8) vrátane popisu hlavných častí.



Obr. 8 ESP32-Devkitc V4

Zaujímavosťou na doske je, že má vyvedené vývody aj miesto pre osadenie WROVER modulu, čo je prídavná RAM pamäť pre zložitejšie projekty, napríklad real-time stream videa. Devkit je možné osadiť do prepojovacieho poľa a následne k nemu pripojiť ďalšie periférie. ESP čip má k dispozícii aj Bluetooth konektivitu s technológiou BLE (Bluetooth Low Energy). Nakoľko čip má iba jednu anténu na frekvencii 2,4 GHz, je možné v reálnom čase používať WiFi, alebo Bluetooth, nie obe technológie naraz.

Pri realizácii bakalárskej práce som využil aj samostatný modul ESP32-WROOM-32 [22], ktorý vychádzal z minimálnej schémy inteligentného relé a bol následne použitý aj v prototype inteligentného relé spolu s ďalšími perifériami. Modul ESP32-WROOM-32 je na obrázku (Obr. 9).

Obr. 9 ESP32-WROOM-32

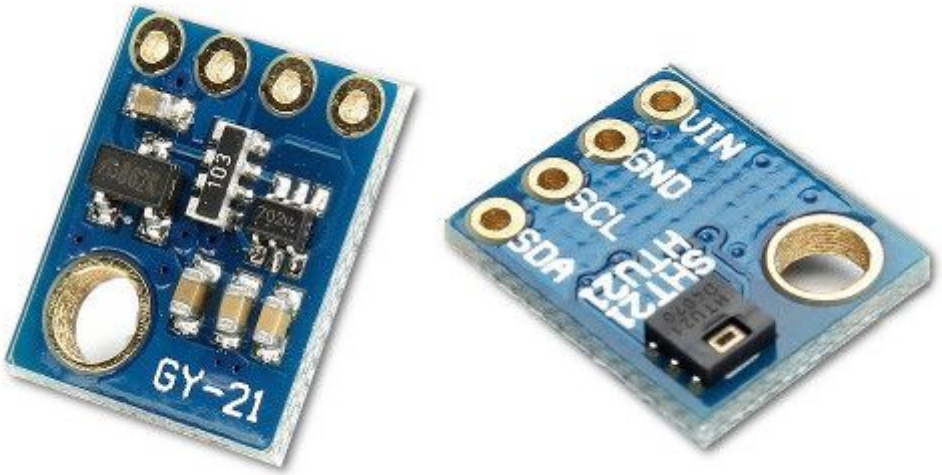
Pri programovaní samostatného čipu som využil samostatný RS232 (komunikačné rozhranie s definovanými napäťovými úrovňami) prevodník [25], ktorý som pripojil na vývody RX, TX. Ďalšie signály prevodníka - CTS, DTR pripojené neboli, keďže sú simulované ručne prostredníctvom tlačidiel na zadanie sekvencie nahrávania programu. V tabuľke (Tab. 2) je zoznam najpoužívanejších čipov ESP32 s dvojjadrovým procesorom. ESP32 má početné zastúpenie aj pri jednojadrových procesoroch, ktoré majú označenie SOLO s hardvérovým vybavením na úrovni WROOM. Jednojadrové procesory ESP-WROOM-02 neobsahujú BLE, úspornejšia prevádzka.

Tab. 2 Prehľad základných modulov ESP32 a kitov s dvojjadrovým procesorom

Modul	**Popis**
ESP32-WROOM-32	Základný modul, dual-core CPU 160/240MHz, WiFi, BLE, PCB anténa
ESP32-WROOM-32D	Vlastnosti ESP32-WROOM-32. Vhodný modul pre low-power aplikácie, stream hudby, kódovanie zvuku.
ESP32-WROOM-32U	ESP32-WROOM-32D + konektor U.FL
ESP32-WROVER	4MB externá SPI flash pamäť + 8MB externá PSRAM pamäť, vhodné pre stream videa
ESP32-WROVER-I	ESP32-WROVER + U.FL anténa
ESP32-PICO-D4	Miniatúrny ESP32 modul v rozmeroch 7x7x0,9mm vo vlastnom puzdre s pamäťou, oscilátorom. Vhodný pre aplikácie do inteligentný hodiniek a pod.
Devkity	**Popis**
Devkit V1-V4	Devkity osadené predovšetkým ESP32-WROOM-32
ESP-CAM / ESP-EYE	Vývojová doska vybavená kamerou a externou RAM pamäťou, real-time stream videa, vhodné pre rozoznanie farby, tváre.

TTGO ESP32	Vývojová doska s ESP32 a čipom SX1276 pre Long range technológiu

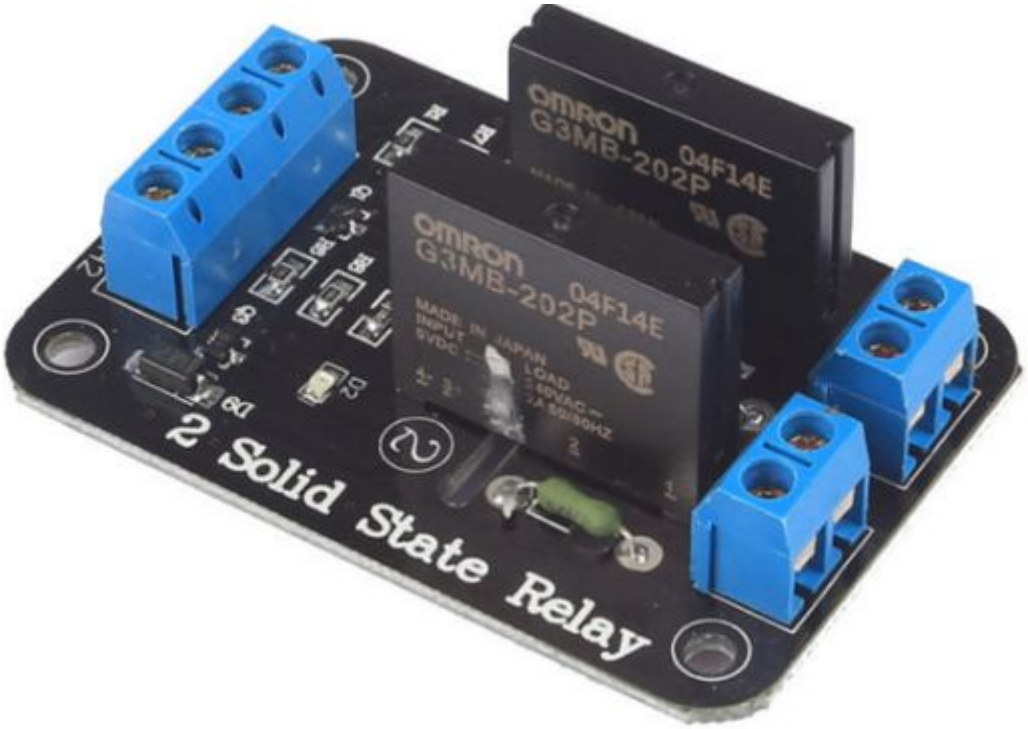
Senzor vlhkosti a teploty je dôležitou periférou celého systému inteligentného relé, od ktorého závisí funkčnosť automatického režimu pre relé, ktorý funguje na základe nameraných teplôt. Senzor je zodpovedný za merania jednotlivých veličín. Aby boli merania rýchle, využil som digitálny modul so senzorom SHT21 [26], ktorý komunikuje po I2C zbernici (komunikačné rozhranie), bližšie priblížený na obrázku (Obr. 10). Rád by som vyzdvihol konštrukciu senzora, ktorý je osadený na malom module Si7021. Modul je možné jednoducho použiť a zasunúť do pinovej lišty, konektora, alebo prostredníctvom skrutky pripevniť do iného materiálu. Pomer ceny a presnosti je výborný, najmä z pohľadu vlhkomera s rozsahom 0-100% relatívnej vlhkosti vzduchu.



Obr. 10 SHT21- senzor vlhkosti a teploty vzduchu

Senzor je potiahnutý plastovou plôškou (neobsahuje senzor na obrázku), ktorá senzor chráni pred vplyvmi okolia, ktoré by spôsobili odchýlky meraní (ohriatie senzora od dotyku, dychu, a podobne). Nakoľko som modul využíval na krátku vzdialenosť, nemusel som využiť externé zvyšovacie (pullup) rezistory, ktoré sa podľa špecifikácie I2C [27] rozhrania používajú. ESP čip má k dispozícii interné 50kΩ zvyšovacie rezistory, ktoré som použil a je ich možné softvérovo zapnúť pri inicializácii I2C zbernice.

Pre demonštráciu výstupu na fyzickom spotrebiči som využil SSR (solid state relay) relé [28] pre jeho ovládanie. SSR relé je typom polovodičového spínacieho prvku bez mechanickým častí, ktorým je možné ovládať spotrebiče a zariadenia s nižším prúdovým odberom. Použité relé sa nachádza na obrázku (Obr. 11) v dvoj-kanálovom prevedení pre ovládanie dvoch spotrebičov.



Obr.11 Relé modul OMRON G3MB-202P

Nakoľko sa v relé nenachádza mechanická časť, z teoretického hľadiska má takmer neobmedzenú životnosť. Spínacím prvkom je polovodičová súčiastka - triak [29], ktorá dokáže ovládať výstup na základe vstupu logickou 0, alebo 1.

Obvod je zopnutý ak je na tranzistor privedený active-low signál, t.j. log 0, čo vo výsledku aktivuje triak privedením 5V na riadiacu elektródu prostredníctvom tranzistora. Pri log 1 je obvod rozopnutý v dôsledku nepriechodnosti triaku. Triak dokáže viesť prúd oboma smermi. Schematicky pripomína dva tyristory so spoločnou riadiacou elektródou. Triak je navrhnutý pre spínanie striedavého napätia a prúdu.

Na jednosmerné obvody triak nefunguje. Dokáže obvod pripojiť ale už nie rozpojiť. Existujú však vyhotovenia, ktoré je možné použiť aj na jednosmerné prúdu a napätia. Relé modul OMRON G3MB-202P [30], ktorý som využil má spínacie charakteristiky pre spínanie maximálne 2A pri 230V striedavých.

Relé modul bol vložený do elektrikárskej krabice, ktorá chráni prevádzkovateľa okrem priamym dotykom so živou časťou aj pred vnikom prachu a striekajúcej vode zo všetkých smerov, spĺňa štandard IP54 (štandard ochrany krytím). Modul je napájaný na 5V. Riadenie z ESP čipu je realizované 3.3V GPIO vývodom pripojeným na NPN (druh tranzistora s dvomi polovodičmi typu N a jedným P) tranzistor. V TTL (logika v tranzistorových digitálnych obvodoch) logike je 3.3V

log 1 rovnako ako 5V [31]. Výstup relé je teda možné ovládať aj cez 3.3V vývod. Pre detekciu pripojenia k sieti bola do obvodu dosadená červená LED dióda pre vizualizáciu tohto stavu. Používateľ ihneď pri pohľade na vývojový kit, alebo čip vie, či je stále pripojený k sieti a relé aktívne. V tabuľke (Tab. 3) je opis pripojenia jednotlivých vývodov čipu ESP32 k periférii. Elektrotechnická schéma zapojenia je dostupná v kapitole 3 pri minimálnej schéme spolu s napájacou časťou v prehľadnej veľkosti.

Tab. 3 Pripojenie vývodov ESP32 k vývodom periférii

ESP32	**SHT21 (senzor teploty, vlhkosti vzduchu)**
3.3V	Vcc
GND	GND
GPIO22 (Hardware SCL)	SCL
GPIO21 (Hardware SDA)	SDA
ESP32	**Relé OMRON G3MB-202P * (externé napájanie)**
GPIO23	CH1
GND	GND
ESP32	**Signalizačná LED dióda**
GPIO18	Anóda
GND	Katóda

2.2 Pripojenie do siete eduroam s ESP32

IoT čip ESP32 plne podporuje pripojenie do WiFi siete eduroam pod štandardom 802.1X. Podporovanými sú aj dve hlavné metódy overenia preukázaním identity a hesla s možnosťou doplnenia o anonymnú identitu, PEAP + MsCHAPv2, EAP-TTLS + MsCHAPv2. Kryptofunkcie pre identitu a heslo sú obsiahnuté v hlavičkovom súbore esp_wpa2.h. Funkcie knižnice sú implementované do programu pre ESP32. Je možné využiť anonymnú identitu spolu s identitou a heslom používateľa. Prostredníctvom krátkej programovej implementácie v jazyku Wiring [32] je možné pripojiť sa k Enterprise WiFi sieti eduroam aj z inej univerzity, organizácie v projekte eduroam, keďže obsahuje kompletný realm domovskej organizácie. Výpis dôležitých informácii je na rozhranie UART - sériovú linku, ktorá je nastavená na rýchlosť 115200 znakov za sekundu. Výpis je možné obohatiť aj o výpis systémových informácii zapnutím Debug/Verbose módu pred nahratím programu. Mód sa volí spolu s doskou v záložke prostredia Arduino IDE - Manažér

dosiek. Kompletné zdrojové kódy s komentárom a dokumentáciou pre mikrokontróler, webové rozhranie sú umiestnené v prílohe A na CD nosiči.

Programová implementácia pripojenia k sieti eduroam pre ESP32 v prostredí Arduino IDE:

```
#include <WiFi.h>
#include "esp_wpa2.h"
#define ANONYMOUS_EAP_IDENTITY "anonymous@tuke.sk"
#define EAP_IDENTITY "id@tuke.sk"
#define EAP_PASSWORD "heslo"
const char* ssid = "eduroam";
void setup() {
Serial.begin(115200);
Serial.print("Pripajam sa na wifi siet: ");
Serial.println(ssid);
WiFi.disconnect(true);
WiFi.mode(WIFI_STA);
esp_wifi_sta_wpa2_ent_set_identity((uint8_t*)ANONYMOUS_EAP_IDENTITY,
strlen(ANONYMOUS_EAP_IDENTITY));
esp_wifi_sta_wpa2_ent_set_username((uint8_t*)EAP_IDENTITY,strlen(EAP_IDENTITY));
esp_wifi_sta_wpa2_ent_set_password((uint8_t*)EAP_PASSWORD,strlen(EAP_PASSWORD));
esp_wpa2_config_t config = WPA2_CONFIG_INIT_DEFAULT();
esp_wifi_sta_wpa2_ent_enable(&config);
WiFi.begin(ssid);
while (WiFi.waitForConnectResult() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi pripojene");
Serial.println("IP addresa nastavena: ");
Serial.println(WiFi.localIP());
}
void loop() {
}
```

2.3 OpenSSL a generovanie certifikátov pre projekt

OpenSSL [33] je voľne dostupný softvér pre SSL (vrstva bezpečných socketov) a TLS protokoly pre komerčné aj nekomerčné využitie pod Apache licenciou. Obsahuje kryptografické knižnice a funkcie, prostredníctvom ktorých je možné generovať aj vlastné certifikáty, kľúče rôznych formátov s rôznymi podporovanými algoritmami. Pre našu web aplikáciu Inteligentné relé v sieti eduroam som generoval 3 druhy certifikátov, ktoré sa používajú.

Certifikáty [34] sú definované štandardom X.509 pod organizáciou ITU-T. Štandard je forma odporúčania, ktorá hovorí o tom, aký všeobecný tvar majú certifikáty spĺňať, definuje to najmä verzia:

1. Verzia 1

- Sériové číslo certifikátu - nezáporné celé číslo, jednoznačne definované bez duplicity
- Algoritmus vytvorenia digitálneho podpisu CA - algoritmus pre podpis certifikátu
- Identifikačné údaje CA - obsahuje meno CA, jedinečné meno autority
- Doba platnosti certifikátu - počet dní platnosti certifikátu
- Identifikačné údaje používateľa - identifikuje držiteľa certifikátu, vlastní súkromný kľúč
- Verejný kľúč používateľa - obsahuje identifikátor algoritmu a verejný kľúč používateľa
- Digitálny podpis CA - hash kód, zašifrovaný súkromným kľúčom CA. Verifikácia iných kľúčov prostredníctvom súkromného kľúča CA.

2. Verzia 2

- Údaje verzie 1 +
- Jednoznačný identifikátor CA - jednoznačne identifikuje CA, ak jej meno bolo využité v iných CA, alebo ak nepostačuje pre jednoznačnú identifikáciu CA
- Jednoznačný identifikátor používateľa - týka sa držiteľa certifikátu, je ho možné doplnkovo jednoznačne identifikovať údajom, napríklad e-mailovou adresou

3. Verzia 3

- Údaje verzie 1, 2 +
- Rozšírenia - obsahuje doplnkové informácie o kľúčoch CA a používateľa, identifikátory CA, používateľa, obmedzenia týkajúce sa vydávania certifikátov.

Certifikát certifikačnej autority je certifikát, ktorým je podpísaný certifikát klienta, servera. Globálne uznávaný a dôveryhodný. Najčastejšie sa tieto certifikáty označujú ako tzv. Root - koreňové certifikačné autority, patria tu napríklad organizácie, ktoré sa zaoberajú vydávaním serverových certifikátov: Digicert, Thawte, TERENA, RapidSSL. V našom prípade som certifikát

vygeneroval s doménovým menom servera s údajmi pre našu organizáciu s nástrojom OpenSSL a jeho kryptografickými knižnicami v Linuxe - prostredí CentOS priamo v príkazovom riadku systému.

Generovali som RSA (kryptosystém Rivest–Shamir–Adleman) certifikáty s dĺžkou 1024 bitov [35]. Problém nastal s implementáciou certifikátu certifikačnej autority a klientského certifikátu na mikrokontróler, ktorý certifikáty nepodporoval a hlásil chybu nekompatibility z dôvodu nesprávneho formátu, bližšie však chyba nebola špecifikovaná. Z dokumentácie krypto knižnice Mbed TLS [36], ktorú ESP32 používa nebolo známe, aký problém sa vyskytol. Riešenie sa podarilo nájsť pri komunite vývojárov, ktorý sa na ESP32 Arduino core podieľajú. Riešenie spočívalo v generovaní certifikátu s dĺžkou minimálne 2048 bitov, ktorý už mikrokontróler podporoval. Certifikáty som generoval vo formáte .pem, ktorý je čitateľný a bolo ho možné aplikovať aj do mikrokontroléru pre implementáciu certifikačnej autority. Certifikáty môže modifikovať a vytvárať jedine certifikačná autorita pomocou jej súkromného kľúča, ktorý nie je známy iným stranám. V mojej implementácii je certifikát certifikačnej autority podpísaný sám sebou. Pre generovanie certifikátu som využil príkazy nástroja OpenSSL:
openssl genrsa -out myCA.key 2048 openssl req -x509 -config certificate-authority-options.conf -new -nodes -key myCA.key -sha256 -days 1825 -out myCA.pem openssl x509 -outform pem -in myCA.pem -out myCA.crt.

Certifikát servera - vydaný pre server, podpísaný certifikačnou autoritou. Server predkladá svoj certifikát klientovi pri hello správe. Certifikát je validný na základe dôvernej certifikačnej autority, ktorej odtlačok je obsiahnutý v certifikáte. Certifikát certifikačnej autority musí byť nainštalovaný na strane servera aj klienta.

Certifikát bol vydaný pre webový server, ktorý slúži na zber dát z vývojovej dosky. Väčšina webových prehliadačov sebou podpísané certifikáty okamžite označuje ako nebezpečné a prístup zamietne a používateľa na danú adresu webu, kde je nastavený takýto serverový certifikát nepustí ani na vlastné riziko. Z toho dôvodu je nutné certifikát certifikačnej autority nainštalovať v .crt (spustiteľnom) formáte do Dôveryhodných certifikačných autorít.

V operačnom systéme Windows je možné certifikát vložiť prostredníctvom sprievodcu dvojklikom na certifikát a výberom umiestnenia certifikátu do Dôveryhodných koreňových certifikačných autorít. Operačné systémy mobilných telefónov, napríklad Android obsahujú jednoduché rozhranie

na inštaláciu certifikátu certifikačnej autority. Serverový certifikát som nástrojom OpenSSL vygeneroval príkazmi:
openssl genrsa -out server.key 2048 openssl req -config options.conf -new -key server.key -out server.csr openssl x509 -req -in server.csr -CA myCA.pem -CAkey myCA.key -CAcreateserial -out server.pem -days 1825 -sha256 -extfile server.ext openssl x509 -outform pem -in server.pem -out server.crt

Certifikát klienta - nakoľko v našom projekte riešim obojstrannú autentizáciu pre klienta i server, generoval som aj klientský certifikát, ktorý preukáže identitu používateľa voči serveru. Klient predkladá svoj certifikát na vyzvanie od servera až po overení servera klientom. Súbor príkazov pre generovanie klientského certifikátu nástrojom OpenSSL:
openssl genrsa -out client.key 2048 openssl req -config options.conf -new -key client.key -out client.csr openssl x509 -req -in client.csr -CA myCA.pem -CAkey myCA.key -CAcreateserial -out client.pem -days 1825 -sha256 -extfile client.ext openssl pkcs12 -inkey client.key -in client.pem -export -out client.pfx

Každý certifikát môže byť svojou certifikačnou autoritou predčasne ukončený, vykoná sa tzv. revoke. Je to cielené znefunkčnenie certifikátu pred dobou jeho expirácie. Napríklad z dôvodu odcudzenia, zle vygenerovanom certifikáte, vojenskom konflikte, alebo z dôvodu ohrozenia bezpečnosti v prípade certifikačných systémov.

2.4 Inštalácia a aplikovanie certifikátov do zariadení

Pre úspešné zabezpečenie web aplikácie bolo potrebné vykonať zmeny na strane webového servera a klientov, či už na strane operačného systému používateľa, tak aj na ESP32. Operačný systém webového servera - CentOS obsahuje balík pre webový server httpd [37].

Balík obsahuje konfiguračné nástroje v priečinku /etc/httpd/conf.d, prostredníctvom ktorých je možné web serveru nastaviť certifikát certifikačnej autority, serverový certifikát, zreťazenie certifikátov, overenie klienta certifikátom, ktorý predloží. Upravoval som súbor ssl.conf, ktorý obsahuje konfiguračné príkazy práve pre overenie certifikátom a HTTPS spojenia, ale aj ďalšie, napríklad nastavenie presmerovania a ďalších zabezpečovacích prvkov.

Ssl.conf súbor som využil aj pre vyžiadanie HTTPS spojenia medzi klientom a serverom vždy z dôvodu vyššej bezpečnosti, to znamená, že ak klient pristupuje na HTTP verziu stránky, automaticky je presmerovaný na HTTPS verziu a spojenie je šifrované a vyžaduje sa jeho

preukázanie certifikátom. Konfiguračný súbor balíka httpd som mohol upravovať priamo z konzoly web servera nástrojom pre úpravu textových dokumentov, napr: nano, či Vim.

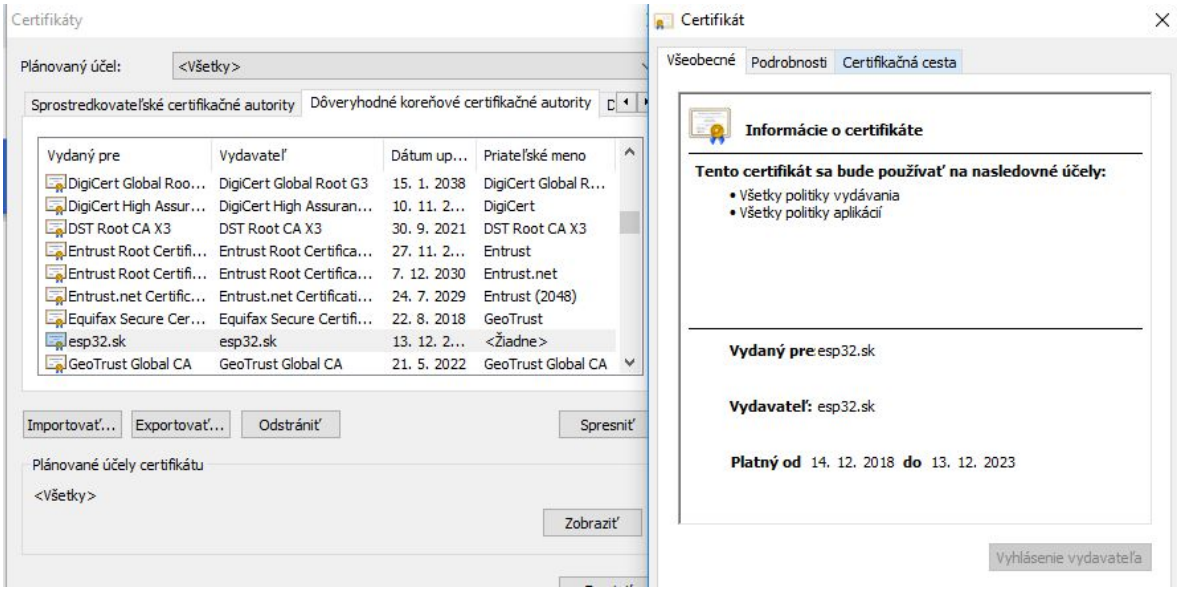
Konfiguračné príkazy pre certifikáty certifikačnej autority a servera vrátane jeho kľúča obsahujú ich umiestnenie. Pre overenie klient je nutné nastaviť pre príkaz SSLVerifyClient možnosť require - vyžadovať. Druhou položkou pri overení klienta, ktorú už nie je nutné nastaviť je hĺbka overenia, t.j. počet skokov po Root CA certifikát.

Konfigurácia pre súbor ssl.conf:

ServerAlias esp32.sk
SSLCACertificateFile /etc/certifikaty/myCA.crt
SSLCertificateFile /etc/certifikaty/server.crt
SSLCertificateKeyFile /etc/certifikaty/server.key
#OVERENIE KLIENTA - komentar
SSLVerifyClient require
SSLVerifyDepth 10

Prvým inštalovaným klientským zariadením pre aplikovanie a dôvernosť certifikačnej autority bol operačný systém Windows 10. Certifikát som si z webového servera stiahol v .crt formáte, t.j. spustiteľný .pem formát, ktorý dokáže nainštalovať sprievodca inštaláciou certifikátov. Pri inštalácii certifikátu som si zvolil jeho umiestnenie do Dôveryhodných koreňových certifikačných autorít.

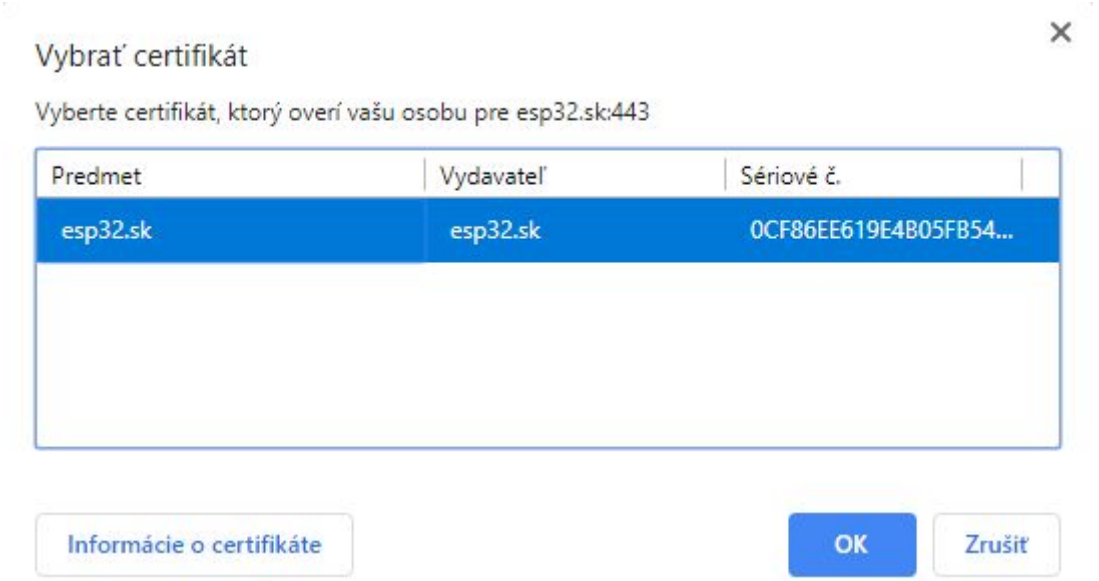
Certifikát je možné inštalovať aj priamo do prehliadača, avšak bude aplikovaný iba v prehliadači. Od tohto momentu je naša vlastná certifikačná autorita dôveryhodná na tomto zariadení a pripojenie na webstránku bolo úspešné (za predpokladu neoverovania klienta). Certifikát sa zaradil do zoznamu certifikačných autorít, ktorým operačný systém Windows dôveruje. Pri pohľade do správcu certifikátov môžeme náš certifikát nájsť, zobraziť si ho a prehliadnuť si jeho informácie. Na obrázku (Obr. 12) je prehľad certifikátov Dôveryhodných certifikačných autorít v prehliadači Google Chrome, kde som nainštaloval predmetný certifikát vlastnej certifikačnej autority.



Obr. 12 Nainštalovaný certifikát certifikačnej autority v prehliadači Google Chrome

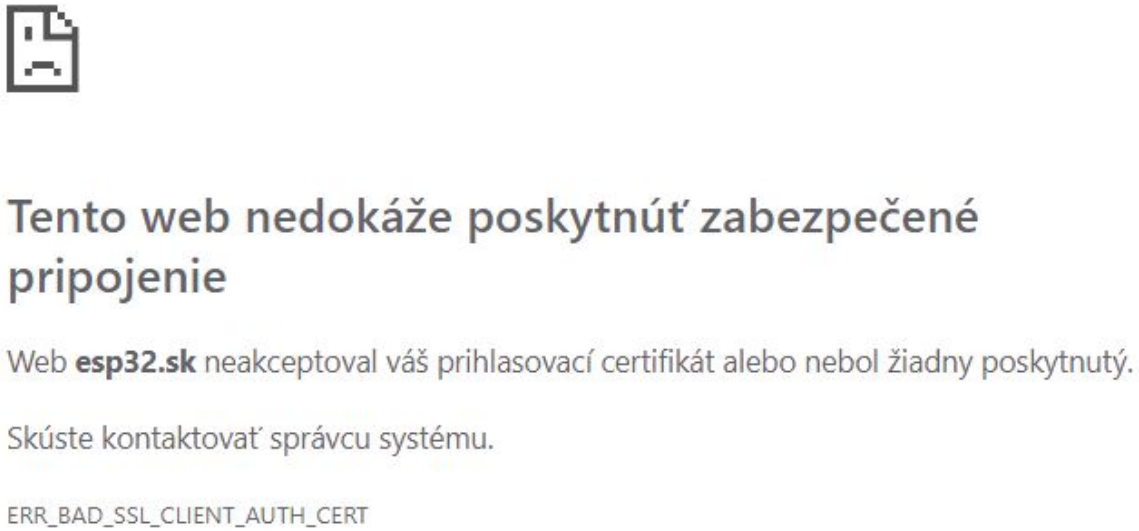
Keďže som chcel overovať aj klienta a zabezpečiť tak obojstrannú autentizáciu, bolo nutné nainštalovať aj klientský certifikát. Certifikát som v nástroji OpenSSL vygeneroval v .pfx formáte, ktorý obsahuje súkromný kľúč klienta a certifikát. Aby bol kľúč chránený, je nutné naň aplikovať heslo. Heslo sa aplikuje priamo pri generovaní nástrojom OpenSSL. Následne ho klient použije pri inštalácii certifikátu na jeho otvorenie.

Certifikát som inštaloval priamo do prehliadača Google Chrome do osobných certifikátov. Od tohto momentu je možné pripojiť sa na web stránku a preukázať sa klientským certifikátom. Vyzvanie na predloženie certifikátu je vyskakovacie okno prehliadača (Obr. 13), ktoré umožní klientovi vybrať si, akým certifikátom sa preukáže. V prípade, že sa preukáže platným certifikátom, úspešne sa pripojí na webstránku, ukáže sa mu obsah a môže stránku plne využívať.



Obr. 13 Výzva na predloženie klientského certifikátu (Windows)

Ak sa klient nemôže preukázať certifikátom, alebo sa preukáže neplatným certifikátom, je automaticky odmietnutý a na webstránku sa nedostane (Obr. 14). Webový server spojenie ukončí.



Obr. 14 Neúspešné overenie klienta serverom

Vývojovú dosku s čipom ESP32 som programoval v zjednodušenou jazyku C, tzv. Wiring vo vývojovom prostredí Arduino IDE. Do prostredia som si doinštaloval podporu vývojových dosiek, kitov s čipom ESP32. V záložke Súbor → Vlastnosti som do okna Manažér dosiek URLs pridal JSON (formát dát) adresu, ktorá načíta dostupné dosky vrátane potrebných knižníc [38]:
https://dl.espressif.com/dl/package_esp32_index.json

V záložke Nástroje → Doska → Manažér dosiek som vyhľadal balík ESP32 a nainštaloval som ho na verziu 1.0.1. Inštalácia je rýchla a okrem podpory dostupných typov ESP32 dosiek sú doinštalované aj knižnice, ktoré sa používajú pri práci s ESP32.

Pre implementáciu certifikátov a samotné HTTPS spojenia existujú dve možnosti. Prvým spôsobom je využitie hlavičkového súboru WifiClientSecure.h. V programovej implementácii touto metódou sa používajú funkcie pre jednorázové načítanie certifikátov .pem formátu, ktoré následne aplikuje do funkcie client.connect(); pri pripájaní na cieľovú lokalitu. Druhou možnosťou je využitie hlavičkového súboru HTTPClient.h. Systém aplikácie certifikátov je odlišný a musia byť vložené manuálne do každej odosielacej funkcie http.begin(); pre príslušný parameter a taktiež obsluha prijatej odpovede zo servera je zložitejšia na prevzatie a následnú prácu s prijatou informáciou.

Po zvážení výhod jednotlivých metód som sa rozhodol pre využitie hlavičkového súboru WifiClientSecure.h. Certifikáty som z .pem formátu prepísal do premenných, ktoré som následne použil do funkcií, ktoré sa používajú pri pripojení na HTTPS lokalitu. Priložená programovaná implementácia ukazuje vzorové nastavenie certifikátov pre certifikačnú autoritu, klienta a klientského súkromného kľúča, ktorý ESP32 používa pri pripojeniach. Certifikáty sú zmenšené na prijateľnú dĺžku pre sprehľadnenie programu.

Programová implementácia certifikátov a HTTPS spojenia:

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "esp_wpa2.h"
#define ANONYMOUS_EAP_IDENTITY "anonymous@tuke.sk"
#define EAP_IDENTITY "id@tuke.sk"
#define EAP_PASSWORD "heslo"
const char* ssid = "eduroam";
const char* host = "www.esp32.sk";
const char* test_root_ca = \ "-----BEGIN CERTIFICATE-----\n" \
"psVaEdVBIz/a4jcB6mkQQm06+KusHmwsc/VNYLD6+pjmIE0=\n"\
"-----END CERTIFICATE-----\n";
const char* test_client_key = \
"-----BEGIN RSA PRIVATE KEY-----\n" \
"Xi6z101s2PqWmL0I8Kky85wIs+HyjJw+jnRqa37z+8AJIQX9xMUVzK4=\n" \
```

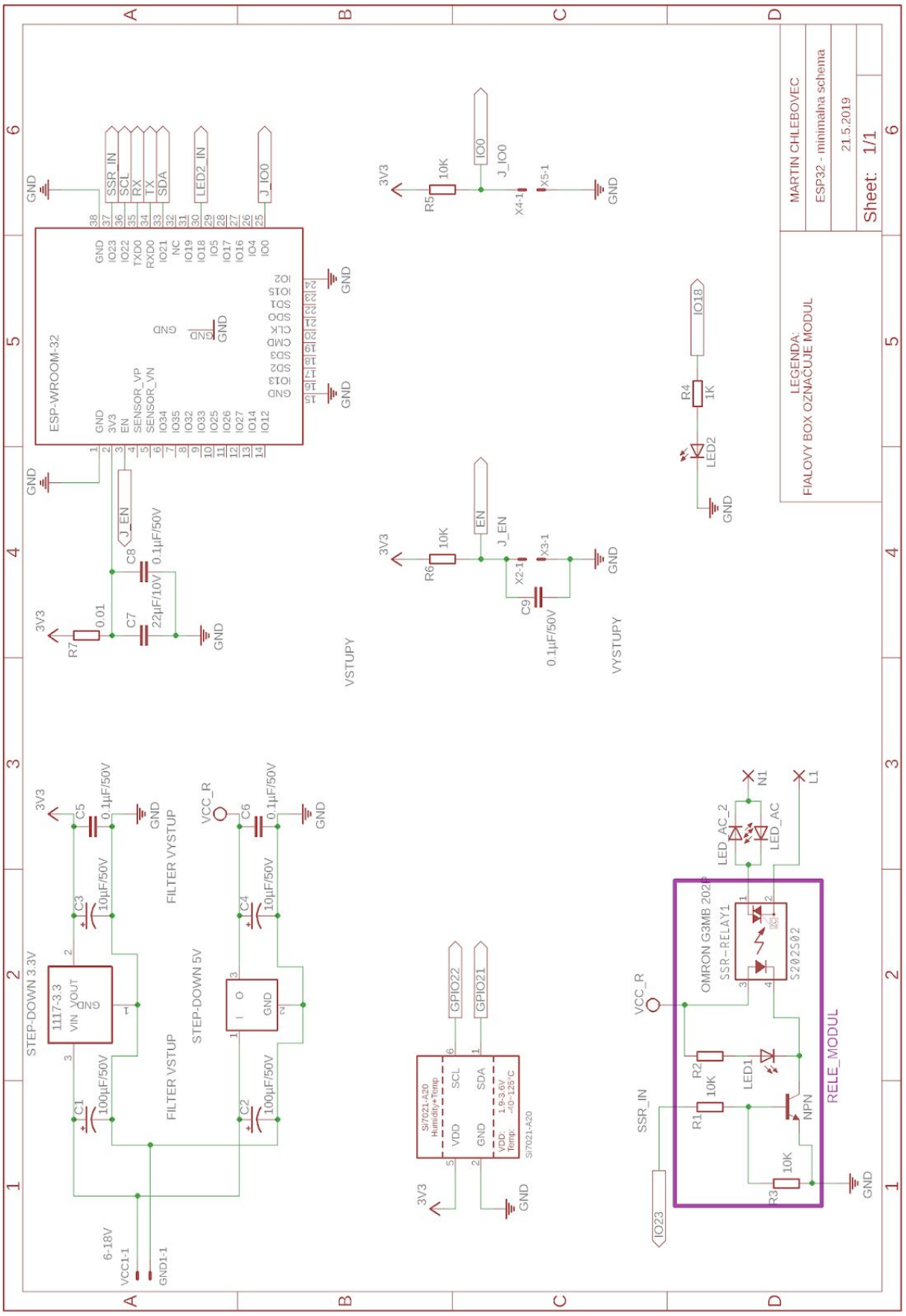
```
"-----END RSA PRIVATE KEY-----\n";
const char* test_client_cert = \ "-----BEGIN CERTIFICATE-----\n" \
"MIIEIzCCAwugAwIBAgIUDPhu5hnksF+1QHC9qFqLPCv8rHMwDQYJKoZIhvcNAQEL\n" \
"DptkN2Jdaw==\n" \
 "-----END CERTIFICATE-----\n";
WiFiClientSecure client;
void setup() {
Serial.begin(115200);
delay(10);
Serial.println();
WiFi.disconnect(true);
WiFi.mode(WIFI_STA);
esp_wifi_sta_wpa2_ent_set_identity((uint8_t*)ANONYMOUS_EAP_IDENTITY,
strlen(ANONYMOUS_EAP_IDENTITY));
esp_wifi_sta_wpa2_ent_set_username((uint8_t*)EAP_IDENTITY,          strlen(EAP_IDENTITY));
esp_wifi_sta_wpa2_ent_set_password((uint8_t *)EAP_PASSWORD, strlen(EAP_PASSWORD));
esp_wpa2_config_t config = WPA2_CONFIG_INIT_DEFAULT();
esp_wifi_sta_wpa2_ent_enable(&config);
WiFi.begin(ssid);
while (WiFi.status() != WL_CONNECTED) {
digitalWrite(led, LOW);
delay(500);
Serial.print(".");
}
client.setCACert(test_root_ca);
client.setCertificate(test_client_cert);
client.setPrivateKey(test_client_key);
Serial.println("");
Serial.println("WiFi pripojene");
Serial.println("IP addresa nastavena: ");
Serial.println(WiFi.localIP());
}
void odoslanie() {
if (client.connect(host, 443)) {
Serial.println("Pripojenie pre odoslanie dat uspesne");
```

```
String url = "/esp32/zapisdata.php?..."; client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" + "User-Agent: ESP32\r\n" + "Connection: close\r\n\r\n");
}
else {
Serial.println("Nepodarilo sa odoslat data");
}
client.stop();
}
void loop() {
odoslanie();
}
```

Overenie klienta prebieha pri každom pripojení na web server v dôsledku čoho sa predlžuje časová odozva jednej bežiacej slučky programu, ktorý odosiela dáta a načítava stav resetu, stav pre relé. Celková doba jednej slučky programu je približne 9 sekúnd.

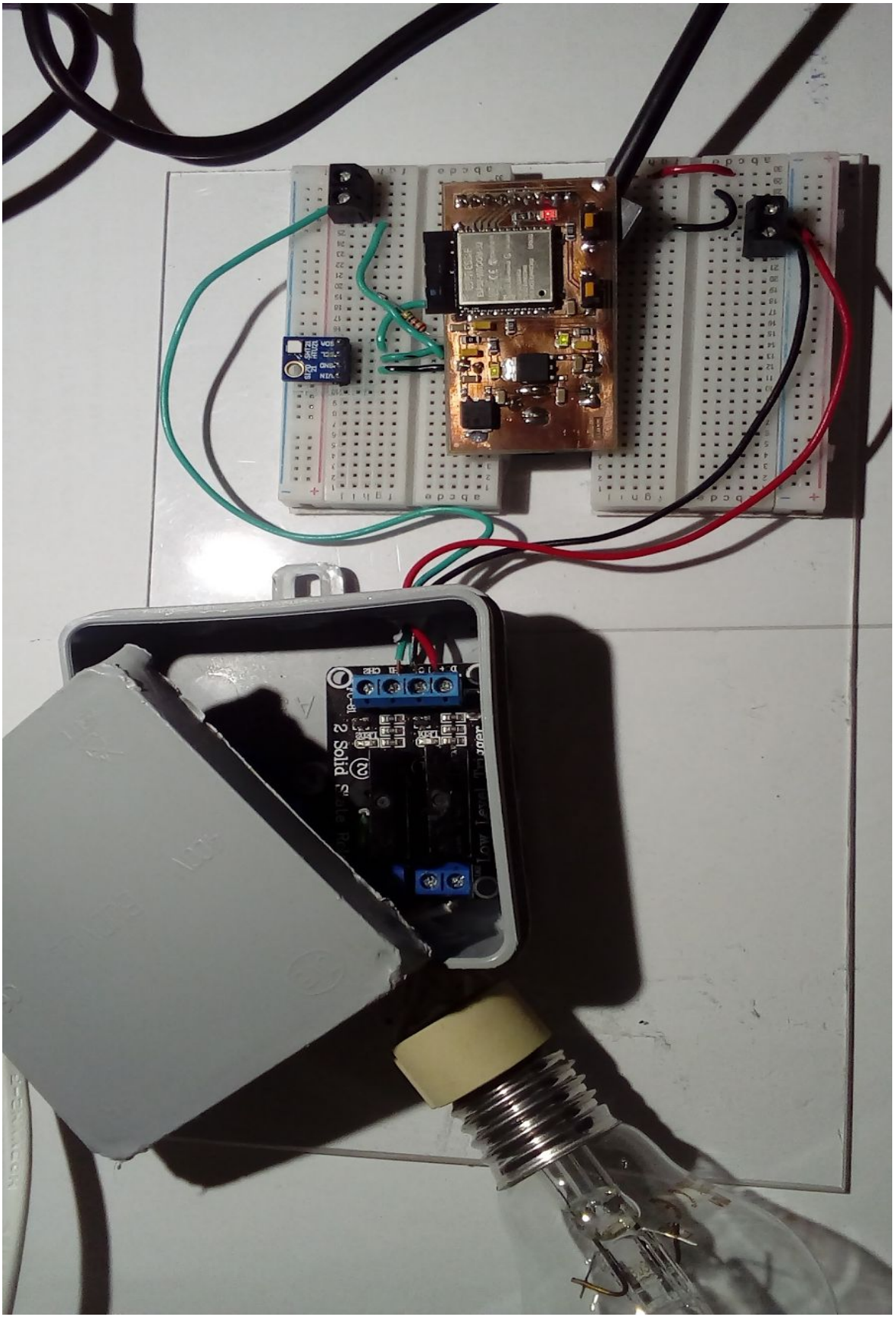
3. Minimálna schéma a prototyp s čipom ESP32-WROOM-32

Súčasťou práce s hardvérom v rámci bakalárskej práce bol aj návrh minimálnej schémy so samostatným čipom ESP32-WROOM-32. Schéma obsahovala kompletný návrh napájacej časti pre ESP čip, zakreslenie relé modulu, vrátane tlačidiel pre ovládanie signálov pre EN (RTS) a BOOT (DTR) vývody, čo je potrebné pri nahrávaní programu do čipu. Na obrázku (Obr. 15) je možné nahliadnuť do elektrotechnickej schémy výkresu, ktorý bol nakreslený v nástroji Autodesk Eagle [39].



Obr. 15 Minimálna schéma s čipom ESP32-WROOM-32

Schéma sa stala základom pre prototyp, ktorý bol vyhotovený na pracovisku Katedry elektroniky a multimediálnych telekomunikácií Ing. Stanislavom Slovákom, PhD., ktorý pripravil aj súborovú dokumentáciu k prototypu. Dokumentácia je obsiahnutá v prílohách a obsahuje návrh DPS, 3D vizualizáciu modelu so súpisom použitých súčiastok, technický výkres osadenia prototypu, súborom pre tlač na papier a výrobu prototypu fotocestou. Prototyp obsahuje napájaciu časť s možnosťou pripojenia externého zdroja pre relé, pričom má vyvedené pinové lišty pre pripojenie senzorov a ovládacieho vývodu relé. Priložený obrázok (Obr. 16) ukazuje náhľad minimálneho prototypu s využitím čipu ESP32-WROOM-32. Prúdová špička pri bootovaní čipu je 480mA pri 3.3V. Nakoľko je napájacia časť dostatočne navrhnutá z pohľadu filtračných kondenzátorov, nestretol som sa s problémami slabo dimenzovaného napájania. Vo výsledku by sa problémy s nedostatočným prúdovým napájaním prejavilo reštartovaním čipu počas procesu bootovania, prípadne odosielania dát.



Obr. 16 Prototyp s čipom ESP32-WROOM-32

Záver

Práca opisuje využitie IoT platformy ESP32 v implementácii inteligentného relé v sieti eduroam. Počas tvorby práce som sa lepšie oboznámil s dostupnými zabezpečovacími prostriedkami, ktoré je možné aplikovať na úrovni LAN siete a taktiež na komunikačnú cestu na vzdialený server či už overením servera s vlastnou certifikačnou autoritou, alebo aj s obojstranným overením klienta a servera.

Kryptografický nástroj OpenSSL som sa naučil používať na úrovni operačného systému v príkazovom riadku ako aplikáciu, ale aj v jazyku C s využitím knižníc tohto nástroja. Hlavným prínosom práce je poukázanie na možnosť využitia ESP32 v sieťach eduroam, ale aj iných pod štandardom 802.1X.

V súčasnosti neexistovala podobná platforma, ktorá by mala tento zabezpečovací mechanizmus implementovaný. V práci som využil dômyselný systém zabezpečenia v spojitosti s reprezentáciou dát a systémom inteligentného relé, ktoré umožňuje vzdialene na základe nameraných teplôt monitorovať a ovládať kúrenie v domácnosti na spôsob inteligentného termostatu prístupného odkiaľkoľvek cez internet.

Okrem on-line riešenia sa mi podarilo vytvoriť aj off-line verziu pre ovládanie UDP datagramami v sieti s využitím asynchrónneho programu pre ESP32. Všetky problémy, ktoré počas riešenia práce nastali sa mi podarilo vyriešiť a tieto poznatky môžem využiť aj v ďalšom postupe vo sfére mikrokontrolérov a ich implementácie do rôznych aplikácii vrátane nadobudnutých znalostí z oblasti kreslenia elektrotechnických schém. Využitie platformy ESP32 je možné aplikovať do výuky aj vďaka kompatibilite so sieťou eduroam pre tvorbu IoT projektov a zaujímavých aplikácii priamo na univerzitách a na výskumných pracoviskách.

Zoznam použitej literatúry

[1]. eduroam [online] Wikipedie [cit 2018-12-01]. Dostupné z: https://cs.wikipedia.org/wiki/eduroam

[2]. ÚVT - Nastavenia [online]. Ústav výpočtovej techniky TUKE [cit. 2019-05-22]. Dostupné z: https://nastavenia.tuke.sk/wifi/

[3]. NTLM [online]. Wikipedie [cit. 2019-02-15]. Dostupné z: https://cs.wikipedia.org/wiki/NTLM

[4]. Sanet [online]. ©Sanet [cit. 2019-05-22]. Dostupné z: http://www.sanet.sk/eduroam.shtm

[5]. MAC adresa [online]. Wikipédia [cit 2016-10-31]. Dostupné z: https://sk.wikipedia.org/wiki/MAC_adresa

[6]. LinuxDays 2016 - eduroam tajemství zbavený - Ondřej Caletka [online]. LinuxDays [cit 2017-01-07]. [video súbor]. Dostupné z: https://bit.ly/2QJ0blS

[7]. Shared Secrets - RADIUS [Book] [online]. Safari Books Online [cit. 2019-05-22]. Dostupné z: https://www.oreilly.com/library/view/radius/0596003226/ch02s04.html

[8]. Suplikant [online]. Wikipedie [cit 2018-09-05]. Dostupné z: https://cs.wikipedia.org/wiki/Suplikant

[9]. EAPoL - Extensible Authentication Protocol over LAN [online]. VOCAL Technologies, Ltd. [cit. 2019-05-22]. Dostupné z: https://www.vocal.com/secure-communication/eapol-extensible-authentication-protocol-over-lan/

[10]. Model OSI [online]. Wikipédia [cit 2018-12-14]. Dostupné z: https://sk.wikipedia.org/wiki/Model_OSI

[11]. IEEE 802.1X [online]. Wikipedie [cit 2018-07-11]. Dostupné z: https://cs.wikipedia.org/wiki/IEEE_802.1X

[12]. RADIUS [online]. Wikipedia [cit 2019-02-16]. Dostupné z: https://en.wikipedia.org/wiki/RADIUS

[13]. IPsec [online]. Wikipedie [cit 2019-05-12]. Dostupné z: https://cs.wikipedia.org/wiki/IPsec

[14]. RadSec [online]. Wikipedia [cit 2015-05-31]. Dostupné z: https://en.wikipedia.org/wiki/RadSec

[15]. Transport Layer Security [online]. Wikipédia [cit 2018-08-01]. Dostupné z: https://sk.wikipedia.org/wiki/Transport_Layer_Security

[16]. Zabezpečený hypertextový prenosový protokol [online]. Wikipédia [cit 2017-02-01]. Dostupné z: https://bit.ly/30zWtwW

[17]. HTTP Methods GET vs POST [online]. W3Schools - Refsnes Data [cit. 2019-05-22]. Dostupné z: https://www.w3schools.com/tags/ref_httpmethods.asp

[18]. PHP Manual [online]. PHP Documentation Group [cit. 2019-05-20]. Dostupné z: https://www.php.net/manual/en/index.php

[19]. jQuery dokumentácia [online]. The jQuery Foundation. [cit. 2019-05-22]. Dostupné z: https://api.jquery.com/

[20]. HTML5 Reference [online]. The World Wide Web Consortium [cit. 2010-08-09]. Dostupné z: https://dev.w3.org/html5/html-author/

[21]. Packet Sender - Documentation [online]. NagleCode, LLC [cit. 2019-05-20]. Dostupné z: https://packetsender.com/documentation

[22]. Katalógový list ESP32-WROOM-32 [online]. ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD. [cit. 2019-05-12]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.

[23]. Katalógový list CP2102-9 [online]. Silicon Laboratories [cit. 2017-01-01]. Dostupné z: https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf

[24]. ESP32-DevKitC V4 Getting Started Guide [online]. ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD. [cit. 2019-05-12]. Dostupné z: https://docs.espressif.com/projects/esp-idf/en/latest/get-started/get-started-devkitc.html

[25]. Katalógový list TTL-232R-3V3 [online]. Future Technology Devices International Ltd [cit. 2016-05-23]. Dostupné z: https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_CABLES.pdf

[26]. Katalógový list SHT21 [online]. Farnell - Elektronické súčiastky [cit. 2019-05-12]. Dostupné z: http://www.farnell.com/datasheets/1780639.pdf.

[27]. I²C [online]. Wikipedia [cit 2019-05-06]. Dostupné z: https://en.wikipedia.org/wiki/I%C2%B2C

[28]. Solid state relé [online]. Wikipedie [cit 2018-12-02]. Dostupné z: https://cs.wikipedia.org/wiki/Solid_state_rel%C3%A9

[29]. Triak [online]. Wikipedia [cit 2017-05-16]. Dostupné z: https://sk.wikipedia.org/wiki/Triak

[30]. Katalógový list OMRON G3MB-202P [online]. OMRON ELECTRONICS LLC. [cit. 2019-05-03]. Dostupné z: https://www.openhacks.com/uploadsproductos/g3mb-ssr-datasheet.pdf

[31]. TTL (logika) [online]. Wikipedie [cit 2019-01-13]. Dostupné z: https://cs.wikipedia.org/wiki/TTL_(logika)

[32]. Wiring (development platform) [online]. Wikipedia [cit 2019-04-24]. Dostupné z: https://en.wikipedia.org/wiki/Wiring_(development_platform)

[33]. OpenSSL: Manual [online] [cit. 2019-05-12]. Dostupné z: https://www.openssl.org/docs/man1.1.1/man1/openssl.html

[34]. LEVICKÝ, D. Kryptografia v informačnej bezpečnosti. Košice: Elfa, 2005. ISBN 80-8086-022-X

[35]. RSA (cryptosystem) [online]. Wikipedia [cit 2019-05-12]. Dostupné z: https://en.wikipedia.org/wiki/RSA_(cryptosystem)

[36]. Mbed TLS [online]. ARM Limited [cit. 2019-05-14]. Dostupné z: https://tls.mbed.org/api/

[37]. Httpd dokumentácia [online]. The Apache Software Foundation. [cit. 2019-05-20]. Dostupné z: https://httpd.apache.org/docs/2.4/

[38]. JSON [online]. Wikipedia [cit 2019-05-07]. Dostupné z: https://en.wikipedia.org/wiki/JSON

[39]. Autodesk Eagle dokumentácia [online]. Autodesk Inc. [cit. 2019-05-20]. Dostupné z: http://eagle.autodesk.com/eagle/documentation

Prílohy

Príloha A: CD nosič - Bakalárska práca, zdrojové kódy pre ESP32, knižnice pre senzor SHT21, sprievodný dokument pre inštaláciu podpory vývojových dosiek do prostredia ArduinoIDE, súbory webovej časti, spustiteľné súbory pre generovanie vlastných certifikátov nástrojom OpenSSL, konfiguračný súbor ssl.conf pre web server, minimálna schéma zapojenia, návrh dosky plošných spojov - vyhotovil Ing. Slovák, PhD.