# E-MAIL SECURITY

Leonardo Querzoni
querzoni@diag.uniroma1.it


SAPIENZA
UNIVERSITÀ DI ROMA

CIS SAPIENZA
CYBER INTELLIGENCE AND INFORMATION SECURITY

# OVERVIEW

The Internet e-mail system

- Architecture and basic functioning
- SMTP, POP, IMAP
- extensions (MIME)
- Email threats
- Infrastructure security: SPF, DKIM, ARC, DMARC
- End-to-end security: PGP, S/MIME

# THE E-MAIL SYSTEM

E-mail is a method of exchanging digital messages from an author to one or more recipients, operating across the Internet/intranet

Among the oldest services available on the internet!

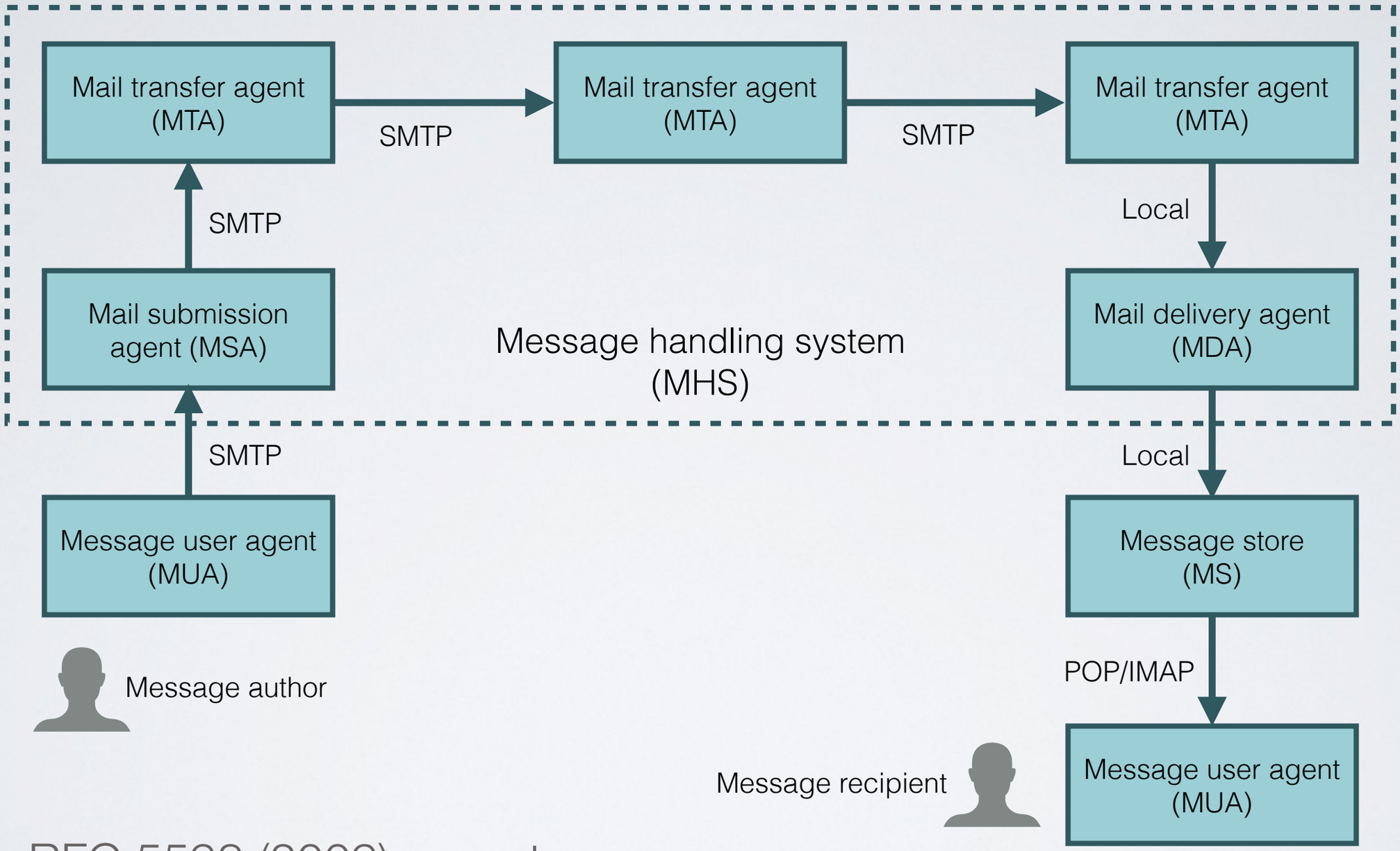
- is around 50 years old! It began in 1971 when Ray Tomlinson, a computer engineer, sent the first electronic message across the ARPANET, the precursor to the Internet, addressing the recipient with the “user@domain” scheme

Modern e-mail systems are based on a **store-and-forward model**: e-mail servers accept, forward, store and deliver messages

- neither the users nor their computers are required to be online simultaneously

Check RFC 5598

# INTERNET E-MAIL ARCHITECTURE




RFC 5598 (2009) + errata

# MUA, MSA, MTA

## Mail User Agent (MUA)

- used to access and manage a user's e-mail

## Mail Submission Agent (MSA)

- receives e-mail messages from a MUA and cooperates with a mail transfer agent (MTA) for delivery of the mail
- it makes sure the message meets the standard format requirements

## Mail Transfer Agent (MTA)

- transfers e-mail messages from one computer to another using a client–server application architecture
- MTAs implements both the client and server portions of the Simple Mail Transfer Protocol

# MDA, MRA

Mail Delivery Agent (MDA)

- responsible for the delivery of e-mail messages to a local recipient's mailbox
- local message delivery is achieved through a process of handling messages from the MTA, and storing mail into the recipient's environment (typically a mailbox)

# E-MAIL EXCHANGE

E-mail transmission across IP networks is carried by the **Simple Mail Transfer Protocol** (**SMTP**, RFC 821, 1982)

- last update: RFC 5321 (2008). Includes the extended SMTP (ESMTP) additions

SMTP communicates delivery parameters using a message envelope separate from the message (header and body) itself

An Internet e-mail address is a string of the form *user@domain*

- the part before the @ sign is the local part of the address
- the part after the @ sign is a fully qualified domain name

# MESSAGE FORMAT

The **Internet Message Format (IMF)** is defined by RFC 5322

- support to MIME (RFC 2045 through RFC 2049), collectively called Multipurpose Internet Mail Extensions

Internet e-mail messages consist of two major sections:

- **Header** — Structured into fields such as From, To, CC, Subject, Date, and other information about the email.
- **Body** — The basic content, as unstructured text; sometimes containing a signature block at the end. This is exactly the same as the body of a regular letter.

The header is separated from the body by a blank line

# HEADER

Each message has exactly one header, which is structured into fields. Each field has a name and a value. RFC 5322 specifies the precise syntax

- Informally, each line of text in the header that begins with a printable character begins a separate field and its name starts in the first character of the line and ends before the separator character ":"
- The separator is then followed by the field value. The value is continued onto subsequent lines if those lines have a space or tab as their first character. **Field names and values are restricted to 7-bit ASCII characters**. Non-ASCII values may be represented using MIME encoded words

Email header fields can be multi-line, and each line must be at most 76 characters long.

# HEADER

Each message is identified through two kinds of IDs

**Message-ID:**

- Pertains to content and is globally unique.
- Its format is similar to that of a mailbox, with two parts separated by @
  - The right side specifies the domain or host that assigns the identifier
  - The left side contains a string that is globally opaque and serves to uniquely identify the message within the domain referenced on the right side.
- Has a variety of uses including threading, aiding identification of duplicates, and DSN (Delivery Status Notification) tracking.
- The MSA assigns the Message-ID:
- Example: Message-ID: <20241001102233.12345@example.com>

# HEADER

Each message is identified through two kinds of IDs

**ENVID**

- Stands for “envelope identifier”
- Used for message-tracking purposes ([RFC3885], [RFC3464]) concerning a single posting/delivery transfer.
- ENVID is used for one message posting until that message is delivered. A re-posting of the message, such as by a MTA, does not reuse that ENVID.

# HEADER

The header contain the following mandatory fields:

- From - Sender's email address of the sender.
- To - Email  recipients. Multiple email addresses can be included.
- Date - The date and time when the message was sent. It follows a specific format (RFC 5322).
- Message-ID
- Subject - A brief summary of the content of the email.

Note that the To: field is not necessarily related to the addresses to which the message is delivered. The actual delivery list is supplied separately to SMTP, which may or may not originally have been extracted from the header content. In the same way, the "From:" field does not have to be the real sender of the email message.

# HEADER

Other headers are optional:

- Cc (Carbon Copy) - Additional recipients who will receive a copy of the email.
- Bcc (Blind Carbon Copy) - Same as Cc, but these recipients are hidden.
- Reply-To - Specifies the email address to which replies should be sent if it's different from the "From" address.
- In-Reply-To - Contains the Message-ID of the email to which this message is a response. This is used for threading in email clients.

# HEADER

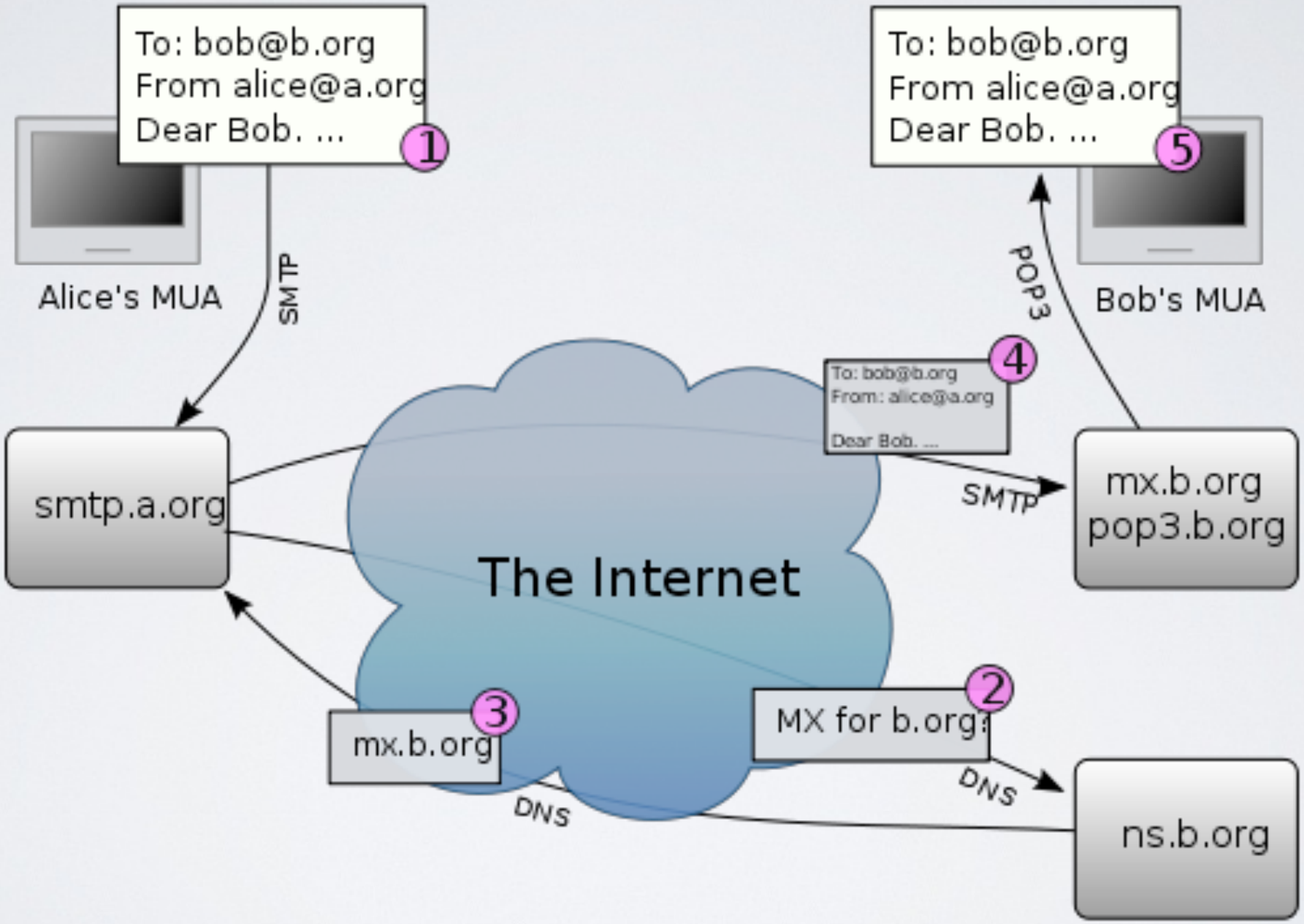
Other headers are optional:

- References - Lists Message-IDs of previous emails in the thread to maintain conversation history.
- Sender - Used when the person sending the email is different from the one listed in the "From" field.
- Return-Path - Specifies the address that will receive bounce-back messages if the email cannot be delivered.
- Received - This field is added by each mail server that processes the message, tracking the route of the email. Each hop adds a new "Received" field.
  - Example: `Received: from mail.example.com by smtp.example.org with ESMTP; Fri, 1 Oct 2024 10:22:33 -0400`

# E-MAIL EXCHANGE

- MUAs (client mail applications) only use SMTP for sending messages to a mail server for relaying
- To access their mail box accounts, MUAs usually use:
  - the **Post Office Protocol (POP)** for mail downloading and offline usage
  - Internet **Message Access Protocol (IMAP)** for online mail reading
  - Proprietary protocols (such as in Microsoft Exchange or Lotus Notes/Domino)

# OPERATION OVERVIEW



# OPERATION OVERVIEW

1. Alice composes a message using her MUA; she enters the e-mail address of her correspondent, and hits the "send" button
2. The MUA formats the message in email format and uses the Submission Protocol (variant of SMTP, see RFC 6409) to send the message to the local MSA
3. The MSA looks at the destination address provided in the SMTP protocol and resolves a domain name to determine the fully qualified domain name of the mail exchange server
4. the DNS server for the b.org domain responds with any MX records listing the mail exchange servers for that domain, in this case mx.b.org, a MTA server run by Bob's ISP

# OPERATION OVERVIEW

5. smtp.a.org sends the message to mx.b.org using SMTP
   - this server may need to forward the message to other MTAs before the message reaches the final message delivery agent (MDA), which delivers it to the mailbox of Bob
6. Bob presses the "get mail" button in his MUA, which picks up the message using either the Post Office Protocol (POP3) or the Internet Message Access Protocol (IMAP4)

# SAMPLE SMTP INTERACTION

| Party | SMTP commands and status codes | Explanation |
|---|---|---|
| Server: | 220 smtp.example.com ESMTP Postfix | After the connection has been established, the SMTP server answers |
| Client: | HELO relay.example.com | The SMTP client logs on with its hostname |
| Server: | 250 smtp.example.com, hello | The server confirms the login |
| Client: | MAIL FROM:<john@doe.com> | The client specifies the sender address of the MUA |
| Server: | 250 OK | The server confirms |
| Client: | RCPT TO:<boss@workplace.com> | The client specifies the recipient address |
| Server: | 250 OK | The server confirms |
| Client: | DATA | The client initiates the transmission of the e-mail |
| Server: | 354 End data with <CR><LF>.<CR><LF> | The server begins the reception and indicates that the e-mail text should be closed with a dot (".") |
| Client: | From: "John Doe" <john@doe.com><br>To: Boss Workplace <boss@workplace.com> | The client transmits the e-mail text, highlights it with a line |

as of RCF 821

# SAMPLE SMTP INTERACTION

| | | |
|---|---|---|
| Client: | DATA | The client initiates the transmission of the e-mail |
| Server: | 354 End data with <CR><LF>.<CR><LF> | The server begins the reception and indicates that the e-mail text should be closed with a dot (".") |
| Client: | From: "John Doe" <john@doe.com><br>To: Boss Workplace <boss@workplace.com><br>Date: Monday, March 12 2018 10:03:42<br>Subject: Sick note<br><br>Hello boss,<br>Unfortunately, I am sick today and cannot come into work. Thank you for your understanding,<br>John Doe | The client transmits the e-mail text, highlights it with a line break after "Subject: Sick note" and ends it with the desired dot |
| Server | 250 OK: queued as 15432 | The server confirms it has successfully received the e-mail and puts it in a queue |
| Client: | QUIT | The client signals the end of the session |
| Server: | 221 Goodbye | The server terminates the connection |

as of RCF 821

# ESMTP

RFC 821 (1982) was obsoleted by RFC 5321 (2008), where ESMTP, an extended version of SMPT, was introduced.

Software agents should stick to ESMTP but for backward compatibility reason a client connecting by SMTP will be also served

The greeting command for ESMTP is EHLO, which gets a (possibly multiline) response listing the supported extended commands

# OTHER ESMTP COMMANDS

- 8BITMIME — 8 bit data transmission, RFC 6152
- ATRN — Authenticated TURN for On-Demand Mail Relay, RFC 2645
- AUTH — Authenticated SMTP, RFC 4954
- CHUNKING — Chunking, RFC 3030
- DSN — Delivery status notification, RFC 3461 (See Variable envelope return path)
- ETRN — Extended version of remote message queue starting command TURN, RFC 1985
- HELP — Supply helpful information, RFC 821
- PIPELINING — Command pipelining, RFC 2920
- SIZE — Message size declaration, RFC 1870
- STARTTLS — Transport layer security, RFC 3207 (2002)
- SMTPUTF8 — Allow UTF-8 encoding in mailbox names and header fields, RFC 6531
- UTF8SMTP — Allow UTF-8 encoding in mailbox names and header fields, RFC 5336 (deprecated)

# MIME (MULTIPURPOSE INTERNET MAIL EXTENSIONS)

Internet standard that extends the format of email to support:

- Text in character sets other than ASCII
- Non-text attachments
- Message bodies with multiple parts
- Header information in non-ASCII character sets

MIME's use has grown beyond describing the content of email to describe content type in general (web, storage)

Virtually all human-written Internet email and a fairly large proportion of automated email is transmitted via SMTP in MIME format

MIME is specified in six linked RFC memoranda

# MIME

Important RFCs

- RFC-822 Standard for the format for ARPA Internet text messages
- RFC-2045 MIME Part 1: Format of Internet Message Bodies
- RFC-2046 MIME Part 2: Media Types
- RFC-2047 MIME Part 3: Message Header Extensions
- RFC-2048 MIME Part 4: Registration Procedure
- RFC-2049 MIME Part 5: Conformance Criteria

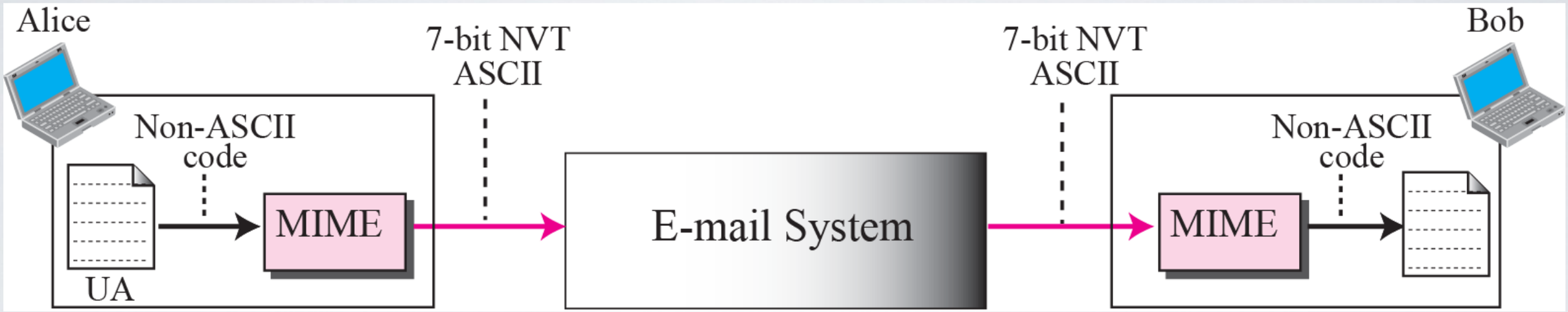
# MIME – WHAT IS IT?

- MIME  refers to an official Internet standard that specifies how messages must be formatted so that they can be exchanged between different email systems.
- MIME  permits the inclusion of virtually any type of file or document in an email message.
- Specifically, MIME messages can contain
  - text
  - images
  - audio
  - video
  - application-specific data
    - spreadsheets
    - word processing documents

# MIME FEATURES

- Support of character sets other than ASCII
- Content type labeling system
- Support of non-text content in e-mail messages
- Support for compound documents

# MIME SCHEME




NVT = network virtual terminal

# MIME HEADERS

MIME headers

| E-mail header |
| --- |
| MIME-Version: 1.1<br>Content-Type: type/subtype<br>Content-Transfer-Encoding: encoding type<br>Content-Id: message id<br>Content-Description: textual explanation of nontextual contents |
| E-mail body |

# NON-ASCII CHARACTER SET SUPPORT

## Message header

- content-type field
  - put in the header by the client program creating the e-mail for use by the client program used to display the received message
  - charset= optional parameter
    - if absent ASCII is assumed

Content-Type: text/plain; charset="ISO-8859-1"

- ISO-8859-1 extends the basic character set of ASCII to include many of the accented characters used in languages such as Spanish, French, German and Italian.
- US-ASCII is the standard character set used in the US

# CONTENT LABELING

A set of registered MIME Types that map to specific file types

- MIME Types consist of :
  - a primary type
  - a sub type separated by a /  ( as text/html)

Common Mime Types:

| FileExtension | MIME Type | Description |
|---|---|---|
| .txt | text/plain | Plain text |
| .htm | text/html | Styled text in HTML format |
| .jpg | image/jpeg | Picture in JPEG format |
| .gif | image/gif | Picture in GIF format |
| .wav | audio/x-wave | Sound in WAVE format |
| .mp3 | audio/mpeg | Music in MP3 format |
| .mpg | video/mpeg | Video in MPEG format |
| .zip | application/zip | Compressed file in PK-ZIP format |

# MIME TYPES/SUBTYPES

| *Type* | *Subtype* | *Description* |
|---|---|---|
| Text | Plain | Unformatted |
| | HTML | HTML format (see Appendix E) |
| Multipart | Mixed | Body contains ordered parts of different data types |
| | Parallel | Same as above, but no order |
| | Digest | Similar to Mixed, but the default is message/RFC822 |
| | Alternative | Parts are different versions of the same message |
| Message | RFC822 | Body is an encapsulated message |
| | Partial | Body is a fragment of a bigger message |
| | External-Body | Body is a reference to another message |
| Image | JPEG | Image is in JPEG format |
| | GIF | Image is in GIF format |
| Video | MPEG | Video is in MPEG format |
| Audio | Basic | Single channel encoding of voice at 8 KHz |
| Application | PostScript | Adobe PostScript |
| | Octet-stream | General binary data (eight-bit bytes) |

# CONTENT-TRANSFER-ENCODING

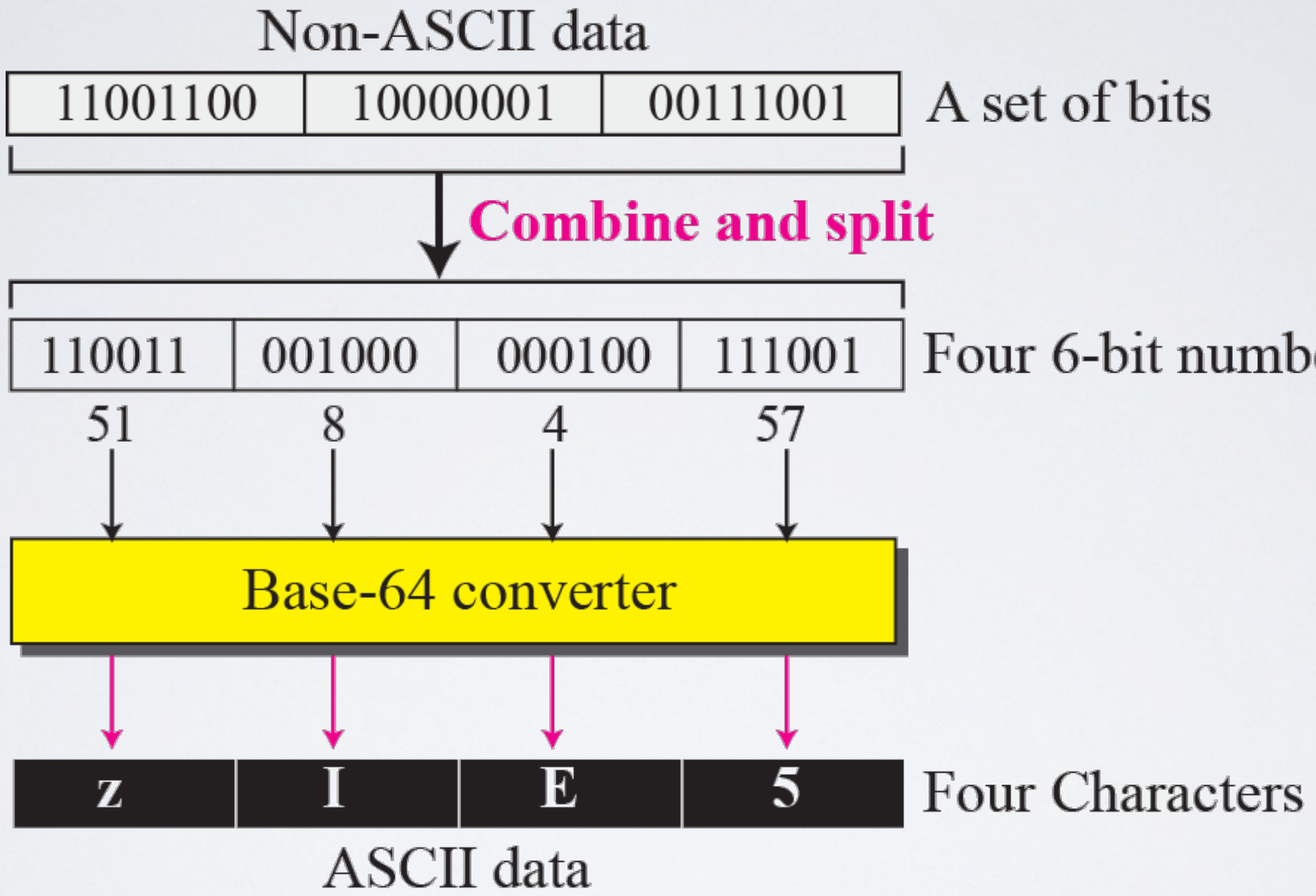
| *Type* | *Description* |
|---|---|
| 7bit | NVT ASCII characters and short lines |
| 8bit | Non-ASCII characters and short lines |
| Binary | Non-ASCII characters with unlimited-length lines |
| Base64 | 6-bit blocks of data are encoded into 8-bit ASCII characters |
| Quoted-printable | Non-ASCII characters are encoded as an equal sign plus an ASCII code |

# NON-TEXT CONTENT

Non-textual content is encoded in ASCII for transmission and decoded back to its original format for display upon receipt

- MIME uses base 64 encoding (RFC 2045)
  - binary to text encoding scheme
  - targets A-Z, a-z, 0-9, +,/
- scheme:
  - take 3 bytes of data, put into a 24 bit buffer
  - extract 4 six-bits values
  - use each value as an index into:
    ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
  - this yields 4 ASCII characters
  - use zero, one or two = symbols for padding (at the end)

# BASE-64 ENCODING SCHEME

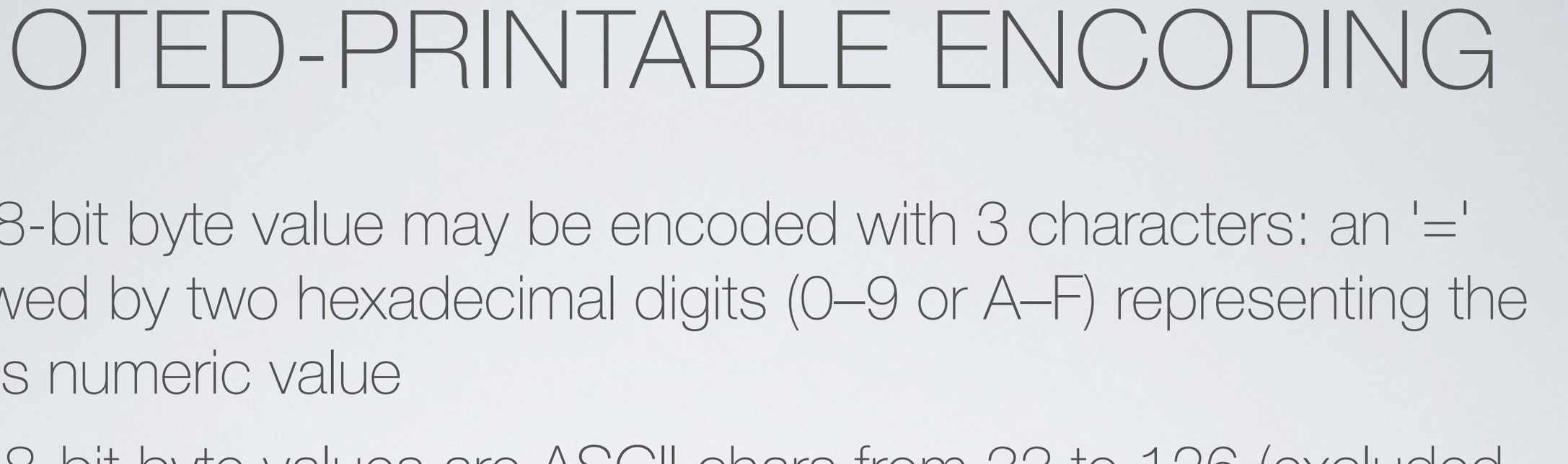


# BASE-64 CONVERTING TABLE

| *Value* | *Code* | *Value* | *Code* | *Value* | *Code* | *Value* | *Code* | *Value* | *Code* | *Value* | *Code* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **A** | 11 | **L** | 22 | **W** | 33 | **h** | 44 | **s** | 55 | **3** |
| 1 | **B** | 12 | **M** | 23 | **X** | 34 | **i** | 45 | **t** | 56 | **4** |
| 2 | **C** | 13 | **N** | 24 | **Y** | 35 | **j** | 46 | **u** | 57 | **5** |
| 3 | **D** | 14 | **O** | 25 | **Z** | 36 | **k** | 47 | **v** | 58 | **6** |
| 4 | **E** | 15 | **P** | 26 | **a** | 37 | **l** | 48 | **w** | 59 | **7** |
| 5 | **F** | 16 | **Q** | 27 | **b** | 38 | **m** | 49 | **x** | 60 | **8** |
| 6 | **G** | 17 | **R** | 28 | **c** | 39 | **n** | 50 | **y** | 61 | **9** |
| 7 | **H** | 18 | **S** | 29 | **d** | 40 | **o** | 51 | **z** | 62 | + |
| 8 | **I** | 19 | **T** | 30 | **e** | 41 | **p** | 52 | **0** | 63 | / |
| 9 | **J** | 20 | **U** | 31 | **f** | 42 | **q** | 53 | **1** | | |
| 10 | **K** | 21 | **V** | 32 | **g** | 43 | **r** | 54 | **2** | | |

# QUOTED-PRINTABLE ENCODING

- Any 8-bit byte value may be encoded with 3 characters: an '=' followed by two hexadecimal digits (0–9 or A–F) representing the byte's numeric value
- Non 8-bit byte values are ASCII chars from 33 to 126 (excluded 61, the '=' sign)
- special cases for SPACE and TAB



# MULTIPART SUBTYPES

- Mixed - For sending files with different "Content-Type" headers.
- Digest - To send multiple text messages.
- Message - Contains any MIME email message, including any headers
- Alternative - Each part is an "alternative" version of the same (or similar) content (e.g., text + HTML)

- more subtypes…

# MIME TYPES/SUBTYPES

```
From: Some One <someone@example.com>
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="XXXXboundary text"

This is a multipart message in MIME format.

--XXXXboundary text
Content-Type: text/plain

this is the body text

--XXXXboundary text
Content-Type: text/plain;
Content-Disposition: attachment;
        filename="test.txt"

this is the attachment text

--XXXXboundary text--
```

# MIME TYPES/SUBTYPES

```
MIME-Version: 1.0
X-Mailer: MailBee.NET 8.0.4.428
Subject: test subject
To: kevinm@datamotion.com
Content-Type: multipart/mixed;
      boundary="XXXXboundary text"

--XXXXboundary text
Content-Type: multipart/alternative;
      boundary="XXXXboundary text"

--XXXXboundary text
Content-Type: text/plain;
      charset="utf-8"
Content-Transfer-Encoding: quoted-printable

This is the body text of a sample message.
--XXXXboundary text
Content-Type: text/html;
      charset="utf-8"
Content-Transfer-Encoding: quoted-printable
<pre>This is the body text of a sample message.</pre>

--XXXXboundary text
Content-Type: text/plain;
name="log_attachment.txt"
Content-Disposition: attachment;
filename="log_attachment.txt"
Content-Transfer-Encoding: base64

TUlNRS1WZXJzaW9uOiAxLjANClgtTWFpbGVyOiBNYWlsQmVlLk5FVCA4LjAuNC4OMjgNClN1Ymp1
```

# PLAIN TEXT AND HTML

- modern graphic email clients allow use of HTML for the body
  - HTML email messages often include a plain text copy as well
- HTML messages should have an additional header: "Content-type: text/html". Most email programs insert this header automatically
- advantages of HTML include the ability to include in-line links and images, etc.
- disadvantages include the increased size of the email

# SUBADDRESSING

Local subaddressing (or “+ subaddressing”):

- Provides support for “tags” in the local part of the email address
- Tags are defined following a separator character that is oftentimes “+”
- The address is an alias of a mailbox defined by the prefix preceding the separator.
  - querzoni+mastercourses@diag.uniroma1.it => querzoni@diag.uniroma1.it
- RFC5233

Domain subaddressing

- The local name can be used as domain subaddress
- Local address can be defined at will
  - mastercourses@querzoni.diag.uniroma1.it => querzoni@diag.uniroma1.it

# EMAIL EXAMPLE

**Delivered-To**: querzoni@diag.uniroma1.it
**Received**: by 2002:a59:cb63:0:b0:49d:7dd7:7af1 with SMTP id c3csp711565vqv;
        Fri, 11 Oct 2024 15:05:46 -0700 (PDT)
**X-Google-Smtp-Source**: AGHT+IGIGOiU4RVgtBjHbzt4mT54JLoQYHfs5UosD6BSJBz6PgBjOqzYVEmmQFbBj
**Return-Path**: <crisis2024@easychair.org>
**Received**: from easychair.org (easychair.org. [213.136.76.235]) by mx.google.com with ESMTPS id 5b1f17b1804b1-43111acd015si33937595e9.60.2024.10.11.15.05.46 for <querzoni@diag.uniroma1.it> (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256); Fri, 11 Oct 2024 15:05:46 -0700
**Received**: from easychair.org (m5801.contaboserver.net [213.136.76.235]) by easychair.org (8.15.2/8.15.2/Debian-18) with ESMTP id 49BM5kCe2465090 for <querzoni@diag.uniroma1.it>; Sat, 12 Oct 2024 00:05:46 +0200
**Message-Id**: <202410112205.49BM5kCe2465090@easychair.org>
**Content-Type**: text/plain; charset="UTF-8"
**MIME-Version**: 1.0
**Date**: Sat, 12 Oct 2024 00:05:46 +0200
**From**: Crisis 2024 <crisis2024@easychair.org>
**To**: Leonardo Querzoni <querzoni@diag.uniroma1.it>
**Subject**: Instructions for Final Paper Submission - CRiSIS 2024
**Sender**: crisis2024@easychair.org

Dear Leonardo Querzoni,