TIN
Techniki Internetowe
lato 2018

Grzegorz Blinowski

Instytut Informatyki

Politechniki Warszawskiej

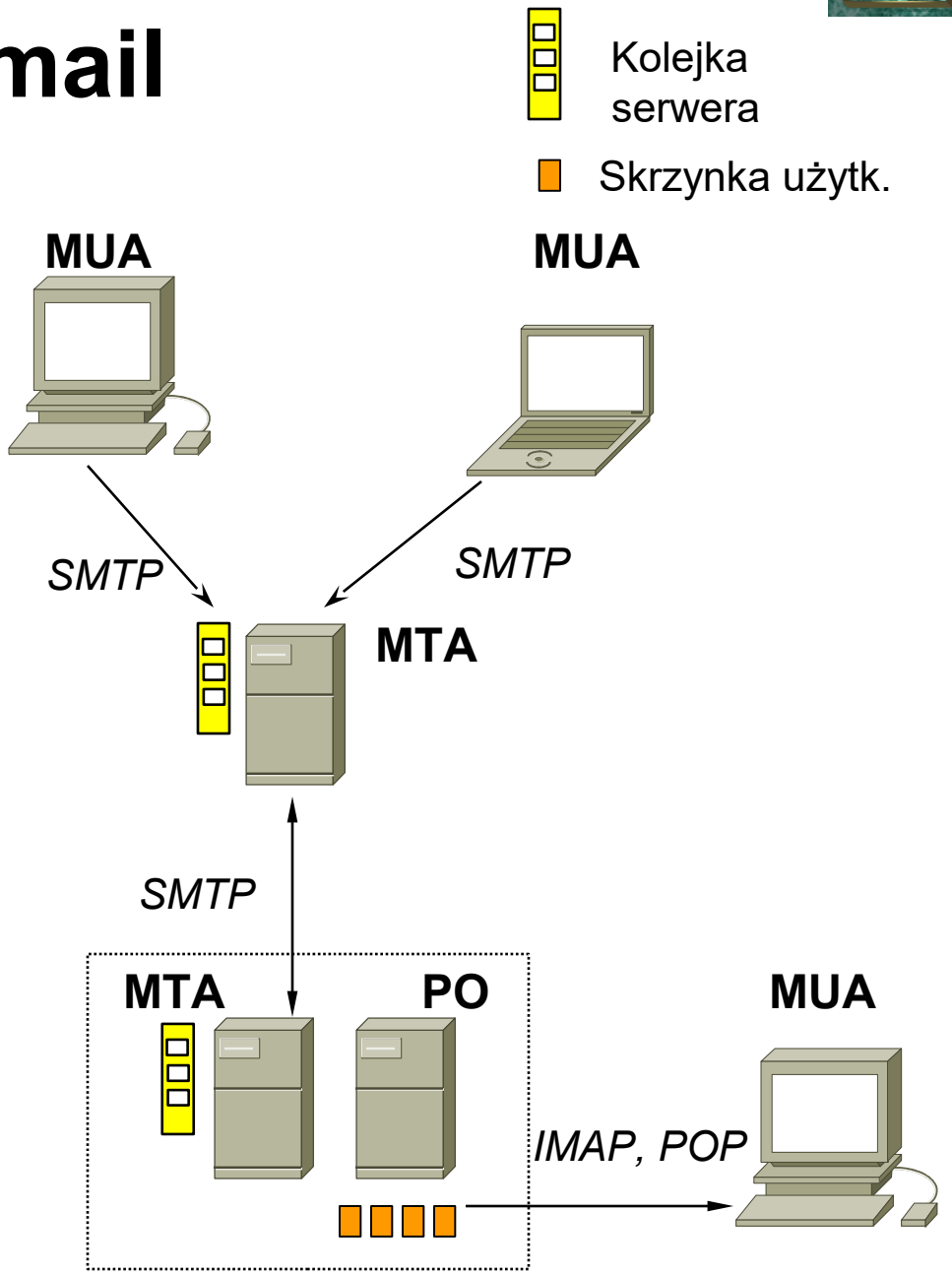
Plan wykładów

2	Intersieć, ISO/OSI, protokoły sieciowe, IP
3	Protokół IP i prot. transportowe: UDP, TCP
4	Model klient-serwer, techniki progr. serwisów
5	Protokoły aplikacyjne: telnet, ftp, smtp, nntp, inne
6	HTTP
7, 8	HTML, XML
9, 10, 11	Aplikacje WWW, CGI, sesje, serwery aplikacji serwlety, integracja z backended SQL
12	Aspekty zaawansowane: wydajność, przenośność, skalowalność; klastering
13	Inne: P2P, SOAP, RDF, WSDL, ontologie
14	Wstęp do zagadnień bezpieczeństwa (IPSec, VPN, systemy firewall) oraz aspekty kryptograficzne (DES, AES, RSA, PGP, S/MIME), tokeny i akceleracja sprzętowa

Poczta elektroniczna: SMTP

E-mail

- **MTA** - Mail Transfer Agent (serwer) - sendmail, postfix, Exchange, …
- **PO** - Post Office - przechowuje skrzynki, może ale nie musi być to ten sam serwer co MUA
- **MUA** - Mail User Agent (klient) - Mozilla, Outlook, Eudora, ...
- Protokół SMTP (**S**imple **M**ail **T**ransfer **P**rotocol) - RFC 821, 1982, szereg późniejszych rozszerzeń
- "maildrop" - POP3, IMAP4; inne - OWA



SMTP

RFC821

RFC5321

- Komunikacja klient do serwera; serwer do serwera
- **SMTP** obejmuje:
  - specyfikację protokołu wymiany danych
  - *uwierzytelnianie (opcja)*
  - adresowanie
  - ruting poczty
  - nie obejmuje: dostarczenia poczty do MUA, sposobu zapisu treści przesyłki
- Port TCP: **25**
- Komunikacja (ASCII):
  - nawiązanie połączenia (greeting)
  - wymiana danych
  - zakończenie sesji

Sesja SMTP

```
S: 220 This is XYZ smtp server at uczelnia.edu
C: HELO nikt.pl
S: 250  Hello nikt.pl, pleased to meet you
C: MAIL FROM: <ktos@nikt.pl>
S: 250 ktos@nikt.pl... Sender ok
C: RCPT TO: <dziekan@uczelnia.edu>
S: 250 dziekan@uczelnia.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Szanowny Panie
C:   Nasza firma oferuje rewelacyjny program do
C:   obslugi dziekanatow oraz calych uczelni ...
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 uczelnia.edu closing connection
```

Polecenia SMTP

- HELO
- EHLO (ESMTP)
  - najpopularniejsze rozszerzenia: pipelining, ...
- MAIL FROM
- RCPT TO
- VRFY, EXPN - zazwyczaj nieaktywne ze względów bezpieczeństwa
- DATA
- QUIT
- NOOP

Więcej o sesji i danych w SMTP

```
S: 220 This is XYZ smtp server at ...
...
C: MAIL FROM: <ktos@nikt.pl>
S: 250 ktos@nikt.pl... Sender ok
C: RCPT TO: <dziekan@uczelnia.edu>
S: 250 dziekan@uczelnia.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "."  ...
C: To: <dziekan@uczelnia.edu>
C: Subject: ciekawa oferta
C:
C: Szanowny Panie
C: Nasza firma oferuje rewelacyjny ...
C: ...
C: .
S: 250 Message accepted for delivery
C: QUIT
```

envelope

header

body

Więcej o sesji i danych w SMTP

- Pola nagłówka wiadomości (header - RFC822) są podobne i mogą (ale nie muszą) mieć zawartość zgodną z polami z envelope - nie jest to jednak to samo!
- Pola nagłówka takie jak np.: *Subject, To, CC* są przez SMTP traktowane na równi z danymi
- Pola nagłówka są przeznaczone dla MUA
- Serwer e-mail może przepisać informacje zawarte w envelope do header
- Serwer e-mail może dodatkowo analizować pola nagłówka, np. realizując funkcje anty-spam

Pola nagłówka E-mail

```
Return-Path: <lhrdina@checkpoint-ee.com>
Delivered-To: grzegorz.blinowski@[81.153.28.98]
Received: from localhost (localhost [127.0.0.1])
        by …….
Received: from freza.core.ignum.cz (217.31.49.12)
  by ….
From: "Ludek Hrdina" <lhrdina@checkpoint-ee.com>
To: "Up-to-date_CHKP" <lhrdina@checkpoint-ee.com>
Subject: Check Point's ….
Date: Tue, 11 Nov 2003 14:08:51 +0100
Message-ID: <LBEBINNFBNIIDCPJCLAA.lhrdina@checkpoint-ee.com>
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-2"
Content-Transfer-Encoding: 8bit
X-Mailer: Microsoft Outlook IMO, Build 9.0.2416 (9.0.2910.0)
X-AntiVirus: OK! AntiVir MailGate Version 2.0.0.9; ...
X-Mozilla-Status: 8001

Dear All,
```

Ruting SMTP

- W transmisji wiadomości serwer SMTP może być serwerem docelowym lub pośrednim
- Ruting SMTP - zestaw reguł pozwalających określić co serwer ma dalej robić z otrzymaną wiadomością
- W najprostszym przypadku:
  - jeśli serwer obsługuje dane konto e-mail (obsługuje domenę i zna użytkownika) to wiadomość zostaje wysłana do PO
  - w przeciwnym wypadku na podstawie rekordu MX pobranego z DNS serwer określa adres serwera do przekazania wiadomości
- Rekordy MX zawierają adres serwera SMTP i priorytet
  - Przykład:

```
smallcom.com MX mail.smallcom.com 10
smallcom.com MX mail-srv.hoster.com 20
```

  - im priorytet niższy tym "ważniejszy" MX

Ruting SMTP

- **Ruting jest w rzeczywistości o wiele bardziej skomplikowany, gdyż:**
  - wewnątrz domeny obsługiwanej przez jeden MX może funkcjonować wiele serwerów SMTP, np. wydziałowych, do których wiadomości rozdzielane są na podstawie wewnętrznych reguł rutingu:
    n.p.: firma.com.pl - SMTP srv.: kadry.firma, finanse.firma, misc.firma
  - serwer STMP może otrzymywać wiadomości z innych systemów E-mail (nie SMTP) - RFC2820 definiuje "SMTP gateway"
  - Serwery obsługują szereg historycznych konwencji takich jak: **`user%domena@domena`** lub **`serwer1!serwer2!serwer3...`**
  - Inne: konieczne jest wykrywanie i usuwanie pętli w rutingu

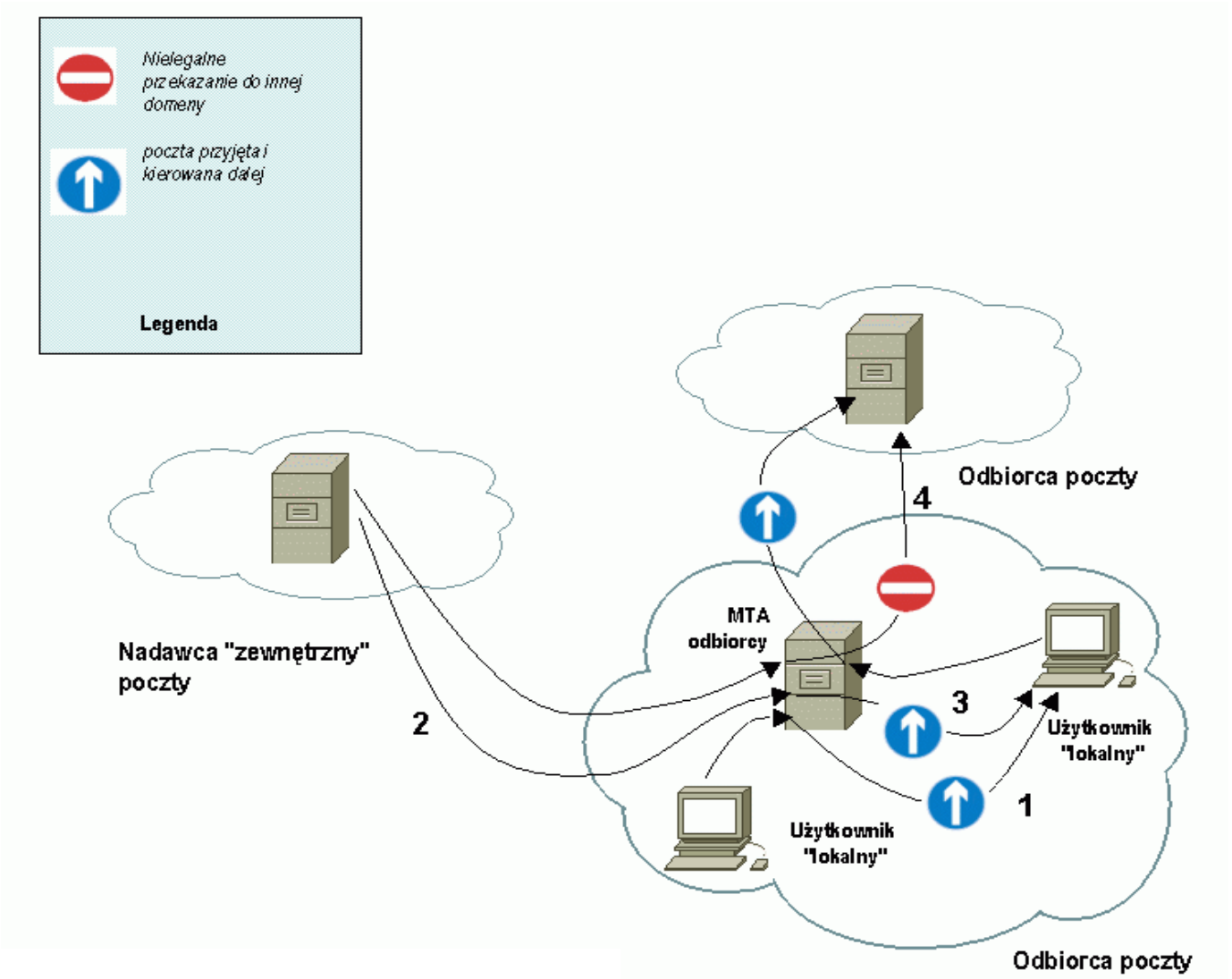
ESMTP

- ESMTP – rozszerzone SMTP
- Sesja SMTP rozpoczyna się poleceniem “EHLO”
- Serwer powinien obsługiwać EHLO nawet jeśli nie obsługuje żadnej z rozszerzonych opcji
- W odpowiedzi na EHLO serwer podaje listę rozszerzonych opcji protokołu, które obsługuje
- Najważniejsze opcje ESMTP:
  - 8BITMIME – transmisja 8 bitowych danych
  - SMTP-AUTH - autoryzacja
  - DSN - powiadomienia
  - STARTTLS – szyfrowanie sesji
  - PIPELINING – klient wysyła polecenia bez konieczności potwierdzenia przez serwer

Przykładowa sesja ESMTP

```
S: 220 m034.host.net.pl ESMTP
C: EHLO [10.0.1.254]
S: 250-m034.host.net.pl
S: 250-AUTH LOGIN PLAIN
S: 250-PIPELINING
S: 250 8BITMIME
C: MAIL FROM:<nadawca@domena.pl>
S: 250 ok
C: RCPT TO:<g.blinowski@host.net.pl>
S: 250 ok
C: DATA
S: 354 go ahead
```

Autoryzacja MUA w MTA



SMTP-AUTH

RFC5321

- Z oczywistych przyczyn organizacyjnych autoryzuje się tylko użytkownik “lokalny”, tj. nadawca, który jest w tej samej domenie administracyjnej co serwer
- Autoryzacja dotyczy przesyłek adresowanych do odbiorców poza domeną obsługiwaną przez dany serwer
- W autoryzacji “AUTH PLAIN” klient podaje nazwę użytkownika i hasło zakodowane w base64:

```
C: EHLO [10.0.1.254]
S: 250-m3.host.net.pl
S: 250-AUTH LOGIN PLAIN
S: 250-AUTH=LOGIN PLAIN
C: AUTH PLAIN BG192uDFGolS5.....==
S: 235 ok, go ahead (#2.0.0)
C: MAIL FROM:<nadawca@domena.pl>
```

SMTP-AUTH CRAM-MD5

RFC2104

- Metoda autoryzacji CRAM-MD5 (Challenge-Responce) jest odporna na podsłuch
- Serwer wysyła zakodowany w base64 ciąg znaków (CH) postaci: <24609.1047914046@host.net.pl>
- Liczby przed znakiem @ są losowe
- Klient musi obliczyć skrót MD5 w następujący sposób: MD5(('secret' XOR opad), MD5(('secret' XOR ipad), CH))
  - ipad, opad – stałe: 0x3636..., 0x5c5c...

```
C: EHLO [10.0.1.254]
S: 250-m3.host.net.pl
S: 250-AUTH LOGIN PLAIN
S: 250-AUTH CRAM-MD5
C: AUTH cram-md5
S: 334 PDI0NjA5LjEwNDc5MTQwNDZAcG9wbWFQ+
C: Zm9vYmFyLm5lE0OTViNGU2ZTczMzRkMzg5MAo=
S: 235 now authenticated as example.org
```

SMTP TLS

RFC5246

- TLS – Transport Layer Security – bezpieczny szyfrowany protokół zapewniający poufność komunikacji
- W wyniki polecenia STARTTLS następuje przełączenie sesji na szyfrowaną (klient i serwer muszą obsługiwać TLS i musi nastąpić jawne przełączenie)
- TLS może być wymagane tylko przez serwer lokalny - obsługujący klientów z tej samej domeny organizacyjnej

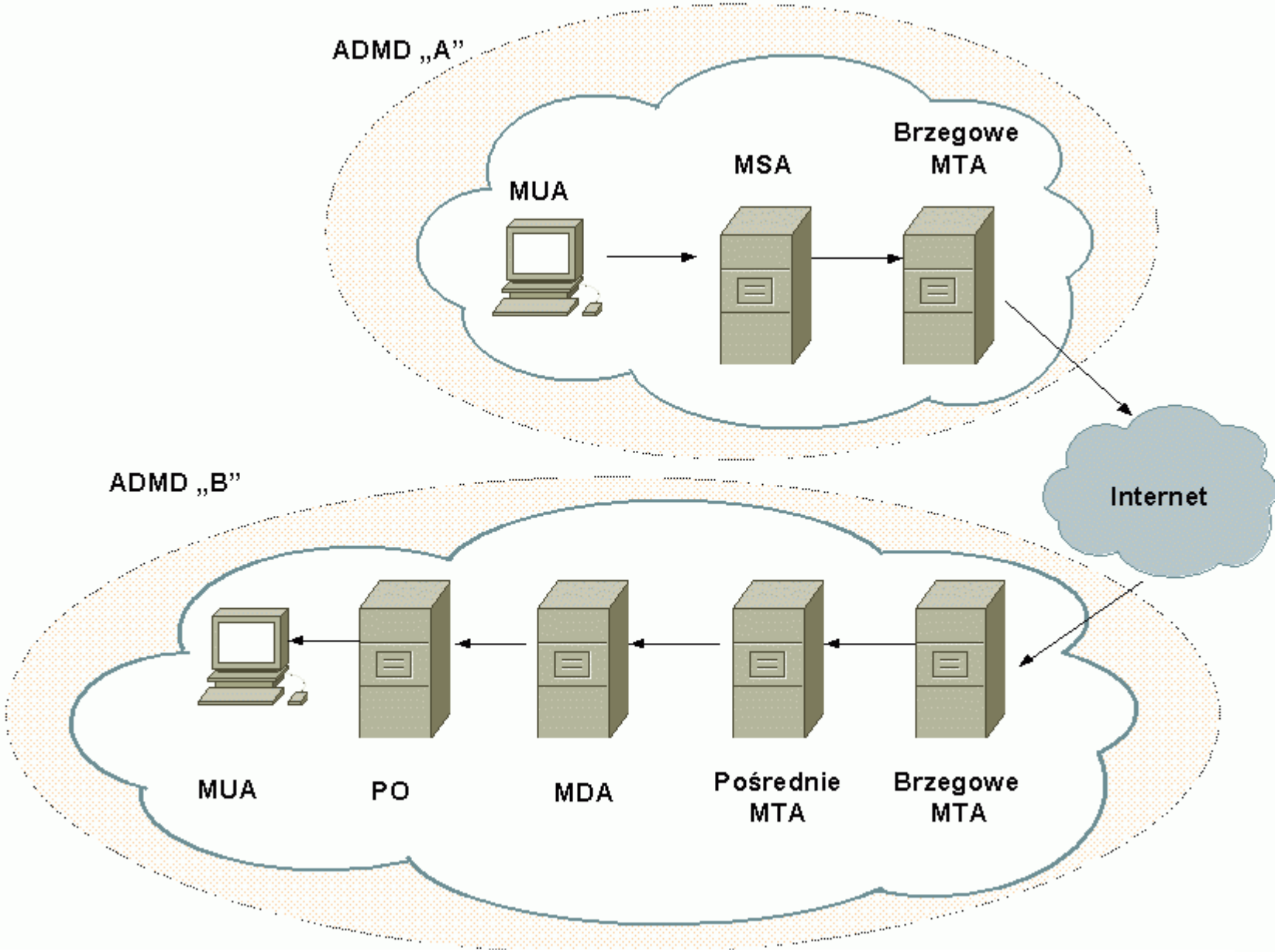
```
C: EHLO mail.example.com
S: 250-mail.imc.org offers a warm hug of welcome
S: 250-STARTTLS
C: STARTTLS
S: 220 Go ahead
C: <starts TLS negotiation>
C & S: <negotiate a TLS session>
C & S: <check result of negotiation>
C: EHLO mail.example.com
S: 250-mail.imc.org touches your hand gently for a moment
```

Dodatkowe uwagi dot. ESMTP

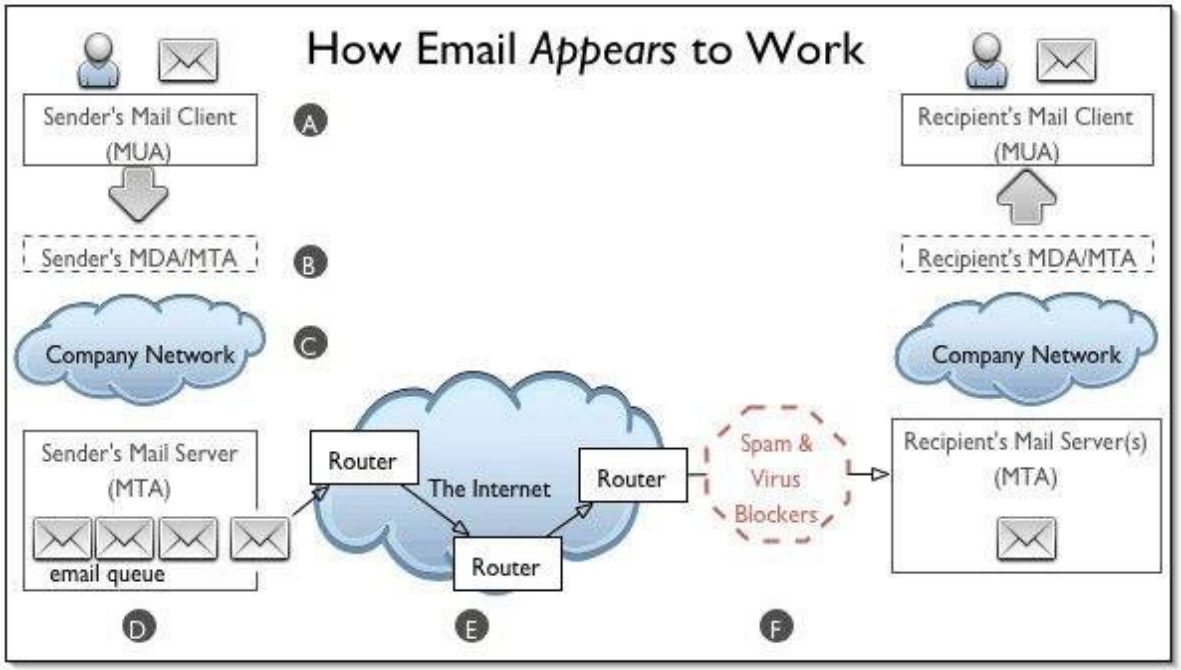
- SIZE:
  - MAIL FROM: <foobar@example.com> SIZE=1512
- Autoryzacja w MAIL FROM:

```
C: MAIL FROM:<e=mc2@example.com> AUTH=e+3Dmc2@example.com
S: 250 OK
```

- Autoryzacja Kerberos
- **DSN** – Delivery Status Notification: dodatkowe polecenia w RCPT TO powodują odesłanie zwrotnego maila w zależności od statusu dostarczenia


ADMD „A”
MUA
MSA
Brzegowe MTA
Internet
ADMD „B”
MUA
PO
MDA
Pośrednie MTA
Brzegowe MTA

:-)


How Email Appears to Work
Sender's Mail Client (MUA)
Recipient's Mail Client (MUA)
Sender's MDA/MTA
Recipient's MDA/MTA
Company Network
Company Network
Sender's Mail Server (MTA)
email queue
Router
The Internet
Router
Router
Spam & Virus Blockers
Recipient's Mail Server(s) (MTA)
A
B
C
D
E
F


How Email Really Works
Sender's Mail Client (MUA)
Recipient's Mail Client (MUA)
A
NSA HEADQUARTERS
RUSSIA
ILLUMINATI
SCARY SHIT
US GOVERNMENT
THE DRACULA

Autoryzacja c.d.

- Autoryzacja MUA → MTA dotyczy relacji: użytkownik → bezpośredni usługodawca (tu: SP) i **nie** "rozciąga" się na cały łańcuch tranmisji MUA_1 → MTA_1 → MTA_2 → … MUA_2
- **Globalna** autoryzacja użytkowników SMTP jest (obecnie) organizacyjnie niemożliwa
- W praktyce chcielibyśmy móc zweryfikować przynajmniej domenę nadawcy, w tym celu opracowano 2 mechanizmy:
  - SPF - Sender Policy Framework
  - DKIM – Domain Keys Identified Mail

SPF

- Weryfikacja MTA źródłowego w docelowym MTA
- Rekordy "SPF" przechowywane w DNS jako tekst (proste rozszerzenie usługi DNS)
- Odbierające MTA sprawdza w DNS-ie czy "domena" nadawcy zezwala źrodłowemu MTA na nadawanie poczty (weryfikacja adresu IP)
- Wpisy SPF budowane są jako proste wyrażenia logiczne odwołujące się do bazy DNS
- SPF pełni swoją funkcję gdy ufamy zawartości DNS, np. gdy zarządza nim zaufany operator, w praktyce SPF wymaga dodatkowo bazy zaufania / reputacji
- SPF służy głównie do eliminacji spamu-u (przemyśleć dlaczego - ?)
- Przykładowe rekordy SPF:

```
"v=spf1 ip4:192.168.0.1/16 -all"
```

Dozwolone są wszystkie adresy 192.168.*.* (maska 16 b.)

```
"v=spf1 a/24 a:openspf.org/24 -all"
```

Dozwolone są wszystkie adresy z domeny nadawcy z maską /24 oraz adresy z domeny openspf.org z maską 24

DKIM

- **DKIM** korzysta z silnej kryptografii, istota działania polega na **podpisywaniu** maili w imieniu nadawcy na poziomie "**domenowym**" (nadającego MTA). UWAGA: nie mylić z PGP, S/MIME gdzie podpis jest indywidualny
- Podpis umieszczany jest w nałówku SMTP
- Odbiorca sprawdza czy podpis zgadza się z kluczem domeny nadawcy
- DKIM pozwala na uniknięcie podszywania się pod znane adresy, dlatego służy głównie do ochrony przed atakami rozsiewającymi mallware i fałszerstwami (przemyśleć dlaczego - ?)

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
      d=gmail.com; s=20161025;
      h=date:from:to:message-id:subject:mime-version:x-original-sender
      :precedence:mailing-list:list-id:x-spam-checked-in-group:list-post
      :list-help:list-archive:list-unsubscribe;
       bh=n908IYtnGQHwoaMiZKfUpPuAsWROpjw/RYLS+uuGCN0=;
       b=sMXuDVMyEMSFOf2rSrvtWa25vK9GUqcwmPUtBdx/aeKJy4MqyQp+W....
```

SMTP DSN

RFC3461

RFC5337

- Nadawca prosi o potwierdzenie “odbioru” wiadomości
- Potwierdzenie zwracane jest jako e-mail generowany po stronie MTA lub MUA odbiorcy
- Serwer informuje czy obsługuje DSN podając kod rozszerzenia “250-DSN” w odpowiedzi na EHLO
- RCPT ... NOTIFY NEVER | SUCCESS | FAILURE | DELAY - żądanie potwierdzenia
- RCPT ... ORCPT – adres odbiorcy (może być inny niż parametr podany w RCPT TO, np. Gdy występuje forward)
- MAIL ... RET FULL | HDRS – sposób obsługi błędu, zwróć całą wiadomość lub tylko nagłówki
- MAIL ENVID – identyfikator wiadomości (tak aby nadawca wiedział czego dotyczy potwierdzenie)
- W przypadku przekazywania wiadomości dalej opcje DSN muszą być także przekazane

SMTP DSN - przykład

```
C:   EHLO Example.ORG
S:   250-Example.ORG
S:   250-DSN
C:   MAIL FROM:<Alice@Example.ORG> RET=HDRS ENVID=QQ314159
S:   250 <Alice@Example.ORG> sender ok
C:   RCPT TO:<Bob@Example.COM> NOTIFY=SUCCESS ORCPT=rfc822;Bob@Example.COM
S:   250 <Bob@Example.COM> recipient ok
C:   DATA
S:   354 okay, send message
C:   (message goes here)
C:   .
S:   250 message accepted
C:   QUIT
S:   221 goodbye
```

SMTP DSN - przykład

```
To: Alice@Example.ORG
From: postmaster@mail.Example.COM
Subject: Delivery Notification (success) for Bob@Example.COM
Content-Type: multipart/report; report-type=delivery-status;
    boundary=abcde
MIME-Version: 1.0

--abcde
Content-type: text/plain; charset=us-ascii
Your message (id QQ314159) was successfully delivered to
Bob@Example.COM.
--abcde
Content-type: message/delivery-status
Reporting-MTA: dns; mail.Example.COM
Original-Envelope-ID: QQ314159
Original-Recipient: rfc822;Bob@Example.COM
Final-Recipient: rfc822;Bob@Example.COM
Action: delivered
Status: 2.0.0
--abcde
Content-type: message/rfc822
(headers of returned message go here)
```

Prokoły maildrop

Maildrop - POP3 (port: 110)

RFC1939

Autoryzacja

- Polecenia klienta:
  - **user**
  - **pass**
- odpowiedzi serwera
  - **+OK, -ERR**

```
S: +OK POP3 server ready
C: user jasio
S: +OK
C: pass iza123
S: +OK user successfully logged on
```

Maildrop

client:

- **list** – nr wiad, wielkość
- **stat** – liczba wiad, suma wielk.
- **retr** – pobierz wiad.
- **dele** – usuń wiad (dopiero po quit)
- **rset** – usuń znaczniki dele
- **top** – początek wiad
- **uidl** – zwraca unikalne id-y
- **quit**

```
C: list
 S: 1 711
 S: 2 654
 S: .
 C: retr 1
 S: <message 1 contents>
 S: .
 C: dele 1
 C: retr 2
 S: <message 1 contents>
 S: .
 C: dele 2
 C: quit
 S: +OK POP3 server signing off
```

POP -inne cechy protokołu

```
S: +OK POP3 server ready
C: user jasio
S: +OK
C: pass iza123
S: +OK user successfully logged
on
C: uidl
S: 1 9fdfe6e8b409ed34
S: 2 2bd45c84aa21aa6d
S: 3 7441b05d7fb2c895
S: 4 04f755d957ddd0cd
S: 5 4cf18c0d46bf0e44
S: 6 9b87e31c0d7e6580
S: 7 b02e81cee8d89f04
```

Od listy wiadomości praktyczniejsza jest lista unikalnych id-ów, pozwala na powoływanie się na indywidualne wiadomości

POP3 - APOP

```
S: +OK POP3 server ready
   <1896.697170952@dbc.mtview.ca.us>
C: APOP username c4c9334bac560ecc979e58001b3e22fb
S: +OK maildrop has 1 message (369 octets)
```

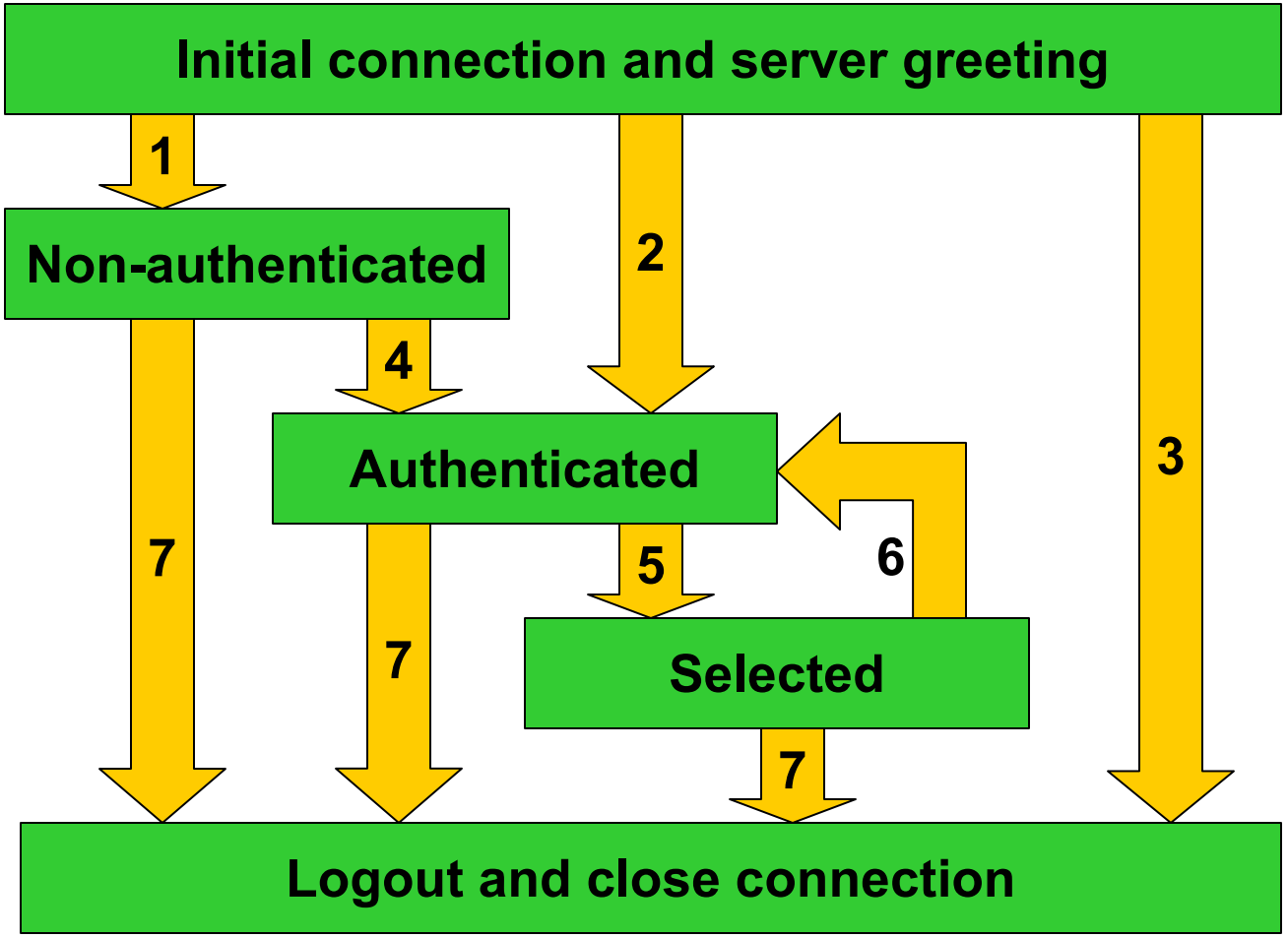
- Prosta weryfikacja tożsamości klienta na podstawie algorytmu Ch/Rp
- Polecenie: **`APOP user <digest>`**
- Odpowiedź: +OK lub –ERR
- Banner powitalny musi zawierać "challenge"
- Challenge ma postać jak niżej:
  - challenge opiera się na timestamp, np. standard proponuje:
    <process-ID.clock@hostname>
    klient oblicza kryptograficzną sumę kontrolną
    md5(challenge+secret)
- Dostępne jest też polecenie **`AUTH`** (RFC1734) - autentykacja Kerberos

Maildrop - IMAP4

RFC2060

- Zaprojektowany pod kątem przechowywania znacznej liczby wiadomości na serwerze
- Wiadomości grupowane w foldery
- Wiadomości posiadają "flagi" (*seen, deleted, answered*, itp.)
- Manipulowanie wiadomościami bez konieczności ładowania całej wiadomości do klienta (tylko nagłówki)
- Zalety: oszczędność pasma, dostęp do poczty z wielu lokalizacji, SSL - bezpieczeństwo
- Wady: skomplikowany protokół, implementacje nie zawsze zgodne ze sobą

Sesja IMAP


Initial connection and server greeting
1
Non-authenticated
2
4
Authenticated
3
7
5
6
7
Selected
7
Logout and close connection

Maildrop - IMAP4

- Protokół - asynchroniczny - odpowiedź na polecenie nie musi nadejść "od razu", czekając na odpowiedź można zlecić inne polecenia
- Aby zidentyfikować odpowiedzi serwera wprowadzono "tagi"
- Podstawowe stany serwera:
  - Non-authenticated
  - Authenticated (po zalogowaniu)
  - Selected – wybrano folder
  - Logout

Sesja IMAP

```
C: a001 LOGIN SMITH SESAME
S: a001 OK LOGIN completed
C: A142 SELECT INBOX
S: * 172 EXISTS
S: * 1 RECENT
S: * OK [UNSEEN 12] Message 12 is first unseen
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * OK [PERMANENTFLAGS (\Deleted
S: A142 OK [READ-WRITE] SELECT completed
```

Polecenia IMAP

• CAPABILITY	• pyt. o opcje funkcjonalne serwera
• NOOP	
• LOGOUT	• serwer zamyka transakcje, odp: “BYE”
• AUTHENTICATE	• opcjonalna autentykacja (np. Kerberos)
• LOGIN	• wymagana autoryzacja
• SELECT	• wybiera mailbox (folder), tylko 1 na raz
• EXAMINE	• j.w. w trybie read-only
• CREATE	• tworzy folder
• DELETE	• usuwa folder
• RENAME	• zmienia nazwę folderu
• SUBSCRIBE	• subskrybuje folder (zob. LSUB)
• UNSUBSCRIBE	• de-subskrybuje folder

Polecenia IMAP c.d.

• LIST	• zwraca listę folderów
• LSUB	• zwraca listę subskrybowanych folderów
• STATUS	• zwraca inf. o folderze (liczba wiad, etc)
• APPEND	• Dodaje wiadomość do folderu
• CHECK	• wymusza "checkpoint"
• CLOSE	• zamyka akt. folder, usuwa ozn. wiadomości
• EXPUNGE	• usuwa ozn. wiadomości (\Deleted)
• SEARCH	• szuka wiad. wg.zadanych kryteriów w akt. folderze
• FETCH	• pobiera (część) wiadomości
• STORE	• uaktualnia wiadomość
• COPY	• kopiuje wiadomości do zadanego folderu
• UID	• wykonuje inne polecenia (SEARCH, COPY, FETCH) w trybie zwracajacym unikalne uid-y

IMAP c.d.

- Flagi wiadomości:
  - \seen: wiadomość "przeczytana"
  - \answered: na wiadomość odpowiedziano
  - \flagged: wiadomość zaznaczono
  - \deleted: wiadomość ma być skasowana
  - \draft: wiadomość w trakcie tworzenia
  - \recent: wiadomość nowa

Sesja IMAP

```
C: A003 CREATE LCryptoGram/
S: A003 OK CREATE completed
C: A004 CREATE LCryptoGram/old
S: A004 OK CREATE completed

C: A682 LIST "" *
S: * LIST () "/" LCryptoGram
S: * LIST () "/" LCryptoGram/old
S: A682 OK LIST completed
C: A683 DELETE OldEmails
S: A683 OK DELETE completed
C: A685 DELETE LCryptoGram/old
S: A685 OK DELETE Completed
C: A683 RENAME SomeEmails OldEmails
S: A683 OK RENAME completed
```

Sesja IMAP – pobieranie wiadomości

```
C: A654 FETCH 2:4 (FLAGS
BODY[HEADER.FIELDS (DATE FROM)])
S: * 2 FETCH ....
S: * 3 FETCH ....
S: * 4 FETCH ....
S: A654 OK FETCH completed


C: A05 FETCH 3 BODY


S: * 3 FETCH (BODY ("TEXT"
"plain" ("charset" "utf-8") NIL
NIL "base64" 15664 202))
C: A05 OK Done FETCH
```

- Wiadomosć pobiera polecenie FETCH
- W argumentach można określić jaka część ma zostać pobrana
- dla dużych wiadomości można wybiórczo pobrac: flagi, nagłówki, ciało, załączniki

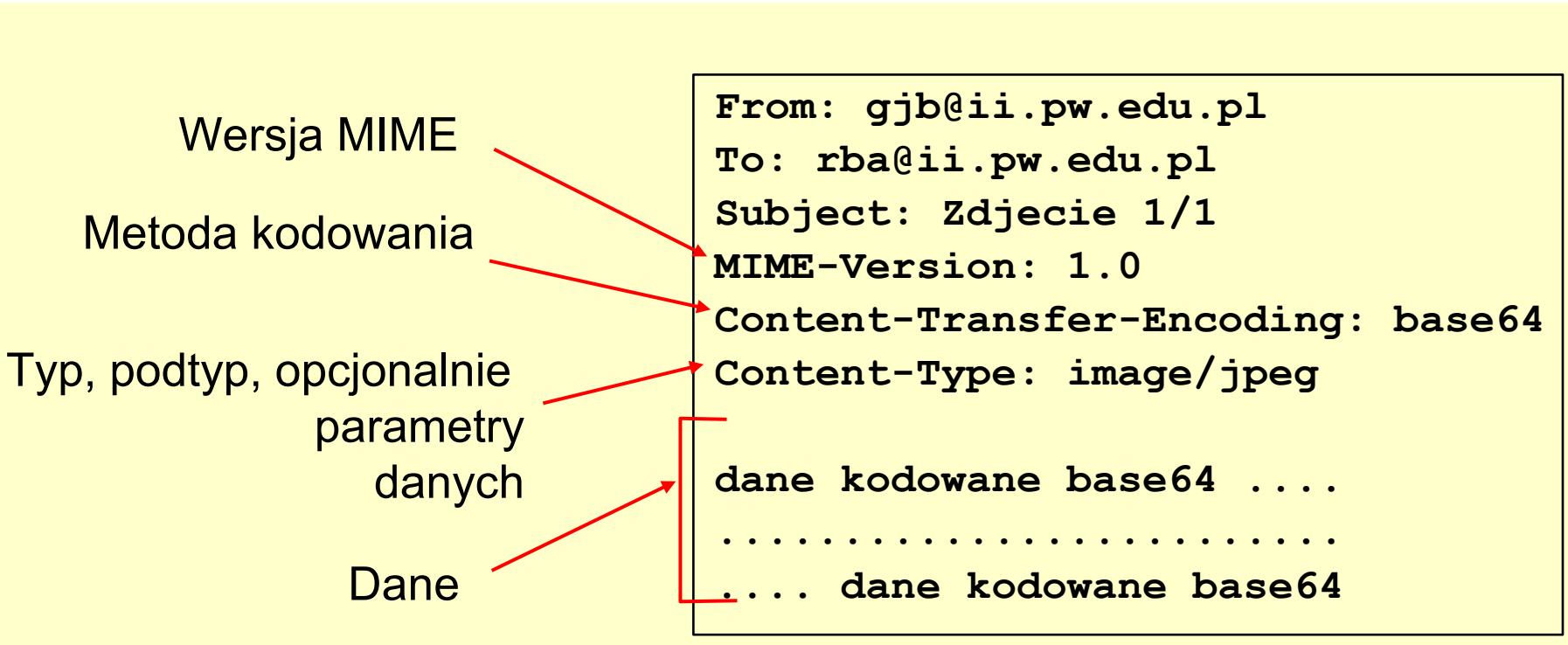
MIME
(Multipurpose Internet Mail Extensions)

MIME - podstawy

RFC2045

RFC2046

- **MIME** - aktualny standard RFC 2045, 2046 (pierwsze wersje RFC1341 - 1992)
- Dodatkowe pola nagłówka definiują format danych
- **Uwaga** - MIME pomyślany dla e-mail, jednak aktualne zastosowanie wykracza poza SMTP! (np. HTTP, XML, ...)



Podstawowe typy MIME

Content-type: type/subtype; parameters

Text

- **`text/plain, text/html`**
  `text/plain; charset=us-ascii`

Image

- **`image/jpeg, image/gif`**

Audio

- **`audio/basic`**
  jeden kanał, 8bit PCM 8000 Hz

Video

- **`video/mpeg, video/quicktime`**

Application

- **`application/msword, application/octet-stream`**

Złożone typy MIME

Content-Type: multipart/mixed; boundary=*bndr-string*

- Wiadomość podzielona na wiele części oddzielonych przez unikalny (ale arbitralny) ciąg znaków "boundary string"
- Generowanie boundary string – heurystyczne (?)
- Każda część może mieć własny content-type, np:
  - pierwsza część może być typu text/plain
  - druga typu image/gif kodowana base64

MIME Multipart/mixed - przykład

```
From: <donald.duck@disney.com>
To: <mickey.mouse@disney.com>
Subject: blah
MIME-Version: 1.0
Content-Type: multipart/mixed;  boundary= "Boundary-
   00=_9W2T/VtQiQcNR1P"

--Boundary-00=_9W2T/VtQiQcNR1P
Content-Type: text/plain; charset=US-ASCII

tekst tekst tekst tekst tekst tekst tekst tekst

--Boundary-00=_9W2T/VtQiQcNR1P
Content-Type: application/octet-stream
...
```

MIME Multipart/mixed - przykład

```
From: <donald.duck@disney.com>
To: <mickey.mouse@disney.com>
Subject: blah
MIME-Version: 1.0
Content-Type: multipart/mixed;  boundary= "Boundary-
   00=_9W2T/VtQiQcNR1P"
...
--Boundary-00=_9W2T/VtQiQcNR1P

Content-Type: application/octet-stream
Content-Transfer-Encoding: base64
 Content-Disposition: attachment;
 filename= "nazwa_pliku.bin"

base64base64base64base64base64...
base64base64base64base64base64...
--Boundary-00=_9W2T/VtQiQcNR1P--
```

Boundry poprzedzone jest --

Ostatnie boundry zakończone jest --

Inne złożone typy MIME

- **multipart/alternative** - ta sama treść przesłana w kilku wariantach, np. w formacie text i HTML, klient wybiera właściwą postać do prezentacji danych
- **multipart/digest** - format identyczny jak w mixed, służy do wysyłania wielu wiadomości formatu message/rfc822
- **multipart/parallel** - format identyczny jak w mixed, służy do równoległej prezentacji danych w kilku formatach
- **multipart/signed, multipart/encrypted – wiadomość podpisana / zaszyfrowana (S/MIME – RFC 1847)**
- **message/partial** - pozwala na przesłanie dużej wiadomości w "kawałkach":

```
Content-Type: Message/Partial; number=2;
total=3;
```

Kodowanie Base64

- Proste kodowanie pozwalające na przesłanie dowolnych danych binarnych w postaci "drukowalnych" znaków ASCII (7bit)
- Wielkość danych zwiększa się o 4/3, np. plik o rozmiarze 3KB ma po zakodowaniu 4KB
- Bloki 3 bajtowe zamieniane są na 4 liczby 6-o bitowe
- Każda liczba 6-o bitowa zamieniana jest na znak drukowalny z przedziału: A-Za-z0-9+/
- Jeśli wielkość danych kodowanych w bajtach nie jest podzielna przez 3 - uzupełnienie zerami/znakami = na końcu
- Wynikowy tekst jest dzielony na linie o długości 76 znaków
- Tak otrzymany tekst jest do zaakceptowania przez każdy MUA/MTA zgodny z podstawowym SMTP

Więcej o MIME

- **RFC:**
  - **RFC 2045** - nagłówki definiowane w standardzie MIME
  - **RFC 2046** - struktura danych MIME i podstawowe typy danych w MIME
  - **RFC 2047** - rozszerzenia MIME w treści nagłówków wiadomości
  - **RFC 2048**, - procedury rejestracji nowych typów MIME w IANA
  - **RFC 2049** - reguły zgodności aplikacji z MIME oraz przykłady

MIME - kodowanie

- Zdefiniowane są następujące typy "Content-Transfer-Encoding":
  - "7bit"
  - "8bit"
  - "binary"

  nie dokonano przekodowania
  - "quoted-printable"
  - "base64"

  Wiadomość jest kodowana w specjalny sposób
  - ietf-token oraz x-token
- Typ "quoted-printable" - do kodowania wiadomości składających się głównie, ale nie wyłącznie, ze standardowych znaków ASCII
- **Uwaga:** RFC 2045 określa dozwolone typy kodowania dla złożonych typów danych (multipart i message) jako wyłącznie: 7/8 bit oraz binary

MIME - "quoted-printable"

- Kodowanie mające na celu zachowanie danych (zwykle tekstu) bez modyfikacji nawet przez serwery niezgodne ze standardami
- każdy oktet może być reprezentowany jako: **=hh**, gdzie **h** - cyfra szesnastkowa: 0123456789ABCDEF
- oktety o kodach: 33-60 (większość znaków przestankowych) oraz: 62-126 (litery, cyfry, [,] , |, ~) mogą być reprezentowane bezpośrednio
- białe znaki reprezentowane są bezpośrednio, **chyba**, że są na **końcu linii**
- znak **=** na końcu linii oznacza "soft line break" (można łamać długie linie)
- Przykład:

```
Dzia=B3aj=B1c w imieniu i na rzecz=
 sp=F3=B3ki "Zielone buraczki" Sp. =
```

RFC 2047 - "Message Header Extensions"

- Do tej pory omówione rozwiązania nie pozwalają na zawarcie znaków **nie ASCII** w nagłówku wiadomości - co może być przydatne np. w nagłówku "Subject" (znaki narodowe)
- RFC 2047 rozwiązuje ten problem wprowadzając tzw. "encoded word"

```
encoded-word = "=?" charset "?" encoding "?" encoded-text "?="
```

- Charset - określa kodowanie (np. ISO-8859-2)
- encoding - **Q** - quoted printable, **B** - base64
- encoded-text - kodowany tekst

Przykład:

```
From: =?ISO-8859-1?Q?Olle_J=E4rnefors?= <ojarnef@admin.kth.se>
Subject: =?ISO-8859-1?B?SWYgeW91IGNhbiByZWFkIHRoaXMgeW8=?=
    =?ISO-8859-2?B?dSB1bmRlcnN0YW5kIHRoZSBleGFtcGxlLg==?=
```

Inne nagłówki MIME

- Content-Disposition: attachment | inline; name="..."
  - określa sposób prezentacji załącznika
  - zawiera dodatkowe wskazówki dot. jego obsługi
  - inne parametry: Size, Creation-Date, Modification-Date
- Content-ID:
  - jak Message-id,
  - muszą być unikalne,
  - pozwalają identyfikować daną pod-zawartość,
  - opcjonalne (poza typem message/external-body)
- Content-Description:
  - opis zawartości - dowolny
  - zawsze opcjonalne

RFC 2049 - Zgodność z MIME

- **RFC 2049**: *"[…] There exist many widely-deployed non-conformant MTAs in the Internet. These MTAs, speaking the SMTP protocol, alter messages on the fly to take advantage of the internal data structure of the hosts they are implemented on, or are just plain broken. […]"*
- RFC 2049 - określa minimalny poziom zgodności, w szczególnosci sytuacje gdy program natrafi na nieznany typ danych (zalecenie - traktować jak *multipart/mixed*), lub nieobsługiwany zestaw znaków (zalecenie - traktować jak application/octed-stream)
- RFC 2049 precyzuje standard, określa kroki jakie musi wykonać aplikacja generując wiadomość MIME

Bezpieczeństwo e-mail

- Cechy bezpiecznej poczty (ogólnie):
  - Integralność
  - Prywatność
  - Niezaprzeczalność
- Jak zapewnić bezpieczeństwo e-mail?
  - w warstwie sieciowej – poprzez protokół VPN wykorzystujący IPsec,
  - poprzez szyfrowanie na poziomie sesji SMTP z wykorzystaniem trybu TLS,
  - poprzez enkrypcję treści przesyłki z wykorzystaniem S/MIME lub PGP,
  - poprzez enkrypcję treści innymi metodami i wysyłkę odsyłacza do zaszyfrowanej treści,
  - poprzez enkrypcję załączników z wykorzystaniem arbitralnych programów szyfrujących

S/MIME

- S/MIME zapewnia:
  - Integralność, niezaprzeczalność – poprzez podpisy
  - Prywatność (tajność) – poprzez szyfrowanie
- Jak działa S/MIME:
  - Dane podpisane lub zaszyfrowane i podpisane są enakpsulowane zgodnie z ogólnymi zasadami MIME
  - Wykorzystuje się normy i standardy PKI (infrastukturę klucza publicznego)
  - System zaufania jest hierarchiczny i zcentralizowany – bazuje na hierarchii certyfikatów x.509v3
  - Podpis zawiera m.in. certyfikat podpisującego: dane go identyfkujące i klucz publiczny
  - Dzięki temu odbiorca może zweryfikować podpis nie posiadając uprzedniej wiedzy o nadawcy
  - Odbiorca musi zweryfikować certyfikat nadawcy – certyfikat powinien być wystawiony przez znane CA lub też łańcuch certyfikatów prowadzących od głównego CA do certyfikatu nadawcy musi pozytywnie przejść taką weryfikację

S/MIME

- Jak działa S/MIME - szyfrowanie
  - jeśli wiadomość ma być przeznaczona „tylko dla oczu” określonego odbiorcy, to powinna być zaszyfrowana jego kluczem publicznym - tak aby tylko on mógł ją odszyfrować przy pomocy swojego klucza prywatnego.
  - Gdy wielu odbiorców wiele kopii?, Nie!
    - wiadomość jest szyfrowana losowo wygenerowanym kluczem symetrycznym
    - klucz symetryczny szyfrowany jest przy pomocy klucza publicznego odbiorcy, który pobierany jest z jego certyfikatu (dla każdego odbiorcy tworzona jest jego „osobista” zaszyfrowana dla niego kopia klucza). Dodatkowo do szyfrowania klucza może być także użyty klucz prywatny nadawcy
    - Zaszyfrowana wiadomość wraz z kompletem zaszyfrowanych kluczy wysyłana jest do odbiorców
  - do przesłania zaszyfrowanej wiadomości potrzebujemy certyfikatu odbiorcy, musi on być wcześniej dostarczony nadawcy

```
Subject: Informacja podpisana
Date: Tue, 26 Oct 2010 13:57:40 +0200
MIME-Version: 1.0
Content-Type: multipart/signed;
        protocol="application/x-pkcs7-signature";
        micalg=SHA1;
        boundary="----=_NextPart_0039_01CB7515.C1444170"
Message-ID: <C3E088AF05CC@jannowak.net.pl>
From: "Jan Nowak" <j.nowak@jannowak.net.pl>
To: "Grzegorz Blinowski" <g.blinowski@example.com.pl>

This is a multi-part message in MIME format.

------=_NextPart_0039_01CB7515.C1444170
Content-Type: text/plain;
        charset="iso-8859-2"
Content-Transfer-Encoding: quoted-printable

Oto tekst wiadomosci, ktory zostanie podpisany.
Pozdrawiam,
Nowak


------=_NextPart_0039_01CB7515.C1444170
Content-Type: application/x-pkcs7-signature;
        name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
        filename="smime.p7s"

MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCG...
BRUvlS2/DxGlYwyxOl0w0YCccSkdWZP5bFMVvFsKIgAAAAAAAA==

------=_NextPart_000_0039_01CB7515.C1444170--
```

PGP

- PGP a S/MIME – podstawowa różnica – model zaufania w PGP jest rozproszony i wzajemny – "web of trust"
- PGP używa własnego formatu danych do zapisu zaszyfrowanej wiadomości, podpisu oraz certyfikatu (OpenPGP).
- Certyfikaty PGP ("klucze PGP"), podobnie jak certyfikaty x509v3, wiążą klucz publiczny z informacją o użytkowniku. PGP wykorzystuje własny format certyfikatu.
- Podpis certyfikatu PGP składany jest przez innego użytkownika - “Web of trust”.
- W PGP są dwa formaty przeznaczone do transmisji danych pocztą: "ASCII armour" oraz PGP/MIME
- Więcej o PGP → przedmiot BSS

```
Message-ID: <4D289E16.6010402@example.net.pl >
Date: Sat, 08 Jan 2011 18:25:42 +0100
From: Grzegorz Blinowski <g.blinowski@example.net.pl >
MIME-Version: 1.0
To: Grzegorz Blinowski <g.blinowski@example.net.pl>
Subject: tym razem sam podpis
X-Enigmail-Version: 1.0.1
Content-Type: multipart/signed; micalg=pgp-sha1;
 protocol="application/pgp-signature";
 boundary="------------enig"

This is an OpenPGP/MIME signed message (RFC 2440 and 3156)
--------------enig
Content-Type: text/plain; charset=windows-1252
Content-Transfer-Encoding: quoted-printable

znowu pisze sam do siebie ale tym razem tylko podpisuje email.

--------------enig
Content-Type: application/pgp-signature; name="signature.asc"
Content-Description: OpenPGP digital signature
Content-Disposition: attachment; filename="signature.asc"

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.11 (MingW32)
Comment: Using GnuPG with Mozilla - http://enigmail.mozdev.org/

iQEcBAEBAg4WAAoJEIg6zkjLMdNevjAIAJzHBxS5QI8VU4Kqngr18DMj
Ds+qDr6Yj3NJFR7MScuP9zVial2EsNPfKnY8pVqxd2nKX8aZ+fXpHLHc
. . .
=nCM7
-----END PGP SIGNATURE-----
```