FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Přenosný přístupový identifikační systém využívající technologii NFC a umožňující komunikaci přes GSM bránu
Student:	Martin Šimůnek
Vedoucí:	Ing. Pavel Kubalík, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Bezpečnost a informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

1) Prozkoumejte existující řešení.
2) Analyzujte technologii NFC a možnost využití platformy Arduino jako ovládacího prvku.
3) Analyzujte a navrhněte bezpečné řešení s ohledem na omezené zdroje platformy Arduino.
4) Navržené zařízení se bude skládat z Arduino mikrokontroléru, GSM modulu pro posílání SMS zpráv, čtečky NFC a LCD displeje.
5) Komunikace se zařízením bude probíhat přes technologii GSM – formou SMS a za pomoci LCD displeje (základní orientační údaje).
6) Navržené řešení zrealizujte a řádně otestujte.

Elektronicky schválil/a prof. Ing. Pavel Tvrdík, CSc. dne 31. ledna 2021 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Přenosný přístupový identifikační systém využívající technologii NFC a umožňující komunikaci přes GSM bránu

Martin Šimůnek

Katedra počítačových systémů
Vedoucí práce: Ing. Pavel Kubalík, Ph.D.

13. května 2021

Poděkování

Tímto bych chtěl poděkovat vedoucímu této práce Ing. Pavlovi Kubalíkovi, Ph.D za rady, časté konzultace a pomoc, kterou mi poskytl při psaní této práce. Dále bych chtěl poděkovat rodině za poskytnutí zázemí pro psaní této práce a svým přátelům a známým, kteří byli velkou oporou a kteří mi pomohli s koretkurou mé práce.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 13. května 2021 . . . . . .

Odkaz na tuto práci

Šimůnek, Martin. *Přenosný přístupový identifikační systém využívající technologii NFC a umožňující komunikaci přes GSM bránu.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Tato bakalářská práce se věnuje využití technologie NFC k identifikaci osoby držící příslušnou kartu/tag se zaměřením na využití platformy Arduino. Jedná se o analýzu již existujících řešení a navržení vlastního řešení. Zařízení komunikuje se serverem po mobilní síti, skrze technologii GPRS. Byl sestaven funkční prototyp, a následně došlo k jeho testování. Jazyk psaní pro Arduino i serveru je C++ a to kvůli rychlosti a malým nárokům. Zařízení komunikuje zašifrovaně na úrovni packetů. Zařízení, které vzniklo, je uživatelsky přívětivé a cenově dostupnější, než podobná zařízení na trhu.

Klíčová slova přístup do budovy, NFC identifikace, bezpečnost řešení, Arduino, GPRS komunikace, PN532, NFC

Abstract

This bachelor thesis deals with the use of NFC technology to identify the holder of the card/tag. Focusing on the use of the Arduino platform. This work is an analysis of existing solutions and designing own solution. The device communicates with the server via a mobile network, through GPRS technology. A functional prototype was built. And then it was tested. The language for both Arduino and the server is C ++ due to speed and small demands on the HW. The devices communicate with encrypted packets. The device is relatively cheap and easy to use compared to other devices available on market.

Keywords building access, NFC identification, security, Arduino, GPRS communication, PN532, NFC

Obsah

Seznam obrázků

Seznam tabulek

Úvod

V posledních několika desetiletích došlo k velkému rozmachu přenositelné elektroniky, nositelné elektroniky a různých karet/čipů zastávajících mnoho funkcí. Společně s tímto došlo k rozvoji bezdrátové komunikace a standardů pro komunikaci bez nutnosti fyzického kontaktu fungující na velmi krátké vzdálenosti. Takovouto technologií je i technologie NFC. Tato technologie byla definována organizací NFC forum.

NFC technologie se vyznačuje krátkým dosahem, který činí pár centimetrů (maximálně 10 cm, ale obvykle se jedná o 4-5 cm). NFC technologie je založena na dřívějších bezdrátových standardech a kompletně je obsažena v ISO/IEC 18092. Zmíněná technologie zažívá velký rozmach v posledních deseti letech, kdy dochází k jejímu masivnímu nasazení. Dnes ji obsahuje velké množství prodávaných nositelných zařízení a smartphonů. Banky vydávají platební karty standardně vybavené touto technologií. NFC technologie je součástí bezkontaktního platebního styku. Prosazuje se ale i jako identifikační médium, například pro přístup do budov, zaznamenání výpůjček a nastavování různých zařízení pomocí přednastavených tagů.

Tato práce se zabývá využitím NFC jako identifikátoru osob pro různé příležitosti, hlavním zaměřením je však přístup do budov. Tato práce prozkoumává možnosti levné a mobilní NFC čtečky, založené na platformě Arduino, schopné identifikovat na základě karty s NFC technologií jejího držitele. Následně je možné provést zvolenou akci, právě na základě identity držitele. Tato práce, spolu s analýzou již nyní existujících řešení, přináší také vlastní řešení zkonstruované pro tyto účely.

V následující kapitole se tato práce zabývá analýzou již existujících řešení, jedná se o řešení, která jsou volně dostupná na trhu a o řešení z akademického či domácího prostředí. Následuje analýza návrhu řešení a prozkoumání tohoto návrhu. Jedná se o obecný návrh řešení, analýzu a výběr dostupného HW. Dále o analýzu dostupného SW, jeho volbu a návrh celkového řešení této práce. Následuje kapitola popisující realizaci celého systému, která referuje

o tom, jaky byl systém sestaven a jaká řešení byla zvolena. Předposlední je kapitola testování, která se zabývá testy sestaveného a naprogramovaného zařízení. Poslední kapitola zdůrazňuje nedostatky prezentovaného řešení.

Kapitola 1

Cíle práce

Hlavním cílem této práce je vytvoření funkčního prototypu čtečky na prototypovací platformě Arduino, která bude schopna identifikovat osobu na základě vlastnictví NFC karty/tagu uživatele této karty.

Mezi dílčí cíle patří analýza existujících řešení přístupů a identifikace spolu s analýzou platformy Arduino jako ovládacího prvku.

Dalším cílem je navrhnout řešení na platformě Arduino. Toto řešení má být bezpečné. Akce jako je vytvoření nové karty či načtení karty se musí logovat. Data nesmí být přenášena v čitelné formě. Celá čtečka musí být snadno použitelná, ovládání musí být intuitivní. Čtečka má podporovat velké množství (kusů) právě aktivních karet.

Celé hotové zařízení se skládá z částí umožňujících dosažení komunikace skrze mobilní síť, LCD panelu schopného zobrazovat základní údaje a komunikovat s uživatelem. Displej je řízen Arduino mikrokontrolérem.

Sestavené řešení musí být otestováno pro svou funkčnost.

Kapitola 2

Existující řešení

NFC technologie je dnes používána běžně a nadále se rozšiřuje její využití. Technologie je součástí mnoha proprietárních řešení přístupů do budov, které cílí především na velké zákazníky, jako jsou firmy nebo různé organizace nebo i na domácí nadšence, kterým nabízí již hotová řešení v rámci IoT. Analýza takovýchto řešení není bohužel možná do hloubky, neboť proprietární řešení nemívají dokumentaci a zdrojové kódy volně k dispozici [1].

NFC jako standard neobsahuje zabezpečení proti útokům na toto médium, pouze popisuje přenosový protokol [2]. Samotné zabezpečení dat je řešené na vyšších vrstvách. Je na výrobci konkrétních NFC karet/zařízení, aby implementoval ochranu před útoky [3]. To vede k vytvoření mnoha proprietárních řešení, kde ne vždy všechny prvky ovládání karet jsou přístupné veřejnosti (např. DESfire) [4]. Existují ale i zcela programovatelné karty, kde si to uživatel může zařídit sám. Takovými kartami jsou třeba JAVA karty [5].

NFC technologie je však živá i v open-source komunitě, hlavně kvůli své otevřenosti. Je možné pořídit poměrně levné NFC čtečky, které lze využít k jakýmkoliv účelům. Objevuje se zde však problém, a to proprietárních částí NFC karet/tagů různých výrobců, kdy nelze využít všechny funkce, které karty nabízejí, jelikož jejich popis a dokumentace není veřejná. Jediná možnost, která zde zbývá, je reverse engineering již existujících aplikací. Příkladem může být společnost NXP semiconductors, která stojí za značkou MIFARE a DESfire, kdy dokumentace je veřejnosti nepřístupná. Aby se jedinec dostal ke kompletní dokumentaci, musí podepsat NDA a navíc se poskytuje pouze firmám. „Je s podivem, že se NXP nepoučilo ze svých vlastních chyb, nebo že stále přetrvává názor, že security-through-obscurity funguje. Je snad Windows bezpečnější než Linux, protože Microsoft zdrojové kódy nezveřejňuje?“ [4].

Tato kapitola se zabývá řešeními v obecné rovině. Co daná čtečka musí obsahovat za povinné části, co daná čtečka obsahovat může, případně jaké doplňky lze běžně k takovémuto systému sehnat nebo jsou běžně žádané. Dále se kapitola zabývá pěti již existujícími komerčními řešeními. Zabývá se ce-

nou, možnostmi těchto řešení. Dále se zabývá výhodami, které výrobce uvádí, spotřebou el. energie a analyzuje použité technologie. Poslední část této kapitoly se zabývá řešeními, která nejsou proprietární. Tato řešení byla vytvořena v rámci akademických prací nebo jako domácí projekty. Výhodou těchto řešení je, že jsou snadněji analyzovatelná díky volně přístupnému kódu nebo práci, která řešení detailně popisuje.

Kapitola byla rozdělena na 3 hlavní části z důvodu přehlednosti a oddělení informací. V první části je seznámení s funkčností přístupových zařízení a jejich stavbou v teoretické rovině. Následuje seznámení s dostupnými řešeními na trhu, která lze zakoupit a instalovat doma. Tato řešení jakožto proprietární nemají otevřenou dokumentaci, a tak jejich analýza vychází pouze z produktových stránek výrobce. Poslední část byla vytvořena jako protiklad proprietárních řešení. Z důvodu snadnější analýzy těchto řešení, které mohou jít i více do hloubky, ilustrují možné funkční celky oproti „blackboxům“ privátních řešení.

2.1 Obecná řešení

Systém, který využívá NFC přístup se obecně bude skládat z několika součástí, které vždy musí být přítomné. Jedná se vždy o čtečku, která je schopna komunikovat s NFC zařízením a dále o kontrolní jednotku, která čtečku řídí a vyhodnocuje údaje [6]. Řešení může být zpravidla jen postavené pro systém, který je malý. Správa takovéhoto systému je jednoduchá a probíhá přímo. Přidávání a odebírání autorizovaných zařízení není řešeno nijak centrálně. Údaje jsou uložené lokálně. Tento systém má však velkou nevýhodu ve správě, už v případě, že takových zařízení je více, se musí úkony správy, jako jsou změna oprávnění nebo zrušení karet, duplikovat tolikrát, kolik zařízení je třeba přenastavit [6]. Tato složitá správa může vést k bezpečnostním rizikům, jako je například neúmyslné ponechání karet v jednom ze spravovaných systémů.

Tyto nedostatky jsou většinou řešeny napojením na domácí síť, kdy přes určité webové rozhraní lze spravovat čtečky, profily duplikovat a centralizovat tím jejich správu [7]. Dalším případným řešením jsou cloudové služby výrobce, které umožňují správu systému odkudkoliv ze sítě [8].

Rozsáhlejší systémy, které jsou většinou určeny pro velké budovy, postavené na míru, mají více čteček, čtečka se skládá z transceiveru a kontroléru. Tento kontrolér sám o sobě může být levnější než samostatné řešení, protože nemusí řešit autorizační logiku. Pouze načte údaje z karty a následně je odešle na server, kde se údaje zpracují a odešle se odpověď do příslušné čtečky [6].

Tento systém je v malém měřítku dražší, než systém samostatných čteček, které obsahují autorizační logiku, nicméně se jedná o řešení, které je snažší na správu a nabízí více možností. V případě aplikace ve větším měřítku se může jednat o levnější řešení, než velké množství samostnatných systémů.

K systémům lze přidávat, či odebírat různé další součásti. Pro větší robustnost systému lze přidat napájení pro případ výpadku el. energie [9], pro zvýšení bezpečnosti lze kromě samotné karty ověřovat například PIN či biometrický údaj, jedná se o takzvanou víceúrovňovou autentifikaci [10].

Bezdrátový elektrický přístup má oproti konvenčním klíčům jednu výhodu, a to, že nemusí být žádné mechanické části dveřního mechanismu přístupné z venku [4]. Jelikož NFC čtečka přečte médium i přes určitou nemetalickou bariéru (neblokující radiové signály) až do vzdálenosti 10 cm [11]. Narozdíl od klíčové dírky. To je velkou výhodou, protože emulace karet a pokusy o prolomení takovéhoto zabezpečení zpravidla vyžadují větší míru znalostí, než klasické metody otevírání zámků, stejně tak pokročilejší nástroje. I když to nemusí znamenat, že tyto nástroje jsou drahé, nebo nedostupné široké veřejnosti. Jednou z nevýhod těchto řešení je snadná zkopírovatelnost média, na kterém jsou informace uloženy. Mnohem snadnější, než u moderních klíčů [12]. Naopak výhodou tohoto řešení je, že lze jednoduše autentifikovanou kartu vyřadit, pokud je třeba, například při odcizení karty, v případě odcizení klíče je potřeba vyměnit celou vložku zámku [13].

2.2 Proprietární řešení

Velkou nevýhodou proprietárních řešení je jejich licencování a ochrana pomocí copyrightů, což omezuje či zcela znemožňuje zásahy do těchto řešení. Většinou jsou zásahy zcela znemožněny v licenčním ujednání [1]. Dále systém může mít omezené rozšiřování, nekompatibilitu s jinými výrobci, často se produkt prodává po omezenou dobu a zajištění podpory po konci výroby bývá problematické. Výhodou těchto systémů pro koncového uživatele je však snadná správa a instalace, kterou výrobce/dodavatel systému nabízí spolu s produktem.

2.2.1 Netamo smart door lock

Příklad řešení, kdy čtečka a celý systém jsou zcela samostatné. Veškerá logika se ukládá v samotném zámku, který je možno spárovat s různými typy řešení chytré domácnosti. Zámek obsahuje Bluetooth low energy 4.2 pro odemykání pomocí mobilního zařízení, případně pro svou správu. Dále pomocí technologie NFC je schopen přečíst a odemknout na základě klíče s NFC čipem. NFC klíče se dají přidávat/odebírat podle potřeb. Zámek slibuje dlouhou životnost baterie. Zámek je možné spárovat s řešením jiných výrobců pro chytrou domácnost [13].

2.2.2 HomeKit Aqara N100

Mezi další řešení se řadí například Aqara N100, v současné době prodávaná pouze na území Číny. Řešení opět obsahuje veškerou logiku v sobě. Jedná se o

Název řešení	Výrobce	Technologie	Cena
Netatmo Smart Door Lock	Netamo	BLE 4.2,NFC	nestanovena
HomeKit Aqara N100	Aqara (Xiaomi)	čtečka otisků prstů, přístupový kód, NFC, Zigbee, Bluetooth	7000-Kč
Nuki Smart lock 2.0	Elektro-System-Technik s.r.o	Bluetooth 5.0, Zigbee, WiFi	7200 Kč
Yale Linus	Yale	BLE 4.2, WiFi	6000 Kč
Danalock V3	Danalock ApS	Bluetooth 4.2, Z-Wave, Zigbee	4900 Kč

Tabulka 2.1: Tabulka dostupných produktů „chytrých“ zámků

řešení přístupu pomocí NFC čtečky, doplněné o čtečku otisků prstů a dotykový panel pro zadání přístupového kódu. Pro komunikaci s mobilním telefonem lze využít i Bluetooth. Systém využívá i Zigbee. Oproti výše zmíněnému systému se jedná o energeticky náročnější systém, který vyžaduje 8 AA baterií [14].

2.2.3 Nuki Smart lock 2.0

Zcela samostatné řešení bezdrátového přístupu, které neobsahuje technologii NFC. Jedná se opět o řešení vše v jednom, napájené bateriemi. Toto řešení se montuje přímo na zámek. Oproti konkurenci nabízí technologii automatického odemnkutí v blízkosti spárovaného zařízení. Lze propojit s dalšími řešeními od jiných výrobců [15].

2.2.4 Yale Linus

Další řešení, které nepoužívá technologii NFC, ale spoléhá na Bluetooth, a to ve verzi 4.2. Jedná se o vše v jednom řešení. Nabízí šifrování AES128 mezi koncovými body. Využití služeb geofencingu pro automatické otevírání zámku při přiblížení zařízení. Pro spárování se zbytkem domácích systémů je vyžadováno dodatkové zařízení [16].

2.2.5 Danalock V3

Opět řešení vše v jednom. Opět bez použití technologie NFC. Veškerá komunikace probíhá pouze pomocí Bluetooth, Z-wave, případně pomocí WiFi, ke kterému je třeba dokoupit další specializovaný HW. Tento HW umožní propojení s dalšími prvky chytré domácnosti [17].

2.3 Ostatní řešení

Mnoho řešení užívajících technologii NFC bylo vytvořeno jako školní práce, různé výzkumy nebo i domácí kutilství. K tomuto je velmi výhodné využití mikrokontrolérů či miniaturních PC, jednak kvůli jednoduchosti a jednak kvůli příznivé ceně. I proto se spousta řešení zaměřuje na kombinaci platformy Arduino a NFC čtečky, často čipu PN532. Takto sestavená čtečka je schopna podporovat velkou část NFC standardu a jedná se o levné řešení.

Problémem řešení, hlavně domácích, je fakt, že NFC standard je pouze popis bezdrátové komunikace mezi zařízeními. V žádném případě neřeší zabezpečení proti odposlechu, nebo modifikaci dat [2]. Tudíž bezpečnost implementace je na každém, jak si ji vyřeší. Mnoho projektů, které byly analyzovány, se zaměřuje na funkčnost a ochranu komunikačních kanálů mezi serverem a čtečkou nebo čtečkou a tagem neřeší. Některé bezpečnostní problémy lze vyřešit, kupříkladu zabezpečení komunikačního kanálu mezi čtečkou a serverem. Některé však naráží na nedostatky levného HW, nebo i nezveřejněnou dokumentaci k využitým technologiím. Například pokročilejší NFC karty s ochranou proti kopírování a silnějším šifrováním, kdy jediným způsobem jak se k dokumentaci dostat, je podepsat NDA a zaplatit poplatky, případně nalézt, pokud dokumentace někde unikla.

Samozřejmě skrytá dokumentace nezabraňuje vytvoření opensourcových řešení, jako je Liblogic [18] a podobné projekty. Tato řešení jsou dostatečná na použití v systémech, která nabízejí dostatečné prostředky pro provoz komplexních řešení. Bohužel tato řešení jsou pro použití v systémech jako je Arduino příliš velká a pomalá. To nutí programátory, aby se uchýlili k osekanějším verzím knihoven, které umí povětšinou jen číst či psát na pár druhů karet na trhu. Typicky se jedná o Mifare Classic a Ultralight.

Ne všechny přístupové systémy nutně musí používat technologii NFC. Rozebírají se zde i práce, které jsou ovládány jinak a jsou na platformě Arduino.

2.3.1 Near field communication (NFC) model for Arduino UNO based security systems office system

Díky nedostatku knihoven, omezeným prostředkům a licenčním podmínkách, má spousta vzniklých řešení nedostatky, ale ve většině případů se nejedná o nedostatky v sestavení samotného HW. Ale již se jedná o nedostatky při čtení či zápisech na kartu, případně při komunikaci v rámci systému. Samotná SW

část a bezpečnost je zcela opominuta například v této [19] práci, kdy kompletní zaměření práce je sestavení funkčního systému. V tomto konkrétním případě se jedná o propojení dvou desek Arduino, toto propojení ale nemá žádné zabezpečení a předpoklad je, že spolu Arduina komunikují v plaintextu. Nicméně pokud kanál, po kterém komunikují, bude fyzický kabel, který vyžaduje fyzické navrtání pro prolomení a nemá žádný mezilehlý systém jako je například switch, nepředstavuje takový bezpečnostní problém.

2.3.2 Smart Door Monitoring and Locking System using SIM900 GSM Shield and Arduino UNO

Pokud je systém ovládán přes veřejnou síť, ať již internetovou či mobilní, absence zabezpečení je zásadní. V případě informativních logů, které se posílají pouze jedním směrem, nejednalo by se o zásadní bezpečnostní hrozbu, pokud by nebyly posílány citlivé informace. Pokud jde i o ovládání skrze síť, posílají se příkazy obousměrně, jako kupříkladu v této práci [20]. Je třeba zajistit bezpečnost přenášených dat a autentizaci těchto dat. Ovládání pouze na základě nijak nechráněných příkazů s sebou přináší riziko. Autor zde na toto myslí a zavádí kontrolu čísla, ze kterého je daný příkaz odeslán. Podvrhnutí samotného telefonního čísla by nemělo být snadné, pokud na tomto nespolupracuje sám operátor mobilní sítě. Dokud není mobilní telefon odcizen, zařízení by mělo být dostatečně zabezpečené.

2.3.3 Classrooms Access Control over Near Field Communication

U některých řešení si je autor vědom bezpečnostních nedostatků při přenášení informací. Příkladem je tato práce [21]. Zde autor jasně zmiňuje nedostatek zabezpečení na přenosovém kanálu mezi hlavním serverem a samotnou čtečkou. Autor tvrdí, že pro nekritické využití ID učitele a ID karty není zabezpečení zvažováno [21]. S tímto tvrzením nelze souhlasit, protože i pokud se jedná o nekritické využití, stále lze dané informace podvrhnout. To může vést k tomu, že se místnost bude tvářit, že je obsazená jiným vyučujícím, než ve skutečnosti je. Ano, lze souhlasit, že není třeba využít silného zabezpečení, které je podporované drahým přídavným zařízením, zvláště pokud jde o málo citlivé informace. Ale i v takovýchto případech by minimálně mělo jít autentifikovat zdroj zprávy. Kvůli možnosti podvržení.

2.3.4 DIY electronic RFID Door Lock with Battery Backup

Nejzajímavější řešení, které tato práce analyzuje, je řešení hobby projektu popsaného v tomto článku [4]. Jedná se o řešení přístupu do budovy. Ač toto řešení má nedostatky, které již byly diskutovány, jako například: správa musí být prováděna přímo na zařízení, karty jsou ukládány přímo v paměti EE-

PROM, tajné klíče jsou ukládány v rámci zařízení bez možnosti bezpečného uložení a tudíž teoreticky zjistitelné. Toto řešení má oproti již zmíněným ostatním řešením výhodu, a to, že využívá moderní kartu, která je odolná proti kopírování a uzpůsobená pro bezpečnou komunikaci se čtečkou. Jedná se o kartu DESfire. Autor řešení byl schopen díky reverznímu inženýrství několika open-source projektů zmapovat karty DESfire EV1 od NXP. Díky tomu vytvořil knihovnu schopnou komunikace s těmito kartami. To znamená využití bezpečnostních prvků, které karta nabízí. Avšak bez přístupu k dokumentaci nelze ověřit správnost tohoto řešení do detailu jinak, než že to „prostě funguje“. Můžou se zde vyskytovat skryté vady, které bezpečnost ohrožují, ale i přesto se jedná o jedinečný projekt. Samotný projekt navíc vyžaduje zdroje, které platforma Arduino nemá k dispozici. Jedinečnost projektu je využití karet DESfire a implementace komunikace s touto kartou.

Kapitola 3

Analýza

Při návrhu systému je nutno vycházet z již funkčních řešení, u kterých je však nutné částečně upravit jejich koncepci tak, aby výsledný projekt odpovídal cílům stanoveným v této práci. Řešení musí být cenově optimální a zároveň bezpečné, a to za použití na trhu volně dostupných prostředků a komponent, a to v rámci prostředků, které jsou k dispozici. Jedním z hlavních úkolů práce je zajištění toho, aby se případný útočník neměl možnost dostat k žádným citlivým údajům, a to i v případě prolomení samotné karty. Dále žádné údaje by neměly být předávány serveru v plaintextu. Jedním z dalších klíčových bodů této práce je správa zařízení. Zařízení by mělo být napojeno na server, který bude spravovat uložené údaje a povolovat/zamítat + logovat. Dalším úkolem je mobilita, zařízení by mělo být v případě potřeby mobilní.

Tato část se zabývá analýzou dostupných komponent a to jak HW tak SW komponent. Analyzuje dostupné součástky pro samotné sestavení zařízení tak, aby byla zajištěna jeho spolehlivá funkčnost. Dále se zabývá SW částí, ve které jde převážně o dostupné programovací jazyky, vývojová prostředí a dostupné knihovny, co nabízejí a jak je možno tohoto využít.

Analýza se dělí do 3 částí, z důvodu přehlednosti a jasného oddělení částí HW a SW. Jedná se o analýzu základních stavebních prvků pro čtečku. Rozebírají se zde možné problémy, kterým může sestavené zařízení čelit a rozebírají se zde různé případy užití. Dále se jedná o analýzu dostupného HW, který bude užit pro stavbu zařízení a analýzu SW, který bude užit pro vytvoření funkčního systému celého zařízení.

3.1 Analýza základních stavebních prvků

Analyzované řešení se skládá ze dvou hlavních částí, a to čtečky a serveru. Čtečka obsahuje veškerou logiku komunikace s kartou. Jedná se o operace čtení, zápisu, šifrování/dešifrování. Zároveň obsahuje logiku a nástroje pro komunikaci se serverem. Server obsahuje a zpracovává informace zaslané čtečkou, na které i následně odpovídá. O povolení či zamítnutí dané karty rozho-

duje server, a to na základě záznamů v databázi. Tímto dojde k oddělení případných citlivých informací, kterými jsou například údaje o vlastníkovi karty, jeho oprávněních apod., od samotné zranitelné části systému - čtečky. Informace jsou uloženy pouze na centrálním serveru, kde se zpracovávají a nejsou odesílány dále. Podobně řešená je funkčnost čtečky při vytváření nové karty, kdy čtečka pouze zapíše údaje, které jí jsou předány ze serveru na kartu, opět i v případě úniků těchto údajů, neuniknou žádné citlivé informace.

Spojení se serverem bude probíhat prostřednictvím internetu. To nabízí velkou mobilitu a flexibilitu zařízení, protože čtečka může mít WiFi modul a pomocí této bezdrátové sítě být mobilně umístěna kdekoliv v areálu, může mít ethernetový modul, či může připojení být řešeno přes mobilní síť. Kvůli zjednodušení systému byla serverová část navržena pouze jako jednovláknová aplikace, která nemá mnoho funkcionalit. Graf návrhu systému je vizualizován pomocí obr. 3.1.

3.1.1 Čtečka-server

Ve velké míře je zde zřejmá inspirace modelem server-klient. V současné době se jedná o velice populární řešení pro různé aplikace. Tento architektonický návrh umožňuje velkou část výpočetní zátěže ze čtečky předat na server, u kterého se počítá s větším výkonem.

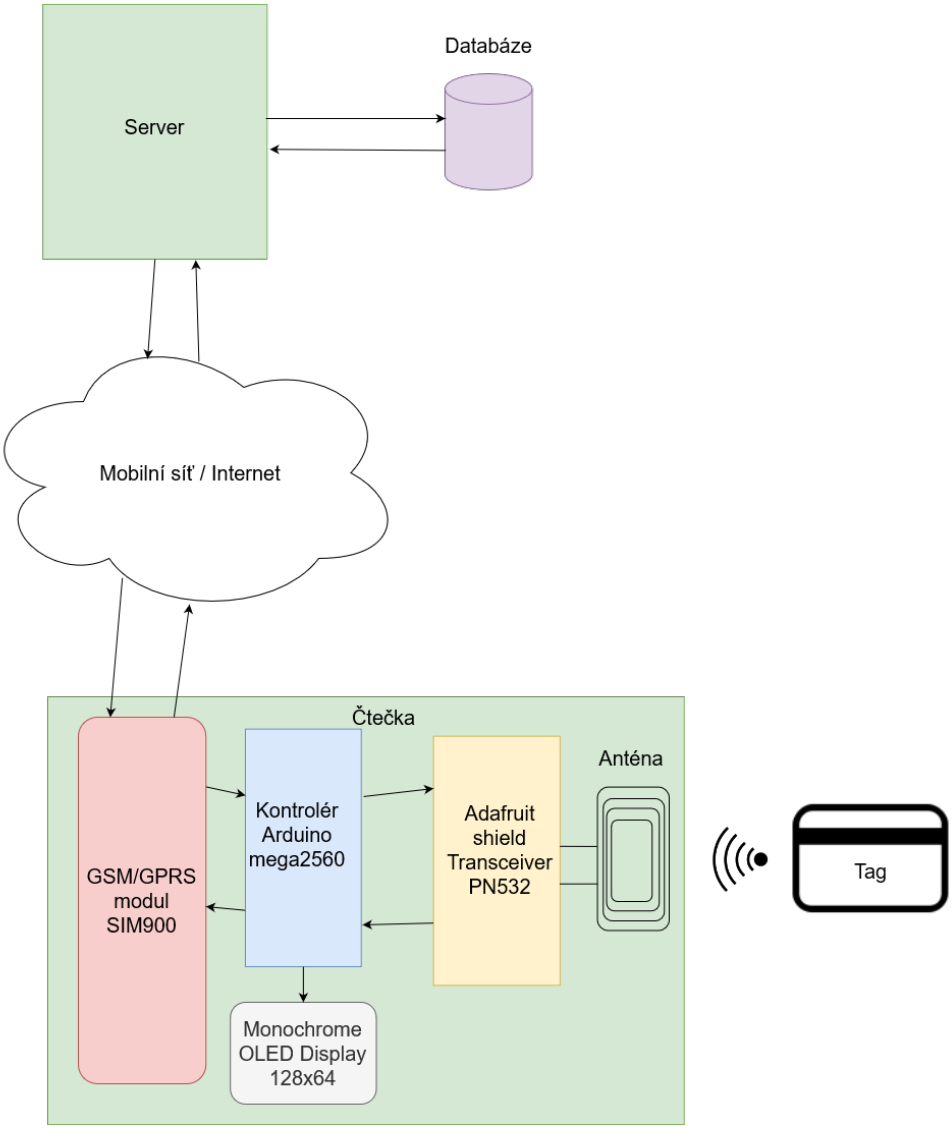
- Klient (čtečka) - Roli klienta v tomto návrhu zastává samotná čtečka, jejíž hlavní úkol je komunikace s kartou, zapisování a čtení. Čtečka by měla být schopna údaje šifrovat/dešifrovat a komunikovat se serverem.
- Server - Role serveru zde zastává výpočetně silnější jednotka, ale nemusí se jednat o drahé zařízení, v tomto případě může jít o levnější zařízení, jako je například Raspebery PI, případně pronajatý server.

3.1.2 Komunikace zařízení

Z návrhu, který je popsán výše, je jasné, že bude probíhat intenzivní komunikace mezi čtečkou a serverem a čtečkou a kartou. Bude se jednat o velmi malé množství dat posílané často. Vzhledem k omezeným možnostem koncových zařízení a propojení mezi nimi, stačí jednoduchá komunikace na úrovni paketů.

Pro tuto komunikaci je lepší využít protokolu TCP na úrovni paketů pro spolehlivost výsledného řešení. Vzhledem k množství posílaných dat není problémem menší rychlost, či navazování spojení. Po navázání spojení se serverem toto spojení bude drženo po celou dobu aktivity zařízení.

Výhodou je, že spojení neřeší Arduino samotné, ale připojený modul, který obhospodařuje toto spojení. V případě UDP spojení by muselo Arduino obsahovat logiku zajišťující spolehlivý příjem paketů, což by vedlo k další zátěži systému.



Obrázek 3.1: Návrh systému

3.1.3 Šifrování a autentifikace dat

Komunikaci se serverem je nutno zabezpečit proti případnému poškození a odposlouchávání přenášených dat, přestože návrh nepočítá s ukládáním citlivých dat na karty, nebo jejich posíláním přes síť čtecímu zařízení. Je nutno řešit jejich šifrování a hlavně autentizaci dat. Je nutno se ujistit, že data, která čtečka čte z karty či přijímá ze sítě, jsou validní a skutečně od subjektu, který má opravnění odpovědět/data vytvořit. Tímto se vyhneme například falešnému potvrzení neexistující karty.

Pro tento účel bylo nejprve rozhodnuto užít šifru AES s módem AEAD, avšak po prozkoumání dokumentace knihovny k Arduinu, byl tento závěr přehodnocen. Nové poznatky vedly k závěru, že se jedná o poměrně pomalou šifru, která bude vytvářet takzvaný „bottleneck“. Navíc tento druh šifry vyžaduje 16 bitový IV, který je problém vytvářet pro pomalý RNG generátor Arduina. Po prozkoumání dalších možných a dostupných šifer na obou stranách bylo rozhodnuto o využití šifry CHACHA20 spolu s POLY1305 algoritmem, který je podle přiložené dokumentace rychlejší než AES, navíc je v dokumentaci poznámka o možných útocích na postranní kanály implememntace AES na Arduinu.

Díky využití této autentifikované šifry jsme schopni zajistit důvěrnost a autentičnost dat.

3.2 Analýza a výběr HW

HW návrh má splňovat požadavky, aby se jednalo o levné zařízení. Což je splněno využitím platformy Arduino, konkrétně Arduino Mega 2560. Cena takového mikrokontroléru se pohybuje okolo 1000 Kč. Další součástí je modul, schopný komunikovat s NFC kartou. Zvolen byl modul Adafruit PN532, který nepatří mezi nejlevnější, ale je kvalitní a ověřený. Cena tohoto modulu činí okolo 1000 Kč. Monochromatický zobrazovací displej se cenově pohybuje kolem 100 Kč. Poslední součást, GSM modul, má cenu v českých obchodech okolo 700 Kč. Tyto součásti se dají sehnat i levněji, ceny jsou převzaty z aktuální nabídky českých e-shopů.

3.2.1 Arduino Mega2560

Platforma Arduino byla vybrána jako všestranná a levná platforma pro snadné vytváření funkčních zařízení. Za svou existenci se tato platforma stala velmi populární. Velmi jednoduše použitelná platforma, na kterou vzniklo velké množství lehce připojitelných modulů ve formě shieldů, disponuje hned třemi jednoduše použitelnými komunikačními rozhraními jako je I2C, UART či SPI pro připojení velkého množství dalších modulů. Teoreticky na sběrnici i2c lze připojit až 127 dalších zařízení s rozdílnou adresou. Bohužel tato adresa je přidělena zařízení při výrobě a není měnitelná. Zařízení je jednoduše programovatelné.

Arduino Mega 2560 bylo zvoleno jako ideální kandidát z důvodu velikosti paměti. Z dostupných desek Arduino má největší paměť, která činí 8kB. Disponuje taktéž velkou možností pinů pro připojení dalších zařízení.

Na druhou stranu Arduino Mega má i své nedostatky v tomto projektu. Jedná se o pomalý procesor, který se projeví hlavně při náročnějších operacích jako je šifrování, dešifrování a autentifikace dat, kdy lze očekávat prodlevy v odezvě zařízení. Dále je zde problém uložení tajného klíče, který bude užíván

pro zapisování na karty a zároveň pro komunikaci se serverovou částí. Platforma nenabízí žádné bezpečné uložení tajného klíče, při fyzickém přístupu k platformě se tajný klíč dá zjistit. Dále samotná platforma nenabízí kvalitní zdroje entropie pro využití kryptograficky bezpečnými generátory náhodných čísel.

3.2.2 Adafruit PN532 NFC/RFID Controller

Adafruit PN532 je shield obsahující čip PN532 a velkou anténu, která je schopná napájet a komunikovat s kartami na frekvenci 13.56 Mhz. Tento shield je schopen díky čipu PN532 zvládat čtení a zápis na karty, emulace samotných karet, komunikaci s NFC zařízeními, jako jsou například mobilní telefon či chytré hodinky.

Shield je díky své velké anténě schopen napájet i energeticky náročnější karty a na teoretickou maximální vzdálenost 10 cm. To poskytuje možnost, aby tento shield mohl komunikovat s kartou skrz nemetalickou překážku. Některé levnější verze shieldů, které obsahují PN532, nejsou schopny zajistit dostatečné napájení skrze překážky. Také nemusí být schopny plně napájet karty jako je DESfire [4] od společnosti NXP.

Tento modul je technologicky schopen naplno využít čipu PN532, bohužel existující knihovny pro Arduino, schopné fungovat s tímto čipem, nepodporují jeho plnou funkčnost. Implementují pouze základní čtení a zápis a jen na některé druhy vyráběných karet. Nejčastěji Mifare Classic. I výrobce upozorňuje na tento fakt na svých stránkách.

3.2.3 OLED displej

Jako modul komunikace s uživatelem čtečky byl zvolen monochromatický OLED displej o velikosti 0,96 palce a se sběrnicí I2C. Výhodou displeje je velká svítivost, je dobře vidět, dokáže zobrazovat základní informace ať již textové či ve formě piktogramů. Jedná se o flexibilní způsob komunikace s uživatelem. Modul není nijak drahý. Díky I2C sběrnici se velmi snadno ovládá. V případě potřeby obsahuje displej několik odlišně barevných řádků pixelů pro zobrazení stavových informací.

3.2.4 SIM900

GPRS/GSM Shield SIM900 byl zvolen pro zachování mobility celého zařízení s možností připojení na server kdykoliv a odkudkoliv. Jedná se o Arduino shield komunikující pomocí UART, který nabízí základní konektivitu do mobilní sítě. Umožňuje přijímat a odesílat SMS zprávy, hovory nebo navázat síťové připojení TCP či UDP. Jedná se o velmi levný modul. Umí pouze základní sítě, nepodporuje sítě 3. a vyšší generace. Pro posílání velmi malého množství dat je toto řešení dostatečné.

Díky využitím sítí pouze 2. generace nabízí shield připojení téměř odkudkoliv. Jen je potřeba počítat s delší odezvou a celkově vysokým RTT a pomalými přenosovými rychlostmi.

Pro ovládání daného modulu se využívají AT příkazy, posílané přes UART. Pro plnou funkčnost modulu je třeba dodat externí napájení, jelikož modul má velkou spotřebu, kterou není Arduino schopno dodat. Parametry externího napájení jsou: napětí mezi 5-12 volty a proud mezi 1-2 ampéry stejnosměrného proudu. Je to kvůli tomu, že modul pro registraci do sítě a odeslání například SMS vyžaduje nárazově velký proud a mohl by se vypnout. Naopak ve standby režimu by modul měl mít nízkou spotřebu. To jej činí ideálním pro bateriový provoz.

3.2.5 Zdroj

Na základě údajů výše byl zvolen zdroj o napětí 12 voltů DC s maximálním odběrem 2 ampéry. Tento zdroj je určen pouze pro napájení SIM900 modulu a měl by být schopen zajistit jeho bezproblémový provoz, a to i přestože se jedná o zdroj, který je v horní hranici tolerovaných napětí.

Místo síťového zdroje, který byl pro tento projekt zvolen, je možné zvolit i zdroj bateriový doplněný o kondezátory, které jsou schopny zajistit právě zmíněný krátkodobý odběr až 2A u modulu. Tím se dosáhne kompletní mobility zařízení. Případně lze baterie využít jako nouzové napájení.

3.2.6 Karty

Vzhledem k omezeným možnostem dokumentace, které znemožňuje využití propracovanějších proprietárních karet, jsou zvolené karty ty, které jsou dobře zmapovány a existují běžně dostupné nástroje pro jejich využití. Jedná se o karty MIFARE Classic 1K a MIFARE Classic 4K a MIFARE Ultralight.

3.2.7 Ostatní

Mezi další použité součástky patří spínač pro ovládání režimů čtečky. Ten slouží k jednoduchému přepínání režimů čtečky. Přímo na čtečce jsou k plné funkčnosti přidány 10K ohm pullup rezistory.

3.3 Analýza SW, výběr knihoven a vývojového prostředí

Celý systém je psán v jazyce C++. Čtečka z důvodu, že jazykem pro Arduino je podmnožina jazyka C/C++. Podmnožina proto, že je nutné ušetřit paměť zařízení. Arduino neobsahuje žádné standartní knihovny C++. Dále je změněná velikost primitivních datových typů, např. int je pouze 16 bitový.

C++ je velice rychlý jazyk s malými paměťovými nároky. Jedná se jazyk, který kombinuje přístupy z nízkého jazyka jako je C a zavádí třídy (class), výjimky (exceptions), stl knihovnu. Tyto věci však kvůli omezeným možnostem mikrokontroléru Arduina nelze využít. Výjimky nejsou v Arduinu možné, musí se spoléhat na vracení návratových hodnot. Stl knihovny také chybí. Kvůli možné paměťové fragmentaci není doporučeno dynamicky alokovat paměť a následně uvolňovat a znovu alokovat.

Na návrh serveru byl zvažován Python, jazyk umožňující velmi rychlé vytvoření serveru TCP, jednoduše bez potřeby řešit různé problémy, které s sebou přináší správa paměti v C. Problémem tohoto jazyka se ukázala být rychlost. Vzhledem k potřebě serveru a co nejlevnějšího řešení Python není dobrá volba. Pro tyto účely je Python velmi těžkopádný a pomalý jazyk, který byl využit jen pro vytvoření rychlého prototypu serveru a poté odložen a dále nerozvíjen.

Konečná serverová část je psaná v C++ pro co největší úspornost. Počítá se s nasazením na slabší HW, případně na nějaký server, který je placený podle požadovaných zdrojů a v takových případech je vhodné, aby výsledná aplikace byla rychlá a nenáročná na výpočetní prostředky. Serverová část napsaná v C++ toto splňuje, výsledek bude rychlý a nenáročný na systémové prostředky. V případě dalšího rozšiřování je však více náchylný k chybám a nedostatkům, které s sebou C++ přináší.

3.3.1 Vývojové prostředí

Vývojové prostředí pro Arduino, tzv. Arduino IDE, není dostatečné vývojové prostředí. Jedná se pouze o editor, co umožňuje kompilaci a zvýrazňování syntaxe, také má správu knihoven v sobě a umí nahrát zkompilovaný kód přímo na samotné Arduino. Nenabízí žádné funkce našeptávání, jako například jiná IDE, zvýrazňování řádků, zvýrazňování chyb při kompilaci je občas nepřesné. Vcelku těžkopádné, uživatelsky nevhodné. Z tohoto důvodu bylo později v projektu zvoleno IDE eclipse, s pluginem sloeber. Jedná se o opensource plugin, který do IDE přidává možnosti Arduino IDE. Umožňuje sledovat serial port a data, která dostává, kompilovat a nahrát program do Arduina, stejně tak umí spravovat knihovny. Krom těchto základních věcí, co umí Arduino IDE, umí vše, co umí eclipse IDE. Umí propojení s gitem, našeptávač, zvýrazňovanou syntaxi. Pro větší projekty se jedná o lepší alternativu. Z tohoto důvodu byla zvolena tato kombinace (sloeber + eclipse ide).

Pro vývoj serveru bylo zvoleno IDE Clion, ke kterému jsou v rámci studia licence přístupné pro studenty. Clion je velmi dobré IDE, umožňující a integrující mnoho nástrojů jako je git, valgrind, cmake. Pro velmi pohodlné užívání a po předchozí zkušenosti autora s tímto nástrojem, byl vybrán pro psaní serverové části projektu.

3.3.2 Komunikace, ukládání dat na karty

Komunikace čtečky je formou jednotlivých packetů po TCP spojení. Tyto packety přenáší inicializační vektor, délku zprávy, šifrovaná data a TAG pro ověření původu šifrovaných dat. Toto schéma je využíváno při každém přenosu dat. Přenos zašifrovaných dat je popsán v této tabulce. 3.1

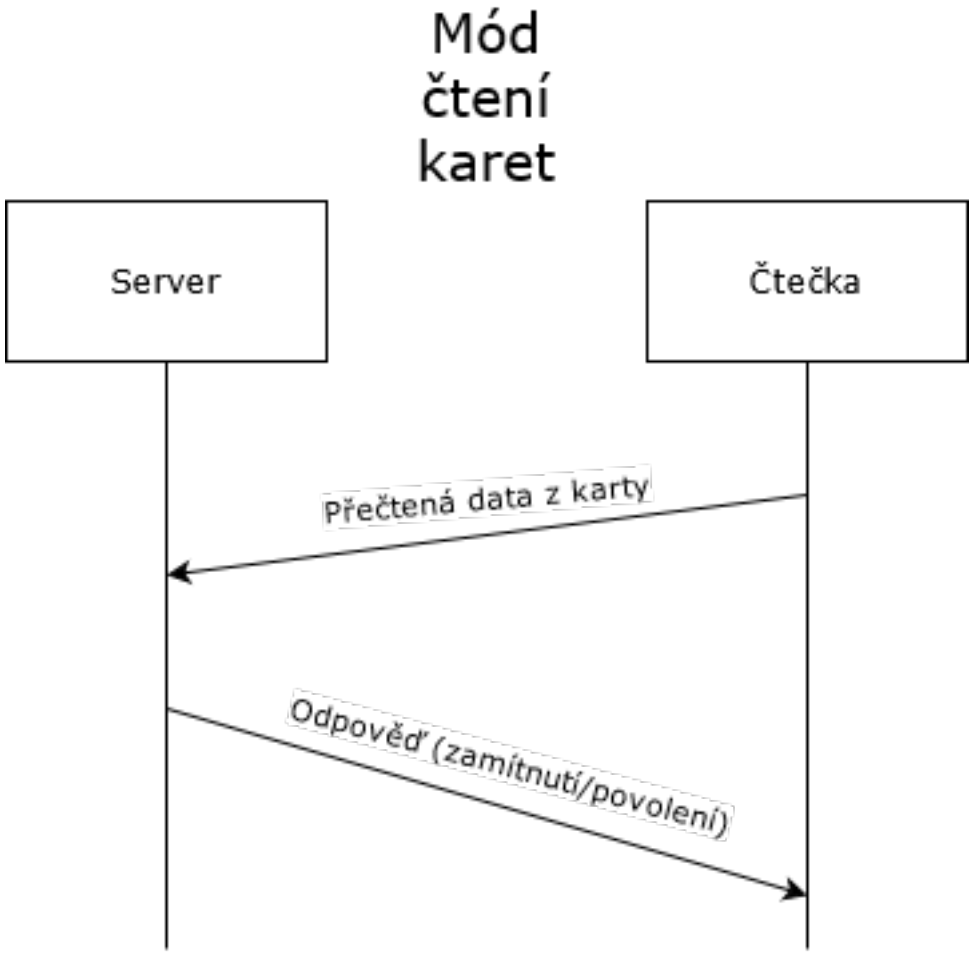
Komunikace mezi čtečkou a serverem při různých modech je zobrazena níže. 3.2 3.3 3.4 Čtečka má dva módy, mód čtení a mód zápisu.

Funkce dat	Uvození dat	Délka dat	Data
Přenos IV	IV:	<délka IV v bytech (16 bit unsigned int)>	<samotná data IV>
Délka šifrovaných dat	SIZE:	<délka bytů určující délku šifrovaných dat (16 bit unsigned int)>	<byty určující délku >
Šifrovaná data	-	-	<data>
TAG pro ověření dat	TAG:	<délka TAGu v bytech (16 bit unsigned int)>	<samotná data IV>

Tabulka 3.1: Popis jednotlivých částí šifrované komunikace

3.3.2.1 Mód čtení

Mód čtení je základní mód čtečky, která je výsledkem tohoto projektu. Mód čtení čeká na kartu, která se přiblíží ke čtečce, aby ji mohl přečíst. Po načtení základních informací jako je UID, SAK a ATQA čtečka porovná tyto karty s kartami, které má umět přečíst. Pokud toto porovnání souhlasí, čtečka se pokusí přečíst data z karty. Pokud je přečte, úspěšně odšifruje a ověří TAG. Získá tak ID, které přidá k datům, která jsou poslána na server. Pokud data obsahují položku ID, čtečka počká na odpověď serveru. Podle té dojde k zamítnutí či povolení vstupu 3.2. V případě absence ID, z jakéhokoliv důvodu nebo neověření TAGu, je karta rovnou zamítnuta, čtečka už na odpověď nečeká 3.3.



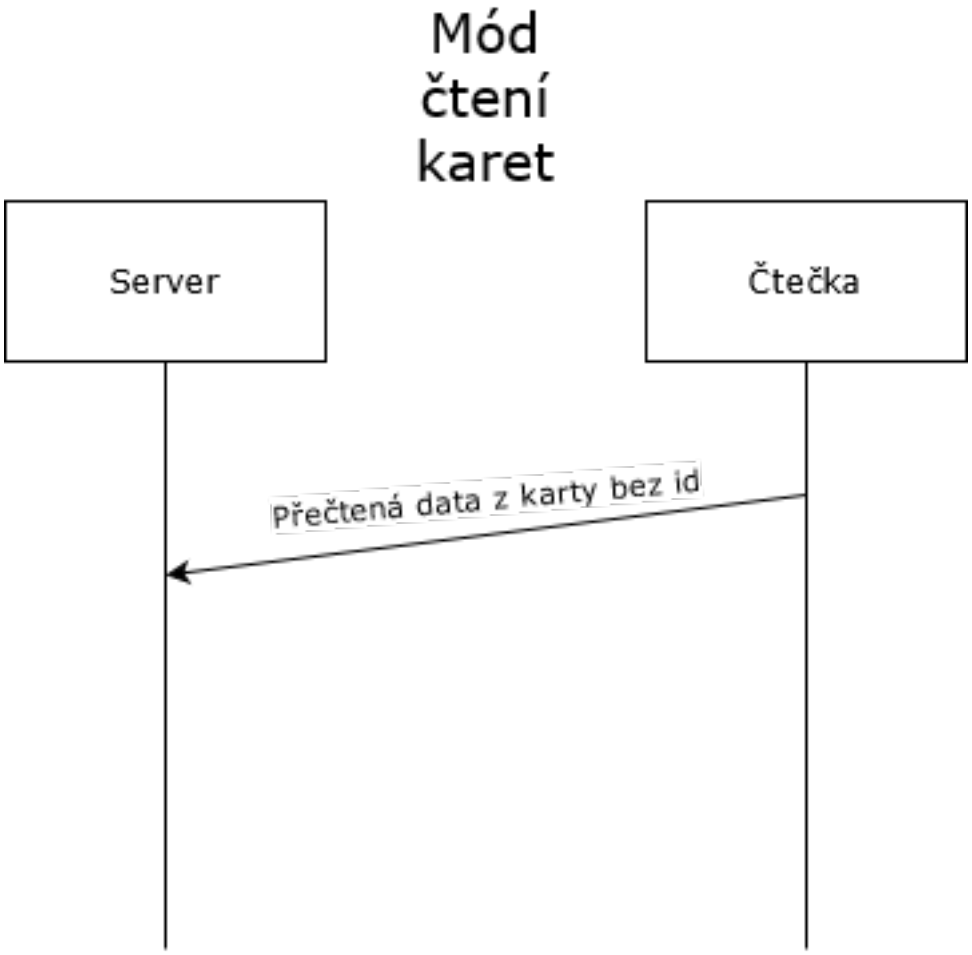
Obrázek 3.2: Komunikace mezi čtečkou a serverem při čtení karet

3.3.2.2 Mód zápisu

Mód zápisu je druhý mód čtečky, která vzniká v rámci tohoto projektu. Tento mód slouží k zápisu nových karet pro přístup. Mód si vyžádá od serveru přidělení nového ID a IV. IV zapíše nezměněné na kartu, ID zapíše v zašifrované podobě a připojí na konec TAG pro ověření původu dat. V případě úspěšného dokončení této operace vyšle čtečka potvrzení, že zapsání proběhlo a ID je zapsáno jako funkční 3.4. Při příštím čtení toto ID projde.

3.3.3 Formát dat

Pro přenos informací mezi čtečkou a serverem byl zvolen formát JSON, jednoduchý formát pro přenos informací po síti, s širokými možnostmi využití. Formát je v lidsky čitelné podobě a jednodušší, než formát XML, který je velmi

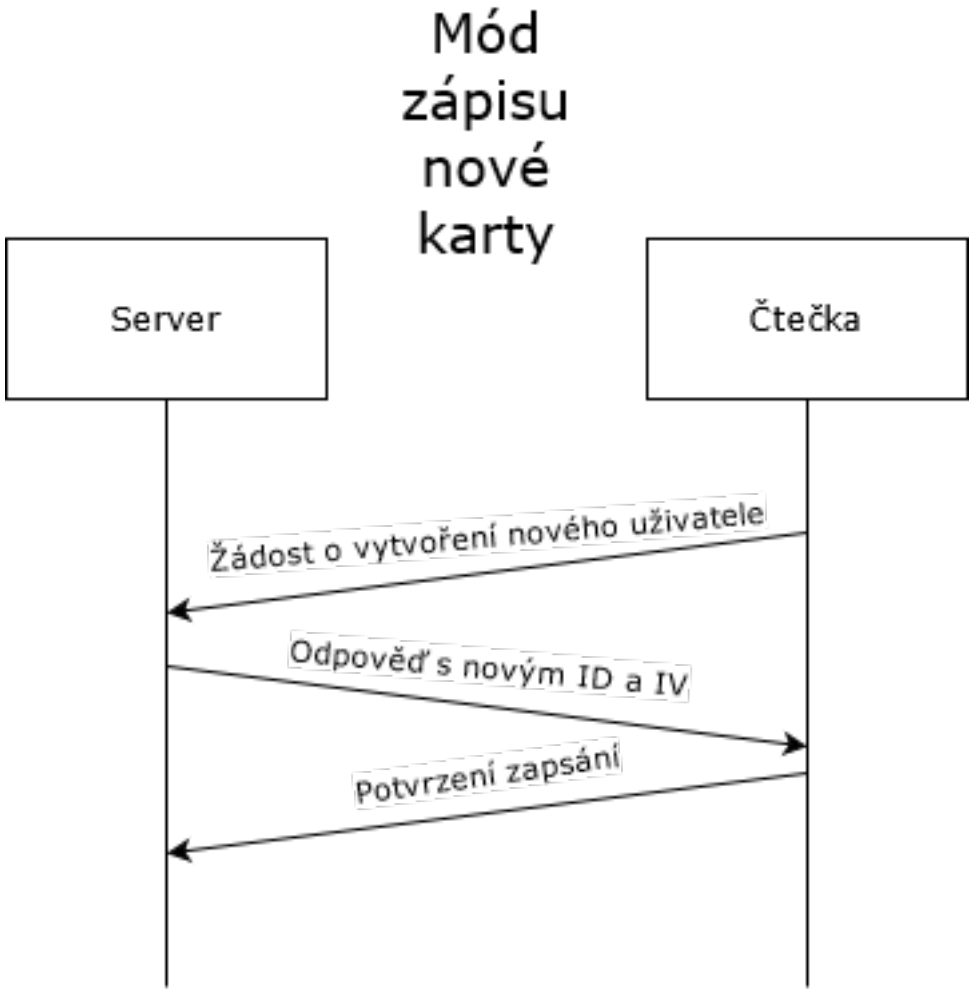


Obrázek 3.3: Komunikace mezi čtečkou a serverem při čtení karet bez ID

robustní, ale náročnější na systémové prostředky a není potřeba složitého parseru. Velkou výhodou JSON formátu je také snadná syntaxe.

3.3.4 Knihovny pro Arduino

Volba knihoven pro projekt na platformě Arduino nebyla přímočará. Z velké části se volba soustředila na to, jaké knihovny existují, co nabízí za funkcionality a jak se používají. Ne vždy byl výběr knihoven velký a často knihovny nenabízely všechny požadované funkcionality. Velkou část knihoven by bylo potřeba značně upravit, což svou náročností však již je zcela mimo rozsah tohoto projektu. Výsledek volby knihoven nemusí být vždy optimální, ať již z důvodů, že lepší knihovny nebyly autorem nalezeny, neexistovaly nebo nedošlo k jejich úspěšnému zprovoznění při jejich aplikaci. Knihovny musely fungovat se zvolenými součástmi, které byly použity v HW části.



Obrázek 3.4: Komunikace mezi čtečkou a serverem při zápisu nové karty

3.3.4.1 Adafruit-PN532

Knihovna [22] od firmy Adafruit byla zvolena, protože se jedná o výrobce konkrétního výše zmíněného PN532 shieldu. Jedná se o knihovnu se základními funkcemi. Nevyužije naplno čip PN532, nemá možnost emulovat karty. Umí pouze číst a zapisovat na karty. Jedná se o karty typu Mifare Classic a Mifare Ultralight. Byla prozkoumána i existence ostatních knihoven, které nabízejí i jiné funkce, avšak bylo shledáno, že tyto funkce nejsou nijak zásadní pro funkčnost tohoto projektu. Dále bylo shledáno, že velké množství z těchto knihoven nadále již není spravováno. Knihovna byla zvolena také pro snadné ovládání a jistotu funkčnosti s konkrétním shieldem.

3.3.4.2 U8g2

Knihovna [23] pro ovládání monochromatického displeje.Knihovna byla zvolena na základě analýzy různých nabízených druhů knihoven. Tato knihovna je snadná na užití, nabízí velké množství fontů, piktogramů a některé obrazce. Knihovna má k dispozici dobrou dokumentaci a příklady, díky čemuž se naučit používat tuto knihovnu bylo snadné. Knihovna jako taková obsahuje vše, co je v rámci tohoto projektu třeba.

3.3.4.3 Crypto

Knihovna [24] od neziskové organizace Operator foundation. Zvolena pro implementaci šifrování a autentifikace dat. Tato knihovna nabízí moderní šifry jako je AES s autentifikačními mody (AEAD) šifer jako je GCM či EAX. Jsou používány šifry a módy, které jsou považovány dnes za bezpečné. Šifry jsou paměťově optimalizované, aby se vešly a fungovaly s platformami Arduino. Krom šifer obsahuje knihovna i kryptograficky bezpečný generátor náhodných čísel. Bohužel tento generátor má na platformě Arduino nedostatek zdrojů entropie a je pomalý. Přes tuto nevýhodu knihovna má dobrou dokumentaci a snadné užití.

Tato knihovna bude obstarávat veškeré šifrování/dešifrování na čtečce.

3.3.4.4 ArduinoJSON

Vzhledem ke zvolenému způsobu komunikace skrze JSON, byla vyhledána knihovna optimalizovaná pro embedded zařízení. V tomto případě bylo nutno se vyhnout knihovnám, které spoléhají na dynamickou alokaci. Což by u Arduina mohlo vést k fragmentaci paměti. Tato knihovna nabízí staticky alokovaný objekt pro JSON + má nástroje online, které spočítají minimální velikost požadované paměti. Knihovna [25] je napsána tak, aby šetrně hospodařila s pamětí a zároveň byla dostatečně flexibilní. Díky snadné serializaci a deserializaci je vhodná pro tento projekt. Její licence je MIT. Krom samotné knihovny je k dispozici obsáhlá dokumentace a příklady.

3.3.4.5 SIM900

Pro ovládání GSM části byla vytvořena vlastní knihovna, která zvládá vytvořit TCP spojení, posílat a přijímat data přes toto spojení a ukončit jej. Knihovna funguje na základě posílání konkrétních AT commandů do modulu SIM900. Oficiální GSM knihovna pro Arduino v tomto případě nešla užít, vzhledem k tomu, že SIM900 není s touto knihovnou kompatibilní [26].

3.3.5 Knihovny pro serverovou část

Pro serverovou část byl výběr knihoven ovlivněn dvěma faktory. Za prvé tím, jaké knihovny autor již použil a za druhé tím, že by se mělo jednat o „open-

source“ knihovny. Knihovny musí být kompatibilní s již zvolenými technologickými řešeními. Serverová část byla inspirována prací na serveru TCP/IP z předmětu BI-PSI.

3.3.5.1 Rapid JSON

Velmi rychlá knihovna pro práci s JSON formátem napsaná v C++. Open-source knihovna, která se řadí mezi nejrychlejší, velice jednoduše použitelné knihovny, ideální pro potřeby tohoto projektu. Díky rychlosti a optimalizaci této knihovny bude server méně náročný na výpočetní výkon.

3.3.5.2 OpenSSL

Známá knihovna poskytující kryptografické funkce v jazyce C. Robustní open-source knihovna, velmi rozšířená. Obsahuje i kryptograficky bezpečný generátor náhodných čísel. S touto knihovnou byl autor seznámen v rámci předmětu BI-BEZ. To také vedlo k volbě této knihovny. Dokumentace + příklady užití na internetu pomáhají pochopit a použít tuto knihovu. Tato knihovna je udržovaná a pečlivě testovaná na chyby.

3.3.5.3 Sqlight3

Knihovna simulující databázi a zpracování příkazů jazyka SQL. Ačkoliv se nejedná o databázi v pravém slova smyslu (cílová databáze je soubor), pro prototyp serveru, který je třeba vytvořit rychle, je postačující. Není třeba zakládat další databázi a složitě ji nastavovat. V případě potřeby databáze pro jiné účely však není doporučeno použít tuto databázi.

3.3.6 Známé nedostatky

Známým nedostakem platformy Arduino v oblastni bezpečnosti je, že platforma neobsahuje dostatečné zdroje entropie pro kryptograficky bezpečné generátory náhodných čísel. Tento problém však není v této práci hlouběji řešen. Je využit generátor náhodných čísel z kryptografické knihovny, který využívá vnitřní stavy mikroprocesoru, ale ve výsledku je tento generátor velmi pomalý.

Druhým problémem platformy je uložení tajného klíče. Platforma Arduino neobsahuje žádný způsob uložení tajného klíče. Klíč může být uložen v kódu, nebo eeprom paměti. Ani jedno řešení není bezpečné a v obou případech se velice snadno dá hodnota klíče zjistit z již běžícího zařízení (pokud je k zařízení fyzický přístup).

Kapitola 4

Návrh řešení

V předchozí kapitole došlo k analýze dostupných prostředků HW a SW, které jsou potřebné k vytvoření funkčního řešení. Po analýze HW a SW je v této kapitole kladen důraz na celkové funkční řešení čtecího zařízení. Prezentuje se tu základní řešení, a dále řešení hlavních částí systému, které je nutné aby systém uměl.

Tato kapitola je rozdělena do 4 částí pro snadnou orientaci a logické vystavění návrhu od základního řešení po řešení v rámci této práce. Nejprve jde o seznámení se základním řešením identifikace na základě UID karty. Následně je uveden upravený návrh modu, při kterém čtečka čte příslušné karty a autentifikuje je. Pokračuje mód čtečky pro vytvoření takovýchto karet a popis daného modu. Poslední částí je komunikační část. Tato část má za úkol navrhnout komunikaci pomocí SMS.

4.1 Základní řešení

Základní řešení je prosté, karta obsahuje číslo, tzv. UID, které by mělo identifikovat přímo danou kartu. Toto číslo kromě identifikace karty jako takové identifikuje i výrobce a je i neměnné. Toto číslo se dá načíst, uložit do systému a poté, pokud se karta identifikuje svým UID, lze porovnat UID karty s UID v databázi a na základě toho tak povolit nebo zamítnout uživateli přístup. Jednoduché a rychlé řešení. Problémem UID je, že jej lze podvrhnout/měnit. Existují karty, které si toto číslo mohou změnit a fakticky se vydávat za někoho jiného. Tímto by se systém stal velmi snadným terčem útoku.

Pro snadnost komunikace se serverem se dá využít normálního packetového spojení, kdy čtečka odešle informace, které se dozvěděla z karty serveru, a ten následně odpoví. Opět snadné řešení, funkční, ale rizikové. V případě útoku „man in the middle“ nic nebrání na síti útočníkovi, aby odchytil packet s odpovědí a místo něj poslal odpověď jinou, která by tak povolila i jinak nevalidní kartu.

Pro ovládání čtečky pomocí SMS příkazů, či sdělování stavu pomocí SMS, stačí SMS přijmout a zkontrolovat její obsah. Pokud odpovídá definovaným akcím, lze tuto akci provést. Pokud ne, akce se neprovede. Z důvodu úspory paměti se SMS smaže ihned po přečtení.

4.2 Čtecí mód

Návrh řešení této práce vychází z velké části ze základního řešení, a dále toto řešení upravuje, aby bylo dosaženo větší bezpečnosti celého systému. První část je komunikace s kartou, identifikace dané karty. Je zachováno čtení UID dané karty, spolu se SAK a ATQA. Tyto údaje však neslouží k identifikaci konkrétní karty, jedná se pouze o určení typu dané karty, zda ji čtečka umí číst a jak. Pro samotnou identifikaci je navženo 64 bitové číslo, které je na dané kartě uloženo v zašifrované podobě spolu s IV a TAGem pro odšifrování a kontrolu původu dat. Díky kontrole TAGu je systém schopen zajistit, že uložené informace přišly od někoho, kdo zná tajný klíč a předpokládat, že údaje jsou validní. Všechny údaje spolu s časovou známkou jsou poslány serveru. Tyto údaje jsou převedeny do formátu JSON a poté zašifrovány a poslány spolu s TAGem a IV. Tímto se zajistí na straně serveru autentifikace dat, a že je nikdo neznalý klíče po cestě nepřečte. Pokud tato data nejsou načtena, nebo selže dešifrování a ověření TAGu, je automaticky karta zamítnuta a na server se odešlou jen údaje o kartě bez ID.

K dosažení šifrování a ověření lze využít nějaký z modu AEAD blokových šifer, případně jinou šifru, která umožňuje také autentifikaci. Možné šifry jsou AES či CHACHA20, projekt musí počítat také s tím, aby server byl schopen pracovat s danými šiframi a mody. Takže nelze využít všech šifer, které daná knihovna nabízí, protože v knihovně využité na serveru nemusí být přítomny. Ačkoliv teoreticky není problém využít na serverové části tu samou knihovnu, co na Arduinu. Výhodou openSSL je, že se jedná o větší projekt, který by měl být pečlivěji kontrolován a teoreticky obsahovat méně kritických zranitelností v implementaci šifer.

Podle tabulky v dokumentaci a doporučeních bylo rozhodnuto využít šifru CHACHA20 s autentifikačním modem POLY1305. Jedná se o proudovou šifru. Její implementace by měla být bezpečná v dané knihovně na Arduinu. Tato šifra je zároveň implementována v knihovně openSSL a je doporučená.

Tato šifra bude obstarávat zabezpečení jak přenosu, tak autentifikaci dat. Také bude využita pro šifrování uložených dat na kartě a jejich autentifikaci při čtení.

Pro každé posílání se vygeneruje nový Inicializační vektor a TAG pro danou zprávu, který se pošle na druhou stranu pro autentifikaci a zašifrování dat. Tímto dosáhneme zabezpečení komunikace mezi čtečkou a serverem.

Server následně data odšifruje, protože má stejný klíč jako čtečka. Klíč je uložen ve čtečce napevno a využívá se jak pro zabezpečení přenosu po síti,

tak pro zašifrování dat na kartě.

Server přijatá data uloží do logu, ve kterém poznamená čas a o jakou operaci se jednalo. Následně jsou přijatá data porovnána s databází a server odpoví čtečce, zda má danou kartu zamítnout či pustit. Opět data splňují výše nadefinovaná pravidla. Data jsou zašifrována.

Na základě obdržených dat čtečka povolí, nebo zamítne vstup.

4.3 Zápisový mód

Čtečka se přepne do zápisového modu. V tomto modu čtečka vyšle serveru požadavek na nová data pro zápis na kartu. Server vygeneruje nové ID na základě dalšího možného ID v tabulce databáze. Následně vygeneruje iniciační vektor, a to z důvodu větší rychlosti kryptograficky bezpečného generátoru na serveru. IV + ID je následně zašifrováno a posláno čtečce, kde jsou tato data odšifrována a uložena pro zápis na kartu. Při přiblížení karty ke čtečce dojde k zašifrování ID, zapsání IV a zapsání vytvořeného TAGu na kartu. Poté je odeslána informace o tom, že data, která čtečka obdržela, byla použita a server je zapíše do databáze jako data, která jsou autentifikována. Poté si čtečka znovu vyžádá další data. A takto celý mód pokračuje až do svého ukončení. V případě, že data nejsou zapsána, čtečka je vymaže a v případě opětovného zapnutí modu je čtečka opět vyžádá od serveru.

4.4 Ovládání skrze SMS

Další forma ovládání čtečky skrze SMS se neliší od základního řešení, pouze přibylo kontrolování telefonního čísla, ze kterého byla SMS přijata. Tímto se zamezí, aby reagovalo Arduino na příkaz od nějakého neautorizovaného čísla. Dále se SMS příkaz načte z paměti modulu SIM900 a porovná se s nadefinovanými příkazy. Pokud nějaký nadefinovaný příkaz odpovídá, provede se. SMS se smaže z paměti zařízení. Takto je čtečka schopna zpracovat jeden příkaz. Kvůli malé paměti SIM modulu, nelze příkazy dávat do fronty. Musí se vyčkat, až je vykonán jeden příkaz, než lze poslat další. Dále je možnost poslat varování na číslo, například když je příliš vysoký počet neplatných vstupních pokusů nebo v případě neočekávaného odpojení od serveru.

Kapitola 5

Realizace

Tato kapitola se zabývá především samotnou realizací návrhu čtečky. Popisuje konkrétní řešení, která byla zvolena, konkrétní třídy, které vznikly pro jaké konkrétní účely. Popisuje ovládání periferií, jak je řešeno na úrovni SW. Dále popisuje propojení všech tříd a knihoven v hlavním souboru. Krom samotné čtečky se v kapitole vyskytuje i řešení serveru, kde se zabývá jeho funkčností.

Kapitola se dělí na 2 části, první část zabývající se samotným řešením čtečky a druhá část zabývající se serverem. Rozdělení této kapitoly je stejné jako zamýšlená funkčnost. Pokud server dodrží schéma komunikace se čtečkou, měl by být snadno vyměnitelný a nezávislý na zvoleném řešení čtečky.

Na obrázku 5.1 lze vidět návrh celého systému čtečky a na následném obrázku lze vidět fotografii konkrétního sestaveného řešení 5.2.

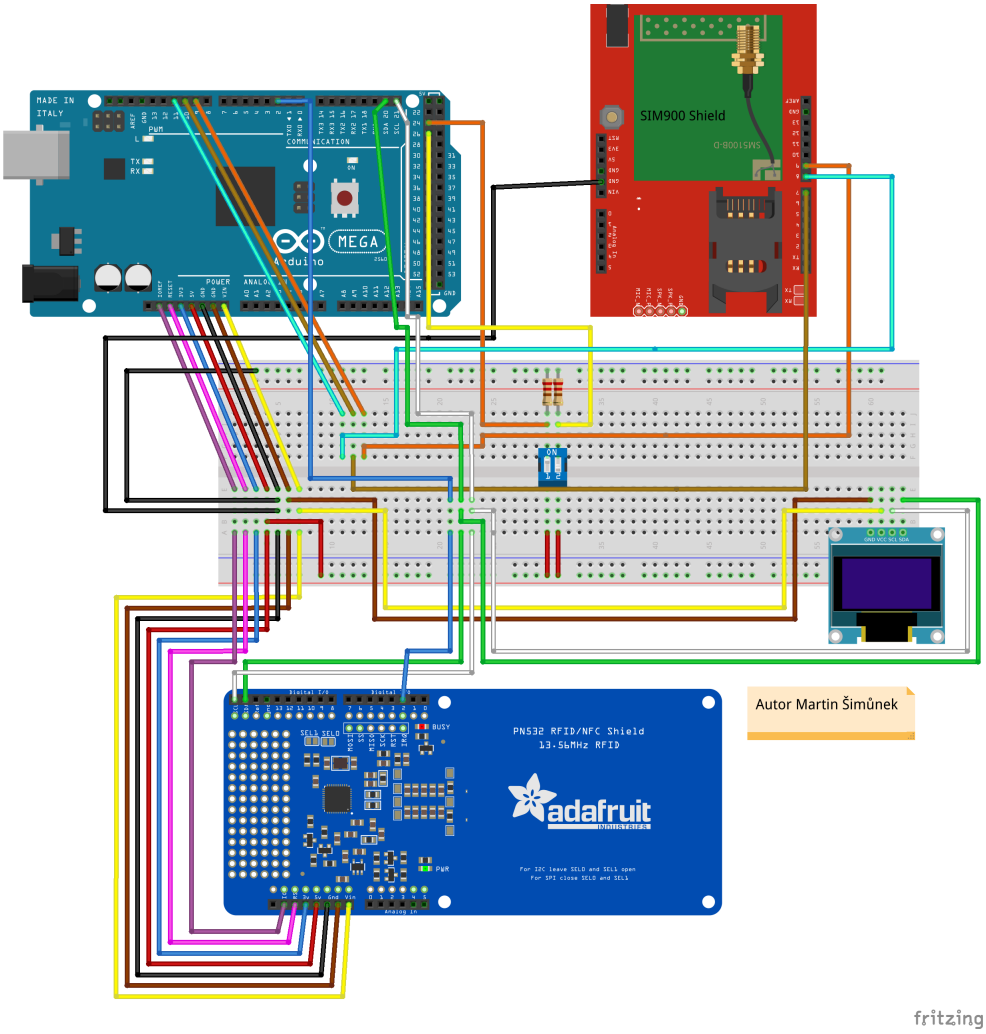
5.1 Řešení čtečky

V této části jde řešení koncového zařízení čtečky. Samotného stroje na čtení a zpracování dat. V této části se řeší propojení HW a ovládání jednotlivých částí pomocí vytvořených tříd. Zaměřuje se na implementaci a využití zvolených knihoven.

Tato sekce je rozdělena aby popisovala nejprve dílčí úkoly řešení, a poté hlavní třídu, která dílčí řešení propojuje.

5.1.1 Propojení zařízení

Fyzické propojení Arduina a ostatních částí je vytvořeno pomocí nepájivého pole. Nelze využít vlastností shieldů, jelikož piny SIM900 shieldu jsou špatně rozvrženy a nejsou kompatibilní s rozložením pinů Arduina Mega. Navíc NFC shield nemá dutinkové lišty, do kterých by šlo zasunout další shield. Propojení NFC shiledu a OLED displeje je řešeno I2C sběrnicí. Propojení se SIM900 je řešeno pomocí softwarově simulovaného UART.



Obrázek 5.1: Návrh obvodu

Napájení Arduina je řešeno pomocí USB. To umožňuje dostatečné napájení všech periferii kromě SIM900 shieldu, který je napájen externě 12V 2A DC zdrojem.

5.1.2 SMS příkazy

SMS příkazy nabízejí možnost získání informací či nastavení čtečky. V rámci přijetí SMS příkazu nejprve dojde k ověření čísla, ze kterého SMS příkaz přišel. Pokud toto číslo souhlasí s číslem uloženým v paměti zařízení, příkaz se dále zpracovává, ověří se, zda příkaz odpovídá některému z nadefinovaných příkazů. Pokud ano, provede se příslušná akce, pokud ne, je odeslána SMS s chybou na autentifikované telefonní číslo. V tabulce 5.1 jsou vypsány jednot-

livé příkazy.

Příkaz	Argument/y	Popis	Odpověď
STATUS	-	zjištění obecných informací o čtečce	obecné informace
STATUS-CONNECT	-	zjištění informací o připojení na server	informace o připojení
RESET	-	restartování čtečky, pokud není zamčená	-
SERVER:	<adresa serveru>:<číslo portu>	změna serveru čtečky	-
SERVER-LOCK	-	uzamknutí serveru proti změně či restartování	-
PHONE:	<telefonní číslo>	změna autentifikovaného telefonního čísla	potvrzení poslané na nové telefonní číslo

Tabulka 5.1: Tabulka SMS příkazů

5.1.3 Ovládání SIM900

K ovládání SIM900 se používají textové AT příkazy, které se posílají pomocí UART, a na které zařízení odpovídá opět textem. K ovládání tohoto zařízení byla vytvořena třída s předdefinovanými příkazy, která umožňuje připojení k internetu a odeslání SMS. Tato třída navíc umí zapnout modul SIM900, pokud je vypnut.

Modul je nastaven vnitřně tak, aby neopakoval právě obdržené příkazy. V továrním nastavení modul všechny příkazy zopakuje. Dále je nastaven, aby ukládal příchozí SMS do SIM karty, kam se vejde až 10 SMS. Dále je nastaveno, aby si modul po zapnutí vzal čas a nastavil se podle času sítě, tímto odpadá nutnost manuálního nastavení hodin.

Komunikace s modulem je zařízená po softwarově emulované sériové lince. Pro přenos dat ze zařízení byl zvětšen v knihovně buffer této linky.

5.1.4 NFC shield

Pro potřeby tohoto projektu bylo v knihovně NFC shieldu upraveno několik funkcí. Jedná se o funkce čtení ID, readPassiveTargetID a readDetectedPassiveTargetID, které oproti své původní implementaci vracejí ATQA a SAK, stejně jako možnost vrácení ATS, které je v rámci tohoto projektu nevyužito.

NFC shield je nastaven tak, aby komunikoval po I2C sběrnici. Shiled čeká v modu pasivním. Když se přiblíží karta, shield se ozve pomocí interuptu na digital pinu 2. Aby funkce čekání na ID v poli shieldu nebyla blokující, je k ní přidán timeout. Tímto se zajistí, že Arduino může vykonat i jiné další funkce, pokud v dosahu shieldu není žádná karta, která by potřebovala přečíst.

Na základě detekované karty při přečtení údajů, je rozhodnuto, co se bude dít s danou kartou dále. Například jestli se pokusí zařízení přečíst danou kartu.

5.1.5 Card IO

Třída obstarávající zápis a čtení a zápis ID na karty, spolu se šifrováním a dešifrováním. Tato třída obsahuje funkce na čtení a zápis karet Ultralight a Classic.

Třída obsahuje dvě následné třídy pro čtení různých typů karet, přečte určený počet dat z karty, rozdělí je dle nadefinovaného chování, pokusí se o odšifrování a ověření TAGu. Pokud se toto povede, metody vrátí 64 bitové ID zapsané na kartě. Pokud se nepodaří, metody vrací 0.

Zápis je proveden tak, že obdržené ID je zašifrováno a je vytvořen TAG. IV,ID a TAG jsou zapsány na kartu. Při zápisu je vráceno true, pokud se povedl.

5.1.5.1 MifareClassic

Třída řešící zápis a čtení z konkrétní karty, z Mifare Classic. Uzpůsobená tak, aby řešila čtení a zápis a přitom přeskakovala konečné sektory, které obsahují informace, jako přístupová práva k daným sektorům a klíče k daným sektorům. Zároveň přeskočí sektor 0, který obsahuje informace od výrobce karty.

Pro uživatele se tato třída tváří, že čte/zapisuje na souvislý blok dat, jen je nutno zdůraznit, že například zápis do 3. sektoru se neudělá, protože se tento sektor přeskočí.

5.1.5.2 MifareUltralight

Další třída, která řeší zápis na konkrétní kartu, a to Mifare Ultralight. Tato třída je přizpůsobená jiné paměťové struktuře Mifare Ultraligh. Pro uživatele se opět tváří jako souvislý blok stránek. Vzhledem k jiné konstrukci je však maximální počet zapisovatelných bajtů mnohem menší, než u předchozí třídy.

5.1.6 Display

Třída pracující s knihovnou U8g2 [23], slouží k jednoduchému vypisování textu na obrazovku, zobrazování jednoduchých piktogramů a tvarů. Velmi jednoduše navržená pro zpřehlednění výsledného kódu. A snadnou komunikaci s uživatelem.

5.1.7 IO Helper

Knihovna vytvořená pro obsah funkcí, které pomáhají transformovat různé druhy výstupů na vstupy a obráceně. Obsahuje funkce pro převádění čísel na pole bytů a pole bytů na čísla. Jedná se o c-funkce pro rychlost. Hlavním účelem je opět zpřehlednění kódu a vytvoření knihovny, která má potřebné funkce pro převádění na jednom místě.

5.1.8 JsonWriter

Třída napsána podle dokumentace k Arduino JSON, jedná se o vlastní třídu implementující funkci streamu. Tato třída zajišťuje dvě základní metody, které podle dokumentace musí zajišťovat. Metodu write ve verzi zapsání jednoho bytu nebo zapsání více bytů. To jsou jediné potřebné požadavky. Tato třída data šifruje a odesílá po síti.

Tato třída je konstruována tak, aby splnila i požadavky odesílaných dat. Odesílá IV, který si nechá vygenerovat kryptograficky bezpečným generátorem náhodných čísel a použije ho na šifrování. Dále umí odeslat předpokládanou délku dat a na závěr odeslat TAG pro autentifikaci dat na druhé straně.

Třída je konstruovaná s bufferem. Do tohoto bufferu se ukládají odeslaná data, aby se zašifrovala a odeslala najednou. To zlepšuje rychlost a efektivitu, protože šifrování a odeslání jednoho bytu po síti je značně neefektivní.

5.1.9 JsonReader

JsonReader je obdobná třída jako JsonWriter. Opět třída implementující streamování dat, tentokrát příjem dat ze sítě. Obsahuje podle dokumentace povinné metody read, pro čtení jednoho či více bytů.

Třída navíc umí přijmout a nastavit IV, očekávanou délku dat, podle které se řídí a přijmout a nastavit TAG pro autentifikaci obdržených dat.

Třída obsahuje buffer, který je při prvním volání metody read zaplněn a odšifrován. Následně data vrací až do doby, než je buffer zcela vyprázdněn a je třeba ho zaplnit znovu. Tímto se zlepší rychlost a efektivita, než pouze po jednom bytu číst z uložených dat v SIM900 a odšifrovávat postupně.

Po načtení všech dat umožňuje třída zkontrolovat TAG. Pokud TAG souhlasí, data se mohou nadále využívat. V opačném případě by se data měla zahodit.

5.1.10 UID

Struktura, která slouží pouze k uložení UID a jeho délky. Struktura nemá žádnou jinou funkci krom této.

5.1.11 Defines

Hlavičkový soubor, obsahující všechna makra na jednom místě pro snadné úpravy. Tento soubor v sobě obsahuje množství definic všech důležitých konstant, které v případě potřeby lze snadno měnit

Tento soubor během implementace narostl více, než bylo plánováno, i přesto zůstává stále přehledným a snadno změnitelným.

5.1.12 Mem

Knihovna s funkcí pro získání informací o volné paměti zařízení Arduino. Slouží pouze pro debugovací účely v rámci projektu.

5.1.13 Hlavní soubor

Hlavní soubor (Arduino_NFC.ino) obsahující celý program Arduina a kombinující všechny výše zmíněné a implementované knihovny do funkčního celku. Inicializace veškerých částí se provádí ve funkci setup, která se volá v rámci zapnutí Arduina pouze jednou. Dochází zde k inicializaci důležitých částí, jako je Serial port, na příslušnou baud rate, dále třídy ovládající SIM900 modul, která využívá Serial. Následně dojde ke zkontrolování stavu modulu SIM900 zda je zapnutý, pokud ne, pokusí se o zapnutí modulu, zkontroluje se připojení NFC shieldu a inicializují některé hodnoty proměnných. Některé důležité hodnoty se inicializují z paměti eeprom. Jedná se o tajný klíč, číslo, ze kterého chodí ovládací SMS a adresa a port serveru.

Poté postoupí program do nekonečné smyčky. Tato funkce s vlastním scopem, je volána pořád dokola. Tato funkce po každém doběhnutí zanikne, a opět se znovu zavolá, proto nelze spoléhat na zachování informací v lokálních proměnných a statických polích vytvořených v této funkci. Vše se musí řešit přes globální proměnné.

Nejprve je získán stav HW přepínačů určujících mód a jestli se má čtečka připojit na server a v jakém módu má čtečka běžet. Následně dojde ke kontrole, zda nepřišla SMS a pokud ano, SMS je přečtena a je zkontrolováno telefonní číslo, ze kterého byla SMS poslána. Pokud toto číslo odpovídá, je SMS porovnána s definovanými akcemi. Pokud akce odpovídá, je vykonána. V opačném případě se nic nestane.

Následně je v loopu zkontrolován stav připojení. Pokud stav připojení odpovídá nastavené hodnotě, nic se neděje. Pokud stav neodpovídá, je podle požadovaného stavu připojení ukončeno či navázáno, nebo je vykonán pokus o tuto akci.

V případě, že je čtečka připojena na server, je umožněno vyzkoušet, zda není v dosahu čtečky karta, která by se dala přečíst. Pokud taková karta není v dosahu, čtečka timeoutuje. Zkontroluje se, zda je nastaven mód zápisu a pokud ano a zároveň čtečka nemá v sobě data určená k zápisu, je požádáno o nová data. V případě, že data jsou obdržena a mód čtečky je stále nastaven pro zápis, při přečtení karty je proveden pokus o zápis dat, v případě, že přiblížená karta je podporována. Výsledek tohoto pokusu je poslán na server.

V případě, že čtečka je ve čtecím modu a je přečtena karta, která patří mezi podporované karty, jsou informace poslány na server s ID a vyčká se na odpověď serveru, která určí zda je karta přijata či zamítnuta. V případě nepřečtení ID nebo neověření TAGu je karta zamítnuta.

5.2 Řešení serveru

Server je řešen pouze jako prototyp pro fungování spolu se čtečkou. Celý server umí pouze přijímat data ze čtečky a je řešen pouze na principu toho nejnutnějšího. Celý server je rozdělen do funkcí a napsán v rámci jednoho main souboru. Server kvůli své jednoduchosti má i tak dostatečně přehledný kód.

Server je pouze jednovláknový, pro ukázku fungování s jednou čtečkou. Server je založen na základě semestrální práce z předmětu BI-PSI. Jedná se o server navazující TCP připojení. Celý server čeká na navázaní spojení ze strany čtečky. Po navázání spojení tento server čeká na přijetí dat ze čtečky. Server pouze pasivně poslouchá a odpovídá čtečce. Sám komunikaci se čtečkou nenavazuje.

Po přijetí dat je server zkontroluje a na jejich základě se rozhodne pro odpovídající akci.

Server je napojen na jednoduchou databázi, která má formu sqlight, proti které ověřuje povolené karty a jejich ID.

Server čeká na data ze čtečky, v případě že je dostane, podle definovaného protokolu si vytáhne z dat IV, velikost dat, data a TAG. Data se pokusí dešifrovat a ověří TAGem. Pokud vše odpovídá, server data deserializuje do JSON formátu a analyzuje jeho obsah. Podle obsahu JSONu vytvoří příslušné události log a pokračuje s příslušnou událostí.

5.2.1 Žádost o data pro novou kartu

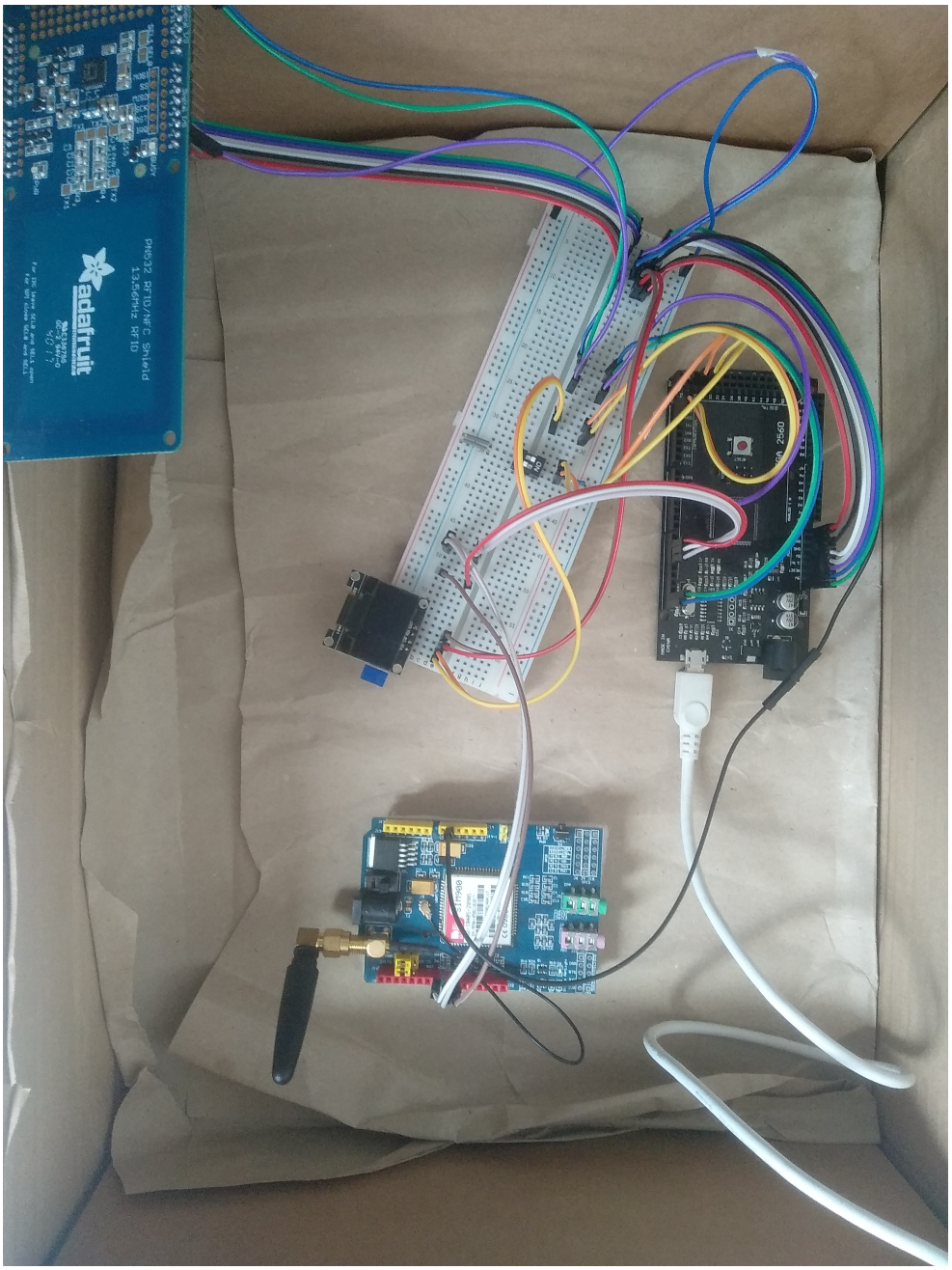
V případě že JSON obsahuje položku INS s hodnotou 1., jedná se o žádost pro získání nového ID a IV pro vytvoření nové karty. Server si poté zjistí jaké další ID je v pořadí v databázi a toto ID vrátí společně s IV, které je vygenerováno pomocí kryptografického generátoru náhodných čísel. Tyto údaje vrátí čtečce.

Následně server čeká na potvrzení, že data byla zapsána. Pokud přijde něco jiného, než potvrzení, je počítáno s tím, že data zapsána nebyla a server

pokračuje ve své funkci. Pokud přijde potvrzení, server vytvoří nový řádek v tabulce databáze. A opět pokračuje ve své funkčnosti.

5.2.2 Žádost o ověření ID

Čtečka dělá implicitně ověřování údajů, které dostane. Konkrétně se jedná o ID. Pokud přijatý JSON neobsahuje INS, a obsahuje položku ID, informace se zalogují a server odpoví, zda je ID validní či není. V případě, že JSON neobsahuje položku ID, server pouze zaloguje příchozí informace a dále již neodpovídá.



Obrázek 5.2: Sestavené řešení podle návrhu

Kapitola 6

Testování

Testování tohoto zařízení probíhalo ve 4 rozdílných fázích. První 3 fáze byly testovány nezávisle na celkovém zařízení a probíhaly v rámci sestavování zařízení a jeho programování. Jedná se o testování AT příkazů, samotných tříd a testování odpovědí na SMS příkazy. Podle tohoto je rozdělena i tato kapitola. Poslední testování probíhalo v rámci již funkčního celku.

Tato kapitola je rozdělena podle jednotlivých testovacích fází. Jednotlivé fáze jsou zde popsány podle postupu testování a jaké výsledky testování přineslo.

6.1 Testování AT příkazů

Potřebné AT příkazy byly testovány za pomocí dokumentace [26], hlavně se jednalo o chování jednotlivých příkazů a možné odpovědi na ně. Arduino přeposílalo data z jednoho sériového portu na druhý. V tomto byly zadávány příkazy ručně a byla sledována reakce shield modulu, zda jeho chování bylo takové, jaké bylo potřeba. Takto byly otestovány všechny příkazy a postupně i jejich navazování v rámci připojování. Příkladem může být posloupnost příkazů pro navázání TCP spojení. Pro TCP spojení a jeho testování bylo využito linuxového nástroje netcat, který tvořil jednoduchý testovací server.

Všechny příkazy byly postupně úspěšně otestovány jako funkční, i když v některých případech byla dokumentace nedostatečně podrobná či obsahovala chyby, protože některé příkazy fungují jen v určitých stavech modulu.

6.2 Testování jednotlivých tříd

Testování jednotlivých tříd probíhalo v rámci jejich vytváření. Pokaždé, když byla napsána nová metoda nebo funkce, byla otestována, zda funguje, jak je předpokládáno. Toto platí ke každé metodě v každé třídě. Metody byly testovány samotné a poté v souladu se zamýšlenou funkcí s dalšími metodami.

Největším problémem v testování bylo poměrně časté selhání metod, které měly ovládat SIM900 modul. Jedná se hlavně o jejich odpovědi. Z tohoto důvodu bylo na SIM900 modulu vypnuto echo, jelikož se ukázalo jako problematické. Po vypnutí funkce echa se většina dosud problematických metod začala chovat tak, jak bylo předpokládáno. Jediná metoda, která občas selhala, je metoda, která má vyzkoušet, zda je zařízení připraveno a napájeno. Občas chybně vyhodnotila stav a zařízení místo zapnutí vypnula.

Tato chyba je výjimečná a již se po pár úpravách neobjevila, přesto teoreticky je možné, že při nějakých stavech by se chyba mohla projevit znovu.

6.3 Testování SMS příkazů

Testování jednotlivých příkazů probíhalo postupně, kdy byl vždy odeslán konkrétní příkaz. Nejprve byly příkazy odeslány z mobilního telefonu s autentifikovaným číslem. Tyto příkazy byly provedeny a ověřen jejich efekt. Následně příkazy byly vyslány z jiného čísla, které již nebylo autentifikováno v čtečce a tyto příkazy již nebyly vykonány.

Při testování došlo k problémům, že některé příkazy občas nereagovaly, nejčastěji po startu zařízení a připojení. Důvod proč tomu tak bylo, není znám, přesto jako jedno z možných vysvětlení se nabízí přijetí SMS, které zabraly volné místo v paměti a zpracovaly se místo poslaného příkazu. Kupříkladu se takto může jednat o SMS od operátora. Toto se objevilo výjimečně, replikování problému se projevilo jako obtížné.

6.4 Testování celku

Celek byl otestován v rámci své předpokládané funkčnosti. Čtečka i modul SIM900 byl vždy vypnut a zapnut v jistém módu, následně byly módy přepínány pomocí přepínačů bez restartování čtečky a byl simulován i výpadek a znovupřipojení na server, případně odpojení od serveru.

Módy se nejprve testovaly odděleně, vždy při přepnutí módu bylo samotné zařízení restartováno. Následně byl proveden finální test, kdy byly módy pouze přepínány bez restartů.

6.4.1 Testování módu čtení

Mód čtení byl otestován s kartami zapsanými, nezapsanými i nepodporovanými.

Testování celkem využilo 3 různých typů karet. Jednalo se o MIFARE Classic, DESfire a Ultralight. Karty Ultralight a Classic byly rozděleny na zaregistrované a nezaregistrované.

První pokus byl s kartou DESfire. Karta byla identifikována jako karta, kterou nelze číst a pouze byla odeslána informace o této kartě. Přístup dle očekávání nebyl povolen.

Další fází bylo otestovat karty, které je zařízení schopno přečíst, ale nejsou na nich zapsány požadované údaje. Dle očekávání byla odezva rychlá a karty byly zamítnuty, zatímco informace o těchto kartách byly poslány na server.

Poslední část testování byla se zapsanými kartami. Dle očekávání karty byly přečteny, následně data odšifrována a odeslána na server k ověření. Server odpověděl a karty byly přijaty, přístup byl povolen. Jediným problémem se ukázala rychlost komunikace karet se serverem. Rychlost komunikace je pomalejší. Trvá zhruba 5 vteřin.

Ve výsledku je tento test vyhodnocen jako úspěšný. Jediný nedostatek zde je pomalá odezva zařízení.

6.4.2 Testování módu zápisu

Mód zápisu byl testován pouze s podporovanými kartami a to Mifare Classic a Mifare Ultralight pro ověření funkčnosti daného modu. Následně tyto karty byly ověřeny, zda prošly v módu čtení.

Čtečka byla ze začátku přepnuta do módu zápisu a restartována. Postupně po sobě byly přiloženy karty Classic a Ultralight, na které bylo úspěšně zapsáno. Tyto karty poté byly ověřeny v resetovaném módu čtení zda prošly. Server tyto karty ověřil jako validní. Rychlost komunikace v tomto případě byla opět poměrně pomalá. Avšak vzhledem k rychlosti a četnosti operací, jako je zápis nové karty, je tento nedostatek zanedbatelný.

Ve výsledku se jedná o úspěšný test.

6.4.3 Testování při přepínání módů

Došlo k testování módu čtení, kdy byly načteny karty, poté byl mód přepnut do módu zápisu a bylo zapsáno na jiné karty, poté opět došlo k přepnutí modu na mód čtení. Toto simulovalo běžný stav používání čtečky.

Byly využity všechny dostupné karty z předchozích testů. Karty prázdné, karty zapsané a karty, které čtečka neumí číst. Nejprve byl pokus přečíst dané karty a validovat je. Tento pokus proběhl hladce a všechny karty, které měly být validovány, validovány byly.

Dále proběhl pokus zapsat nové karty a přepnout zpět do módu čtení bez jakýchkoliv restartů mezi změnou módů. Tento test byl úspěšný a nově zapsané karty byly přijaty.

Ještě proběhlo jedno přepnutí módu na zápis, zapsání karty a přepnutí módu zpět na mód čtení a přečtení karty. Opět bez problému.

6.4.4 Shrnutí

Čtečka prošla základními uživatelskými testy v pořádku. Během těchto testů se neobjevily žádné zásadní chyby ze strany uživatelského použití čtečky. Jediným nedostatkem čtečky se ukazuje pomalá odezva na čtení karet a komunikace se serverem. Důvod této pomalé odezvy lze hledat v šifrování dat, využití pomalého a nestálého připojení přes mobilní sítě 2. generace a pomalému získávání entropie pro kryptograficky bezpečný generátor náhodných čísel.

Kapitola 7

Možná vylepšení

Vytvořené řešení obsahuje prostor pro navázání v podobě dalších možných řešení či rozšíření stávajícího řešení o další doplňující funkce, které však již nejsou z důvodu omezeného rozsahu této práce její součástí. Jedním z hlavních nedostatků je zdroj entropie pro generátor náhodných čísel, kdy současné řešení je velmi pomalé a do jisté míry by se mohlo stát i předvídatelným.

Nejlepším vylepšením by byla implementace knihovny, schopné komunikovat s kartami s větší úrovní zabezpečení, jako je karta DESfire. Tato implementace čistě závisí na vyšších vrstvách a SW. HW je schopen tuto implementaci využít, pokud bude rozumně náročná pro omezené prostředky Arduina. Dalším možným vylepšením by mohla být integrace bezpečného úložiště tajných klíčů.

Samotná serverová část, která je pro účely projektu do značné míry obsahově omezená, by mohla doznat také vylepšení. Díky komunikaci ve formátu JSON by teoreticky mohla snadno vzniknout serverová část, která by nabízela snadnou správu daného zařízení, popřípadě správu více zařízení naráz, různé uživatelské profily a atp.

Závěr

Cílem práce bylo vytvoření funkčního přenosného zařízení schopného identifikovat osobu na základě vlastnictví NFC karty. Takovéto zařízení má být přenosné, lehké na užívání a má komunikovat skrze mobilní síť. Samotné zařízení musí být dobře zabezpečené.

V analytické části šlo o prozkoumání již existujících řešení, která jsou nabízena na trhu nebo sestavena v rámci akademické činnosti.

Všechny zadané cíle byly splněny.

V této práci se v souladu s úvodními cílí práce podařilo vytvořit funkční čtečku na platformě Arduino. Navržené řešení je bezpečné, co se týče útoků, jako je odposlouchávání či manipulace s přenášenými daty, řešení splňuje ovládání pomocí SMS a komunikuje se serverem skrze GPRS síť. Vytvořené řešení by mohlo mít uplatnění pro identifikaci osob v rámci společných prostor bytového domu, otevřeného areálu, jakým je kupříkladu botanická zahrada, zoo, apod. Jedná se o funkční prototyp čtečky, postavený na levných komponentech. Samotná čtečka dokazuje, že funkční bezdrátový přístup do budov lze sestavit levně. Toto řešení nabízí větší úroveň zabezpečení než obyčejný klíč. Zejména pro společně přístupná místa. Zařízení je odolné vůči útokům, které hrozí konvenčním mechanickým zámkům. Další výhodou tohoto zařízení je, že není třeba vyměňovat vložku a vytvářet nové klíče při odcizení karty, tak jako u mechanického zámku. Stačí pouze odcizenou kartu deaktivovat.

Analytická část rozebírá existující řešení včetně těch, která nevyužívají technologii NFC. Značné množství řešení využívá Bluetooth pro spárování s mobilním zařízením. Daná řešení bývají samostatná, avšak umožňují napojení na rozsáhlejší systémy jiných výrobců v rámci sítě IoT pro domácnosti. Tato rozšíření nabízí například používání a posílání dočasných klíčů, nebo sledování stavu zařízení vzdáleně prostřednictvím internetu.

Bibliografie

1. TECHOPEDIA. *What is Proprietary Software? - Definition from Techopedia* [online]. Techopedia [cit. 2021-04-07]. Dostupné z: https://www.techopedia.com/definition/4333/proprietary-software.
2. HASELSTEINER, Ernst; BREITFUSS, Klemens. Security in near field communication (NFC). In: *Workshop on RFID security*. 2006, sv. 517, s. 517.
3. CHATTHA, Naveed Ashraf. NFC — Vulnerabilities and defense. In: *2014 Conference on Information Assurance and Cyber Security (CIACS)*. 2014, s. 35–38. Dostupné z DOI: 10.1109/CIACS.2014.6861328.
4. ELMUE. *DIY electronic RFID Door Lock with Battery Backup* [online]. CodeProject, 2018-05-07 [cit. 2020-03-22]. Dostupné z: https://www.codeproject.com/Articles/1096861/DIY-electronic-RFID-Door-Lock-with-Battery-Backup?msg=5794076%5C#xx5794076xx.
5. CHEN, Zhiqun. *Java card technology for smart cards: architecture and programmer's guide*. Addison-Wesley Professional, 2000.
6. *NFC Application: Access Control* [online] [cit. 2021-04-25]. Dostupné z: https://www.nxp.com/design/training/nfc-application-access-control:TIP-NFC-ACCESS-CONTROL.
7. INC., Apple. *HomeKit* [online] [cit. 2021-04-25]. Dostupné z: https://developer.apple.com/homekit/.
8. SOLIMAN, Moataz; ABIODUN, Tobi; HAMOUDA, Tarek; ZHOU, Jiehan; LUNG, Chung-Horng. Smart Home: Integrating Internet of Things with Web Services and Cloud Computing. In: *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*. 2013, sv. 2, s. 317–320. Dostupné z DOI: 10.1109/CloudCom.2013.155.
9. *Backup Batteries* [online]. Kisi [cit. 2021-04-07]. Dostupné z: https://www.getkisi.com/guides/backup-batteries-for-access-control.

10. DASGUPTA, Dipankar; ROY, Arunava; NAG, Abhijit. Multi-factor authentication. In: *Advances in User Authentication*. Springer,Cham, 2017, s. 185–233. ISBN 978-3-319-58806-3. Dostupné z DOI: https://doi.org/10.1007/978-3-319-58808-7_5.

11. INDUSTRIES, Adafruit. *Adafruit PN532 NFC/RFID Controller Shield for Arduino Extras* [online]. adafruit industries, [n.d.] [cit. 2021-04-23]. Dostupné z: https://www.adafruit.com/product/789.

12. *Keycard Entry Systems: Kisis Guide to Card Access* [online]. Kisi [cit. 2021-04-15]. Dostupné z: https://www.getkisi.com/keycard-access-systems.

13. *Inteligentní Zámek* [online]. Netatmo®, 2020 [cit. 2021-04-07]. Dostupné z: https://www.netatmo.com/cs-cz/security/doorlock.

14. AQARA. *Aqara Smart Door Lock N100* [online]. 2020 [cit. 2021-04-25]. Dostupné z: https://www.aqara.com/en/smart_door_lock_n100.html.

15. *NUKI SMART LOCK 2.0* [online] [cit. 2021-04-25]. Dostupné z: https://www.nuki-lock.cz/.

16. YALELOCK. *Produkty* [online] [cit. 2021-04-25]. Dostupné z: https://www.yalelock.cz/cs/yale/cz/produkty/smart-living/linus/.

17. *Danalock · Danalock V3 - The Smart Home Enabler* [online] [cit. 2021-04-25]. Dostupné z: https://danalock.com/.

18. ISLOG. *liblogicalaccess* [soft.]. 2020. 2020-03-17 [cit. 2021-04-24]. Dostupné z: http://liblogicalaccess.islog.com/.

19. CHAIRUNNAS, A; ABDURRASYID, I. Near field communication (NFC) model for arduino uno based security systems office system. *IOP Conference Series: Materials Science and Engineering* [online]. 2018, roč. 332, s. 012006 [cit. 2020-03-19]. Dostupné z DOI: 10.1088/1757-899x/332/1/012006.

20. LEEKONGXUE, Sialee; LI, Li; PAGE, Tomas. Smart Door Monitoring and Locking System using SIM900 GSM Shield and Arduino UNO. *International Journal of Engineering Research and Technology (IJERT)*. 2020, roč. V9, s. 47–52. ISSN 2278-0181. Dostupné z DOI: 10.17577/IJERTV9IS040011.

21. PALMA, Daniel; AGUDO, Juan Enrique; SÁNCHEZ, Héctor; MACÍAS, Miguel Macías. An Internet of Things Example: Classrooms Access Control over Near Field Communication. *Sensors* [online]. 2014, roč. 14, č. 4, s. 6998–7012 [cit. 2020-03-19]. ISSN 1424-8220. Dostupné z DOI: 10.3390/s140406998.

22. ADAFRUIT INDUSTRIES, LLC. *Adafruit-PN53* [soft.]. 2020. 2020-12-11 [cit. 2021-04-23]. Dostupné z: https://github.com/adafruit/Adafruit-PN532.

23. OLIKRAUS. *U8g2* [soft.]. 2021. 2021-04-19 [cit. 2021-04-23]. Dostupné z: https://github.com/OperatorFoundation/Crypto.

24. OPERATOR FOUNDATION. *Crypto* [soft.]. 2018. 2018-11-20 [cit. 2021-04-23]. Dostupné z: https://github.com/OperatorFoundation/Crypto.

25. BLANCHON, Benoît. *ArduinoJson* [soft.]. 2021. 2021-02-15 [cit. 2021-04-23]. Dostupné z: https://github.com/bblanchon/ArduinoJson.

26. LTD., SIMCom wireless solutions. *SIM900_AT Command Manual_V1.03* [online]. SIMCom wireless solutions Ltd., 2010-12-24 [cit. 2021-04-23]. Dostupné z: https://www.espruino.com/datasheets/SIM900_AT.pdf.

27. BELLARE, Mihir; ROGAWAY, Phillip; WAGNER, David. A conventional authenticated-encryption mode. *manuscript, April*. [N.d.].

28. *What is Arduino* [online]. Arduino, 2018-02-05 [cit. 2020-03-19]. Dostupné z: https://www.arduino.cc/en/guide/introduction.

29. B_E_N. *What is an Arduino* [online]. sparkfun, 2013-02-26 [cit. 2020-04-15]. Dostupné z: https://learn.sparkfun.com/tutorials/what-is-an-arduino/all.

30. ZHICONG, Qian; DELIN, Luo; SHUNXIANG, Wu. Analysis and Design of A Mobile Forensic Software System Based on AT Commands. In: *2008 IEEE International Symposium on Knowledge Acquisition and Modeling Workshop* [online]. IEEE, 2009, s. 597–600 [cit. 2021-04-05]. Dostupné z DOI: 10.1109/KAMW.2008.4810559.

31. SEMICONDUCTORS, NXP. MIFARE DESFire EV1. In: [online]. 2015 [cit. 2020-03-19]. Dostupné z: https://www.nxp.com/design/documentation:DOCUMENTATION%5C#/collection=documents&start=0&max=12&language=en&keyword=MF3ICDX21_41_81_SDS&depth=1.

32. SPACEY, John. *What is Data Authentication?* [Online]. Simplicable, 2016-12-21 [cit. 2021-04-15]. Dostupné z: https://simplicable.com/new/data-authentication.

33. NOTES, Electronics. *What is GPRS? - General Packet Radio Service Tutorial* [online]. Electronics Notes [cit. 2021-04-15]. Dostupné z: https://www.electronics-notes.com/articles/connectivity/2g-gprs/what-is-gprs-tutorial.php.

34. NDUNGU, Solomon; MIXON, Erica. *What is GSM (Global System for Mobile communication)?* [Online]. TechTarget, 2021-03 [cit. 2021-04-15]. Dostupné z: https://searchmobilecomputing.techtarget.com/definition/GSM.

35. B.V., NXP. *2C-bus specification and user manual* [online]. NXP B.V., 2007-06-19 [cit. 2021-04-15]. Dostupné z: https://web.archive.org/web/20120207003629/http://www.nxp.com/documents/user_manual/UM10204.pdf.

36. *About the Technology* [online]. NFC Forum, 2019-10-23 [cit. 2020-03-19]. Dostupné z: https://nfc-forum.org/what-is-nfc/about-the-technology/.

37. KOISHIGAWA, Kris. *What is NFC? Near Field Communication Uses, Chips, Tags, and Readers Explained* [online]. freeCodeCamp.org, 2020-11-03 [cit. 2021-04-15]. Dostupné z: https://www.freecodecamp.org/news/what-is-nfc-near-field-communication-uses-chips-tags-and-readers-explained/.

38. SEMICONDUCTORS, NXP. *PN532/C1* [online]. 2017-11-27 [cit. 2020-03-19]. Dostupné z: https://www.nxp.com/design/documentation:DOCUMENTATION%5C#/collection=documents&start=0&max=12&language=en&keyword=pn532&depth=1.

39. LOSHIN, Peter; COBB, Michael. *What is Encryption and How Does it Work?* [Online]. TechTarget, 2019-10-16 [cit. 2021-04-15]. Dostupné z: https://searchsecurity.techtarget.com/definition/encryption.

40. MARELI, M.; RIMER, S.; PAUL, B.S.; OUAHADA, K.; PITSILLIDES, A. Experimental evaluation of NFC reliability between an RFID tag and a smartphone. In: *2013 Africon*. 2013, s. 1–5. Dostupné z DOI: 10.1109/AFRCON.2013.6757740.

41. *Transceiver* [online]. Merriam-Webster, 2021-04-03 [cit. 2020-04-15]. Dostupné z: https://www.merriam-webster.com/dictionary/transceiver.

42. NANDA, Umakanta; PATTNAIK, Sushant Kumar. Universal Asynchronous Receiver and Transmitter (UART). In: *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)* [online]. Coimbatore, India: IEEE, 2016, sv. 01, s. 1–5 [cit. 2020-04-23]. Dostupné z DOI: 10.1109/ICACCS.2016.7586376.

Příloha **A**

Seznam použitých pojmů a zkratek

AEAD Authenticated encryption with asociated data. Mód blokových šifer, který zajišťuje důvěrnost dat, stejně tak jejich autentifikaci [27].

AES Advanced encryption standard druh blokové šifry.

Arduino Jedná se o open source platformu postavenou tak, aby její užívání a programování bylo jednoduché a zvládl ho i někdo, kdo není expertem z oboru. Tato platforma se objevila v roce 2005 a byla zamýšlena jako platforma rychlého prototypování pro studenty. Od té doby se platforma rozšířila a nachází uplatnění i mezi profesionály [28] Arduino je založené na jednočipových počítačích (mikrokontrolérech) od firmy Atmel. Jedná se o 8 bitový mikročip [29].

AT commands Znakové příkazy pro ovládání modemu, standardizované, ale dnes rozšířené o nestandardní příkazy [30].

ATS Odpověď na výběr podle specifikace ISO/IEC 14443-4 [31].

ATQA Odpověď na požadavek REQA podle normy ISO/IEC 14443-4 [31].

Autentizace dat Ověření, že přijatá data nebyla během přenosu změněna. Ať se již může jednat o poškození při přenosu, nebo záměrnou změnu [32].

GPRS Služba, která umožňuje přenos dat po mobilní síti. Rozšíření GSM s maximální teoretickou rychlostí okolo 172kbit/s [33].

GSM Standard celosvětové digitální mobilní sítě [34].

CHACHA20 Druh proudové šifry.

I2C Inter-Integrated Circuit, někdy také označováno jako TWI (Two Wire Interface), protože I2C je chráněné označení. Jedná se o sdílenou sériovou sběrnici , realizovanou dvěma vodiči. Umožňuje propojit až 128 zařízení, kdy každé zařízení má svou adresu, která je daná z výroby [35].

IoT Internet of things.

Kombinace SAK+ATQA+ATS Jednoznačně může určit výrobce a typ výrobku, podle dokumentace se však na toto nedá spolehnout a nedoporučuje se využívat to k identifikaci, jelikož odpovědi může uživatel změnit [31].

NFC Standard bezdrátové komunikace, fungující na frekvenci 13,56 MHz celosvětově. Maximální dosah NFC je přibližně 10 cm. NFC standart umožňuje napájení koncového zařízení z elektromagnetického pole, vytvořeného čtecím zařízením, takže není potřeba zdroj el. energie v jednom ze zařízení. Standart NFC je definován v normě ISO/IEC 18092 a postaven na základě starších existujících norem. Maximální přenosová rychlost je 424 Kbit/s [36], i když se většinou používá rychlost nižší. Technologie svým charakterem není stavěná na přenos objemých dat, ačkoliv technologicky to možné je.

NFC čtečka Zařízení schopné komunikace s tagem [37].

PN532 Transceiver, modul pro bezdrátovou komunikaci, který je v současné době velmi používaný pro svou příznivou cenu. Zvládá velkou část NFC strandardu, umožňuje komunikovat přes I2C sběrnici, UART nebo SPI, což je vhodné zejména pro mikrokontroléry typu Arduino. Modul operuje na frekvenci 13,56 MHz. Tato část obsahuje jen komunikační logiku [38].

POLY1305 MAC funkce pro ověření integrity dat.

SAK Potvrzení Výběru [31].

Shield Periferie pro snadné připojení k mikrokonktroléru Arduino [29].

Šifrování Proces utajení dat, jehož výsledkem je šifrovaný text, opakem je proces Dešifrování [39].

Tag Zařízení bez vlastního zdroje energie. Jedná se o pasivní zařízení, které začne být napájeno až v době, kdy se přiblíží do elektromagnetického pole čtecího zařízení, ze kterého pak čerpá energii pro svůj chod [40]. Tímto procesem se aktivuje zařízení a začne pracovat. V současné době je toto řešení velmi využívané jako přístupové, identifikační karty. Většinou obsahuje malou paměť.

Transceiver Prvek kombinující přijímač i vysílač. Tento prvek sám obsahuje pouze komunikační logiku [41].

UART Universal asynchronous receiver-transmitter slouží jako sériový port, jedná se o konfigurovatelnou počítačovou sběrnici s nastavitelnou rychlostí přenosu, slouží pro sériový asynchronní přenos dat [42].

UID Unique identificator, jednoznačné, unikátní číslo karty, které by mělo být dáno výrobcem a běžně neměnné. Avšak na neměnnost a neklonovatelnost tohoto identifikačního čísla nelze spoléhat, jelikož existují klony ostatních výrobců, které umožňují UID libovolně změnit [31].

Příloha **B**

Obsah přiloženého CD

- `CD` — Kořenová složka
  - `readme.txt` — Readme obsahuje strom všech souborů a krátký popisek
  - `bin` — Obsahuje zkompilované binární soubory
    - `Arduino_NFC.hex`
    - `server`
  - `src` — Obsahuje složky se zdrojovými kódy
    - `impl`
      - `libs` — Obsahuje zdrojové kódy vlastních tříd rozděleně
        - ...
      - `reader` — Obsahuje zdrojové kódy čtečky
        - ...
      - `server` — Obsahuje zdrojové kódy serveru
        - ...
    - `thesis` — Obsahuje zdrojové soubory bakalářské práce pro LaTeX
      - ...
  - `text` — Složka s textem bakalářské práce
    - `BP_Martin_Simunek_2021.pdf` — Text bakalářské práce