Návrh a implementace e-mailového klienta

Dominik Vodička

Bakalářská práce
2022

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Dominik Vodička**
Osobní číslo:	**A19124**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Softwarové inženýrství**
Forma studia:	**Prezenční**
Téma práce:	**Návrh a implementace e-mailového klienta**
Téma práce anglicky:	**Design and Implementation of an E-mail Client**

Zásady pro vypracování

1. Srovnejte dostupná řešení e-mailových klientů.
2. Popište způsoby legitimizace odesílatele.
3. Popište možnosti k odhalení nežádoucích zpráv.
4. Popište protokoly pro komunikaci s poštovními servery.
5. Navrhněte a implementujte vlastní řešení e-mailového klienta.
6. Demonstrujte komunikaci klienta s poštovním serverem.
7. Vyhodnoťte navržené řešení.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. LOSHIN, Pete. Essential email standards: RFCs and protocols made practical. New York: John Wiley, 1999. ISBN 978-0471345978.
2. HUGHES, Lawrence. Internet E-mail: protocols, standards, and implementation. Boston: Artech House, 1998. ISBN 978-0890069394.
3. Simple Mail Transfer Protocol[online]. IETF[cit. 2021-11-22]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc5321.
4. Internet Message Access Protocol (IMAP) – Version 4rev2[online]. IETF[cit. 2021-11-22]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc9051.
5. Post Office Protocol – Version 3[online]. IETF[cit. 2021-11-22]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1939.
6. DomainKeys Identified Mail (DKIM) Signatures[online]. IETF[cit. 2021-11-22]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc6376.

Vedoucí bakalářské práce: **Ing. Stanislav Kovář, PhD.**
Ústav bezpečnostního inženýrství

Datum zadání bakalářské práce: **3. prosince 2021**
Termín odevzdání bakalářské práce: **23. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 24. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 23.5.2022

Dominik Vodička v. r.
podpis studenta

ABSTRAKT

Cílem bakalářské práce je vytvoření e-mailového klienta, jenž je schopen na straně klienta rozpoznat nevyžádanou zprávu, a tím odstranit nedostatky poštovních systémů bez filtrace nevyžádaných zpráv. Poštovní klient bude informovat uživatele o důvěryhodnosti přijaté e-mailové zprávy a bude mít vlastní implementaci komunikace s poštovními servery dle standardů. Dále bude umožněno čtení e-mailových zpráv bez internetového připojení, pomocí integrované vnitřní databáze pro ukládání e-mailových zpráv. Práce též podrobně popisuje základní poštovní protokoly, a srovnává významně využívané poštovní klienty na trhu. Nakonec budou popsány metodiky legitimizace a odhalení nežádoucích zpráv.

Klíčová slova:
IMAP, SMTP, DKIM, SPF, RDNS, PTR, PGP, digitální podpis

ABSTRACT

The bachelor thesis aims to create an e-mail client that can detect unsolicited messages on the client-side and thus eliminate the shortcomings of mail systems without filtering unsolicited messages. The mail client will inform the user about the trustworthiness of the received e-mail message and will have its implementation of communication with mail servers with respect to the standards. Furthermore, it will be possible to read e-mail messages without an Internet connection, using an integrated internal database for storing e-mail messages. The thesis will also take a detailed look at the basic mail protocols and compare the prominently used mail clients on the market. Finally, methodologies for legitimizing and detecting unwanted messages will be described.

Keywords:
IMAP, SMTP, DKIM, SPF, RDNS, PTR, PGP, digital signature

Chtěl bych velmi poděkovat svému vedoucímu práce, panu doktoru Stanislavu Kováři za vedení mé bakalářské práce, veškeré rady a všechen čas, po který se mi věnoval.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD

V dnešní internetové době, přesycené informacemi, je velmi důležité být schopen interpretovat předmětné přijaté e-mailové zprávy, od nežádoucích. Ovšem tyto zprávy nelze vždy ověřit lidskou intuicí, a to především z důvodu, že se na špatně nakonfigurovaných poštovních systémech se může podvodník vydávat za libovolnou adresu odesílatele, a navodit tak dojem, že je přijatá zpráva předmětná, tedy důvěryhodná. Proto se autor práce, ze zmíněných důvodů, rozhodnul vytvořit aplikaci řešící dotyčné nedostatky některých poštovních systémů. Součástí práce je rovněž představit způsoby legitimizace, které mají v dnešní době značný smysl.

V teoretické části bude vypracována rešerše, zastřešující významně používané e-mailové klienty, kteří budou srovnáni z hlediska funkčností. Následovat bude podrobný popis základních komunikačních protokolů pro komunikaci s poštovními systémy. Nakonec budou vypracovány metody pro legitimizaci odesílaných zpráv a postupy k odhalení nežádoucích zpráv.

Praktická část bude zaměřena na implementaci e-mailového klienta, snažící se vyplnit nedostatky špatně nakonfigurovaných poštovních systémů, který bude naimplementován bez využití externích knihoven pro komunikaci s poštovními servery. Klient též bude podporovat čtení e-mailových zpráv bez internetového připojení, díky lokální databázi, ve které budou uložený všechny stažené zprávy. Výsledkem práce budou principy a zásady pro bezpečný a důvěryhodný přenos e-mailových zpráv.

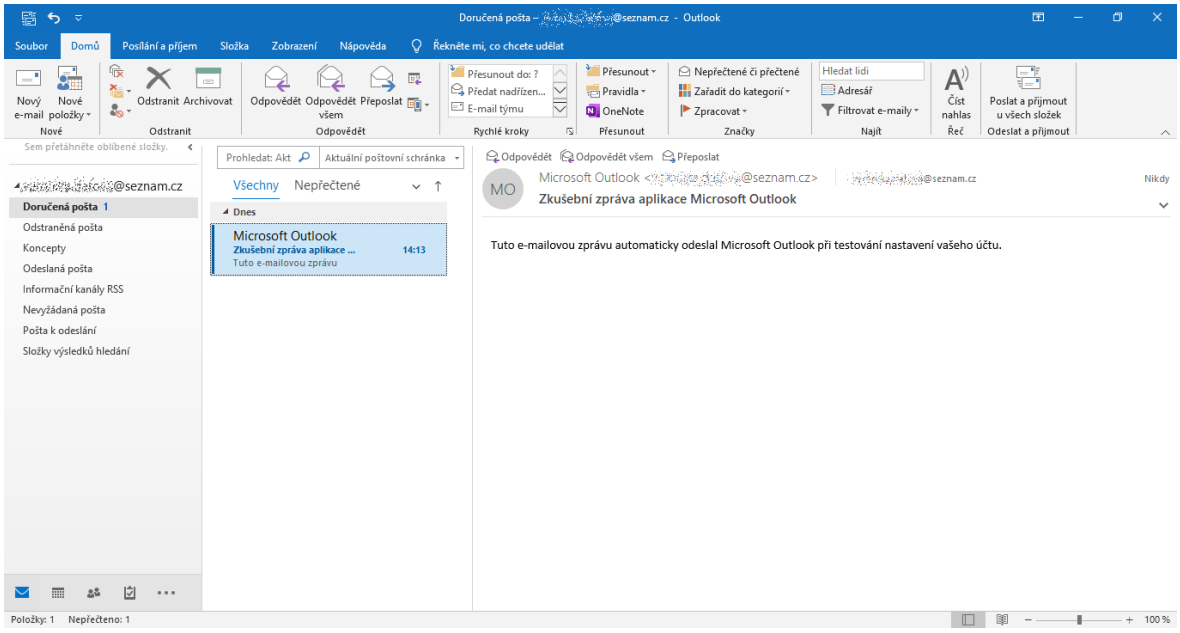
I. TEORETICKÁ ČÁST

1 SROVNÁNÍ DOSTUPNÝCH E-MAILOVÝCH KLIENTŮ NA TRHU

Tato kapitola nabízí srovnání vybraných e-mailových klientů pro operační systém Microsoft Windows, zastupující významnou část trhu e-mailových klientů. Srovnání bude zaměřeno především na bezpečnost.

1.1 Microsoft Outlook

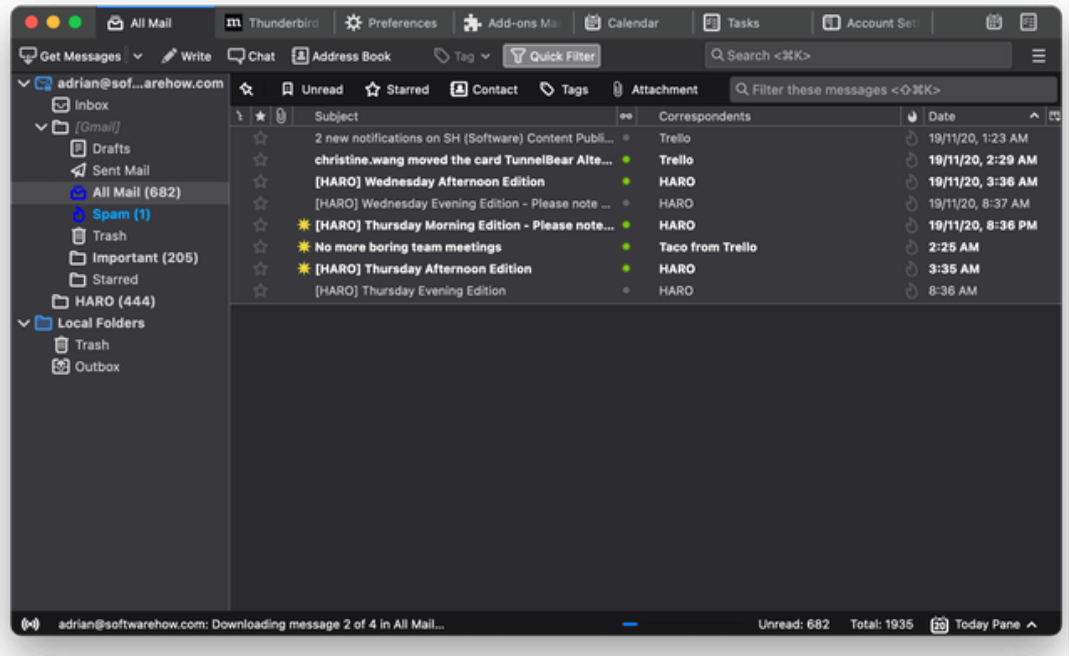
Microsoft Outlook je e-mailový klient vyvíjen společností Microsoft Corporation. Program je součástí balíku aplikací Microsoft Office, který funguje na bázi předplatného. Avšak v určitých intervalech, Microsoft sadu aplikací vydává jako samostatnou verzi, na bázi jednotné licence, která je především vhodná pro uživatele bez internetového připojení. Outlook podporuje všechny tři základní protokoly pro komunikaci s poštovním severem. Může se chlubit plně vestavěným kalendářem, či přímou integrací digitálních podpisů. Klient si bohužel s šifrovanými PGP zprávami sám neporadí, naštěstí ovšem existuje komunitní doplněk, který tento nedostatek dokáže odstranit. V poslední řadě zbývá podotknout, že aplikace bohužel nenabízí bezplatnou verzi. [1-9]



Obrázek 1. Aplikace Microsoft Outlook 2019. [1]

1.2 Mozilla Thunderbird

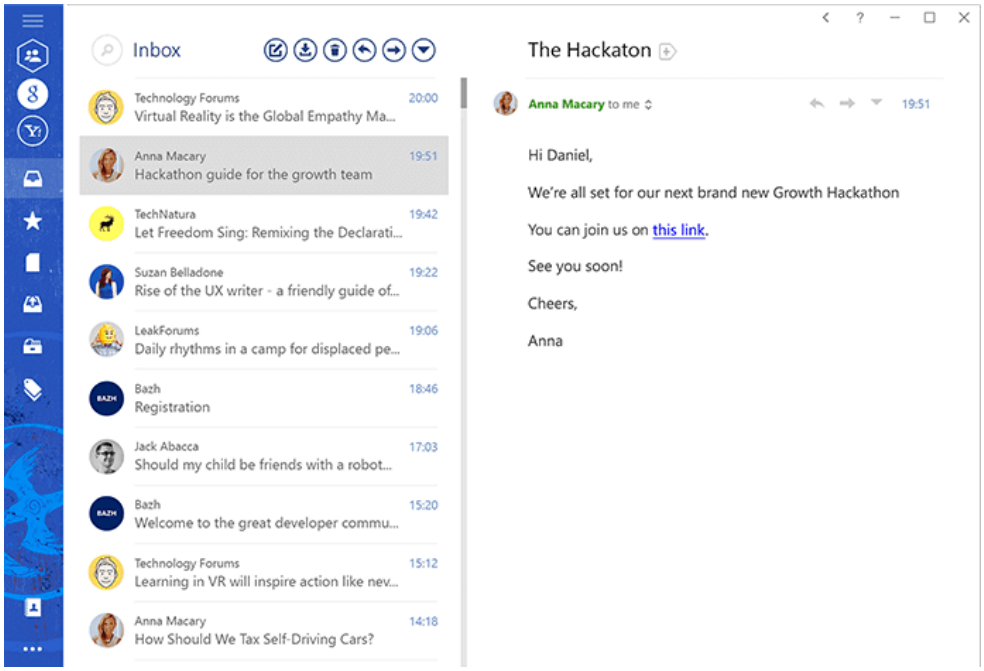
Poštovní klient Mozilla Thunderbird je vyvíjen komunitou, vývojáři otevřeného softwaru, pod záštitou organizace Mozilla. Klient nabízí širokou škálu přídavných doplňků, vytvořených komunitou. Thunderbird se vypořádá se základními e-mailovými protokoly, zvládne i napojení na vícero poštovních schránek současně. Klient plně podporuje PGP, umožní tak bezpečný přenos zpráv. Nejpodstatnější protokoly pro kontrolu proti nežádoucím zprávám klient bohužel ve výchozím stavu nepodporuje, nicméně díky komunitním doplňkům, je možné jejich přidání. Aplikace je nabízena bezplatně ke stažení. [11-20]



Obrázek 2. Aplikace Mozilla Thunderbird. [12]

1.3 Mailbird

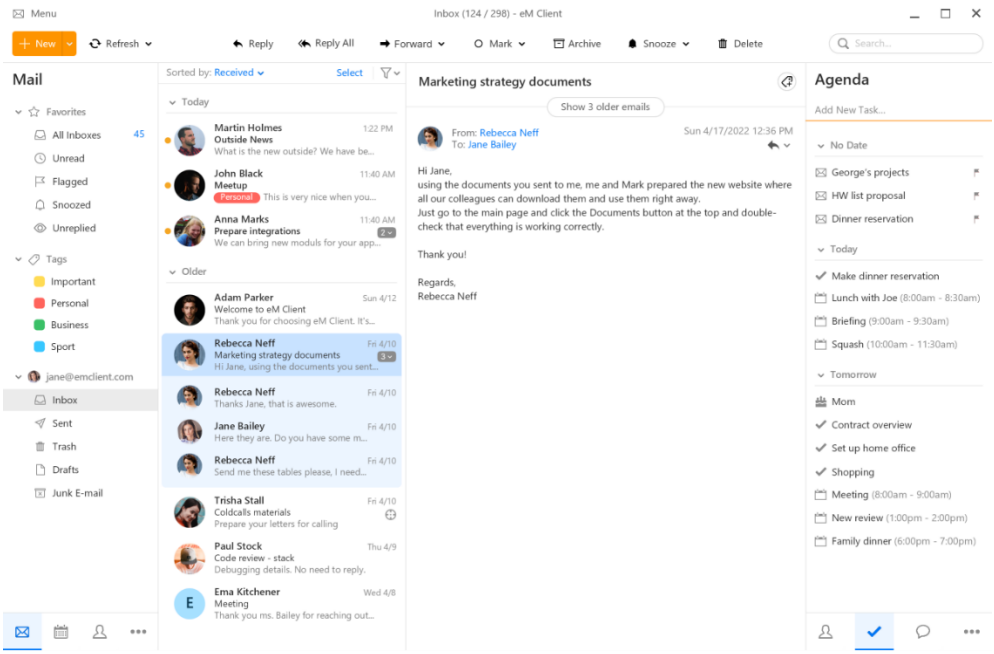
Mailbird je poštovní klient vyvíjen stejnojmennou společností. Aplikace je dostupná na základě měsíčního předplatného. Klient podporuje veškeré základní protokoly pro komunikaci. Aplikaci nechybí nativní kalendář či možnost využívat více poštovních schránek v jednu dobu. Dále se může pyšnit integrací komunikátoru mnoha externích služeb pro přenos zpráv. [22-26]



Obrázek 3. Aplikace Mailbird. [18]

1.4 eM Client

E-mailový klient je vyvíjen stejně se jmenující firmou. Aplikace využívá modelu „freemium“, což je obchodní model, ve kterém je uživateli nabízena základní verze produktu zdarma. Program má integrovaný nativní kalendář, a zvládne vykomunikovat všechny základní poštovní protokoly. Podporuje taktéž více e-mailových schránek, digitální podpis a dokonce i PGP šifrování. [28-33]



Obrázek 4. Aplikace eM Client. [25]

1.5 Shrnutí

Všichni porovnávaní klienti dosahovali velmi dobrých výsledků z hlediska podpory více stránek současně a implementaci nativního kalendáře. Ovšem u podpory metod pro legitimizaci odesílatele a rozpoznání nežádoucích zpráv byla situace o poznání horší. Ze vzájemného srovnání vyplývá, že nejlepších výsledků dosahuje poštovní klient Mozilla Thunderbird. Tato skutečnost je dána zejména kvůli komunitním doplňkům, které vykompenzovaly nedostatky aplikace.

Tabulka 1. Shrnutí srovnání e-mailových klientů.

Software	Podpora základních protokolů	Podpora více schránek	Kalendář	Digitální podpis	PGP	SPF	DKIM	Bezplatná verze
Microsoft Outlook	Ano [26]	Ano [27]	Ano [28]	Ano [29]	Ano* [30]	-	-	Ne [31]
Mozilla Thunderbird	Ano [8]	Ano [9]	Ano [4]	Ano [10]	Ano [11]	Ano* [5]	Ano* [6]	Ano [9]
MailBird	Ano [16]	Ano [17]	Ano [15]	-	-	-	-	Ne [13]
eM Client	Ano [21]	Ano [32]	Ano [22]	Ano [23]	Ano [23]	-	-	Ano [24]

*Pouze s doplňkem

2 PROTOKOLY PRO KOMUNIKACI S POŠTOVNÍMI SERVERY

2.1 Internetový protokol

Internetový protokol je sada pravidel, které popisují komunikaci mezi síťovými subjekty přes síť Internet nebo jiné TCP/IP sítě. Protokol se stane standardem sítě Internet až v době, kdy bude zapsán v dokumentu STD-1. Dokument STD-1 bývá publikován přibližně každým 100. RFC standardem sítě Internet. [33]

2.2 RFC

Dokumenty RFC (Request For Comments) reprezentují nejdůležitější standardy sítě Internet. RFC je zpráva, ve které výzkumníci sdílejí své výsledky, teorie a aktivity pro zpětnou vazbu od ostatních výzkumníků, kteří rozumí dané problematice. Zveřejněním RFC dokumentu, může být daný dokument zvážen jako standard. Po zveřejnění RFC dokument už nelze upravit či zaktualizovat. Zapojit se do psaní RFC dokumentů může kdokoliv, kdo se vyzná v problematice počítačových sítí, od studentů přes zaměstnance firem se zaměřením do počítačových sítí. [33]

2.3 E-mailové protokoly

Pro komunikaci v prostředí sítě Internet bylo zapotřebí jednotné implementace komunikačních pravidel pro předávání zpráv, proto aby si mezi sebou různí klienti a servery rozuměli. Díky této potřebě, postupem času přišly na svět standardizované protokoly organizací IETF. [33]

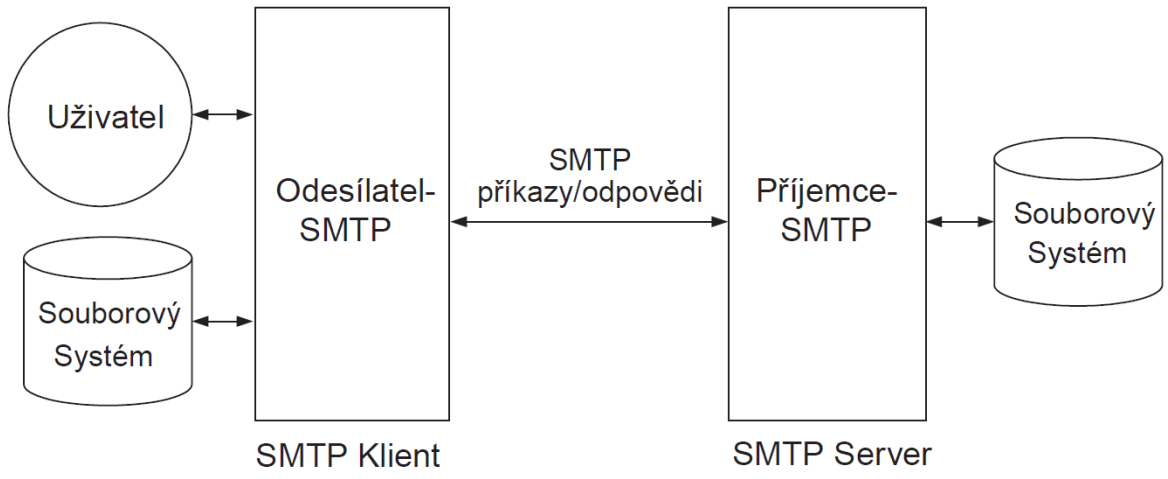
2.3.1 SMTP

SMTP je protokol, definován dle dokumentu RFC821, jehož cílem je spolehlivě a efektivně přenést e-mailovou zprávu přes libovolné přenosové médium k SMTP serveru. Poštovní protokol je postavený nad síťovým protokolem TCP a využívá výchozí port 25. Důležitý aspekt SMTP protokolu je schopnost přenést zprávu v jakémkoliv síťovém prostředí, izolovaném intranetu, i v síti do Internetu. [33], [34]

2.3.1.1 SMTP model

SMTP model v Obrázku 5 vyobrazuje, že se SMTP protokol zajímá pouze o způsob přepravy e-mailové zprávy, od klienta k serveru. SMTP formuluje principy inicializace spojení, přenosu zpráv a uzavírání spojení, co nastane s přijatou zprávou na straně klienta, není záležitostí SMTP protokolu, ale e-mailového klienta nebo jiného mechanismu, který je schopen vkládat zprávu do přenosového prostředí. SMTP systémy se dle standardu rozdělují na systémy odesílací a přijímací. Přijímací SMTP systém přijímá zprávy, odesílací SMTP systém zprávy odesílá. Od vydání dokumentu RFC821bis [33] byly tyto termíny nahrazeny klientem SMTP a serverem SMTP. Kde SMTP klient nahrazuje odesílající SMTP systém a SMTP server nahradil přijímací SMTP systém. Často se však stává, že SMTP systém přebírá obě role pro stejnou zprávu. [33]

Komunikace dle SMTP modelu v obrázku níže probíhá následovně: Uživatel zadává požadavek klientovi na odeslání e-mailové zprávy. Klient naváže spojení se vzdáleným serverem, vytvoří tak obousměrný přenosový kanál, pomocí něhož klient přenese e-mailovou zprávu na přijímající server. Přijímacím systémem může být konečný příjemce zprávy, nebo se může jednat o pouhého prostředníka. [35]



Obrázek 5. SMTP model. [33]

2.3.1.2 Procedury

Procedury jsou skupina stavových bloků, které mají na práci funkčnost SMTP serveru. [36]

- **E-mailové transakce**

Procedura obsahuje tři transakční kroky. Transakce započne s příkazem „MAIL“, ve kterém se odesílající identifikuje. Poté následuje jeden nebo více příkazů udávající informace o příjemcích zprávy. Dále následuje příkaz „DATA“, po jehož zadání bude SMTP server vyčkávat na SMTP klienta, než odešle veškerá data e-mailové zprávy. Nakonec následuje zaslání indikátoru konce dat e-mailové zprávy, po jehož zaslání se transakce potvrdí a tím se e-mailová zpráva odešle. [36]

První krok „MAIL“ procedury, dá SMTP serveru vědět, že začíná nová transakce. S tímto krokem si dále server vynuluje veškeré stavové tabulky, zásobníky, včetně příjemců a datové části zprávy. Dále se zkontroluje, zdali je adresa příjemce přijatelná SMTP serverem. V případě, že bude všechno v pořádku, server vrátí kód „250 OK“ [36]

„MAIL <SP> FROM:<reverse-patch> <CRLF>“ [36]

Druhý krok „MAIL“ procedury vyžaduje zadání adres příjemců zprávy. Tento krok může být neomezeně opakován, pro zadání vícero příjemců. [35]

„RCPT <SP> TO:<forward-path> <CRLF>“ [36]

Třetí krok „MAIL“ procedury se zaktivuje po zadání příkazu níže. Příkaz zahájí zápis dat do e-mailové zprávy. Tento krok také umožňuje zápis položek hlavičky: časové razítko odeslání zprávy, předmět zprávy, příjemce, příjemce kopie, odesílatel. Nakonec se tímto krokem ukončí transakce po zaslání indikátoru konce dat e-mailové zprávy. [36]

„DATA <CRLF>“ [36]

Následující příklad procedury MAIL zobrazuje e-mailovou zprávu zasílanou odesílatelem Smith na adrese „Alpha.ARPA“, příjemcům Jones, Green, a Brown na adrese „Beta.ARPA“. Ve vyobrazené transakci se předpokládá, že systém „Alpha“ zkontaktuje systém „Beta“ napřímo. [36]

Příklad SMTP procedury MAIL:

„S: MAIL FROM:<Smith@Alpha.ARPA>
R: 250 OK

S: RCPT TO:<Jones@Beta.ARPA>
R: 250 OK

S: RCPT TO:<Green@Beta.ARPA>
R: 550 No such user here

S: RCPT TO:<Brown@Beta.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: <CRLF>.<CRLF>“ [36]

- **Směrování**

Při doručování zpráv může nastat problém, kdy odesílající SMTP klient nezná trasu k příjemci, ale přijímací SMTP server tuto trasu zná. V takových případech se využije jedna ze stavových zpráv níže, pro informování odesílatele o špatné adrese příjemce. Odesílatel bude taktéž informován, kdo v této situaci převezme zodpovědnost za doručení zprávy. [36]

Odpověď níže označuje, že přijímací SMTP server zná umístění uživatelské schránky, která je v jiném systému. Proto vypíše odesílateli správnou adresu, kterou by měl využít v budoucnu. Přijímací SMTP server převezme zodpovědnost za doručení zprávy. [36]

„251 User not local; will forward to <forward-path>“ [36]

Tato odpověď níže indikuje znalost přijímacího SMTP serveru, o umístnění uživatelské schránky v jiném systému. Protože zná správnou adresu, kterou by měl použít, přijímací SMTP server odmítá přijmout e-mailovou zprávu pro tohoto uživatele. Odesílatel musí zprávu sám přesměrovat dle získaných informací. [36]

„551 User not local; please try <forward-path>“ [36]

- **Ověřování a rozšiřování**

Přídavné rozšíření SMTP protokolu umožňuje zkontrolovat existenci e-mailové schránky na daném SMTP serveru, či dopodrobna rozepsat e-mailovou skupinu adres, pomocí příkazů „*VRFY“* a „*EXPN“*. Příkaz *„ VRFY“* přijímá pouze přezdívku e-mailové schránky. Po zadání příkazu SMTP server zkontroluje, zdali taková schránka existuje a odpoví SMTP klientovi. V případě, že schránka existuje, SMTP server může odpovědět i s celým jménem adresáta. [36]

Příklad procedury „VRFY“:

„S: VRFY Smith

R: 250 Fred Smith <Smith@USC-ISIF.ARPA>“ [36]

Příkaz „EXPN“ přijímá přesný řetězec e-mailové skupiny. V případě existence zadané e-mailové skupiny, budou SMTP klientovi vypsány veškeré adresáty. SMTP server může odpovědět celými jmény adresátů. [35]

Příklad procedury EXPN:

*„**S**: EXPN Example-People*

***R**: 250-Jon Postel <Postel@USC-ISIF.ARPA>*

***R**: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>*

***R**: 250-Sam Q. Smith <SQSmith@USC-ISIQ.ARPA>*

***R**: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>*

***R**: 250-<joe@foo-unix.ARPA>*

***R**: 250 <xyz@bar-unix.ARPA>"* [36]

- **Otevírání spojení**

V době, kdy se přenosový kanál otevře, server přivítá klienta a vypíše mu která konkrétní rozšíření protokolu SMTP server podporuje. Dále proběhne výměna informací mezi SMTP serverem a SMTP klientem tak, aby se server ujistil, že klient komunikuje se správným systémem. Příkazem níže se odesílající SMTP klient identifikuje svojí plně kvalifikovanou doménou (FQDN). Pokud odesílatel nemá plně kvalifikovanou doménu, může využít jiný libovolný identifikátor. [33], [35]

Příklad identifikace odesílatele:

„HELO <SP> <domain> <CRLF>“ [36]

Příkaz „HELO“ byl později nahrazen novější SMTP implementací, příkazem „EHLO“, který nově odesílateli vypíše, spolu se stavovým kódem, jaké SMTP rozšíření podporuje. [33]

- **Uzavírání spojení**

Jakmile připojený SMTP klient předá SMTP serveru veškeré potřebné informace, obousměrný kanál se uzavře příkazem „QUIT“. SMTP server musí tento příkaz potvrdit stavovým kódem. [33]

Příklad uzavření spojení:

„S: QUIT

R: 221 BBN-UNIX.ARPA Service closing transmission channel“ [36]

2.3.1.3 Role SMTP serveru

V SMTP standardu existují 4 role, které může SMTP server plnit. [33]

- **SMTP systém původce (originator system)**

SMTP server přijímá zprávy pro doručení od poštovních uživatelských agentů (MUA), které dále přeposílá do sítě Internet. V tomto případě se SMTP server stává původcem původní zprávy v síti Internet. [33]

- **Doručovací systém (delivery system)**

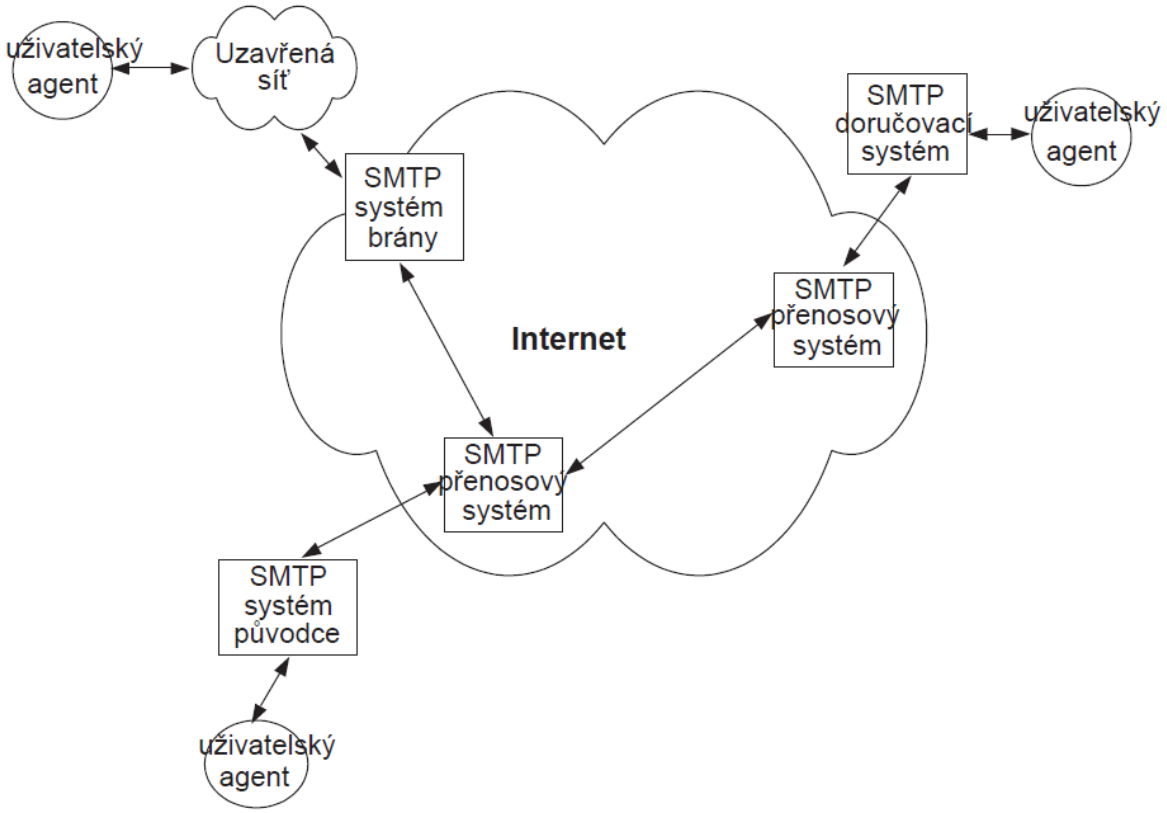
Systém přijímá e-mailové zprávy od systému přenosu zpráv, které pak předá poštovním agentům (MUA) nebo zprávy uloží v souborovém systému serveru. [33]

- **Přenosový systém (relay system)**

Přijímá zprávy od SMTP klientů, které pak předává dalšímu SMTP serveru. Přenosový systém upravuje původní zprávu, pouze zanecháním svého otisku v trasové části e-mailové zprávy. [33]

- **Systém brány (gateway system)**

Tento systém přijímá zprávu od SMTP klienta v jedné sítí a v síti druhé danou zprávu přeposílá. [33]



Obrázek 6. Role systému původce, doručovacího systému, přenosového systému a systému brány SMTP. [33]

2.3.2 IMAP

IMAP je spolehlivý protokol, který pracuje nad síťovou vrstvou TCP. Tento protokol slouží pro čtení přijatých e-mailových zpráv a využívá port 143 pro komunikaci ve formě nešifrovaného prostého textu (cleartext), nebo port 993 pro komunikaci s implicitní přenosovou vrstvou zabezpečení (TLS). [37]

2.3.2.1 Atributy zpráv

- **Číslování zpráv**

  K emailovým zprávám se přistupuje pomocí unikátních identifikátorů zpráv nebo za pomoci sekvenčního čísla zpráv. [37]

- **Příznaky zpráv**

  Každá e-mailová zpráva může mít přirazeno více příznaků. Příznaky se rozdělují na systémové příznaky a klíčové příznaky. Příznak jakéhokoliv typu může být trvalý či existující pouze v průběhu relace

Aktuálně definované systémové příznaky standardem:

- **\Seen**
  Zpráva byla přečtena. [37]
- **\Answered**
  Na zprávu bylo zodpovězeno. [37]
- **\Flagged**
  Zpráva je označena pro speciální pozornost. [33]
- **\Deleted**
  Zpráva byla označena pro odstranění. [33]
- **\Draft**
  Označuje zprávu, která ještě nebyla dopsána. [33]
- **\Recent**
  Zpráva byla přijata nedávno. Tento příznak se využije pouze v první relaci, po jejíž trvání byla zpráva přijata. [33]

- **Interní časové razítko**

Časové razítko obsahuje datum a čas přijetí zprávy serverem. Pokud zpráva byla doručena SMTP protokolem, jedná se o konečný datum a čas přijetí zprávy. [33], [37]

- **Velikost zpráv**

Atribut obsahuje velikost zprávy v bajtech. [33]

2.3.2.2 Stavy

Spojení se může nacházet v jednom ze čtyřech stavů. V počátečním stavu relace, server předá klientovi prvotní zprávu s pozdravem. Příkazy IMAP protokolu jsou omezovány dle toho, v jakém stavu se klient zrovna nachází. [37]

- **Neautentizován**

Klient se do tohoto stavu dostane po navázání spojení k serveru, pokud nevyužil předběžnou autentizaci. V tomto stavu nelze využívat většinu IMAP příkazů. [37]

- **Autentizován**

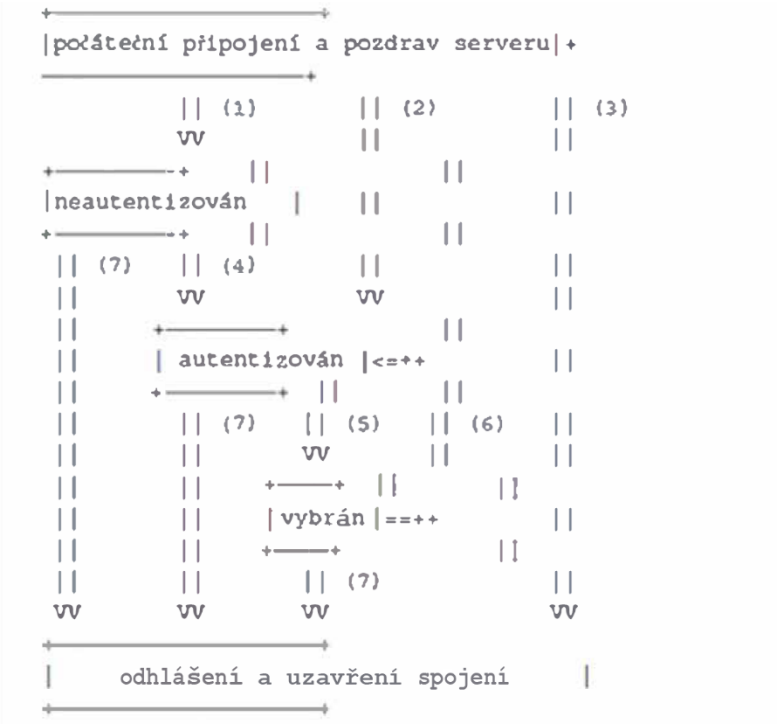
Mezi klientem a serverem došlo k autentizaci nebo předběžná autentizaci. Většinu příkazů pro práci s e-mailovými zprávami může klient využívat, až zvolí poštovní schránku, se kterou chce klient manipulovat. [37]

- **Vybrán**

Tento stav se zaktivuje, jakmile klient zvolí poštovní schránku pro manipulaci. [37]

- **Odhlášení a uzavření spojení**

Stavu odhlášení, klient dosáhne pomocí příkazu „LOGOUT“, či přes jednostranné jednání ze strany klienta nebo serveru.



Obrázek 7. Stavový diagram protokolu IMAP. [33]

2.3.2.3 Atributy hlavičky e-mailové zprávy

Hlavičkové atributy jsou řádky postavené z dvojic, jména atributy a hodnoty atributy, které jsou odděleny pomocí dvojtečky. Jméno atributu je podmíněno tisknutelnými znaky kódovací sady „US-ASCII“, přičemž hodnota atributy může využívat všechny znaky ze zmíněného kódování. [38]

- **Date**
  Časové razítko toho, kdy byla zpráva připravena vstoupit do poštovního doručovacího systému. [39]

- **From**
  Popisuje adresu schránky původce zprávy. [39]

- **Sender**
  Popisuje poštovní schránku původce, který odvysílal danou zprávu. [39]

- **Reply-To**
  Atributa popisující na jakou adresu by měl příjemce zprávy odpovídat. [39]

- **To**
  Obsahuje adresy hlavních příjemců e-mailové zprávy. [39]

- **CC**
  Zahrnuje adresy ostatních příjemců zprávy, na které obsah zprávy nemusí cílit. [39]

- **Bcc**
  Adresy příjemců zprávy, jejíchž adresa není prozrazena ostatním příjemcům dané zprávy. [39]

- **Message-ID**
  Unikátní identifikátor odkazující na určitou verzi dané zprávy. Pokud je zpráva zaslána znovu, beze změn, identifikátor se zachová. [39]

- **In-Reply-To**
  Identifikátor původní zprávy, na kterou je tato zpráva odpovědí. [39]

- **References**

  Identifikátory e-mailových zpráv, se kterými může být aktuální zpráva spjata. [39]

- **Subject**

  Obsahuje krátký popis předmětu aktuální zprávy. [39]

- **Comments**

  Součástí jsou případné komentáře k tělu zprávy. Někteří poštovní klienti hodnotu tohoto pole nezobrazují. [39]

- **Keywords**

  Zahrnuje klíčová slova aktuální zprávy, která mohou být pro příjemce důležitá. [39]

- **Resent-Date**

  Časové razítko obsahující datum a čas doby, kdy byla zpráva znovu přidána do přenosového poštovního systému. [39]

- **Resent-From**

  Adresa schránky, která aktuální zprávu přidala do přenosového poštovního systému. [39]

- **Resent-To**

  Zahrnuje adresy schránek, pro které by měla být aktuální zprávy přeposlána. [39]

- **Resent-CC**

  Adresy schránek, pro které by měla být zaslána kopie, při znovuzaslání zprávy. [39]

- **Resent-Bcc**

  Zahrnuje adresy příjemců zprávy, jejíchž adresa není prozrazena ostatním příjemcům dané zprávy, při znovuzaslání zprávy. [39]

- **Resent-Reply-To**

  Zahrnuje adresu, na kterou by měl příjemce znovu zaslané e-milové zprávy odpovídat. [39]

- **Resent-Message-ID**

  Obsahuje identifikátor zprávy, pro její znovuzaslání. [39]

- **Return-Path**

  Návratová adresa pro diagnostiku odezvy na zprávu. [39]

- **Received**

  Obsahuje informace o přijetí aktuální zprávy přenosových poštovních systémem (MTA). [39]

- **Disposition-Notification-To**

  Popisuje adresu schránky, na kterou odesílatel zprávy chce dostávat oznámení o událostech aktuální zprávy - její přečtení, zpracování, apod. [39]

- **Disposition-Notification-Options**

  Umožňuje nastavit volitelné modifikátory oznámení. [39]

- **Accept-Language**

  Prozrazuje, v jakém jazykem odesílatel vyžaduje odpověď od příjemce. Atribut bývá využíván při generování automatických odpovědí. [39]

- **Original-Message-ID**

  Identifikátor původní zprávy, využívaný při opětovném odeslání zprávy s alternativním formátem obsahu. [39]

- **PICS-Label**

  Atribut ohodnocení využívaný pro filtrování zpráv protokolem PICS. [39]

- **Encoding**

  Obsahuje kódování aktuální zprávy. [39]

- **Message-Context**

  Poskytuje informace o kontextu a charakteristiky aktuální zprávy. [39]

- **Alternate-Recipient**

  Nastavuje, zdali může být zpráva přesměrována na alternativní adresu příjemce, pokud schránka určitého příjemce nebude dostupná. [39]

- **Generate-Delivery-Report**

  Konfigurace, zdali hlášení o úspěšném doručení zprávy je žádáno. Ve výchozím stavu není hlášení generováno. [39]

- **Prevent-NonDelivery-Report**

  Určuje, zda odesílatel žádá o hlášení při neúspěšném doručení zprávy. Hlášení je generováno ve výchozím stavu. [39]

- **Delivery-Date**

  Časové razítko, kdy byla zpráva úspěšně doručena příjemci. [39]

- **Expires**

  Stanovuje čas, kdy e-mailová zpráva ztrácí platnost. [39]

- **Reply-By**

  Čas, do kterého odesílatel vyžádal odpověď. [39]

- **Importance**

  Nastavuje důležitost zprávy. Nabízí se stupně: vysoká, normální a nízká důležitost. [39]

- **Priority**

  Určuje prioritu aktuální zprávy. Existují následující priority: normální, naléhavá, nenaléhavá. [39]

- **Sensitivity**

  Určuje jak citlivé je sdělení zasílané zprávy jiným adresátům, než jsou zadaní příjemci. Hodnota může nabýt: osobní, soukromé, firemní tajemství. [39]

- **Deferred-Delivery**

  Poskytuje informace o případech odloženého doručení k příjemci. [39]

- **Latest-Delivery-Time**

  Popisuje příjemci informace o požadovaném doručení. [39]

2.3.2.4 MIME

MIME je standard, jehož vznik započal odesílání a přijímání e-mailových zpráv s prakticky libovolným obsahem. S tímto standardem přišly nové e-mailové atributy hlavičky, která specifikují, jakým způsobem by měl uživatelský agent (MUA) interpretovat přijatou zprávu. Atribut hlavičky „Content-Type“ specifikuje typ, případný podtyp dat v těle zprávy, a hlavičkový atribut „Content-Transfer-Encoding“ popisuje kódování dat. [40], [39]

Atributy standardu MIME:

- **MIME-Version**

  Atribut obsahuje verzi používaného MIME ve zprávě, Taktéž stanovuje, že je zpráva formátována dle standardu MIME. [39]

- **Content-ID**

  Unikátní identifikátor těla MIME zprávy. [39]

- **Content-Description**

  Popisuje část těla MIME zprávy. [39]

- **Content-Transfer-Encoding**

  Určuje využité kódování v části těla MIME zprávy. [39]

- **Content-Type**

  Znázorňuje formát dat v MIME zprávě. [39]

- **Content-Disposition**

  Značí, zdali část těla MIME zprávy by měla být zobrazena metodou „inline“ nebo se jedná o přílohu. V případě přílohy může atribut obsahovat název přílohy. [39]

- **Content-Language**

  Stanovuje jazyk použitý v MIME zprávě. [39]

- **Content-MD5**

  Otisk MIME zprávy, sloužící pro ujištění, že zpráva nebyla upravena. [39]

2.3.3 POP3

POP3 je jednoduchý protokol pro stahování a odstraňování e-mailových zpráv z POP3 serveru. Standard pracuje nad síťovým protokolem TCP, s výchozím portem 110. POP3 je významně jednodušší protokol než IMAP, protože server neumožňuje žádnou manipulaci s přijatými zprávy jako IMAP, a ve standardním případě e-mailovou zprávu smaže hned, jakmile ji uživatel stáhne z POP3 serveru. [33], [41]

2.3.3.1 Stavy

POP3 relace v průběhu spojení může nabýt dvou stavů. [33]

- **Autorizace**

  V tomto stavu POP3 server očekává přihlašovací údaje od klienta, pro jeho autorizaci. Klient se může přihlásit standardně, „PLAINTEXT“ metodou za pomocí přihlašovacího jména a hesla, či využít příkaz kryptografické funkce „APOP“, která uživatele autorizuje za pomocí hashe, tvořeného z přihlašovacích údajů, algoritmem MD5. [33]

  Příklad autorizace klienta „PLAINTEXT“ metodou:

  „C: USER pete
  S: +OK pete likes spinach
  C: PASS swordfish
  S: +OK pete's maildrop has 12 messages (21320 octets)“ [33]

- **Transakce**

  Do aktuálního stavu se klient dostane po své úspěšné autentizaci, ve kterém může získat informace o e-mailových schránkách, stáhnout zprávy či zprávy smazat. [33]

2.3.3.2 Příkazy

- **STAT**

  Příkaz zobrazující statistiku e-mailové schránky. Výpis předává informaci, kolik nových zpráv schránka přijala a kolik bajtů tyto všechny zprávy zabírají.

  Příklad příkazu:

  „C: STAT
  S: +OK 12 21320“ [33]

- **LIST**

Tento příkaz vypisuje informace o všech přijatých zprávách, čekajících na stažení. Výpis příkazu obsahuje počet přijatých zpráv s informací relačního čísla zprávy a velikosti každé jednotlivé zprávy.

Příklad příkazu:

„C: LIST
S: +OK 12 messages (21320 octets)
S: 1 1220
S: 2 2300
S: 3 240
S: 4 5580“ [33]

- **RETR**

Příkaz je využíván ke stažení zprávy z POP3 serveru. Do argumentu příkazu se zadává relační číslo zprávy. Příjem zprávy je ukončován tečkou. [33]

Příklad příkazu:

„C: RETR 1
S: +OK 120 octets
S: <the POP3 server sends message 1>
S: . „ [33]

- **DELE**

Příkaz se používá pro mazání zpráv z POP3 serveru. Při odstraňování zprávy, se zpráva identifikuje pomocí relačního čísla zprávy. [33]

Příklad příkazu:

„C: DELE 1
S: +OK message 1 deleted“ [33]

- **NOOP**

Tento příkaz slouží pro ujištění klienta, že POP3 server má stále navázané spojení. Po zadání příkazu, POP3 server odpovídá zprávou *„+OK“*. [33], [41]

- **RSET**

Zadáním příkazu se zruší značky na e-mailových zprávách, které byly označeny pro jejich odstranění. [33]

- **QUIT**

Příkaz, po jehož zadání se uzavře obousměrný přenosový kanál. V případě, že byla relace ve stavu transakce, pak se před uzavřením spojení uloží veškeré změny. [33]

3 ZPŮSOBY LEGITIMIZACE ODESÍLATELE

Aktuálně v doposud používaných poštovních protokolech neexistují žádná omezení v zadávání adresy původce zprávy (MAIL FROM), ani v identifikaci identity původce (HELO) pomocí plně kvalifikované domény (FQDN), proto je snadné zneužívat libovolnou adresu odesílatele. Nicméně postupem času se objevily metody, pro ujištění příjemce, že e-mailová zpráva doopravdy pochází ze zdroje, z kterého danou zprávu očekává. [42]

3.1 SPF (Sender Policy Framework)

Jedná se o validační systém, který se snaží předcházet e-mailovým zprávám od neautorizovaných odesílatelů, vydávající se pod určitou doménou. [42]

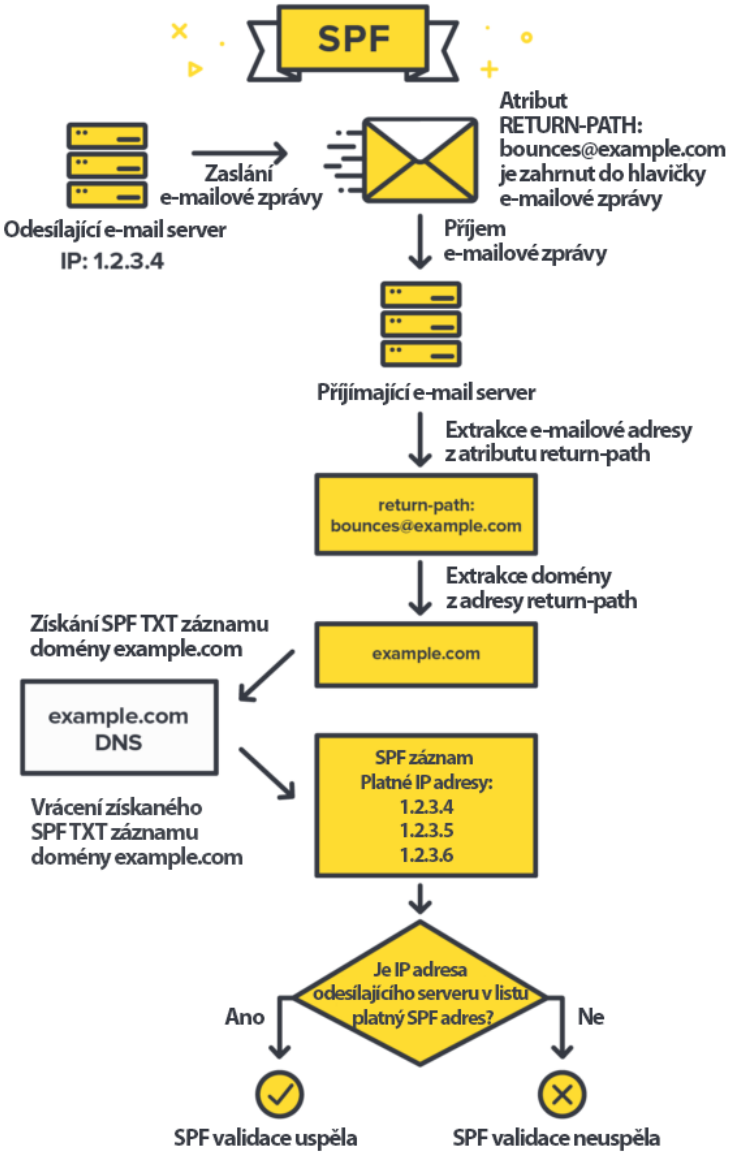
3.1.1 Publikace pravidel autorizace

Pravidla SPF doménového jména se nastavují na patřičném DNS serveru. SPF záznam se publikuje pod DNS záznam typu TXT. Příklad záznamu v DNS, který povoluje zasílání zpráv za doménu „example.com“ IP adresám 192.0.2.1 a 192.0.2.129: [42]

„example.com. IN TXT "v=spf1 ip4:192.0.2.1 ip4:192.0.2.129 -all"“ [42]

3.1.2 Ověřování autorizace

Ověření spočívá v kontrole autorizace zadané identity odesílatele (HELO) a také kontrole autorizace adresy odesílatele e-mailové zprávy (MAIL FROM) vůči IP adrese odesílatele. Ověřování autorizace odesílatele probíhá většinou při přijetí zprávy přenosovým agentem (MTA), avšak může být provedeno kdekoliv v cestě doručení. [42]



Obrázek 8. Diagram validace SPF. [43]

3.1.3 Výsledné stavy ověření

- **None**

  Žádný SPF záznam v DNS dané domény nebyl nalezen. [42]

- **Neutral**

  DNS server explicitně zakázal vyhodnocování autorizaci odesílatele. [42]

- **Pass**

  Odesílatel je autorizován využívat příslušnou doménu. [42]

- **Fail**
  Odesílatel není autorizovaný využívat příslušnou doménu. [42]

- **Softfail**
  Odesílatel pravděpodobně není autorizován. [42]

- **Temperror**
  Značí chybu při spojování s DNS serverem. [42]

- **Permerror**
  Záznam v DNS serveru nelze interpretovat. [42]

3.1.4 Zranitelnost

Jelikož SPF systém validace spoléhá na DNS, potencionální útočník by mohl napadnout DNS server, který využívá SMTP systém pro ověřování SPF a výslednou odpověď upravit tak, aby SPF validace byla platná. [42]

3.2 DKIM (DomainKeys Identified Mail)

Doménové klíče umožňují přenášet zodpovědnost zprávy za organizaci či člověka, jež je vlastníkem dané domény. Ověření zodpovědnosti zpráv se provádí skrze kryptografický podpis přijaté zprávy a veřejného klíče domény v systému DNS. [44]

3.2.1 Generování klíčů

Pro podepisování e-mailových zpráv je zapotřebí vygenerovat veřejný a soukromý klíč. Soukromý klíč se na e-mailovém serveru využije pro kryptografické podepisování odesílaných zpráv a veřejný klíč bude využit pro ověřování podpisu zprávy. [44], [45]

3.2.2 Publikace veřejného klíče

Veřejný klíč DKIM podpisu se ukládá do systému DNS vlastněné domény. DKIM standard požaduje, aby název záznamu v DNS měl předponu „_domainkey“, pokud přijatá zpráva s podpisovým parametrem „s“, nespecifikuje jinak. Příklad DKIM záznamu v systému DNS, kde parametr „p“ je veřejný klíč a parametr „k“ označuje šifrovací algoritmus: [44], [45]

„v=DKIM1;k=rsa;t=s;p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDDm zRmJRQxLEuyYiyMg4suA2SyMwR5MGHpP9diNT1hRiwUd/mZp1ro7kIDTKS8ttkI6z6 eTRW9e9dDOxzSxNuXmume60Cjbu08gOyhPG3GfWdg7QkdN6kR4V75MFlw624VY3 5DaXBvnlTJTgRg/EW72O1DiYVThkyCgpSYS8nmEQIDAQAB“ [45]

3.2.3 Kanonikalizace e-mailové zprávy

Při doručování zprávy, může nastat k nepatrným modifikacím hlavičky či obsahu e-mailové zprávy. V takových případech by se podepsaný e-mail stal neplatným. DKIM má naštěstí vůči těmto mírným modifikacím opatření, nazývající se kanonikalizace. Kanonikalizace dokáže upravit zprávu tak, aby byla i při mírné úpravě přenášené zprávy, přijatá zpráva platná. Na úpravu zpráv se využívají dva kanonikalizační algoritmy – „simple“ a „relaxed“. „Simple“ je algoritmus, který netoleruje téměř žádnou úpravu původní zprávy. „Relaxed“ je protiklad předchozího algoritmu, upravuje zprávu tak, aby mírné úpravy v průběhu přenosu nezpůsobily neplatnost validace. Je schopný zpracovat přeházenou hlavičku e-mailové zprávy a především mezery. [44], [45]

3.2.4 Podepisování a validace zprávy

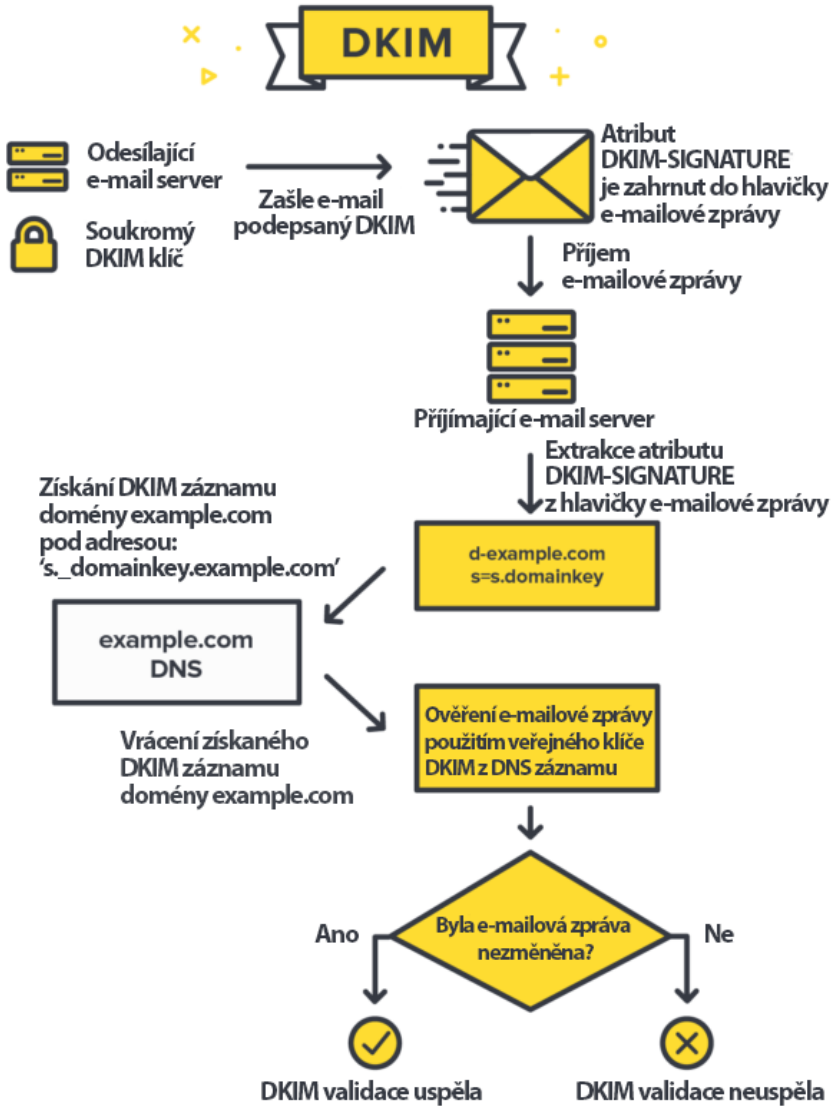
- **Podepisování zprávy**

Před podepsáním zprávy klíčem DKIM, či ověřením DKIM podpisu, je potřeba odesílanou či přijímanou zprávu upravit, dle zvoleného kanonikalizačního algoritmu na straně serveru odesílatele. Po kanonikalizaci se zpráva podepisuje, na to se využívají aktuálně podporované podepisovací algoritmy jako jsou „RSA-SHA1“ a „RSA-SHA256“. Nicméně standard doporučuje zejména algoritmus „RSA-SHA256“, který kombinuje „SHA256“ pro výpočet hashe a algoritmus „RSA“ pro zašifrování. Pro podpis e-mailových zpráv odesílající server či systém brány využije nastavený soukromý klíč a přidá podpis do hlavičky e-mailové zprávy atribut „DKIM-Signature“, který může vypadat následovně: [44], [45]

„*DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=seznam.cz; s=beta;t=1578641730;bh=/TD0lk+sB1gnEN1dbkIDJAt59PhptcTvRhtuO1OrpNs=;h=Received:From:To:Subject:Date:Message-Id:Mime-Version:X-Mailer:Content-Type;b=LfWqIXU+MPBXIgfdHqh5y6TGGK/avFzMX6Lu7mFKNE8sPSCIGV8Vy5iFIyU MqFvyjUnWwctBxpDJ8TuaCOHB3dMp0PD9CdmZGa/0vOJhFE/SxVHFVs/iUDR7HLvTv fG2PA9hwLQxdZAuX4PIDOGTgERxJ/FB9fxb0H3ga7FPuno=“* [45]

- **Validace zprávy**

Validace zprávy může nastat u doručovacích systémů (MDA), přenosových systémů (MTA) či dokonce na straně uživatelských agentů (MUA). Před validací zprávy, se e-mailová zpráva upraví dle požadovaného kanonikalizačního algoritmu. Z kanonikalizované zprávy se vytvoří hash, který se porovná s hashem v atributu podpisu. Pokud hashe nejsou identické, DKIM validace je neplatná. Následuje získání veřejného klíče z DNS systému domény odesílatele. Pomocí veřejného klíče se z atributu podpisu dešifruje očekávaný hash hlavičky, který se porovná s hashem kanonikalizované hlavičky. Pokud hashe nejsou identické, DKIM validace je neúspěšná. [44], [45]



Obrázek 9. Diagram validace DKIM. [46]

3.3 Digitální podpis

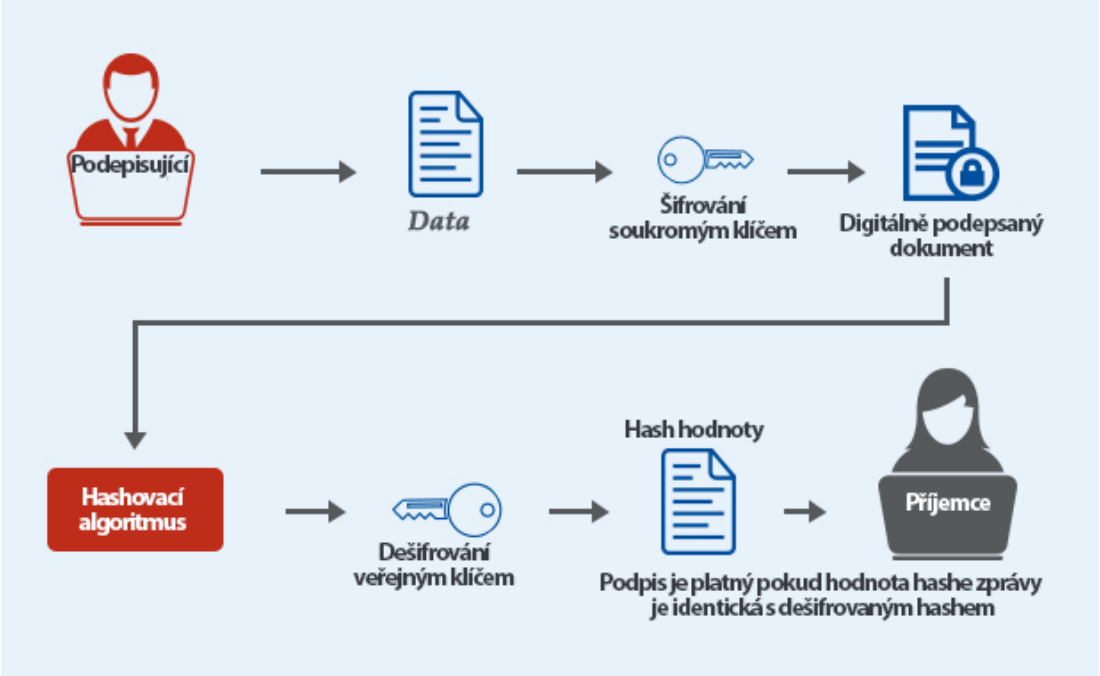
Digitální podpis se používá k potvrzení toho, že přenášená data nebyla po jejich podepsání odesílatelem nijak upravena. Digitální podpisy využívají kryptografický protokol zvaný PKI (Public Key Infrastructure), který vyžaduje dvojici klíčů, soukromého a veřejného klíče, jež jsou vystaveny důvěryhodnou certifikační autoritou (CA). V e-mailových protokolech se digitální podpis objevil díky standardu MIME, který umožnil přenos digitálního podpisu v e-mailových zprávách, viz. 2.3.2.5. [43], [42], [47], [48], [49]

3.3.1 Podepisování

Pro podepisování dat digitálním podpisem je zapotřebí soukromý klíč. Podepisující vytvoří hash odesílaných dat, který se zašifruje privátním klíčem, čímž vznikne digitální podpis. [48], [50]

3.3.2 Ověřování

Ověření platnosti digitálního podpisu spočívá v dešifrování zprávy kontrolního hashe veřejným klíčem, který se porovná s hashem vytvořeným z přijatých dat. [48], [50]



Obrázek 10. Diagram digitálního podpisu - podepisování a ověřování. [46]

3.4 PGP (Pretty Good Privacy)

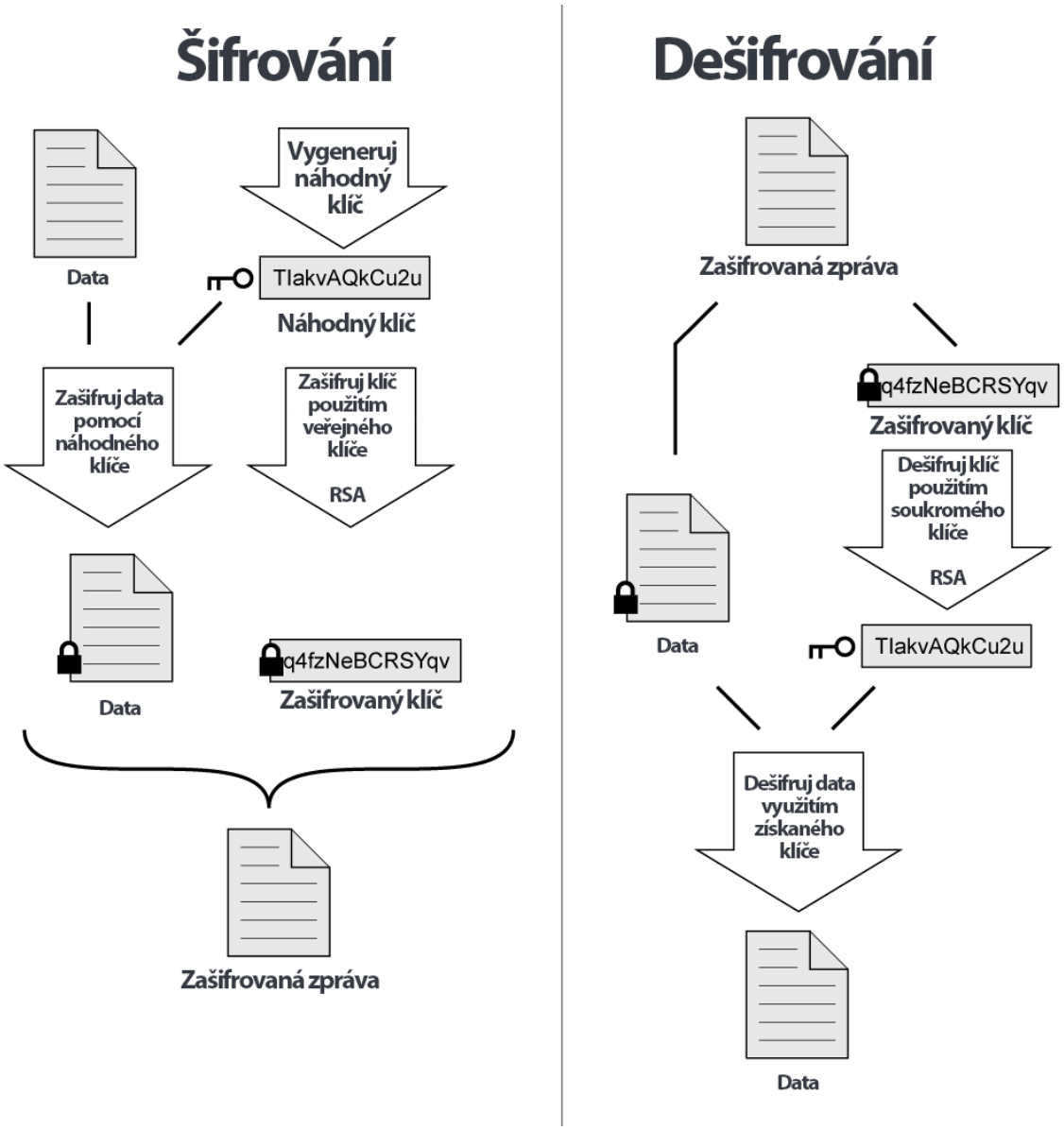
PGP umožňuje odesílateli důvěryhodně, za pomocí kryptografie, přenést zprávu, aniž by ji kdokoliv, mimo cíleného příjemce, mohl jakkoliv přečíst. [7]

3.4.1 Šifrování

Utajovaná zpráva je před odesláním zašifrovaná symetrickou šifrou, jejíž klíč je zašifrován veřejným klíčem. [7]

3.4.2 Dešifrování

Získaný zašifrovaný klíč se dešifruje za pomocí soukromého klíče, a poté využitím dešifrovaného klíče se dešifrují data. [7]



Obrázek 11. Diagram PGP - šifrování a dešifrování. [51]

4 MOŽNOSTI K ODHALENÍ NEŽÁDOUCÍCH ZPRÁV

Nevyžádaná pošta, známá pod termínem „SPAM“, se stala jednou z největších problémů sítě Internet. V dubnu roku 2003 se poskytovateli internetu AOL podařilo zablokovat 2,37 miliard nevyžádaných zpráv. Naštěstí vůči této pandemii, existuje mnoho způsobů jak takové zprávy potlačit. [52]

4.1 SPF

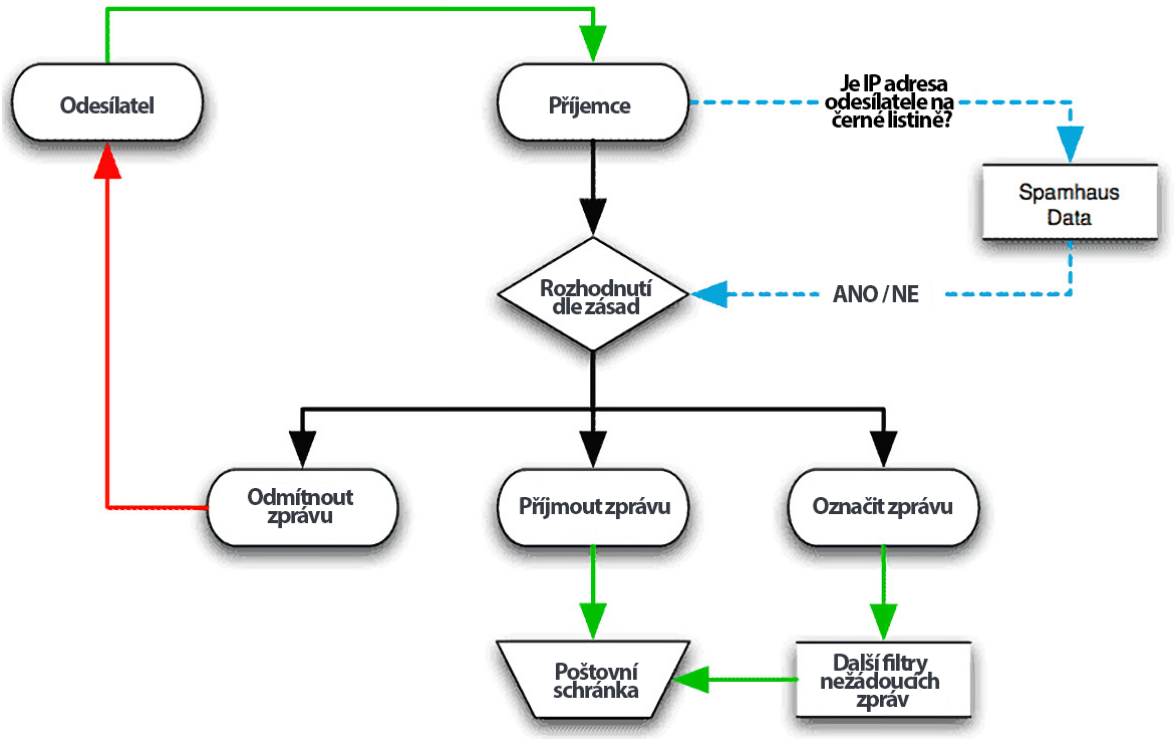
SPF je validační mechanismus autorizace odesílatele, na základě sady pravidel v DNS systému dané domény, který opravňuje odesílat zprávy z určitých domén či IP adres, viz. 3.1. Při doručování zprávy přenosovým poštovním systémem (MTA), může přijímací systém přijímanou zprávu, před jejím uložení do e-mailové schránky, zkontrolovat pomocí SPF validace. Pokud by validace neuspěla, znamenalo by to, že systém odesílatele (MTA) nemá oprávnění posílat e-mailové zprávy za danou doménu, a dle nastavených zásad na straně SMTP serveru by mohla být daná zpráva zamítnuta. [42]

4.2 DKIM

DKIM je metoda zajišťující důvěryhodné zasílání zpráv za organizaci či osobu, viz. 3.2. Při příjmu e-mailové zprávy, podepsané podpisem DKIM, může být zpráva přijímajícím serverem ověřena. Pokud by validace DKIM podpisu nebyla úspěšná, mohla by být daná zpráva dle nastavené politiky SMTP serveru odmítnuta či označena jako nežádoucí. [44]

4.3 Reputace odesílatele

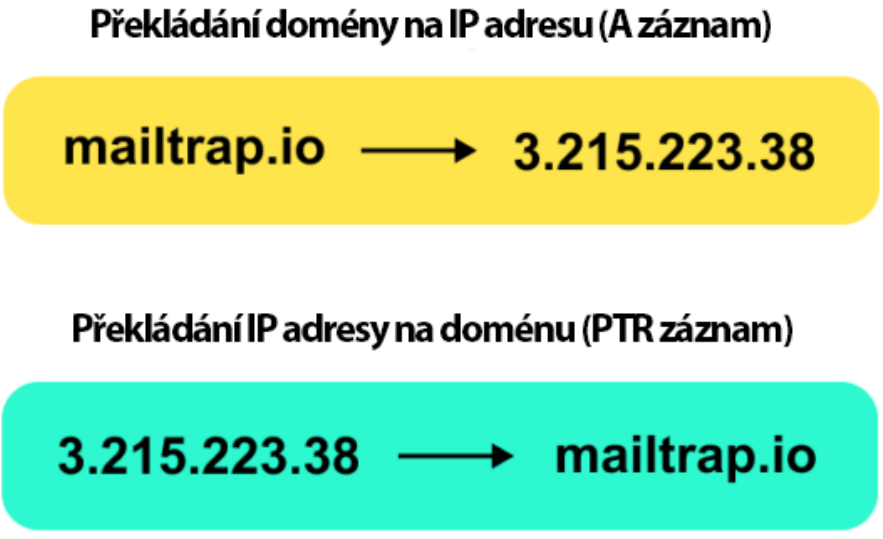
DNSBL je veřejná databáze obsahující názory na IP adresy odesílatelů e-mailových zpráv. Při přijímání e-mailové zprávy může být IP adresa odesílajícího SMTP serveru zkontrolována vůči veřejným černým listinám DNSBL, které obsahují IP adresy známých odesílajících, zasílající nežádoucí zprávy. Pokud by na takové listině byla nalezena IP adresa odesílatele, zpráva by se mohla považovat jako nežádoucí. [53], [54]



Obrázek 12. Diagram DNSBL – rozhodnutí o zprávě dle reputace. [54]

4.4 Nepřímý PTR záznam

PTR záznam je typ DNS záznamu, který převádí IP adresu na doménové jméno. Převedené doménové jméno poté může přijímající server porovnat s doménou v adrese odesílatele. Pokud by doména odesílatele neseděla s doménou z PTR záznamu, mohlo by se jednat o odesílatele nežádoucí zprávy. [55], [56]



Obrázek 13. PTR záznam - převod domény na IP adresu a nazpět. [56]

II. PRAKTICKÁ ČÁST

5 IMPLEMENTACE ŘEŠENÍ

E-mailový klient byl naprogramován v jazyce Python se síťovou komunikační vrstvou TCP/IP pomocí knihovny Sockets.

5.1 Spojení s IMAP serverem

Aplikace využívá implementovanou knihovnu ImapLib. Tato knihovna podporuje PLAINTEXT, STARTTLS i SSL komunikaci se vzdáleným IMAP serverem. Dále si poradí s jakýmkoliv typem zprávy – doplněné přílohami, zakódované v BASE64, či s quoted-printable kódováním.

V následujícím snímcích, Obrázku 14 a Obrázku 15, jde vidět, jak naimplementovaný e-mailový klient komunikuje s IMAP serverem, který podporuje SSL komunikaci. Po připojení klienta k serveru, klient vyčká na uvítací zprávu a zpětně pošle přihlašovací údaje uživatele. Pokud bylo přihlášení úspěšné, server vrátí stavový kód „OK“, pro indikaci úspěšného přihlášení. Dále klient požádá server o vypsání všech dostupných složek e-mailové schránky, kvůli možnosti přecházení klienta mezi složkami. Následně se klient serveru dotáže na všechny unikátní identifikátory e-mailových zpráv ve výchozí složce, které dále porovná s identifikátory, již stažených e-mailů v lokální SQLite databázi. Nakonec se začne postupně dotazovat na nové e-mailové zprávy, které poté uloží v místní SQLite databázi.

```
-Server - Client-
---> b'* OK Gimap ready for requests from **.**.***.** f11mb104283994edr\r\n'
-Server - Client-
<--- b'. login ********@gmail.com *******\r\n'
-Server - Client-
---> b'* CAPABILITY IMAP4rev1 UNSELECT IDLE NAMESPACE QUOTA ID XLIST CHILDREN X-GM-EXT-1 UIDPLUS COMPRESS=DEFLATE ENABLE
MOVE CONDSTORE ESEARCH UTF8=ACCEPT LIST-EXTENDED LIST-STATUS LITERAL- SPECIAL-USE APPENDLIMIT=35651584\r\n. OK evilcorp9@
gmail.com authenticated (Success)\r\n'
-Server - Client-
<--- b'. LIST "" "*"\r\n'
-Server - Client-
---> b'* LIST (\\HasNoChildren) "/" "INBOX"\r\n* LIST (\\HasChildren \\Noselect) "/" "[Gmail]"\r\n* LIST (\\HasNoChildren
 \\Important) "/" "[Gmail]/D&AW8-le&AX4-it&AOk-"\r\n* LIST (\\Drafts \\HasNoChildren) "/" "[Gmail]/Koncepty"\r\n* LIST (\
\HasNoChildren \\Trash) "/" "[Gmail]/Ko&AWE-"\r\n* LIST (\\HasNoChildren \\Sent) "/" "[Gmail]/Odeslan&AOE- po&AWE-ta"\r\n
* LIST (\\Flagged \\HasNoChildren) "/" "[Gmail]/S hv&ARs-zdi&AQ0-kou"\r\n* LIST (\\HasNoChildren \\Junk) "/" "[Gmail]/Spa
m"\r\n* LIST (\\All \\HasNoChildren) "/" "[Gmail]/V&AWE-echny zpr&AOE-vy"\r\n. OK Success\r\n'
-Server - Client-
<--- b'. SELECT "INBOX"\r\n'
-Server - Client-
---> b'* FLAGS (\\Answered \\Flagged \\Draft \\Deleted \\Seen $NotPhishing $Phishing NonJunk)\r\n* OK [PERMANENTFLAGS (\\
Answered \\Flagged \\Draft \\Deleted \\Seen $NotPhishing $Phishing NonJunk \\*)] Flags permitted.\r\n* OK [UIDVALIDITY 1]
 UIDs valid.\r\n* 2099 EXISTS\r\n* 0 RECENT\r\n* OK [UIDNEXT 2109] Predicted next UID.\r\n* OK [HIGHESTMODSEQ 251980]\r\n
. OK [READ-WRITE] INBOX selected. (Success)\r\n'
-Server - Client-
<--- b'. UID SEARCH ALL\r\n'
-Server - Client-
---> b'* SEARCH 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 7
9 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114
 115 116 117 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145
```

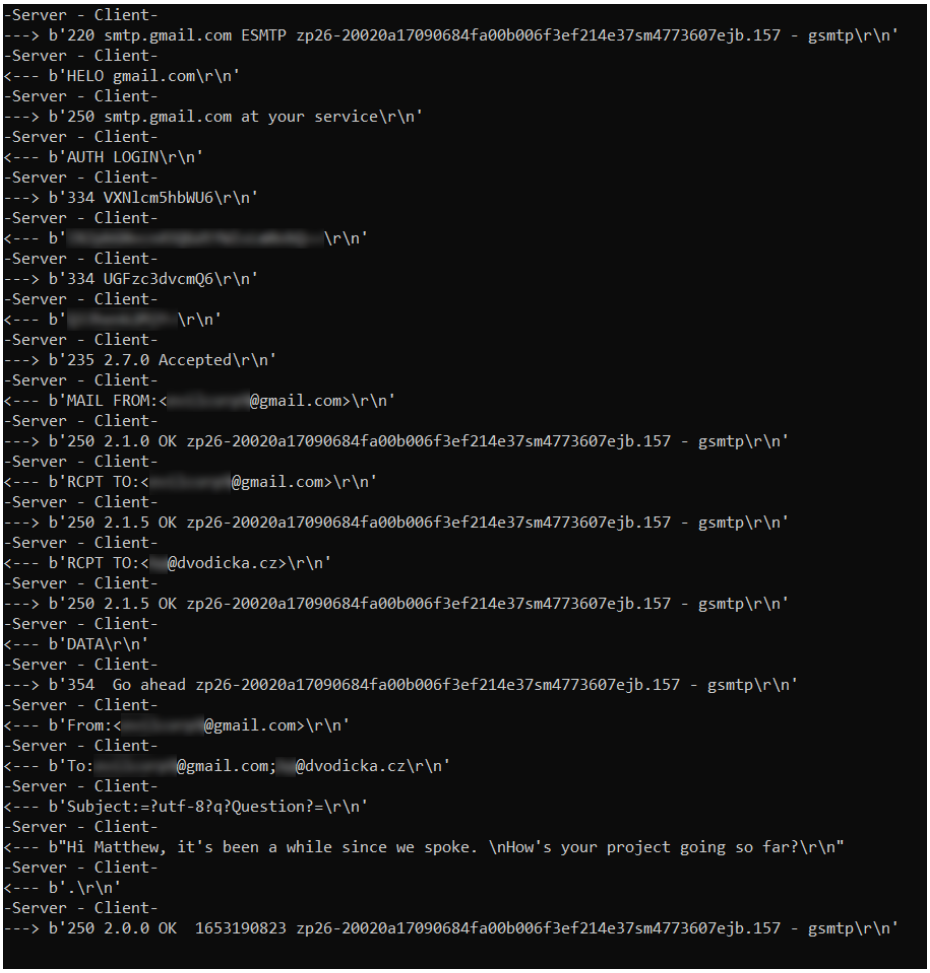
Obrázek 14. Spojení s IMAP serverem část 1. – přihlášení, načtení složek a unikátních identifikátorů e-mailových zpráv.

Obrázek 15. Spojení s IMAP serverem část 2. – načtení nové e-mailové zprávy ve formátu RFC822.

5.2 Spojení s SMTP serverem

Aplikace využívá implementovanou knihovnu SmtpLib. Knihovna podporuje PLAINTEXT, STARTTLS i SSL komunikaci s SMTP serverem.

V Obrázku 16 je vyobrazeno, jakým způsobem implementovaná knihovna SmtpLib komunikuje s SMTP serverem přes SSL komunikaci. Po navázání spojení klient dostává pozdravnou hlášku, kterou se SMTP server identifikuje. Dále se klient poštovnímu serveru identifikuje odesílatelem využívanou doménou, v aktuálním případě to je „gmail.com“. Server odpoví, že z dané domény e-maily přijímá a klient započne autentizaci. Přihlašovací jméno i heslo je serveru odesíláno v BASE64 kódování. Po přihlášení, klient začne vyplňovat e-mailovou transakci. Následuje zadání adresy odesílatele, poté příjemce. V této ukázce jsou uvedeni dva příjemci stejné zprávy, proto je příkaz „RCPT TO“, opakován dvakrát. Stejná data se ještě zapíší do formálních hlaviček „From“ a „To“, dělá se to především z důvodu využití aliasu odesílatele, či příjemce. Poté se zadává předmět odesílané zprávy, a znakem „ . “ se ukončí transakce, čímž se zpráva se odešle.



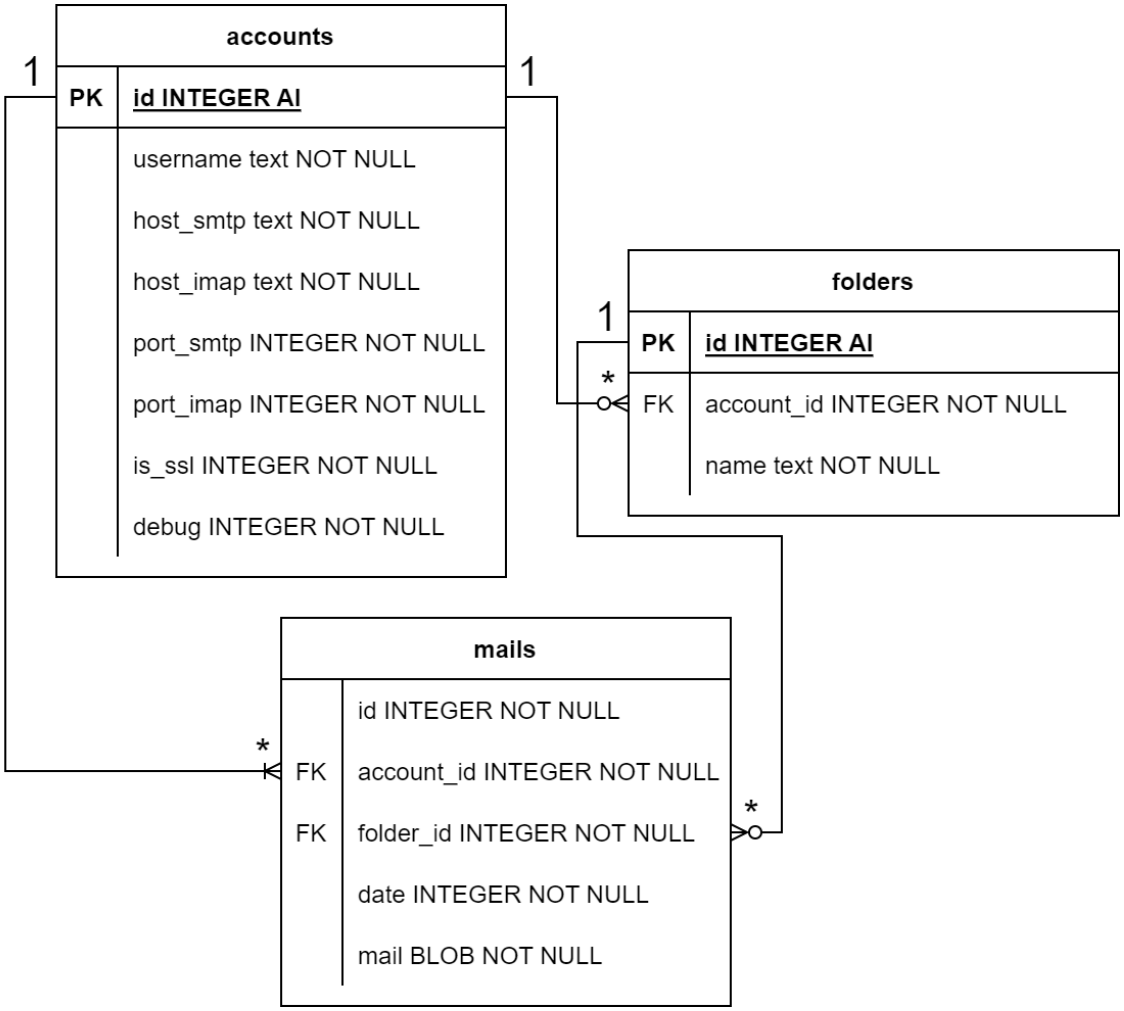
Obrázek 16. Spojení s SMTP serverem.

5.3 SQLite databáze

Aplikace využívá místní SQLite databázi pro ukládání konfigurace uživatele a přijatých e-mailových zpráv pro možnost práce bez internetového připojení.

5.3.1 Model databáze

Databázový model využívá vztahy tabulek vyobrazené v Obrázku 17. Model je navržen tak, aby nemohlo nastat uložení špatně načtených dat, proto jsou všechny položky databáze omezeny podmínkou „NOT NULL“. V obrázku je znázorněno, že nekonečně mnoho záznamů tabulky *mails* může mít vazbu na jeden záznam uživatele v tabulce *accounts*. Totéž platí i u tabulky *folders*, kde mnoho záznamů této tabulky může mít návaznost na jeden záznam tabulky *accounts*. To samé platí u vztahu tabulky *accounts* s tabulkou *folders*, jeden uživatel může vlastnit nekonečně mnoho složek e-mailové schránky.



Obrázek 17. Vztahy tabulek v databázi SQLite.

5.3.2 Tabulka accounts

Tabulka obsahuje konfiguraci uživatele – přihlašovací jméno, heslo, adresy a porty pro příchozí a odchozí poštu.

Tabulka: accounts — Filter in any column

	id	username	password	host_smtp	host_imap	port_smtp	port_imap	is_ssl	debug
	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr
1	1	********@gmail.com	*******	smtp.gmail.com	imap.gmail.com	465	993	1	1

Obrázek 18. Výpis tabulky *accounts* z databáze SQLite.

5.3.3 Tabulka folders

Tabulka obsahuje názvy, identifikátor vlastníka složky a identifikátory složek e-mailové schránky.

Tabulka: folders — Filter in any column

	id	account_id	name
	Filtr	Filtr	Filtr
1	1	1	INBOX
2	2	1	[Gmail]
3	3	1	[Gmail]/Důležité
4	4	1	[Gmail]/Koncepty
5	5	1	[Gmail]/Koš
6	6	1	[Gmail]/Odeslaná pošta
7	7	1	[Gmail]/S hvězdičkou
8	8	1	[Gmail]/Spam
9	9	1	[Gmail]/Všechny zprávy

Obrázek 19. Výpis tabulky *folders* z databáze SQLite.

5.3.4 Tabulka mails

Tabulka obsahuje všechny doposud přijaté e-mailové zprávy ve formátu standardu RFC822, spolu s identifikátorem zprávy a časovým razítkem přijetí.

Tabulka: mails — Filter in any column

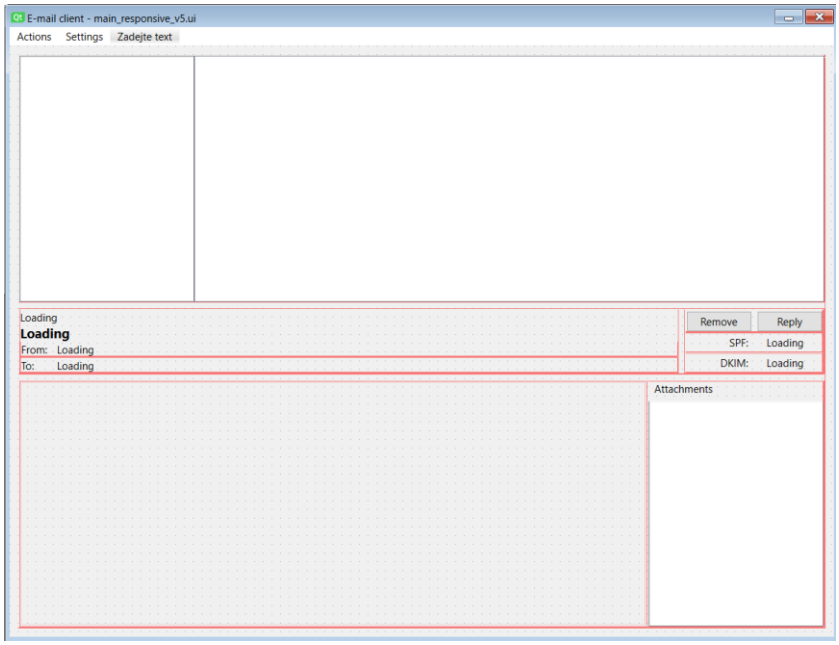
	id	account_id	folder_id	date	mail
	Filtr	Filtr	Filtr	Filtr	Filtr
1	2108	1	1	1652115194	* 2099 FETCH (UID 2108 RFC822 ...
2	2107	1	1	1651741664	* 2098 FETCH (UID 2107 RFC822 ...
3	2106	1	1	1651709218	* 2097 FETCH (UID 2106 RFC822 ...
4	2104	1	1	1651491695	* 2096 FETCH (UID 2104 RFC822 ...
5	2103	1	1	1651516859	* 2095 FETCH (UID 2103 RFC822 ...
6	2102	1	1	1651516829	* 2094 FETCH (UID 2102 RFC822 ...
7	2101	1	1	1651189385	* 2093 FETCH (UID 2101 RFC822 ...
8	2100	1	1	1650931228	* 2092 FETCH (UID 2100 RFC822 ...
9	2099	1	1	1650808213	* 2091 FETCH (UID 2099 RFC822 ...
10	2098	1	1	1650641651	* 2090 FETCH (UID 2098 RFC822 ...
11	2097	1	1	1650418157	* 2089 FETCH (UID 2097 RFC822 ...
12	2096	1	1	1650191112	* 2088 FETCH (UID 2096 RFC822 ...
13	2095	1	1	1650190911	* 2087 FETCH (UID 2095 RFC822 ...
14	2094	1	1	1649792199	* 2086 FETCH (UID 2094 RFC822 ...

Obrázek 20. Výpis tabulky *mails* z databáze SQLite.

5.4 Grafické rozhraní

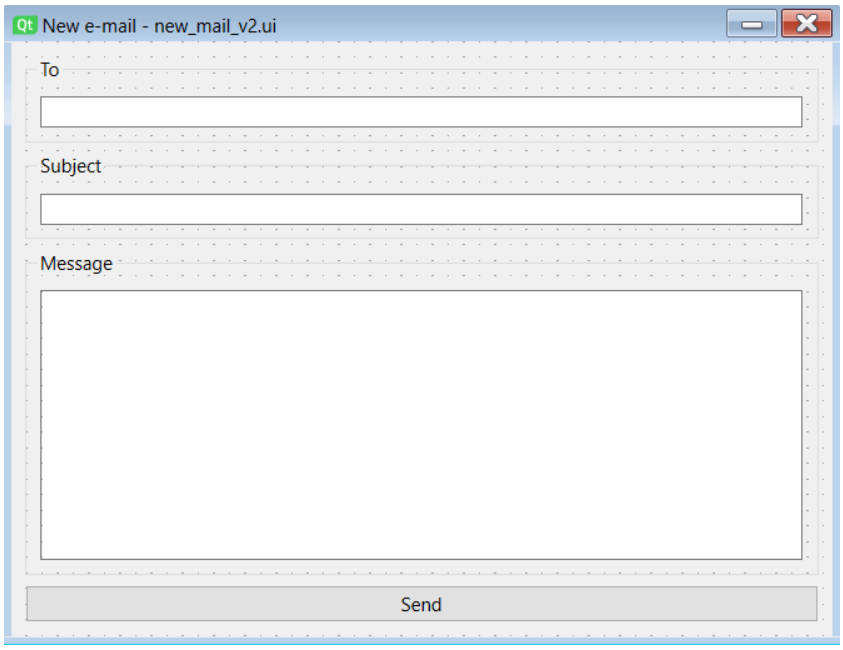
Grafické rozhraní bylo implementováno pomocí knihovny PySide6. Design aplikace byl navržen v aplikaci Qt Designer a je plně responzivní, obsah aplikace se přizpůsobuje velikosti okna programu.

- **Úvodní obrazovka**



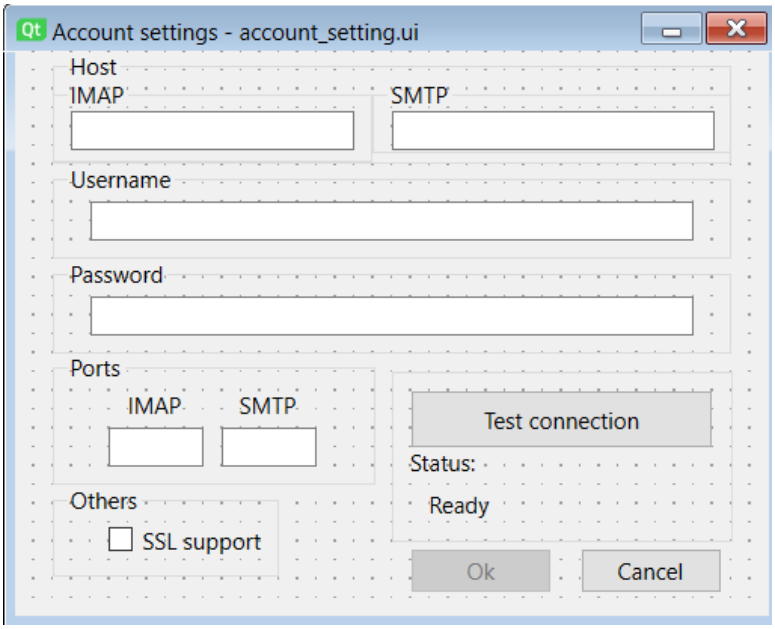
Obrázek 21. Návrh designu aplikace – Úvodní obrazovka aplikace.

- **Odesílání nové zprávy**



Obrázek 22. Návrh designu aplikace – odesílání nové zprávy.

- **Nastavení uživatelské konfigurace**



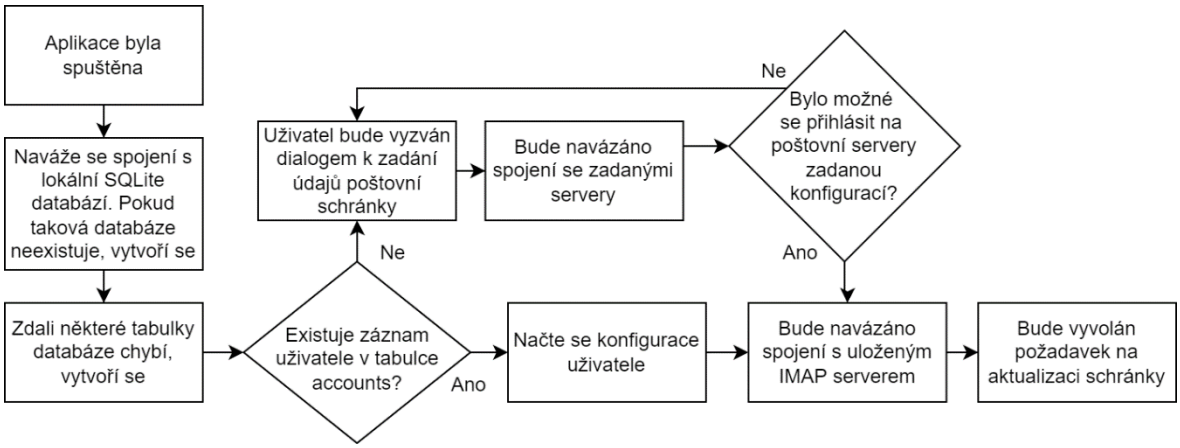
Obrázek 23. Návrh designu aplikace – nastavení uživatelské konfigurace.

5.5 Vývojové diagramy implementace

Tato část obsahuje popsané základní funkce implementovaného programu, s pomocí diagramů.

- **Spuštění aplikace**

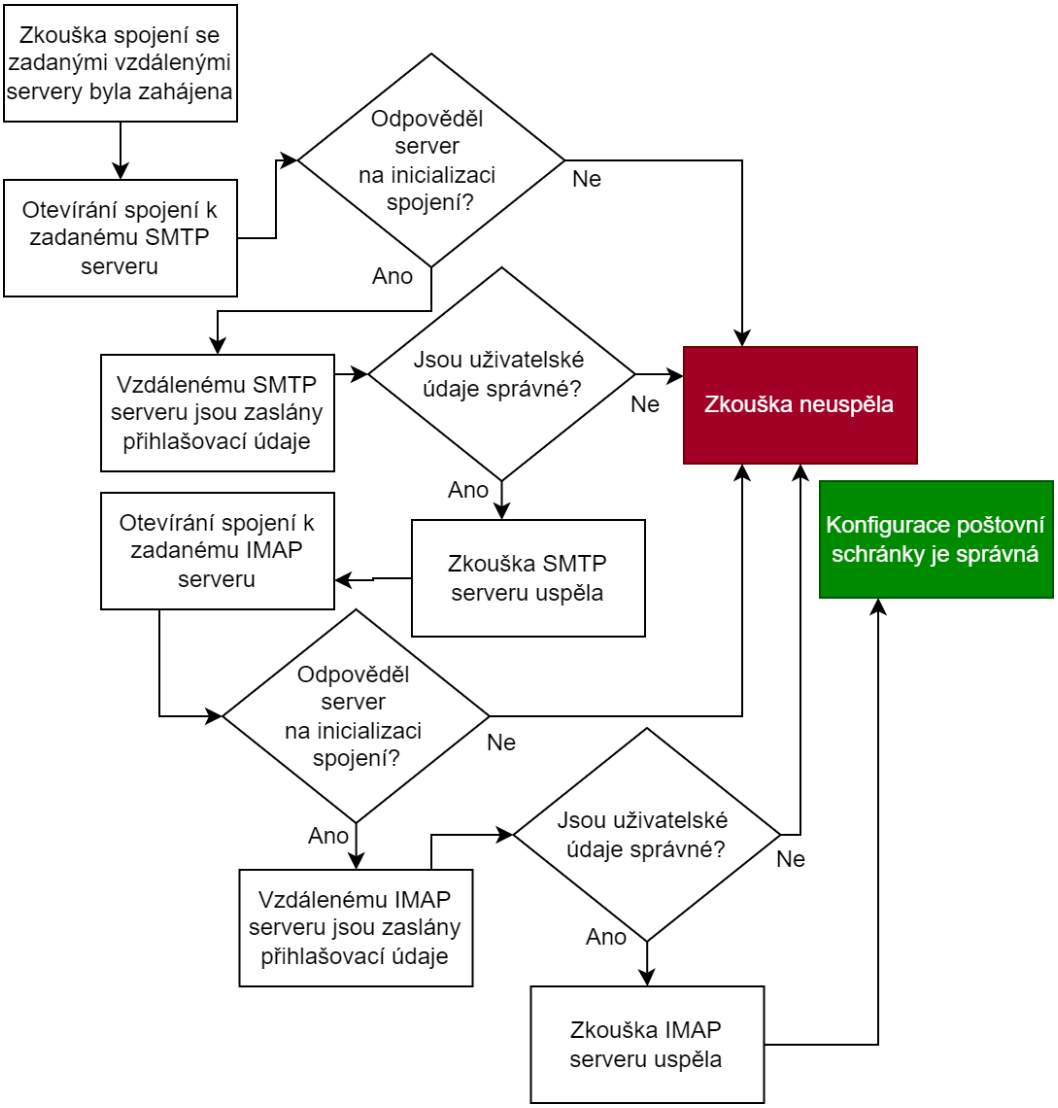
  Diagram v obrázku níže vyobrazuje spouštěcí logiku aplikace.



Obrázek 24. Diagram spuštění aplikace.

- **Zkouška spojení**

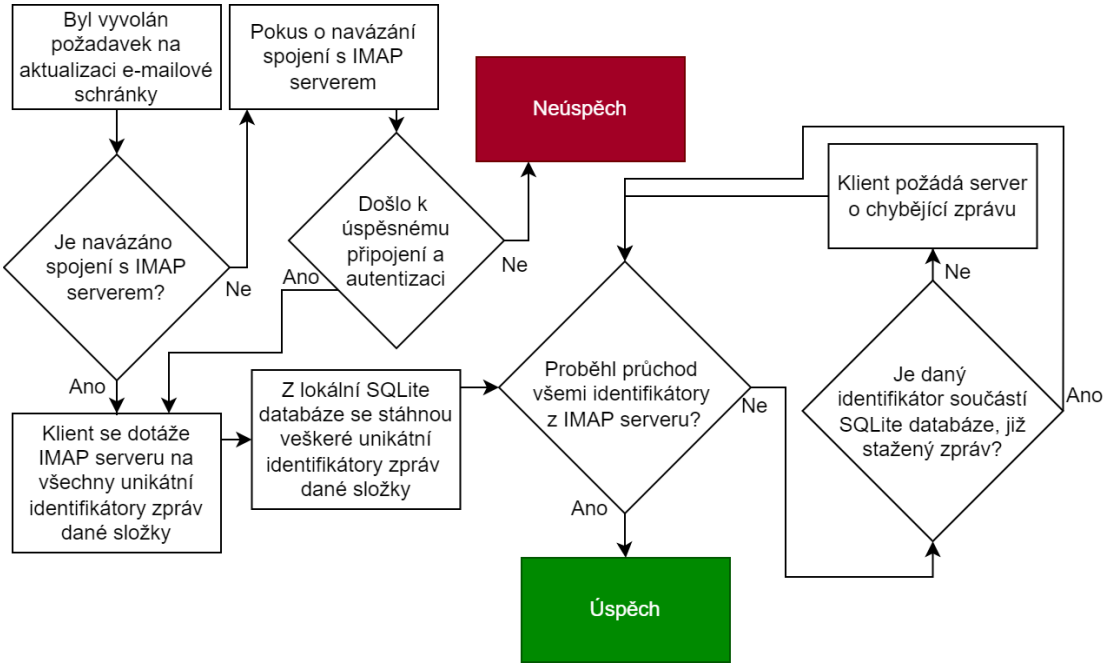
Diagram v Obrázku 25 znázorňuje, jakým způsobem probíhá zkouška nové uživatelské konfigurace.



Obrázek 25. Diagram zkoušky konfigurace.

- **Aktualizace poštovní schránky**

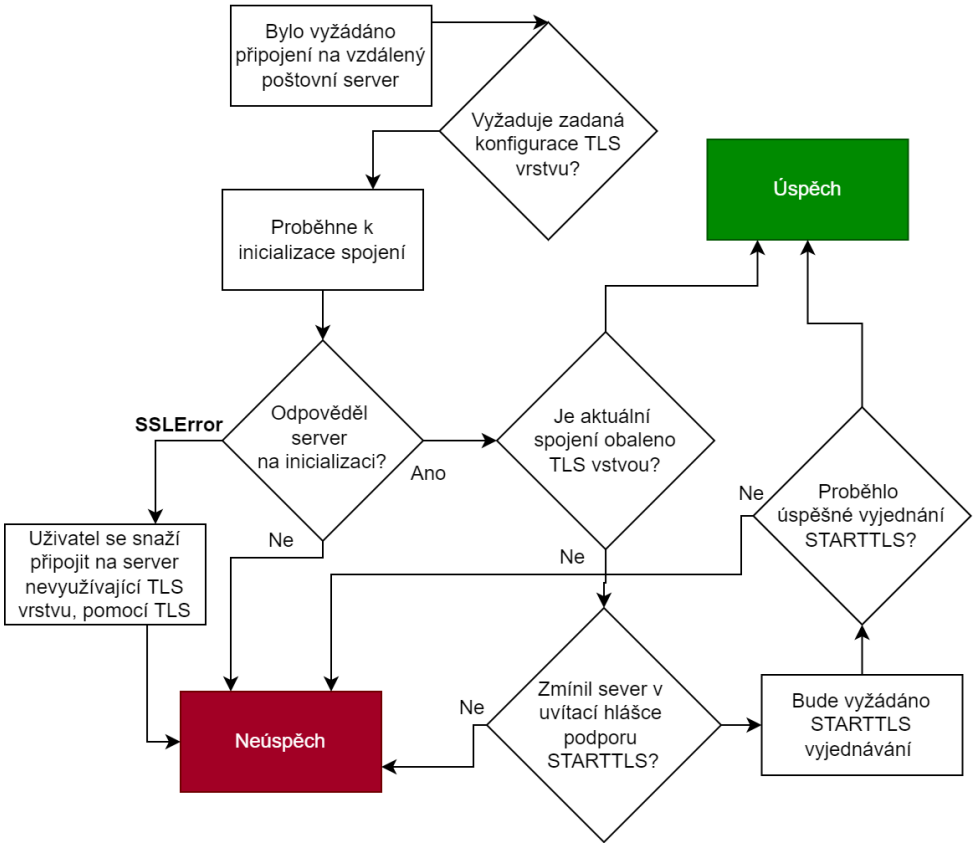
Diagram níže vysvětluje průběh funkce, pro aktualizaci e-mailové schránky.



Obrázek 26. Diagram aktualizace poštovní schránky.

- **Vyjednávání šifrovaného spojení**

Tento diagram popisuje logiku vyjednávání o zašifrované spojení. Pokud se vyjednání nezdaří, spojení nebude využíváno, z důvodu bezpečnosti.



Obrázek 27. Diagram vyjednávání šifrovaného spojení.

5.6 Manuál aplikace

Tento úsek práce je zaměřen na jednotlivé funkce implementovaného e-mailového klienta. Rovněž bude vysvětlena jejich podstata a funkce v celku dané třídy. Aplikace je rozdělena do čtyř souborů.

5.6.1 Main

Soubor s třídy pro práci s grafickým rozhraním aplikace.

- **Třída *UI***

  Jedná se o hlavní třídu, jejíž instance je vytvořena hned po spuštění programu. Třída obsahuje celou logiku hlavního okna aplikace, výpis složek, e-mailů, případných příloh e-mailových zpráv a též vykresluje zvolenou zprávu.

- **Třída *Account***

  Tato třída zastupuje konfiguraci uživatele ve své instanci. Taktéž se stará o načtení konfigurace z lokální databáze, či o přidání konfigurace do databáze.

- **Třída *DB***

  Rozhraní pro práci s lokální SQLite databází.

- **Třída *NewMail***

  Třída starající se o logiku grafického rozhraní pro psaní nové e-mailové zprávy.

- **Třída *AccountSettings***

  Daná třída pečuje o logiku grafického rozhraní pro nastavení uživatelské konfigurace.

- **Třída *MailTableModel***

  Je instancí abstraktní třídy *QAbstractTableModel*, starající se o vykreslování položek seznamu e-mailových zpráv.

- **Třída *WebEnginePageModel***

  Jde o abstrakci třídy *QWebEnginePage*, která se stará o zpracování kliknutí na hypertextový odkaz, aby byl otevřen v systémovém prohlížeči.

5.6.2 Imap

Skupina tříd starající se o protokol IMAP

- **Třída *ImapLib***

  Veškerou komunikaci s IMAP servery spravuje tato třída.

- **Třída *Mail***

  Objekt zastupující načtenou e-mailovou zprávu.

- **Třída *MailAttachment***

  Tato třída zastupuje přílohy e-mailových zpráv.

- **Třída *ImapFolder***

  Třída zastupující složku na poštovním serveru.

- **Třída *ImapFlags***

  Objekt typu ENUM obsahující všechna dostupná značení e-mailových zpráv.

- **Třída *ImapFolderFlags***

  Objekt typu ENUM obsahující všechna značení složek e-mailového serveru.

5.6.3 Smtp

Soubor tříd SMTP protokolu

- **Třída *SmtpLib***

  Třída starající o všechnu komunikaci s SMTP servery.

5.6.4 Utils

Statické, pomocné funkce

- **Třída *Utils***

  Statická třída obsahující pomocné funkce využívané napříč celým klientem.

- **Třída *UiUtils***

  Statická třída obsahující pomocné funkce grafického rozhraní.

5.7 Zhodnocení implementace

Řešení je schopné informovat uživatele o důvěryhodnosti přijaté zprávy. Implementace funguje velmi spolehlivě, ale pouze v případě kontroly čerstvě přijatých e-mailů. U starších zpráv může nastat problém, kdy vlastník domény nebo správce e-mailového serveru nahradí veřejný DKIM klíč, nově vygenerovaným, v tom případě by validace neuspěla. Totéž by mohlo nastat i u SPF záznamu, v případě přesunu serveru na novou IP adresu, nastavená privilegia v DNS systému pro zasílání za danou doménu by se v tomto případě přepsaly, a zpráva by ztratila legitimitu. Bohužel tato implementace plně nenahradí ověření na straně poštovního přenosového systému (MTU), nicméně na poštovních systémech, kde takové ověření nastaveno není, ji tato implementace celkem spolehlivě zastoupí.

ZÁVĚR

Cílem bakalářské práce bylo vytvořit e-mailového klienta schopného ujistit příjemce, že přijatá zpráva je legitimní.

V teoretické části byla vypracována rešerše o aktuálním stavu klientů na trhu, spolu s porovnáním funkcionalit jednotlivých poštovních klientů. Z porovnání vyplynulo, že je e-mailový klient Mozilla Thunderbird, je nejlépe připraven na bezpečné, důvěryhodné čtení e-mailových zpráv. Následně byly popsány základní komunikační protokoly poštovních serverů. Poté byly vylíčeny postupy legitimizace odesílatele zprávy a metodiky k odhalení nežádoucích zpráv.

V praktické části byla vyobrazena spojení s poštovními servery, skrze implementované komunikační knihovny. Rovněž byla znázorněna konfigurace lokální SQLite databáze, společně s využívaným rozhraním implementované aplikace. V aplikaci byly implementovány principy pro legitimizaci přijímaných zpráv, jejichž zásluhou je klient schopen informovat o legitimitě zprávy. Dále byla prezentována logika aplikace za pomocí diagramů základních funkcí aplikace. Nakonec byly vykresleny jednotlivé třídy implementovaného klienta, s popisem jejich funkce. Výstupem práce je aplikace ujišťující příjemce o legitimnosti zprávy. Další částí výstupu jsou metodiky pro legitimizaci odesílatele, principy k odhalení nežádoucích zpráv a souhrn základních protokolů pro komunikaci poštovními servery.

SEZNAM POUŽITÉ LITERATUTY

[1] *Microsoft Outlook 2019* [online]. T-Mobile Czech Republic, 2022 [cit. 2022-05-19]. Dostupné z: https://www.t-mobile.cz/podpora/technicka-podpora/internet-a-e-mail/nastaveni-e-mailoveho-klienta-v-pocitaci/microsoft-outlook-2019

[2] *Thunderbird vs Outlook: Which One is Better?* [online]. TEMOK, 2020 [cit. 2022-05-19]. Dostupné z: https://www.temok.com/blog/thunderbird-vs-outlook-which-one-is-better/

[3] *My Top 5 Thunderbird Add-ons* [online]. Romain Vuillemot [cit. 2022-05-19]. Dostupné z: https://romain.vuillemot.net/posts/my-top-5-thunderbird-add-ons/

[4] *Lightning Calendar* [online]. Mozilla Corporation, 2022 [cit. 2022-05-17]. Dostupné z: https://www.thunderbird.net/en-US/calendar/

[5] *Sender Verification Anti-Phishing Extension* [online]. Mozilla Corporation, 2008 [cit. 2022-05-17]. Dostupné z: https://addons.thunderbird.net/en-US/thunderbird/addon/sender-verification-anti-phish/

[6] *DKIM Verifier* [online]. Mozilla Corporation, 2022 [cit. 2022-05-17]. Dostupné z: https://addons.thunderbird.net/en-US/thunderbird/addon/dkim-verifier/

[7] *OpenPGP Message Format* [online]. IETF, 2007 [cit. 2022-05-17]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc4880

[8] *Automatic Account Configuration* [online]. Mozilla Corporation, 2022 [cit. 2022-05-18]. Dostupné z: https://support.mozilla.org/en-US/kb/automatic-account-configuration

[9] *Thunderbird FAQ* [online]. Mozilla Corporation, 2022 [cit. 2022-05-19]. Dostupné z: https://support.mozilla.org/en-US/kb/thunderbird-faq

[10] *Digitally Signing and Encrypting Messages* [online]. Mozilla Corporation, 2022 [cit. 2022-05-17]. Dostupné z: https://support.mozilla.org/en-US/kb/digitally-signing-and-encrypting-messages

[11] *OpenPGP in Thunderbird - HOWTO and FAQ* [online]. Mozilla Corporation, 2022 [cit. 2022-05-17]. Dostupné z: https://support.mozilla.org/en-US/kb/openpgp-thunderbird-howto-and-faq

[12] *Mailbird vs. Thunderbird* [online]. Softwarehow, 2021 [cit. 2022-05-19]. Dostupné z: https://www.softwarehow.com/mailbird-vs-thunderbird/

[13] *Mailbird Pricing* [online]. Mailbird, 2022 [cit. 2022-05-19]. Dostupné z: https://www.getmailbird.com/pricing/

[14] *Mailbird Features That Turn You Into An Email Ninja* [online]. Mailbird, 2022 [cit. 2022-05-20]. Dostupné z: https://www.getmailbird.com/features/

[15] *Mailbird Native Calendar* [online]. Mailbird, 2021 [cit. 2022-05-17]. Dostupné z: https://support.getmailbird.com/hc/en-us/articles/360016953673-Mailbird-Native-Calendar

[16] *Access your Email.it Account from an Email Program using IMAP* [online]. Mailbird, 2022 [cit. 2022-05-18]. Dostupné z: https://www.getmailbird.com/setup/access-email-it-via-imap-smtp

[17] *Multiple Email Accounts in Mailbird* [online]. Mailbird, 2021 [cit. 2022-05-20]. Dostupné z: https://support.getmailbird.com/hc/en-us/articles/220106747-Multiple-Email-Accounts-in-Mailbird

[18] *Manage multiple accounts with the best email client 2022* [online]. Mailbird, 2022 [cit. 2022-05-20]. Dostupné z: https://www.getmailbird.com/

[19] *EM Client* [online]. 2017 [cit. 2022-05-20]. Dostupné z: https://www.openpgp.org/software/emclient/

[20] *Freemium* [online]. TechTarget, 2008 [cit. 2022-05-20]. Dostupné z: https://www.techtarget.com/searchitchannel/definition/freemium

[21] *Otázky a Odpovědi* [online]. eM Client, 2022 [cit. 2022-05-18]. Dostupné z: https://cz.emclient.com/otazky-a-odpovedi-zaciname

[22] *Calendar and tasks* [online]. eM Client, 2022 [cit. 2022-05-17]. Dostupné z: https://www.emclient.com/features-calendar

[23] *Email Encryption and Digital Signature* [online]. eM Client, 2022 [cit. 2022-05-17]. Dostupné z: https://www.emclient.com/email-encryption

[24] *Pricing* [online]. eM Client, 2022 [cit. 2022-05-19]. Dostupné z: https://www.emclient.com/pricing

[25] *EM Client 9 - The next version of eM Client is here!* [online]. eM Client, 2022 [cit. 2022-05-20]. Dostupné z: https://www.emclient.com/blog/em-client-9---the-next-version-of-em-client-is-here-485

[26] *How to set up an Internet email account in Outlook 2013 or 2016* [online]. Microsoft, 2022 [cit. 2022-05-18]. Dostupné z: https://support.microsoft.com/en-us/topic/how-to-set-up-an-internet-email-account-in-outlook-2013-or-2016-3d900107-3c86-a326-6b8c-f214d10a6017

[27] *Add an email account to Outlook* [online]. Microsoft, 2022 [cit. 2022-05-20]. Dostupné z: https://support.microsoft.com/en-us/office/add-an-email-account-to-outlook-6e27792a-9267-4aa4-8bb6-c84ef146101b

[28] *Introduction to the Outlook Calendar* [online]. Microsoft, 2022 [cit. 2022-05-17]. Dostupné z: https://support.microsoft.com/en-us/office/introduction-to-the-outlook-calendar-d94c5203-77c7-48ec-90a5-2e2bc10bd6f8

[29] *Secure messages by using a digital signature* [online]. Microsoft, 2022 [cit. 2022-05-17]. Dostupné z: https://support.microsoft.com/en-us/office/secure-messages-by-using-a-digital-signature-549ca2f1-a68f-4366-85fa-b3f4b5856fc6

[30] *How to use PGP encryption with Outlook* [online]. Comparitech, 2018 [cit. 2022-05-17]. Dostupné z: https://www.comparitech.com/blog/information-security/pgp-encryption-with-outlook/

[31] *Upgrade to Outlook with Microsoft 365* [online]. Microsoft, 2022 [cit. 2022-05-19]. Dostupné z: https://www.microsoft.com/en-us/microsoft-365/outlook/outlook-personal-email-plans

[32] *Frequently Asked Questions* [online]. eM Client, 2022 [cit. 2022-05-20]. Dostupné z: https://www.emclient.com/faq-getting-started

[33] LOSHIN, Pete. *Essential email standards: RFCs and protocols made practical*. 1. vyd. New York: John Wiley, 1999. ISBN 978-0471345978.

[34] HUGHES, Lawrence. *Internet E-mail: protocols, standards, and implementation*. 1. vyd. Boston: Artech House, 1998. ISBN 978-0890069394.

[35] *Simple Mail Transfer Protocol* [online]. IETF, 2008 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc5321

[36] *Simple Mail Transfer Protocol* [online]. IETF, 1982 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc821

[37] *Internet Message Access Protocol (IMAP) - Version 4rev2* [online]. IETF, 2021 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc9051

[38] *Internet Message Format* [online]. IETF, 2001 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc2822

[39] *Registration of Mail and MIME Header Fields* [online]. IETF, 2005 [cit. 2022-05-22]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc4021

[40] *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* [online]. IETF, 1996 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc2045

[41] *Post Office Protocol - Version 3* [online]. IETF, 1996 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1939

[42] *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1* [online]. IETF, 2014 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc7208

[43] *SPF record: Protect your domain reputation and email delivery* [online]. ActiveCampaign, 2022 [cit. 2022-05-16]. Dostupné z: https://postmarkapp.com/guides/spf

[44] *DomainKeys Identified Mail (DKIM) Signatures* [online]. IETF, 2011 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc6376

[45] *Ověřování emailů pomocí DKIM - DomainKeys Identified Mail* [online]. Petr Bouška, 2020 [cit. 2022-05-16]. Dostupné z: https://www.samuraj-cz.com/clanek/overovani-emailu-pomoci-dkim-domainkeys-identified-mail/

[46] *DKIM: What is it and why is it important?* [online]. ActiveCampaign, 2022 [cit. 2022-05-16]. Dostupné z: https://postmarkapp.com/guides/dkim

[47] *What is Digital Signature? How does it Work?* [online]. Rapid Web Services, 2022 [cit. 2022-05-16]. Dostupné z: https://comodosslstore.com/blog/what-is-digital-signature-how-does-it-work.html

[48] *What are digital signatures?* [online]. DocuSign, 2022 [cit. 2022-05-16]. Dostupné z: https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq

[49] *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile* [online]. IETF, 2008 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc5280

[50] *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification* [online]. IETF, 2019 [cit. 2022-05-16]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc8551

[51] *What Is PGP And How Is It Used?* [online]. groovyPost, 2020 [cit. 2022-05-17]. Dostupné z: https://www.groovypost.com/howto/what-is-pgp-and-how-is-it-used/

[52] *A No Soliciting Simple Mail Transfer Protocol (SMTP) Service Extension* [online]. IETF, 2004 [cit. 2022-05-19]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc3865

[53] *DNS Blacklists and Whitelists* [online]. IETF, 2010 [cit. 2022-05-17]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc5782

[54] *Understanding DNSBL Filtering* [online]. The Spamhaus Project, 2022 [cit. 2022-05-17]. Dostupné z: https://www.spamhaus.org/whitepapers/dnsbl_function/

[55] *DNS PTR's as an AntiSpam mechanism* [online]. David E Lares S, 2021 [cit. 2022-05-19]. Dostupné z: https://medium.com/analytics-vidhya/dns-ptrs-as-an-anti-spam-mechanism-b62b858440f4

[56] *PTR Records and Why They Matter for Emails* [online]. Railsware Products Studio, 2019 [cit. 2022-05-19]. Dostupné z: https://mailtrap.io/blog/ptr-record/

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

IMAP	Internet Message Access Protocol
POP3	Post Office Protocol
SMTP	Simple Mail Transfer Protocol
IP	Internet Protocol
PTR	Pointer
SSL	Secure Sockets Layer
TLS	Transport Layer Security
STARTTLS	Opportunistic TLS
TCP	Transmission Control Protocol
<CRLF>	The characters carriage return and line feed
<SP>	The space character
MUA	Mail User Agent
MTA	Mail Transfer Agent
DNS	Domain Name Systém
MIME	Multipurpose Internet Mail Extensions
PKI	Public Key Infrastructure
CA	Certificate Authority
PGP	Pretty Good Privacy
PICS	Protocol implementation conformance statement

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: OBSAH CD

- fulltext.pdf
- E-mail_client.zip