Bachelor Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Testing Tool for Generation and Verification of Emails Secured by S/MIME and PGP Standards

Kirill Khakimov

Supervisor: Ing. Tomáš Vaněk, Ph.D.
Study programme: Software Engineering and Technology
May 2022

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Khakimov Kirill**	Personal ID number:	**495548**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Computer Science**		
Study program:	**Software Engineering and Technology**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Tool for automated generation and verification of emails secured by S/MIME and PGP protocols

Bachelor’s thesis title in Czech:

Nástroj pro generování a ověřování zabezpečených emailů ve standardech S/MIME a PGP

Guidelines:

Bibliography / sources:

RFC 4880 - https://datatracker.ietf.org/doc/html/rfc4880
RFC 3156 - https://datatracker.ietf.org/doc/html/rfc3156
RFC 8551 - https://datatracker.ietf.org/doc/html/rfc8551

Name and workplace of bachelor’s thesis supervisor:

Ing. Tomáš Vaněk, Ph.D. Department of Telecommunications Engineering FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **11.02.2022** Deadline for bachelor thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

Ing. Tomáš Vaněk, Ph.D. Supervisor’s signature	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt	Student’s signature

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Acknowledgements

I want to express my sincere gratitude and appreciation to my supervisor, Ing. Tomáš Vaněk, Ph.D., for his valuable guidance and feedback and also for his excellent lectures on cryptography and information security.

Declaration

I declare that I elaborated this thesis on my own and that I mentioned all the information sources that have been used in accordance with the Guideline for adhering to ethical principles in the course of elaborating an academic final thesis.

Prague, May 20, 2022

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.

V Praze, 20. května 2022

Abstract

This paper presents an overview of S/MIME and PGP, the two most prominent technologies in the field of email that provide cryptographic privacy and authentication for message exchange. The last part of the paper is dedicated to the implementation of educational portal utilizing S/MIME and PGP.

Keywords: email, security, authentication, encryption, s/mime, pgp, clients, mua

Supervisor: Ing. Tomáš Vaněk, Ph.D.
České vysoké učení technické v Praze,
Fakulta elektrotechnická,
B3-602,
Technická 2,
166 27 Praha 6

Abstrakt

Tento dokument představuje přehled S/MIME a PGP, dvou nejvýraznějších technologií v oblasti e-mailů, které zajišťují kryptografickou důvěrnost a autentizaci pro výměnu zpráv. Poslední část dokumentu je věnována implementaci výukového portálu využívajícího S/MIME a PGP.

Klíčová slova: email, bezpečnost, autentizace, šifrování, s/mime, pgp, klienty, mua

Překlad názvu: Nástroj pro Generování a Ověřování Zabezpečených Emailů ve Standardu S/MIME a PGP

Contents

Figures

Tables

Introduction

E-mail has been around for half a century, and during this period it has completely transformed our everyday lives. To a large extent, it has replaced traditional postal mail and several other messaging mediums and become an essential part of personal, as well as commercial and organisational communication.

With the ever-increasing reliance on e-mail, the privacy and authenticity of electronic correspondence have become increasingly relevant. This thesis focuses on modern ways of achieving message confidentiality and guaranteeing message authentication.

The goal of the thesis is to introduce two major technologies aimed to address this issue, Secure Multipurpose Internet Mail Extensions (S/MIME) and Pretty Good Privacy (PGP), describe their practical application in terms of digital signing and encryption, and finally analyse and summarise e-mail clients with regard to their support for mentioned technologies.

The primary motivation for examining the subject from a broader perspective is the need to have a better technical understanding of S/MIME and PGP for implementation of the tool that processes secured emails as a part of this bachelor thesis.

Chapter 1 starts with an overview of electronic mail messages underlying formats. The following sections of Chapter 1 include the detailed description of each security technology, S/MIME and PGP, along with their comparison. The support of e-mail clients, as the main method of sending and receiving e-mail, for S/MIME and PGP is examined in Chapter 2. Chapter 3 describes the tool that was developed while working on this bachelor thesis.

Chapter 1

Email digital signing and encryption

Electronic mail is the most widely used network-based service in practically all systems. People expect to send and receive emails to and from others connected to the Internet being anywhere in the world. As people's dependence on email increases, so does the demand for authentication and confidentiality services. Pretty Good Privacy (PGP) and S/MIME are two protocols that stand out as widely used techniques. Both of them are introduced and described in this chapter.

1.1 Email formats

To comprehend S/MIME and PGP protocols, we first must understand the process of creating non-protected messages in the MIME format and its underlying Internet Message Format. Therefore, this section begins with an overview of these two standards and then proceeds to a description of S/MIME and PGP.

1.1.1 Internet Message Format

The basic format of internet messages is defined in RFC 5322, which was published in 2008 and replaced the now obsolete RFC 2822 from 2001, and its predecessor RFC 822 published way back in 1982 [1].

According to the standard [1, Section 2.1], an "electronic mail" message comprises a header section and an optional body section. The header section contains a number of fields that are used to transfer technical details about the message. These fields provide information such as the name of the sender and the recipient of the message, the subject of the letter, the date it was sent, and other data. The text of the message itself is placed in the body section.

In terms of syntax, each header field is a line of characters that ends with `CRLF` and consists of a field name and a field body separated by a colon `":"` [1, Section 2.2]. Headers are followed by a message body, separated from them by a blank line. The specification does not set any strict rules for the body, so basically, it is just lines of text [1, Section 2.3].

The following example (Figure 1.1) demonstrates an RFC 5322 compliant email:

```
From: "David V. Henderson" <david.v.henderson@example.com>
To: "Paul W. Burgess" <paul.w.burgess@example.com>
Cc: "Kelli R. Eaton" <kelli.r.eaton@example.edu>
Date: Tue, 02 Nov 2021 08:20:15 +0100
Subject: An email that conforms RFC 5322

Hello! Here starts the actual body of an email, which
is just plain text. A blank line delimits it from
the header. Moreover, it can be multi-line as well.
```

Figure 1.1: Example RFC 5322 message structure

1.1.2 MIME

The original RFC 822 specification for electronic mail only allowed messages to be transmitted in the form of relatively short lines of flat ASCII data, limiting potential applications. Consequently, it was not possible to send messages with attachments or to use accented letters common in many European languages and other special characters for a long time.

To address these problems, the Internet message format has been extended with the Multipurpose Internet Mail Extensions (MIME) standard in 1992. The MIME specification, originally defined in RFC 1341[1], enabled the exchange of messages with more complex content than RFC 822 had allowed [2, Section 1], introducing support for:

- character encodings other than ASCII for plain-text message content,
- various textual and binary attachments (images, audios, videos, programs, etc.),
- multi-part message bodies, and
- header information in non-ASCII character encodings.

MIME introduces five new message header fields to describe the content within the message body: **MIME-Version**, **Content-Type**, **Content-Transfer-Encoding**, **Content-ID** and **Content-Description** [2, Section 1]. With an eye to the goals of this thesis, two of them, namely **Content-Type** and **Content-Transfer-Encoding**, are worth taking a look at. These are defined in MIME as follows:

[1]The latest version of MIME is currently defined in RFCs 2045 through 2049, as well as in several other updating papers.

- **Content-Type:** Indicates the type of media (text, audio, video, ...) contained in the message body. It gives additional details to the recipient's email program processing the content of the email, so that the data could be correctly represented to the end user [2, Section 5].

- **Content-Transfer-Encoding:** Specifies what kind of conversion mechanism was used to encode the body of the message with the intention of seamless transmission over mail transfer protocols [2, Section 6].

Content types in MIME

Altogether, there are available 7 various top-level content types, which are further divided into a total of 15 minor types (subtypes). Main types can also be segregated into two groups: discrete and composite. The former ones are relatively self-describing: **Text**, **Image**, **Audio**, **Video** and **Application**, while the latter ones are **Message** and **Multipart** [2, Section 5][3]. The **Message** type is just a representation of an encapsulated message, which means that a body is itself a whole message or a piece of a message of some type. The remaining media type called **Multipart** deserves a couple of paragraphs of its own.

The **Multipart** type, as the name implies, is used to represent a compound content made up of a number of separate parts. To transmit a multipart email, a boundary parameter (boundary value) is added to the Content-Type header field, which defines the sequence of characters utilized to clearly separate individual parts of the message. Two hyphens must be prepended to each boundary value, except the final one, that has two trailing hyphens in addition the leading ones. Optionally, each message part can have its own header section to characterize the nature of its content [3, Section 5.1].

RFC 2046 provides a simple example of a multipart email that contains two plain-text parts (Figure 1.2).

As the example above shows, **multipart/mixed** is one of four initial subtypes defined for multipart type. The other three are **multipart/parallel**, **multipart/alternative** and **multipart/digest** [3, Section 5.1].

Transfer encodings in MIME

Some communication protocols have certain limitations related to the format of the transmitted data. Binary data, for example, cannot be sent over 7-bit mail protocols, such as SMTP (without extensions) [4], which, in addition, restricts the length of lines to 1000 characters. Transfer encodings are an important concept featured in the MIME specification designed to solve this issue [2, Section 6].

There are [2, Section 6.2] six mechanism types available as Content-Transfer-Encoding field's value. Three of them (**7bit**, **8bit** and **binary**) only provide some clues about the nature of the data and indicate that original data were not transformed in any way. Non-standardized vendor-specific values,

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Date: Sun, 21 Mar 1993 23:56:48 -0800 (PST)
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple boundary"

This is the preamble.  It is to be ignored, though it
is a handy place for composition agents to include an
explanatory note to non-MIME conformant readers.

--simple boundary

This is implicitly typed plain US-ASCII text.
It does NOT end with a linebreak.
--simple boundary
Content-type: text/plain; charset=us-ascii

This is explicitly typed plain US-ASCII text.
It DOES end with a linebreak.

--simple boundary--

This is the epilogue.  It is also to be ignored.
```

Figure 1.2: Example MIME multipart message structure [3, Section 5.1.1]

collectively called **x-tokens**, must use `"X-"` prefix to emphasize their non-standard origin [2, Section 6.3]. Remaining two schemes actually perform data transformations:

- **quoted-printable** transfer encoding is designed for cases when ASCII data prevail in text, but it is also necessary to encode non-safe characters. Printable ASCII characters remain intact while remaining bytes are represented by hexadecimal digits of their numeric value [2, Section 6.7].
- **base64** encodes arbitrary binary data as ASCII characters by transforming them into a radix-64 representation [2, Section 6.8].

1.2 S/MIME

Secure/Multipurpose Internet Mail Extension (S/MIME) is a security extension to the MIME message format standard for sending and receiving encrypted and digitally signed messages. The most recent S/MIME version 4.0 is standardized by IETF (Internet Engineering Task Force) in RFC 8551 which is the principal document describing how to create and process S/MIME messages.

S/MIME provides four cryptographic security-related capabilities: *confidentiality* (by means of encryption), *integrity*, *authenticity and non-repudiation of origin* (via digital certificates). Additionally, it offers the message compression service [5].

1.2.1 Basics

The cryptographic techniques used in S/MIME make it an excellent example of a hybrid cryptosystem[2]. The specification allows securing a message with a signature, encryption or both. Figure 1.3 depicts a simplified S/MIME usage scenario combining encryption and signing when Alice wants to securely send a message to Bob in a way that guarantees all four security properties mentioned earlier [6].

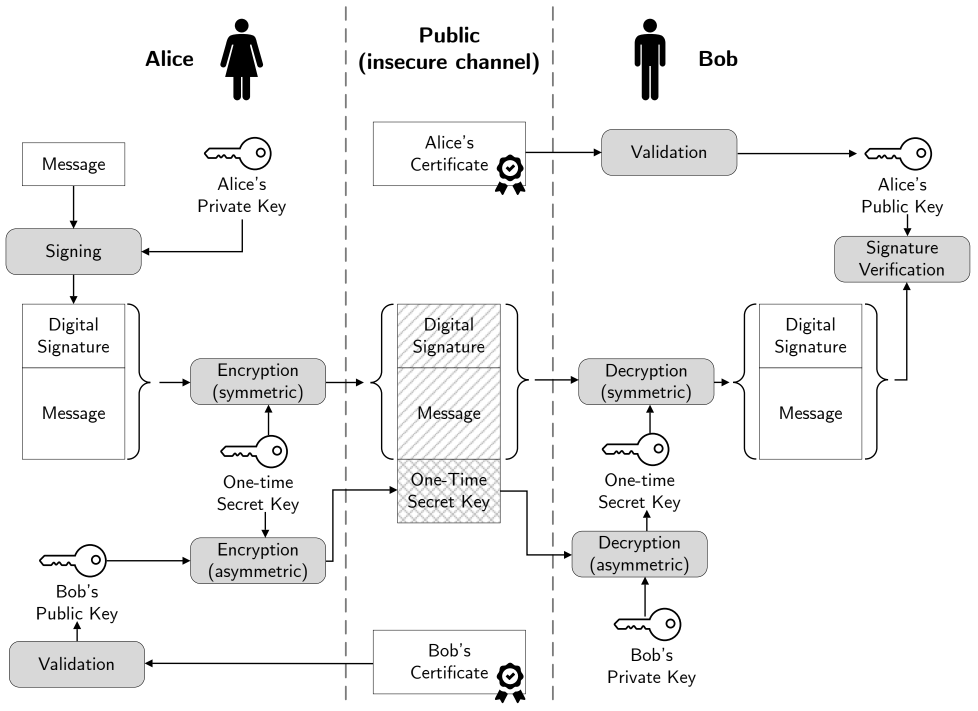


Figure 1.3: Simplified S/MIME functional flow diagram

Operations performed in Figure 1.3 can be split into eight steps, as follows:

1. Alice creates a message.

2. Alice's private key is used to create a digital signature for the message which is attached to the original message.

[2]Hybrid cryptosystem is a cryptographic system that combines the performance of symmetric algorithms with the advantages of public-key cryptography.

3. Alice generates a random one-time secret key to be used as a content-encryption key for this message only.
4. The message (including signature) is encrypted using symmetric-key algorithm with the one-time secret key.
5. The one-time secret key is encrypted with Bob's public key and is attached to the encrypted message.
6. Bob uses his private key to decrypt and recover the one-time secret key.
7. The one-time secret key is used to decrypt the message with the signature.
8. Bob verifies the signature using Alice's public key.

1.2.2 Content types

S/MIME [5, Section 2.4] makes use of the following message content types from the *Cryptographic Message Syntax (CMS)* standard which is defined in RFC 8933:

- **Data** type refers to the "inner" MIME message content which is subject to being secured with S/MIME. This content type is usually encapsulated in the other CMS types.
- **SignedData** must be used when user wants to apply a digital signature in order to provide *authentication*, *integrity* and *non-repudiation of origin*.
- **EnvelopedData** is used to ensure message *confidentiality* by encryption.
- **AuthEnvelopedData** is an additional CMS content type defined in RFC 5083 that provides message *confidentiality* and message *integrity*. It does not provide authentication or non-repudiation because no digital signatures are involved [5, Section 2.4.4][7].
- **CompressedData** is used to reduce the size of the data and doesn't protect message in any way.

One can also apply signing, encrypting or compressing operations at the same type and in any order. Each way of combining these operations has its benefits and drawbacks, and it is a user's choice how to perform it [5, Section 3.7].

1.2.3 Messages format

S/MIME is used to secure MIME entities composed of MIME message headers (not including the RFC 5322 headers) and a MIME body. A MIME entity can be an entire message, as well as one or more parts of it in the case of using the multipart content type. Constructed MIME entity in conjunction with other necessary security data (such as certificates and identifiers of algorithms) is processed by S/MIME and packed in a CMS object also known as *PKCS #7*. Then *Basic Encoding Rules (BER)*[3] format is used to represent *PKCS* object as binary data. After that, the appropriate MIME transfer encoding, which is base64 in most cases, is applied to the *BER-encoded PKCS* object. Finally, the base64-encoded data are treated as message content and wrapped in MIME message with *application/pkcs7-mime* subtype [5, Section 3.1][8]. An optional *smime-type* parameter conveys which S/MIME data type of those described in Section 1.2.2 was applied [5, Section 3.2.2].

A sample message of **EnvelopedData** type would be:

```
Content-Type: application/pkcs7-mime; name=smime.p7m;
   smime-type=enveloped-data
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m

MIIBHgYJKoZIhvcNAQcDoIIBDzCCAQsCAQAxgcAwgb0CAQAwJjASMRAwDgYDVQ
QDEwdDYXJsUlNBAhBGNGvHgABWvBHTbi7NXXHQMA0GCSqGSIb3DQEBAQUABIGA
C3EN5nGIiJi2lsGPcP2iJ97a4e8kbKQz36zg6Z2i0yx6zYC4mZ7mX7FBs3IWg+
f6KgCLx3M1eCbWx8+MDFbbpXadCDgO8/nUkUNYeNxJtuzubGgzoyEd8Ch4H/dd
9gdzTd+taTEgS0ipdSJuNnkVY4/M652jKKHRLFf02hosdR8wQwYJKoZIhvcNAQ
cBMBQGCCqGSIb3DQMHBAgtaMXpRwZRNYAgDsiSf8Z9P43LrY4OxUk660cu1lXe
CSFOSOpOJ7FuVyU=
```

Figure 1.4: S/MIME EnvelopedData sample message [5, Section 3.3]

1.2.4 Cryptographic algorithms

The S/MIME specification includes a list of several cryptographic algorithms for use. These are listed in Table 1.1. The terms MUST and SHOULD specifying the requirement level have a meaning in accordance with RFC 2119 [9]. S/MIME also includes a set of rules and recommendations for deciding which encryption algorithm should be used [5, Section 2].

1.3 PGP

An alternative to S/MIME is Pretty Good Privacy (PGP) protocol which provides almost the same functionality. The creator of PGP, Phil Zimmermann, released the first version of PGP as a program in 1991. It was available at no charge and was distributed with the complete source code, and quickly acquired a sizable following all around the world.

[3]Basic Encoding Rules (BER) is a format for encoding ASN.1 data structures encoding format. Abstract Syntax Notation One (ASN.1) is an abstract standard interface description language for defining data structures.

Operation	Requirement
Content digesting	MUST support: – SHA-256 – SHA-512
Signing[1]	MUST support: – ECDSA P-256 with SHA-256 – Ed25519 using PureEdDSA mode – RSA PKCS #1 v1.5 with SHA-256 SHOULD support: – RSASSA-PSS with SHA-256
Session-key encryption	MUST support: – ECDH ephemeral-static mode for P-256 – ECDH ephemeral-static mode for X25519[2] – RSA PKCS #1 v1.5 SHOULD support: – RSAES-OAEP
Content encryption	MUST support: – AES-128 GCM and AES-256 GCM – AES-128 CBC SHOULD support: – ChaCha20-Poly1305

[1] Sending agents as opposed to receiving agents MUST support either both ECDSA P-256 with SHA-256 and Ed25519 using PureEdDSA mode or at least one of them, in addition to other listed algorithms.
[2] Using HKDF-256 for key derivation function (KDF).

Table 1.1: Cryptographic algorithms for S/MIME [5, Section 2]

The initial protocol used patented cryptographic algorithms (RSA and IDEA at that moment) and was published in 1996 as informational RFC 1991. Thanks to Zimmermann's conviction of the need for PGP as an open standard, a specification called OpenPGP was developed by the forces of IETF in 1998 and defined in RFC 2440. Currently, OpenPGP Message Format is defined in RFC 4880.

Besides the original PGP implementation, which is now proprietary and owned by Symantec Corp., there is the GNU Privacy Guard (GnuPG, GPG) software, a free and open-source implementation of the OpenPGP standards compatible with PGP [10][11].

This section will focus on examining the MIME security with OpenPGP specification described in RFC 3156. Even though there is also RFC 2015 (MIME Security with PGP), I will generally refer to RFC 3156 as PGP/MIME within this thesis.

1.3.1 Basics

Principally PGP/MIME has almost the same set of capabilities as S/MIME. Basic functional flow is identical to the one described in Section 1.2.1. Yet, there are two considerable distinctions between S/MIME and PGP/MIME:

- **Key certification:** S/MIME uses X.509[4] certificates issued by Certificate Authority (CA), while OpenPGP is based on self-generated public and private keys. A user trusts a specific X.509 certificate if there is a chain of trust to a root certificate of CA. OpenPGP uses a concept called *Web of Trust*, where we trust some user's public key if it is signed by another user that we have designated as trusted.

- **Key distribution:** In contrast to S/MIME, OpenPGP does not automatically send a user's public key with every message. Recipients of PGP must then somehow get the sender's public key if they want to verify signed messages. This can be accomplished by using OpenPGP public key servers. One can also publish his public key on his own website if it ensures an end-to-end secure channel for the person willing to download it [12].

The following subsections take an overview of other minor differences from S/MIME.

1.3.2 Content types

Three new content types are used for PGP/MIME purposes [13]:

- "**application/pgp-encrypted**" content type supports *confidentiality* via encryption.

- "**application/pgp-signature**" ensures *authentication* and *integrity* via digital signatures.

- "**application/pgp-keys**" is used for public key distribution as described in [13, Section 7].

1.3.3 Messages format

Same as S/MIME, PGP/MIME operates on top of MIME messages. It also utilizes RFC 1847 which defines security multipart formats. Data to be protected is prepared and transformed into a MIME entity. The prepared MIME entity is then encrypted (or signed), and is encapsulated in security multipart message according to RFC 1847 using radix-64 encoding[5] [14, Section 2.4]. A required *protocol* parameter denotes whether data was encrypted or signed (see Section 1.3.2). The *micalg* parameter must (in case of signing) contain algorithm used to generate the signature [13].

Figure 1.5 illustrates an example of **application/pgp-encrypted** type PGP/MIME message:

[4] X.509 is an International Telecommunication Union (ITU) standard defining the format of public key certificates.

[5] Radix-64 is base64 encoding with a 24-bit checksum at the end [14, Section 6].

```
Content-Type: multipart/encrypted; boundary=foo;
   protocol="application/pgp-encrypted"

--foo
Content-Type: application/pgp-encrypted

Version: 1

--foo
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----
Version: 2.6.2

hIwDY32hYGCE8MkBA/wOu7d45aUxF4Q0RKJprD3v5Z9K1YcRJ2fve87lMlDlx4
OjeW4GDdBfLbJE7VUpp13N19GL8e/AqbyyjHH4aS0YoTk10QQ9nnRvjY8nZL3M
PXSZg9VGQxFeGqzykzmykU6A26MSMexR4ApeeON6xzZWfo+0yOqAq6lb46wsvl
dZ96YAAABH78hyX7YX4uT1tNCWEIIBoqqvCeIMpp7UQ2IzBrXg6GtukS8Nxbuk
LeamqVW31yt21DYOjuLzcMNe/JNsD9vDVCvOOG3OCi8=
=zzaA
-----END PGP MESSAGE-----

--foo--
```

Figure 1.5: PGP/MIME encrypted message example [13, Section 4]

1.3.4 Cryptographic algorithms

Underlying OpenPGP requirements for algorithms greatly differ from those in S/MIME as Table 1.2 shows. Implementations may, however, implement any other algorithm.

Operation	**Requirement**
Content digesting	MUST support: – SHA-1 MAY support: – RIPE-MD/160 – SHA256, SHA384, SHA512, SHA224
Signing	MUST support: – DSA SHOULD support: – RSA
Session-key encryption	MUST support: – Elgamal SHOULD support: – RSA
Content encryption	MUST support: – TripleDES (DES-EDE, 168 bit key) SHOULD support: – AES-128 – CAST-128 (CAST5) – IDEA[1] MAY support: – Blowfish (128 bit key, 16 rounds) – AES-192, AES-256 – Twofish with 256-bit key

[1] If interoperability with PGP 2.6 or earlier is required.

Table 1.2: Cryptographic algorithms for OpenPGP [14, Section 6]

Chapter 2

S/MIME and PGP support in applications

Both technologies S/MIME and PGP are reliant on the implementation provided by the email clients. As will be seen in the next sections, most Mail User Agents (MUAs) available at the moment (but not all of them) offer support for these technologies.

This chapter summarises popular actively maintained MUAs available on different devices and platforms with a focus on support for the security mechanisms mentioned earlier. Provided tables and data are intended to be illustrative rather than exhaustive.

2.1 Desktop email clients

One would argue that desktop email clients should be considered old-fashioned in the days of rapid popularity growth of web-based applications and webmail providers. However, despite the potential benefits web applications can provide (such as ubiquitous accessibility, for example), native clients have their own crucial advantages, particularly when it comes to security-related features.

The commonly used mail user agents on three major desktop platforms and their support for S/MIME and PGP are given below in Table 2.1 (for Windows), Table 2.2 (for macOS) and Table 2.3 (for Linux). Following paragraphs provide more detailed information about some of them.

Thunderbird, a free, cross-platform and one of the most popular email programs, has been offering integrated support for S/MIME for a long time. OpenPGP encryption and signing, however, is available only since version 78.2.1. Prior to that version users had to install extension such as *Enigmail* [15]. Since OpenPGP functionality became a part of Thunderbird core, the author of *Enigmail* decided to discontinue the development of the add-on for Thunderbird, [40] and users are encouraged to migrate to newer versions of the Thunderbird to have a built-in OpenPGP support.

Another free and open-source MUA available on all three operating systems with integrated support for both S/MIME and PGP is **Claws Mail**. But like many of Claws Mail's advanced features, PGP is added via plug-ins. The official PGP (*PGP/Core* and *PGP/Mime*) and *S/MIME* plugins are developed by The Claws Mail Team and can be freely installed to take

MUA	Distribution		Support	
	Free	Open-Source	S/MIME	PGP
Thunderbird [15]	✓	✓	✓	✓
Microsoft Outlook [16][17]	✗	✗	✓	✗[1]
Claws Mail [18]	✓	✓	✓	✓
eM Client [19]	✓[2]	✗	✓	✓
The Bat! [20]	✗	✗	✓	✓
Postbox [21][22]	✗	✗	✓	✗[1]
Mail (Windows Mail) [23]	✓	✗	✗[3]	✗
Mailbird [24]	✗	✗	✓	✓
Mailspring [25]	✓	✓	✗	✗
Bluemail [26]	✓	✗	✗	✗

[1] Available via third-party plugin.
[2] The free version has some limitations.
[3] Available for Exchange ActiveSync (EAS) accounts.

Table 2.1: Windows native MUAs

MUA	Distribution		Support	
	Free	Open-Source	S/MIME	PGP
Thunderbird [15]	✓	✓	✓	✓
New Outlook for Mac [27]	✗	✗	✓	✗
Claws Mail [18]	✓	✓	✓	✓
eM Client [28]	✓[1]	✗	✗	✓
Postbox [21][22]	✗	✗	✓	✗[2]
Mail (Apple Mail) [29]	✓	✗	✓	✗[2]
Airmail [30]	✓	✗	✗	✗
MailMate [31]	✗	✗	✓	✓
Canary Mail [32]	✗	✗	✗	✓
Mailspring [25]	✓	✓	✗	✗
Bluemail [26]	✓	✗	✗	✗

[1] The free version has some limitations.
[2] Available via third-party plugin.

Table 2.2: macOS native MUAs

advantage of benefits of confidential communication and emails validation.

An email client developed by Microsoft, **Microsoft Outlook**, that is available on Windows, has support for S/MIME [17][16]. The Outlook version for macOS called **New Outlook for Mac** had also received S/MIME support in 2021 [27]. Despite the long history of development (since the '90s) still neither version of Outlook can encrypt or sign message using PGP. However, there are several Windows-only extensions (both free and paid), that have the required functionality, for example: *GpgOL*, *gpg4o* or *Encryptomatic OpenPGP for Outlook*. The disadvantage of using such extensions is degraded user experience related to integration issues because of proprietary nature of Outlook.

MUA	Distribution		Support	
	Free	Open-Source	S/MIME	PGP
Thunderbird [15]	✓	✓	✓	✓
Claws Mail [18]	✓	✓	✓	✓
KMail [33]	✓	✓	✓	✓
Evolution [34]	✓	✓	✓	✓
Geary [35]	✓	✓	✗	✗
Astroid [36]	✓	✓	✗	✓
Mailspring [25]	✓	✓	✗	✗
Bluemail [26]	✓	✗	✗	✗
Mutt / NeoMutt [37]	✓	✓	✓	✓
Alpine / Pine [38]	✓	✓	✓	✗[1]
mu / mu4e / gnus [39]	✓	✓	✓	✓

[1] Requires *ez-pine-gpg* integration scripts.

Table 2.3: Linux native MUAs

eM Client built for Windows and macOS has S/MIME standard implemented for several years while the support of PGP was added in 2020. Version for macOS was released relatively recently, in early 2019, and still lacks support for S/MIME, but not for PGP.

The Ritlabs, SRL company is positioning its app **The Bat!** as "the best secure email client software" for security-conscious users, so it is not odd that S/MIME and PGP are among the many features available. The Bat! is offered as shareware and is compatible with Microsoft Windows systems only.

Pre-installed MUAs. Modern operating systems commonly have some default email client preinstalled. Vendors usually design built-in email management applications for a broad audience of operation systems; hence they are not usually feature-rich. They have a few options and limited functional capabilities available to users.

Microsoft corporation includes a software called **Mail** (formerly **Windows Mail**) in the basic set of applications that comes with the Windows OS. It has no PGP capability, and no general S/MIME support unless a user plans to send and receive messages using a proprietary Exchange ActiveSync (EAS) protocol.

Unlike Microsoft's Mail, **Apple's Mail** for Mac surprisingly supports Secure/MIME out of the box. To handle PGP, one nevertheless would need to install the *GPG Mail* (a part of *GPG Suite* package), which is a paid add-on for Apple Mail.

Various Linux distributions usually have built-in email clients depending on the desktop environment they use, for example: KDE has **KMail**, whereas GNOME has **Evolution** and **Geary**.

Terminal-based apps. Unix-like operating systems are characterized by extensive use of command-line applications. Although nowadays terminal apps have a few key advantages over GUI's, we can still find several email clients with a text-based user interface. These include: **Mutt / NeoMutt**, **Pine / Alpine**, **mu / mu4e / gnus** (as extensions to the Emacs editor). Most of them support both technologies.

2.2 Web email clients

Webmail is a web application that allows users to access their email through a browser and thus is accessible from everywhere. Unlike desktop email applications, webmail software is usually deployed on a web server and may be either self-hosted or externally-hosted. Table 2.4 shows an overview of webmail clients.

MUA	**Distribution**		**Support**	
	Free	**Open-Source**	**S/MIME**	**PGP**
Horde IMP [41]	✓	✓	✓	✓
RoundCube [42]	✓	✓	✗	✓
SquirrelMail [43]	✓	✓	✗	✗
Mailpile [44]	✓	✓	✗	✓
Rainloop [45]	✓	✓	✗	✓

Table 2.4: Webmail clients

2.3 Webmail providers

A *webmail service provider* is a provider of email service that can be accessed using a standard web browser. Although many providers these days allow access to messages through desktop email clients using IMAP and POP protocols, a significant number of people continue to use services through web browsers thanks to the accessibility of web applications.

Webmail providers generally have no support for digital signatures and end-to-end encryption because an attempt to use server-side signing or encryption of messages would compromise users' secret keys. In order to prevent providers from accessing private keys, a number of browser extensions were developed which allow PGP encryption and signing on the client side:

- **Mailvelope** add-on available for Firefox and Chromium-based browsers adds OpenPGP features to the user interface of common webmail providers. Users have also an option to manually add additional email providers.
- **Flowcrypt** extension for Firefox and Chromium-based browsers integrates with *Gmail* to offer OpenPGP capabilities.
- **Psono** offers a PGP integration with *Gmail*, *Outlook on the Web* and *Yahoo! Mail*.

A small part of providers do not require the installation of additional browser plugins; instead, they directly provide OpenPGP support through JavaScript. These include **Protonmail**, **Mailfence**, and **Startmail**, among others. While they are more straightforward to use, some people argue that they do not provide end-to-end security [46].

Chapter 3
Implementation

In order to demonstrate the process of encryption, decryption, signing and verification of electronic messages using two previously described methods in practice and to contribute to the process of the education of students at the Faculty of Electrical Engineering at Czech Technical University, it was decided to redesign and extend the functionality of an existing learning portal for generating and evaluating assignments in the field of secure email communication that is used within Cryptography and Information Security course.

3.1 Existing portal and problems

The original learning portal app was developed back in 2016. The purpose of creating the portal was to find a way to automatically verify students' practical knowledge in the field of electronic correspondence protection and cryptography in general.

Generally, the portal is a system for generating tasks and verifying their solutions. The interaction process of students with the portal is as follows:

1. Student sends a digitally signed email.
2. Portal verifies whether the message's signature is valid.
3. Portal, in case of successful signature validation, generates a number of email responses, each of which contains an encrypted short text string. The email is then signed with the server's digital signature; however, all signatures but one are intentionally invalidated. All created emails are then sent back to the student.
4. Student, having received replies from the server, must determine which email completely satisfies the conditions of "correctly signed message".
5. Student deciphers a message that he thinks is correct.
6. Student then uses the web part of the portal, where he specifies the string he obtained after the decryption process.

7. Portal verifies submitted string, and if the student correctly picked a message from the set received from the portal and correctly deciphered it, the portal reports the correctness of the solution.

To provide an overview of students' solutions, the portal also has a simple administrator panel, where information about the student is presented:

1. Email address of student,
2. Date and time when the solution was submitted,
3. Whether it was correct,
4. Content of the string submitted by the student in case of a incorrect solution.

Initially, the portal was created solely to verify one specific assignment described above. It wasn't designed as a modular or potentially extensible solution for future growth, and partly for this reason, no new features have been added to the project since then. The learning portal no longer meets the needs of the Cryptography and Information Security course. It was decided that the entire project should be rewritten from scratch using modern frameworks because implementing the requirements accumulated over several years within the framework of an already existing project would be quite unwise and problematic.

3.2 Alien Mail

The project developed as a part of the presented thesis was named **'AlienMail'**, where the first part of the name refers to the original name of the portal **'Alien'**.

The structure of the entire solution is quite complex and is therefore divided into three sub-projects:

- **AlienMail.Service:** Core part
- **AlienMail.Web:** Web part
- **AlienMail.Tool:** Helper tool

The following subsections describe each of the sub-projects in more detail.

3.2.1 AlienMail.Service

The **AlienMail.Service** sub-project is the central part and is the core of the entire portal. Its main tasks include:

- Connecting to the mail server to keep track of incoming messages.

- Receiving a message from the mail server.
- Message processing: identifying the task, verifying the correctness of the submitted solution, saving the results, and composing the answer.
- Connecting to a mail server to send a reply message.

3.2.2 AlienMail.Web

User web interface is implemented in **AlienMail.Web** sub-project. As the requirements of the project imply, web interface is divided into two parts: a part for students and the administrator panel.

Students' part consists of two pages:

- **Tasks Overview:** Page with a table of all assignments available to the student: assignment name, assignment description, status of assignment, link to student's submissions for a certain assignment (see Figure 3.1).

- **Submission Overview**: Page where all (successful and failed) submissions of student for a certain tasks are listed. In addition to the informational data such as date of submission user can also see here causes of unsuccessful submission attempts (see Figure 3.2).

The panel for administrator also consists of two pages:

- **Students Overview:** Page with a list of all students, where an administrator can have a grasp of tasks completion status for a certain student and go to student's submission page (see Figure 3.3).

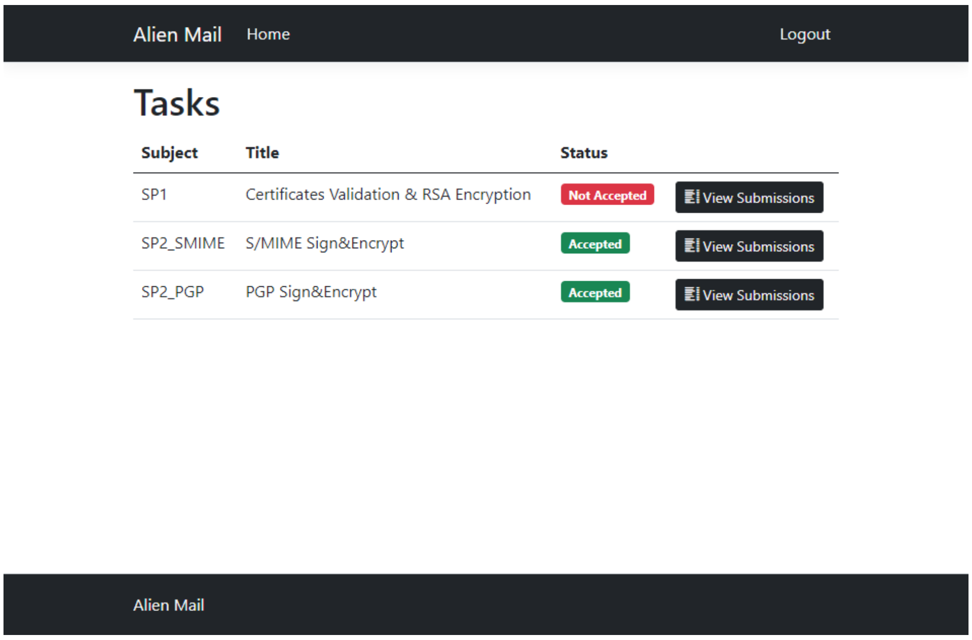
- **Submission Overview**: Page is almost identical to the student's 'Submission Overview' page, except for the option to download original message sent by student (see Figure 3.4).

3.2.3 AlienMail.Tool

AlienMail.Tool is auxiliary console utility tool that should be used by administrator to import new S/MIME, PGP and CA certificates into the project's certificate repository.

A set of different import commands is available to administrator:

- `smime import crl`: Imports a CRL (Certificate Revocation List),
- `smime import pem`: Imports a non-encrypted PEM (aka base64-encoded DER) certificate,
- `smime import pkcs12`: Imports a PKCS12-encoded certificate,
- `pgp import public`: Imports a PGP public keyring,
- `pgp import secret`: Imports a PGP secret keyring.



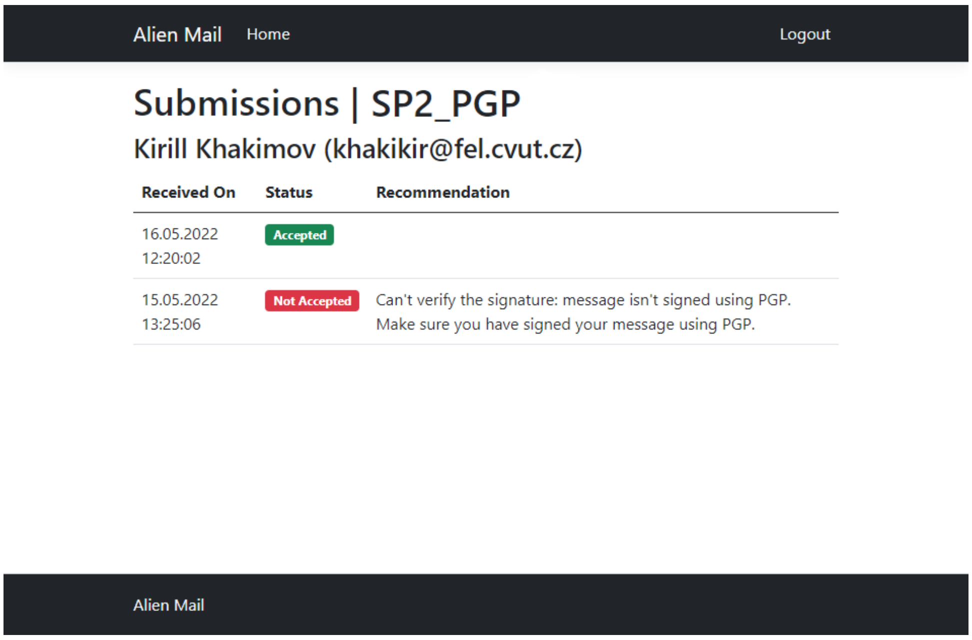
Alien Mail Home Logout

Tasks

Subject	Title	Status	
SP1	Certificates Validation & RSA Encryption	Not Accepted	View Submissions
SP2_SMIME	S/MIME Sign&Encrypt	Accepted	View Submissions
SP2_PGP	PGP Sign&Encrypt	Accepted	View Submissions

Alien Mail

Figure 3.1: Student's Tasks Overview Page



Alien Mail Home Logout

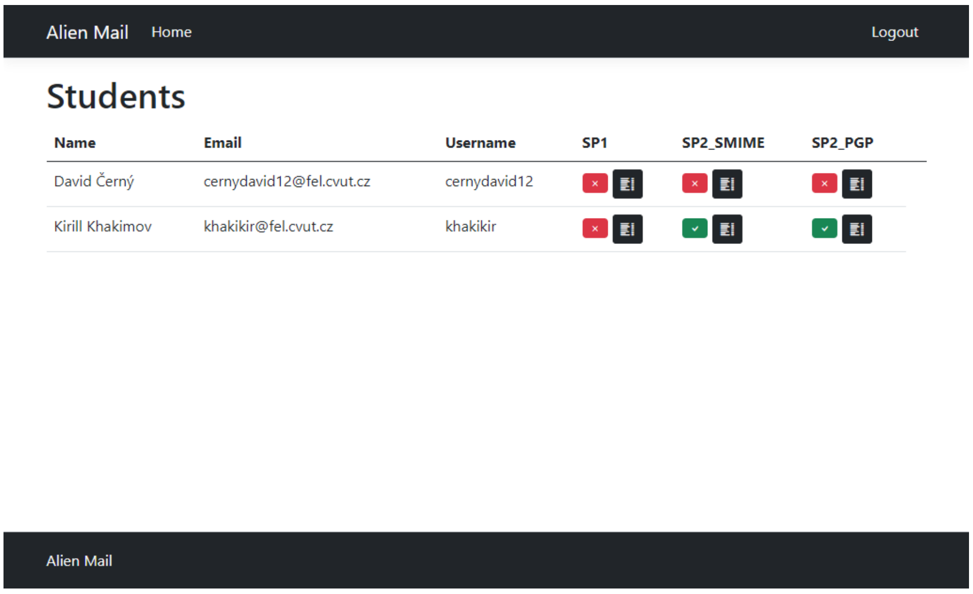
Submissions | SP2_PGP

Kirill Khakimov (khakikir@fel.cvut.cz)

Received On	Status	Recommendation
16.05.2022 12:20:02	Accepted	
15.05.2022 13:25:06	Not Accepted	Can't verify the signature: message isn't signed using PGP. Make sure you have signed your message using PGP.

Alien Mail

Figure 3.2: Student's Submission Overview Page



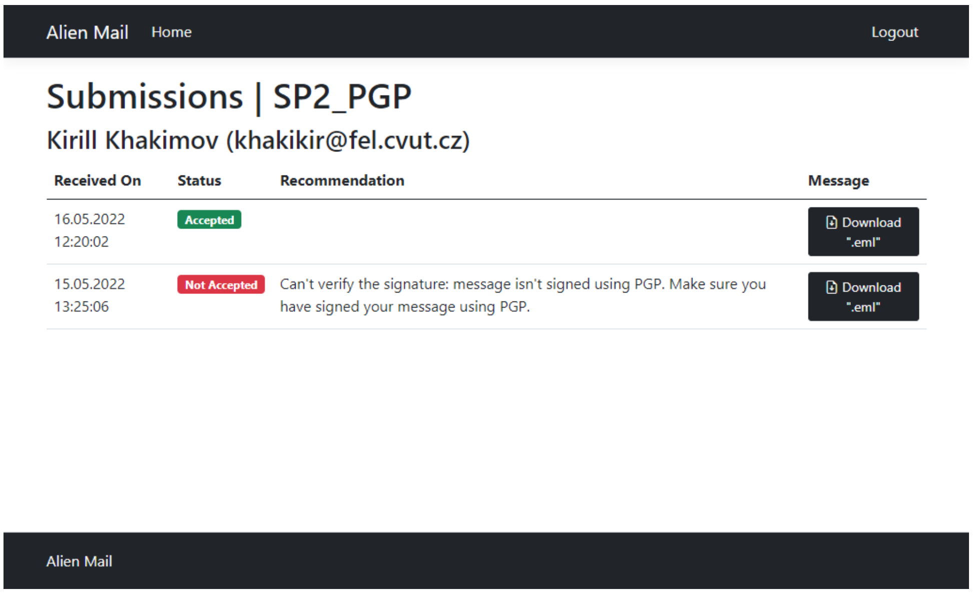
Alien Mail Home Logout

Students

Name	Email	Username	SP1	SP2_SMIME	SP2_PGP
David Černý	cernydavid12@fel.cvut.cz	cernydavid12	✗	✗	✗
Kirill Khakimov	khakikir@fel.cvut.cz	khakikir	✗	✓	✓

Alien Mail

Figure 3.3: Administrator's Students Overview Page



Alien Mail Home Logout

Submissions | SP2_PGP

Kirill Khakimov (khakikir@fel.cvut.cz)

Received On	Status	Recommendation	Message
16.05.2022 12:20:02	Accepted		Download ".eml"
15.05.2022 13:25:06	Not Accepted	Can't verify the signature: message isn't signed using PGP. Make sure you have signed your message using PGP.	Download ".eml"

Alien Mail

Figure 3.4: Administrator's Submission Overview Page

Conclusion

The first goal was to learn about techniques used for securing email communication, namely S/MIME and PGP protocols. Chapter 1 contains the theory and provides an overview of formats used in email, starting from the most basic one up to security-enhanced (PGP and S/MIME), and also describes their functionalities and capabilities. Sections 1.2.3 and 1.3.3 elaborate upon message formats and illustrate (with the examples) how secured messages can be constructed, achieving the next goal. The final goal was to study mail user agents available on different platforms on the subject of S/MIME and PGP support. It was accomplished in Chapter 2. The knowledge gained was used for the last goal which was the implementation of educational portal utilizing PGP and S/MIME and it was eventually described in Chapter 3.

Technically both S/MIME and PGP schemes provide a high level of security in an email exchange in terms of message confidentiality, integrity and authenticity. Nevertheless, it becomes clear that despite the availability of such technologies, their presence in implementations leaves much to be desired. Moreover, some examples indicate that the effectiveness of the security mechanisms highly depends on the correct use.

Bibliography

[1] RESNICK, Pete. *Internet Message Format* [RFC 5322]. RFC Editor, 2008. Request for Comments, no. 5322. Available from DOI: `10.17487/RFC5322`.

[2] FREED, Ned; BORENSTEIN, Nathaniel S. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* [RFC 2045]. RFC Editor, 1996. Request for Comments, no. 2045. Available from DOI: `10.17487/RFC2045`.

[3] FREED, Ned; BORENSTEIN, Nathaniel S. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types* [RFC 2046]. RFC Editor, 1996. Request for Comments, no. 2046. Available from DOI: `10.17487/RFC2046`.

[4] KLENSIN, John C. *Simple Mail Transfer Protocol* [RFC 5321]. RFC Editor, 2008. Request for Comments, no. 5321. Available from DOI: `10.17487/RFC5321`.

[5] SCHAAD, Jim; RAMSDELL, Blake C.; TURNER, Sean. *Secure / Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification* [RFC 8551]. RFC Editor, 2019. Request for Comments, no. 8551. Available from DOI: `10.17487/RFC8551`.

[6] DE CLERCQ, Jan. *Windows Server 2003 Security Infrastructures: Core Security Features*. 1st ed. Elsevier, 2004. ISBN 9780080521121.

[7] HOUSLEY, Russ. *Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type* [RFC 5083]. RFC Editor, 2007. Request for Comments, no. 5083. Available from DOI: `10.17487/RFC5083`.

[8] HOUSLEY, Russ. *Cryptographic Message Syntax (CMS)* [RFC 5652]. RFC Editor, 2009. Request for Comments, no. 5652. Available from DOI: `10.17487/RFC5652`.

[9] BRADNER, Scott O. *Key words for use in RFCs to Indicate Requirement Levels* [RFC 2119]. RFC Editor, 1997. Request for Comments, no. 2119. Available from DOI: `10.17487/RFC2119`.

[10] LUCAS, Michael. *PGP & GPG: Email for the Practical Paranoid*. No Starch Press, 2006. ISBN 9781593270711.

[11] THE GNUPG PROJECT. *GnuPG: Using the GNU Privacy Guard*. Samurai Media Limited, 2015. ISBN 9789888381159.

[12] STALLINGS, William. *Cryptography and Network Security: Principles and Practice*. 7th ed. Pearson Education, 2016. ISBN 9781292158587.

[13] ROESSLER, Thomas; ELKINS, Michael; LEVIEN, Raph; TORTO, Dave Del. *MIME Security with OpenPGP* [RFC 3156]. RFC Editor, 2001. Request for Comments, no. 3156. Available from DOI: `10.17487/RFC3156`.

[14] FINNEY, Hal; DONNERHACKE, Lutz; CALLAS, Jon; THAYER, Rodney L.; SHAW, David. *OpenPGP Message Format* [RFC 4880]. RFC Editor, 2007. Request for Comments, no. 4880. Available from DOI: `10.17487/RFC4880`.

[15] SIPES, Ryan. *OpenPGP in Thunderbird 78* [online]. The Thunderbird Blog, 2020-09-07 [visited on 2021-12-31]. Available from: `https://blog.thunderbird.net/2020/09/openpgp-in-thunderbird-78/`.

[16] MICROSOFT CORP. *Encrypt email messages* [online]. Microsoft Office Support, 2021-11-23 [visited on 2022-01-11]. Available from: `https://support.microsoft.com/en-us/office/encrypt-email-messages-373339cb-bf1a-4509-b296-802a39d801dc`.

[17] MICROSOFT CORP. *Secure messages by using a digital signature* [online]. Microsoft Office Support, 2021-03-08 [visited on 2022-01-11]. Available from: `https://support.microsoft.com/en-us/office/secure-messages-by-using-a-digital-signature-549ca2f1-a68f-4366-85fa-b3f4b5856fc6`.

[18] THE CLAWS MAIL TEAM. *Claws Mail Features* [online]. Claws Mail, 2021-04-24 [visited on 2022-01-11]. Available from: `https://www.claws-mail.org/features.php`.

[19] EM CLIENT S.R.O. *Email Encryption and Digital Signature* [online]. eM Client [visited on 2022-01-11]. Available from: `https://www.emclient.com/email-encryption`.

[20] RITLABS, SRL. *The Bat! - Secure Desktop Email Client for Windows 10* [online]. The Bat! [Visited on 2022-01-11]. Available from: `https://www.ritlabs.com/en/products/thebat/`.

[21] POSTBOX, INC. *Obtaining an S/MIME Certificate to Sign Emails* [online]. Postbox Support, 2015-10-21 [visited on 2022-01-11]. Available from: `https://support.postbox-inc.com/hc/en-us/articles/202200540-Obtaining-an-S-MIME-Certificate-to-Sign-Emails`.

[22] POSTBOX, INC. *How to use OpenPGP to Encrypt Messages* [online]. Postbox Blog, 2021-01-14 [visited on 2022-01-11]. Available from: `https://www.postbox-inc.com/blog/entry/how-to-use-openpgp-to-encrypt-messages`.

[23] MICROSOFT CORP. *Configure S/MIME for Windows* [online]. Microsoft Documentation, 2021-03-12 [visited on 2022-01-11]. Available from: `https://docs.microsoft.com/en-us/windows/security/identity-protection/configure-s-mime`.

[24] MAILBIRD, INC. *Mailbird | Email made easy & beautiful* [online]. Mailbird [visited on 2022-01-11]. Available from: `https://www.getmailbird.com/`.

[25] FOUNDRY 376. *Mailspring - The best free email app* [online]. Mailspring [visited on 2022-01-11]. Available from: `https://getmailspring.com/`.

[26] BLIX INC. *BlueMail - The Best Email Management App for Windows, Mac, Linux, Android, and iOS* [online]. BlueMail, 2022 [visited on 2022-01-11]. Available from: `https://bluemail.me/`.

[27] JEELANI, Faisal. *Ever evolving, the new Outlook for Mac adds more.* [Online]. Outlook Blog, 2021-04-24 [visited on 2022-01-11]. Available from: `https://techcommunity.microsoft.com/t5/outlook-blog/ever-evolving-the-new-outlook-for-mac-adds-more/ba-p/2196988`.

[28] EM CLIENT S.R.O. *eM Client for Mac is officially here!* [Online]. eM Client Blog, 2019-01-23 [visited on 2022-01-11]. Available from: `https://emclient.com/blog/em-client-for-mac-is-officially-here-295`.

[29] APPLE INC. *Sign or encrypt emails in Mail on Mac* [online]. Mail User Guide [visited on 2022-01-11]. Available from: `https://support.postbox-inc.com/hc/en-us/articles/202200540-Obtaining-an-S-MIME-Certificate-to-Sign-Emails`.

[30] BLOOP S.R.L. *Airmail - Lightning Fast Mail Client for Mac and iOS* [online]. Airmail [visited on 2022-01-11]. Available from: `https://airmailapp.com/`.

[31] FRERON SOFTWARE. *MailMate IMAP email client for macOS* [online]. MailMate [visited on 2022-01-11]. Available from: `https://freron.com/`.

[32] MAILR TECH LLP. *Canary Mail | Best Email App for Apple iPhone, iPad & Mac* [online]. Canary Mail [visited on 2022-01-11]. Available from: `https://canarymail.io/`.

[33] KDE E.V. *KMail* [online]. Kontact Suite, 2022-01-06 [visited on 2022-01-11]. Available from: `https://kontact.kde.org/components/kmail.html`.

[34] THE GNOME PROJECT. *Mail encryption and certificates* [online]. GNOME Help [visited on 2022-01-11]. Available from: `https://kontact.kde.org/components/kmail.html`.

[35] THE GNOME PROJECT. *Geary - send and receive email* [online]. GNOME Wiki [visited on 2022-01-11]. Available from: `https://wiki.gnome.org/Apps/Geary`.

[36] ASTROIDMAIL. *Astroid: A graphical threads-with-tags style, lightweight and fast, e-mail client for Notmuch* [online]. GitHub [visited on 2022-01-11]. Available from: `https://github.com/astroidmail/astroid`.

[37] BLOSSER, Jeremy; ELKINS, Michael R. *The Mutt E-Mail Client* [online]. Mutt, 2021-12-30 [visited on 2022-01-11]. Available from: `http://www.mutt.org/`.

[38] CHAPPA, Eduardo. *Alpine Email Program* [online]. Alpine Home Page, 2021-09-18 [visited on 2022-01-11]. Available from: `https://alpine.x10host.com/`.

[39] FREE SOFTWARE FOUNDATION, INC. *Emacs: Message - Signing and encrypting commands* [online]. A GNU Manual [visited on 2022-01-11]. Available from: `https://www.gnu.org/software/emacs/manual/html_node/message/Signing-and-encryption.html`.

[40] BRUNSCHWIG, Patrick. *Future OpenPGP Support in Thunderbird* [online]. Enigmal News, 2020-09-07 [visited on 2022-01-11]. Available from: `https://enigmail.net/index.php/en/home/news/70-2019-10-08-future-openpgp-support-in-thunderbird`.

[41] HORDE LLC. *IMP - Internet Messaging Program* [online]. The Horde Project [visited on 2022-01-11]. Available from: `https://www.horde.org/apps/imp/`.

[42] BRÜDERLI, Thomas. *About the Roundcube webmail project* [online]. Roundcube [visited on 2022-01-11]. Available from: `https://roundcube.net/about/#features`.

[43] THE SQUIRRELMAIL TEAM. *SquirrelMail - Webmail for Nuts!* [Online]. SquirrelMail [visited on 2022-01-11]. Available from: `https://squirrelmail.org/`.

[44] MAILPILE TEAM. *Subject: mailpile is an e-mail client* [online]. Mailpile [visited on 2022-01-11]. Available from: `https://www.mailpile.is/`.

[45] RAINLOOP. *Features* [online]. Rainloop Home Page, 2013-12-30 [visited on 2022-01-05]. Available from: `https://www.rainloop.net/features/`.

[46] ARCIERI, Tony. *What's wrong with in-browser cryptography?* [Online]. Tony Arcieri's Blog, 2013-12-30 [visited on 2022-01-05]. Available from: `https://tonyarcieri.com/whats-wrong-with-webcrypto`.