L@ POST@ ELETTRONIC@

Gli standard per la posta elettronica: POP3, SMTP, IMAP e problematiche di sicurezza

- **La posta elettronica**
  - **MUA e MTA**

- **Architettura**
  - **Il formato**
    - **RFC 822**
    - **MIME (RFC 1341)**
  - **I protocolli**
    - **SMTP**
      - **Conversazione SMTP**
      - **Comandi principali**
      - **I "Reply number"**
    - **POP**
      - **Conversazione POP**
      - **Comandi principali**
    - **IMAP**
      - **Pop3 vs Imap4**
    - **NNTP**
      - **Comandi principali**

- **La sicurezza**
  - **Tecniche crittografiche**
    - **Algoritmi**
    - **Message digest**
    - **Firme digitali**
  - **Certificati**
  - **PGP**
  - **S/MIME**

Di Francesco Magagnino

LA POSTA ELETTRONICA

La posta elettronica è uno dei servizi più consolidati ed usati nelle reti.
In Internet è una realtà da almeno vent'anni e può essere considerato uno dei fautori del suo successo.

Gli indirizzi di posta elettronica, in Internet, hanno la forma:

username@hostname

dove username è la stringa che identifica l'interlocutore (mittente o destinatario) che è univoca per hostname; hostname è un nome DNS oppure un indirizzo IP.
Gli indirizzi di posta elettronica sono forniti dagli ISP (Internet Service Provider) o da altre realtà che supportino una banda di comunicazione verso Internet e verso gli utenti.
Va notato che l'indirizzo di posta elettronica non è associato ad una persona, ma ad una casella postale elettronica; ogni utente può possedere diverse caselle come è anche possibile associare nomi utente diversi (*alias*) alla stessa casella di posta.

La posta elettronica viene implementata in Internet attraverso la cooperazione di due tipi di sottosistemi:

- Mail User Agent (MUA)
- Mail Transport Agent (MTA)

Il MUA

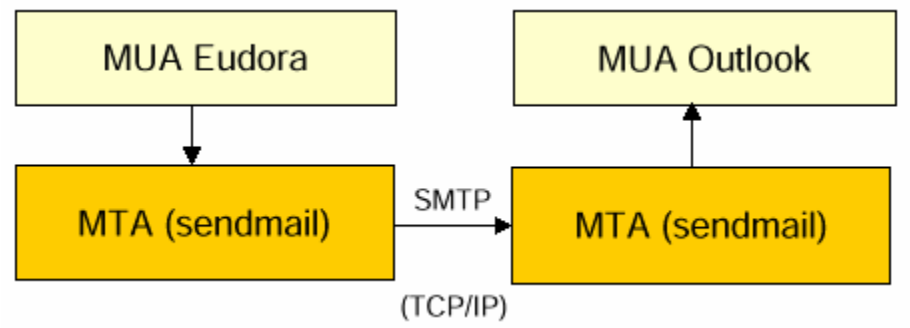
Un MUA è un programma di gestione di posta (Outlook, Eudora,…), operativo sul client, che deve:

- Possedere un'interfaccia utente per l'inserimento, la composizione, la ricezione e la lettura dei messaggi.
- Conoscere il protocollo per spedire i messaggi (SMTP) e quindi consegnarli ad una MTA per la trasmissione
- Conoscere il protocollo POP3 e IMAP4
- Conoscere la sintassi di composizione dei messaggi (RFC822 e MIME)

L'MTA

L'MTA funge da ponte tra due MUA, si occupa della ricezione di tutti i messaggi e del suo recapito, può essere paragonato ad una centralina telefonica dove vengono smistate le chiamate.
L'MTA può essere:

- un server SMTP (porta 25) che gestisce la spedizione e la ricezione dei messaggi verso e da altri server SMTP
- un server POP3 (porta 110) che gestisce la spedizione dei messaggi al client
- un server IMAP4 (porta 143) che permette la gestione dei messaggi sul server dal client



ARCHITETTURA

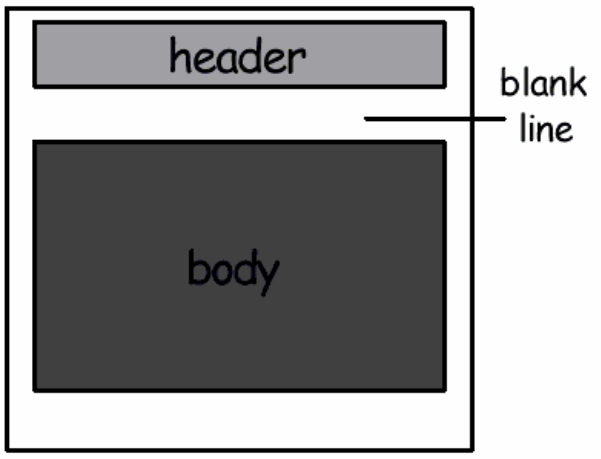
RFC 822, MIME RFC 1341

Inizialmente il protocollo per la rappresentazione dei documenti di posta elettronica (e-mail) era definito nel documento *RFC 822*, del 1982; in questo documento veniva specificato il formato per i messaggi di posta e ci si limitava a messaggi esclusivamente di tipo testo ASCII, senza alcun riferimento a messaggi di altro tipo (ad esempio, le immagini). Nel giugno 1992 è stato presentato un nuovo documento, l'*RFC 1341*, in cui viene descritto lo standard MIME. In RFC 1341 vengono presentati i meccanismi per superare le limitazioni contenute in RFC 822; viene specificato come definire il formato sia di messaggi testuali (ASCII e non) sia di messaggi multimediali (cioè contenenti video, suono, immagini, ecc.). Una delle principali limitazioni del protocollo descritto in RFC 822 si ha nel fatto che il contenuto dei messaggi è limitato a simboli (caratteri) di 7 bit; questo impone che ogni messaggio non costituito da solo testo ASCII, debba essere convertito in questo formato prima di essere inviato in rete. Per risolvere questo problema è stato proposto il documento RFC 1341.
Il MIME è stato decisivo per implementare dei servizi di sicurezza, nello specifico il S/MIME (Secure/MIME, RFC 1847) che da la possibilità di inviare messaggi corredati di firma digitale, di crittografia o di autenticazione.

La RFC (Request for Comments) è un voluminoso documento ("*http://www.rfc-editor.org*", "*http://www.ietf.org*", "*http://www.faqs.org*") solo in lingua inglese che contiene tutte le specifiche che un server deve avere per essere considerato "standard" ovvero utilizzabile da tutti i client che sono stati programmati seguendo il medesimo documento.
La necessità è dovuta al fatto che bastano piccole modifiche per far si che un client sia inutilizzabile in un server, e danno maggiore, che non sia possibile la comunicazione tra due server diversi.

RFC 822

L'RFC 822 è una stringa di testo costituita da un header e da un body separati da una linea vuota.
L'header contiene le informazioni per il trasporto:



To: lista di destinatari
From: mittente
Cc: lista di destinatari per conoscenza
Bcc: lista nascosta di destinatari per conoscenza
Date: data di spedizione
Reply- to: indirizzo diverso dal mittente
Subject: titolo del messaggio

Keywords non previste sono comunque spedite e interpretate dagli agenti utente: es. Organization

RFC 1341 o MIME (Multipurpouse Internet Mail Extensions)

Quando due programmi dialogano tra loro attraverso la rete Internet (uno invia un file e l'altro lo riceve), il programma che invia il file deve specificarne il tipo secondo lo standard MIME; in questo modo il programma che riceve i dati può capire come trattarli. Con lo standard MIME è possibile inserire in un qualsiasi messaggio di e-mail, oltre al testo, anche files contenenti immagini, segnali audio e video; il software che gestisce la posta non si preoccupa del contenuto del messaggio, è l'utilizzatore finale a preoccuparsi della sua opportuna decodifica in base alle specifiche di tipo inserite nel messaggio stesso.

In MIME, come descritto in RFC 1341, vengono introdotti dei meccanismi che permettono di risolvere i problemi di RFC 822, senza introdurre delle incompatibilità con i documenti scritti secondo il vecchio standard. Un documento MIME contiene una testata in cui si trovano i seguenti campi:

MIME version
Content-Transfer-Encoding
Content-Type
Content-ID
Content-Description

MIME version
Identifica la versione dello standard MIME usato nel messaggio. Questo permette di indicare se un messaggio è conforme allo standard, in modo tale che il software che lo riceve possa distinguerlo da quei messaggi scritti secondo il vecchio standard (in cui questo campo è assente).

Content-Transfer-Encoding
Specifica un modo di codifica dei dati accessorio a quello principale, utilizzato per permettere loro di passare attraverso tutti i meccanismi di trasporto della posta elettronica, i quali potrebbero avere delle limitazioni nel set di caratteri ammessi (questa codifica aggiuntiva deve essere applicata ai dati prima della loro trasmissione). I

documenti, per essere trasportati in rete, necessitano di un'ulteriore codifica e nel *content-transfer-encoding* viene specificato qual è la relazione tra i dati nella loro forma originale ed il formato con cui vengono trasmessi. La maggioranza dei documenti può così essere trasportata nella rete senza problemi, anche nel caso in cui i dati debbano attraversare una rete conforme soltanto allo standard RFC 822 e non alle sue successive versioni (come il MIME); un esempio è un sistema di posta compatibile col protocollo SMTP (Simple Mail Transfer Protocol), in cui è necessario che il documento venga opportunamente codificato in caratteri ASCII, a 7 bit, con non più di 1000 linee. Il campo *Content-Transfer-Encoding* è appunto usato per specificare come possono essere manipolati i dati per essere trasportati in rete; in esso viene specificato un meccanismo, invertibile, per trasformare il documento originario e non ha alcuna influenza sul tipo di dati trasportati.

I valori di *Content-Transfer-Encoding* sono: 7bit, 8bit, binary, quoted-printable, base64. Il loro significato è questo:

- ***7bit, 8bit, binary:*** questi valori stanno a significare che nessuna operazione di codifica è stata effettuata sul contenuto del messaggio e, allo stesso tempo, forniscono una indicazione sul tipo di dati contenuti nel messaggio stesso (quindi forniscono un'indicazione sul tipo di codifica che potrebbe rendersi necessaria per trasmettere il messaggio in determinati sistemi di trasmissione). Il valore "7bit" significa che in questo caso i dati possono essere rappresentati in gruppi di sette bit, ognuno dei quali rappresenta un carattere ASCII; questo è anche il valore assunto come default se il campo non viene specificato. Il valore "8bit" significa che possono essere presenti caratteri non appartenenti al set ASCII; cioè, suddividendo il messaggio in linee di 8 bit ciascuna e associando ad ogni linea un carattere ASCII, si possono ottenere delle sequenze di caratteri apparentemente senza significato. Il valore "binary" indica che il contenuto del messaggio è in formato binario (un'immagine, un file audio, ecc.). La differenza tra "8bit" e "binary" può sembrare irrilevante ma, può assumere grande significato in quei sistemi di trasferimento dati che non sono conformi alle restrizioni della RFC 821.
- ***quoted-printable:*** questo valore significa che un'operazione di codifica è già stata applicata ai dati, in modo da trasformare il messaggio in una sequenza di caratteri ASCII (se il messaggio originario era già costituito da un testo ASCII, questa codifica lo lascia sostanzialmente inalterato). Lo scopo principale di questa codifica è di mettere i dati in un formato che difficilmente subirà delle trasformazioni da parte dei vari sistemi che è costretto ad attraversare, prima di giungere a destinazione.
- ***base64:*** questo valore significa che sui dati è stata effettuata un'operazione di codifica, detta base64; con questa operazione il messaggio viene trasformato in una sequenza di caratteri appartenenti ad un sottogruppo del set di caratteri ASCII (le lettere maiuscole da "A" a "Z", quelle minuscole da "a" a "z", I numeri da"0" a "9", il carattere "+" ed il carattere "\|"). In questo modo, ogni carattere codificato può essere rappresentato con sei bit. L'operazione di codifica consiste nel suddividere la sequenza dei bit in ingresso (il messaggio) in gruppi di 24 bit; ogni gruppo di 24 bit viene diviso in quattro gruppi di sei bit, ad ognuno dei quali si associa il corrispondente carattere ASCII appartenente al sottogruppo specificato.
- ***x-token:*** viene usato per specificare uno schema di codifica esterno, non standard, scelto da chi trasmette il messaggio (*token* coinciderà col nome dato a questa codifica); si deve fare attenzione al fatto che questa codifica deve essere nota

anche a chi riceve il messaggio, in modo che questo possa essere ricostruito correttamente

Content-Type
Specifica il tipo ed il sottotipo di dati contenuti nel messaggio (MIME type), in modo che il software che riceve il messaggio possa immediatamente capire come sono codificati i dati ricevuti.
Questo campo ha la forma:
Content-Type: tipo/sottotipo;[parametro]
dove *tipo* specifica la forma generale dei dati, mentre *sottotipo* specifica il particolare tipo di dati trasmessi. Il campo *parametro* è opzionale.
Text, per esempio, è un *tipo* usato per rappresentare informazioni in forma testuale, scritte secondo un certo linguaggio. Un *sottotipo* per *text* è ***plain***, che indica un testo non formattato, un altro è ***richtext***, usato per testi con una semplice formattazione. Un *parametro* usato per i messaggi *text* è ***charset***, che è usato per indicare il set di caratteri utilizzato.
Ad esempio, per la posta elettronica in Internet, il campo Content-Type assume la forma:
Content-Type: text/plain; charset=us-ascii
Nel caso non sia specificato il *parametro*, viene usato come default il set di caratteri *US-ASCII.*

Content-ID
Identifica il messaggio in modo univoco (opzionale).

Content-Description
Descrizione testuale del contenuto del messaggio (opzionale).
Tutti i tipi di dati MIME, specificati nel campo *Content-Type*, devono essere registrati presso la IANA (Internet Assigned Numbers Authority); i nuovi tipi di dati, non ancora ufficialmente riconosciuti dalla IANA devono essere indicati con "*x*-",come ad esempio *multipart/x-mixed*, oppure *application/x-httpd-cgi.*

Esempio di header e body tra due utenti inf.unitn.it

Received: from 193.205.204.5 (odino.inf.unitn.it [193.205.204.6])
by cevedale.inf.unitn.it (8.9.3/8.9.3) with SMTP id QAA14260
for <ezanoni@inf.unitn.it>; Thu, 31 Jan 2002 16:57:11 +0100 (MET)
Message-Id: <200201311557.QAA14260@cevedale.inf.unitn.it>
To: ezanoni@inf.unitn.it
From: fmagagni@inf.unitn.it
Subject: Ciao
Date: Thu, 31 Jan 2002 16:57:11 MET
X-Mailer: Endymion MailMan Standard Edition v3.0.22
Content-Type: text
X-UIDL: d16f7ac9bfcd49467c84888f8440b62b
Status: RO

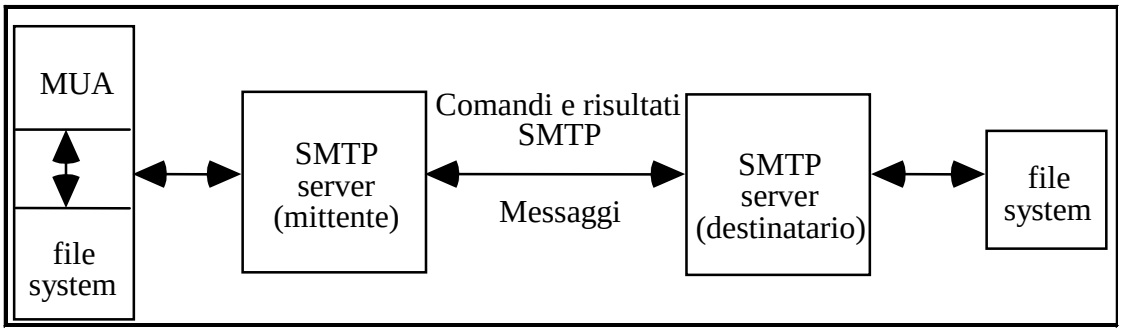
Ciao, oggi ho lavorato proprio molto.
Oggi è una bella giornata.

I PROTOCOLLI

I protocolli che garantiscono la maggior parte degli attuali servizi di posta elettronica sono l'SMTP, il POP3 e l'IMAP; tutti e tre sul livello Applicazione (cinque) come l'http, l'ftp e il dns.

SMTP - Simple Mail Transfer protocol (RFC 821)

E' il protocollo standard per la trasmissione di e-mail su Internet. E' un protocollo di tipo TCP/IP che definisce il formato del messaggio e l'agente di trasferimento messaggio (MTA), il quale memorizza e invia il messaggio. SMTP era stato definito in origine per il solo testo ASCII, ma la definizione del metodo di codifica MIME ed altri metodi di codifica abilitano il protocollo alla possibilità di effettuare attachments in formato multimediale o di programmi ai messaggi SMTP standard.
I server SMTP effettuano l'inoltro dei messaggi di posta elettronica attraverso Internet ai mail server quali IMAP e POP3
Lo standard SMTP è un protocollo "furbo", perché prevede alcune astuzie come quella di non obbligare il trasferimento di un messaggio più di una volta qualora questo abbia più destinatari, o quella di verificare prima l'esistenza del luogo di destinazione, evitando così di lasciare trasferire inutilmente i dati sul server per poi scoprire che il messaggio non è giunto a destinazione, per esempio a causa di un errore nella digitazione dell'indirizzo.



Conversazione SMTP

E' importante vedere il funzionamento di una conversazione SMTP via telnet sulla porta 25 (la porta 25 è indicata dalla rfc numero 1060: "smtp 25/tcp mail").

La procedura per spedire un'e-mail prevede delle fasi standard:

- Contrattazione: il server saluta con un tipico banner:"220 dominio SMTP Service ready".
- Dialogo: botta-risposta tra il client ed il server.
- Chiusura.

Esempio di conversazione Smtp su inf.unitn.it

```
220 odino.inf.unitn.it ESMTP Sendmail 8.10.2+Sun/8.10.2; Thu, 31 Jan
2002 16:33:35 +0100 (MET)
helo fmagagni
250 odino.inf.unitn.it Hello lab3pc15.inf.unitn.it [193.205.204.145],
pleased to meet you
vrfy rflor
250 2.1.5 Roberto Flor <rflor@odino.inf.unitn.it>
mail from:<fmagagni@inf.unitn.it>
250 2.1.0 <fmagagni@inf.unitn.it>... Sender ok
rcpt to:<franz@ewave.it>
250 2.1.5 <franz@ewave.it>... Recipient ok
rcpt to:<rflor@odino.inf.unitn.it>
250 2.1.5 <rflor@odino.inf.unitn.it>... Recipient ok
data
354 Enter mail, end with "." on a line by itself
date: 21 febbraio
ciao come va? sto facendo una conversazione con il server STMP
dell'università
.
250 2.0.0 g0VFYZZ08321 Message accepted for delivery
quit
221 2.0.0 odino.inf.unitn.it closing connection
```

I comandi principali sono (porta 25):

Hello (helo)

Questo comando serve per identificare il client (sender-SMTP), l'argomento contiene l'host name dello stesso client. Il server si identifica nella fase di connessione (con l'apposito banner di saluto, un biglietto da visita); il server risponderà con un messaggio di OK, in questo caso i buffers e le tabelle sono tutti puliti.

Mail (mail)

Questo comando è utilizzato per inizializzare la vera e propria transazione. In questo caso si deve indicare il mittente (reverse-path), a cui è riferita la creazione della e-mail; ci possono essere più host mittente a cui è riferita la e-mail, che in caso di mancata consegna, riceveranno l'annotazione.
Questo comando pulisce i buffer, e inserisce nello stesso buffer dedicato al mittente, l'e-mail del client.

Recipient (rcpt)

Questo comando è usato per identificare il destinatario della Email. Si possono creare più destinatari con l' uso multiplo di questo comando.
La forward-path consiste in una lista opzionale di hosts destinatari, ma almeno una mail-box obbligatorio. Può anche capitare che il dominio della Email non venga riconosciuto, in questo caso si verificherà un errore (550 mail box unaviable).

Data (data)

Una volta digitato questo comando il server capirà che i successivi dati fanno parte del body della e-mail; questo comando, inoltre, prepara l'apposito buffer al ricevimento; c'è

da ricordare che il campo in cui vi sono i dati può essere composto solo dei 128 caratteri ASCII, in pratica tutti i caratteri di una tastiera.
I dati della Email terminano quando vi è una linea con solo un punto, quindi bisogna seguire questa sequenza"<CRLF>.<CRLF>", questo rappresenta la fine del body. La fine della Email data indica al server di poter processare immediatamente i dati nei buffers raccolti durante la transazione. Al termine di questa operazione i buffers verranno puliti. Se il processo è andato a buon fine il server risponderà con un OK, se invece fallisce si spedirà un'e-mail di reply.
data<CRFL>
Start mail input; end with <CRLF>.<CRLF>
ciao come va....
.
Mail Accepted

Ogni volta che la mail deve passare per un server-SMTP, quest'ultimo lascerà negli header della stessa un così detto "Time Stamp", con varie informazioni sull'host, etc., Una volta arrivati all'ultimo server, quest'ultimo scriverà all'inzio la c.d. return path, con la mail-box del mittente.

Send (send)
Questo comando serve per spedire una Email ad uno o più terminali. Questo comando di seguito necessita la Email del mittente. Questo comando pulisce i buffers, e inserisci le informazioni nel buffer dedicato alla Email del mittente

Send or mail (soml)
Questo comando serve per spedire un messaggio ad un terminale, oppure alla sua mail-box. Se il terminale è attivo allora riceverà il messaggio, se no verra inviato nella sua mail-box.

Send and mail (saml)
Questo messaggio, invece, spedisce ad ambedue, sia al terminale che alla sua mailbox, ricordiamo che alla fine di questo comando c'è sempre la pulizia dei buffers e il loro riempimento.

Reset (rset)
Questa procedura indica la volontà di voler abbortire la transazione Email, ogni buffer verrà immediatamente ripulito e si ricomincerà da capo, il server dovrà spedire un OK di conferma.

Verify (vrfy)
Questo comando vuole avere la verifica della e-mail di un utente; non si portano modifiche ai buffers.

Expand (expn)
Questo comando vuole ricevere conferma che il suo argomento (quello che si inserisce dopo il comando), faccia parte di una mailing list, se è così si restituirà gli appartenenti alla mailing list; si restituirà il nome dell'utente e della sua mailbox, come il comando VERIFY non si apportano modifiche ai buffer.

Help (help)
Questo comando restituisce la lista degli aiuti forniti dal server; di solito se si vogliono sapere informazioni su un particolare comando, quest'ultimo diventa l'argomento dell'HELP, questo comando non causa effetti ai buffers.

Noop (noop)
Questo comando specifica di non poter fare nessuna operazione, il server spedirà un OK di conferma; non si apportano modifiche ai buffers.

Quit (quit)
Questo comando è la fine della transazione, il server risponderà con un messaggio di conferma e chiuderà la transazione stessa.

Turn (turn)
Questo comando è molto particolare, perché serve per invertire i ruoli tra il server ed il client. Se il client spedisce questo comando il server lo può rifiutare, oppure può rispondere con un messaggio di conferma e diventare lui un client-SMTP.

I "Reply number":

211 System status, or system help reply
214 messaggio di Help
220 <domain> Servizio pronto
221 <domain> Servizio ha chiuso il canale
250 Richieste di azione completata, OK.
251 Utente non locale; si spedirà a <forward-path>
354 Inizia l'input dei dati; finisce con <CRLF>.<CRLF>
421 Servizio non avviabile
450 Richiesta di azione non avviabile [E.g., mailbox piena]
451 Richiesta di azione abbortita: errore locale durante il processo
452 Richiesta di azione abbortita: spazio di sistema insufficiente
500 Errore di sintassi, comando non riconosciuto
501 Errore di sintassy nei parametri o negli argomenti
502 Comando non implementato
503 Cattiva sequenza del comando
504 Parametri del comando non implementati
550 Richiesta di azione non presa: mailbox inavvibile
551 Utente non locale; perfavore prova <forward-path>
552 Richiesta di azione abbortita: si eccede l'allocazione di spazio
553 Richiesta di azione non presa: nome della mailbox non permesso
554 Transazione fallita

POP3 - Post Office Protocol 3 (RFC 1225)

La maggior parte dei sistemi di posta elettronica sfrutta il protocollo POP per il trasferimento dei messaggi tra la propria mailbox e il client di posta.
Del protocollo POP esistono tre versioni diverse, ma l'unica di fatto utilizzata è la versione 3 (POP3).
Fornisce un magazzino messaggi che trattiene gli e-mail ricevuti fino a che l'utente effettua il log-on e li scarica. POP3 è un sistema molto semplice con poca selettività. Tutti i messaggi in arrivo e gli attachment devono essere scaricati insieme. POP3 accetta messaggi formattati ed inviati tramite protocollo messaggi SMTP.

Conversazione POP3

Come abbiamo visto prima la conversazione tra client e server Smtp, ora vediamo la conversazione tra client e server Pop3 (sulla porta 110)

La conversazione POP3 avviene in tre fasi.

- Autorizzazione (AUTHORIZATION): il client si identifica e il server verifica che abbia le dovute autorizzazioni per accedere alla casella postale. Effettuata la verifica, il server accede alla casella postale del client, bloccandone l'utilizzo da parte di altri programmi:
- Transazione (TRANSACTION): questa è la fase nella quale la posta viene effettivamente scaricata; dopodiché il client lancia il comando QUIT.
- Aggiornamento (UPDATE): nella quale il server elimina dalla casella postale tutti i messaggi per i quali era stata richiesta la cancellazione nella fase precedente, toglie il blocco alla casella postale, e chiude la connessione

Esempio di conversazione Pop3 su inf.unitn.it

```
+OK QPOP (version 2.53) at cevedale.inf.unitn.it starting.
user fmagagni
+OK Password required for fmagagni.
pass #######
+OK fmagagni has 1 message (646 octets).
stat
+OK 1 646
list
+OK 1 messages (646 octets)
1 646
.
noop
+OK
retr 1
+OK 646 octets
//Header

//Body

.
list
+OK 1 messages (646 octets)
1 646
.
```

```
dele 1
+OK Message 1 has been deleted.
list
+OK 0 messages (0 octets)
.
rset
+OK Maildrop has 1 messages (646 octets)
list
+OK 1 messages (646 octets)
1 646
.
quit
+OK Pop server at cevedale.inf.unitn.it signing off.
```

Quando il destinatario vuole scaricare la posta, deve stabilire una connessione TCP sulla porta 110 del server POP3. Stabilita la connessione, il server si presenta e i due cominciano a scambiarsi comandi e risposte fintanto che la posta non è stata scaricata; se richiesto, e in genere è così, la posta scaricata viene cancellata dalla casella postale, quindi la connessione viene chiusa.
I comandi non sono altro che delle stringhe di caratteri in formato ascii (Byte) terminate dalla sequenza di chiusura (13,10). Ogni stringa contiene una parola chiave più eventualmente una serie di parametri. Come separatore viene usato un singolo carattere di spaziatura. In genere, le parole chiave sono formati da tre o quattro caratteri e non ha importanza se sono spedite in maiuscolo o minuscolo. Viceversa i parametri possono essere lunghi fino a 40 caratteri e la cassa del testo viene rispettata. Anche le risposte a un comando sono stringhe di caratteri ascii terminate dal solito (13, 10). Tuttavia, esse contengono prima un indicatore di stato e poi una parola chiave con eventuali ulteriori informazioni, per un massimo di 512 caratteri, spazi e caratteri di controllo compresi. Inoltre, è possibile che a fronte di un comando arrivi più di una risposta, o meglio una risposta formata da più stringhe. In tal caso si usa la tecnica dove l'ultima stringa deve essere una sequenza formata dal carattere "." inserito fra due sequenze di chiusura (13, 10). Eventuali punti a inizio riga che facciano parte del testo vengono raddoppiati in partenza e rinormalizzati all'arrivo secondo un classico meccanismo di trasparenza. Attualmente esistono solo due possibili indicatori di stato, e precisamente +OK e -ERR, da inserire rigorosamente in maiuscolo.
La gestione degli errori è essenziale. Se il client spedisce una richiesta errata o non supportata, oppure una richiesta valida al momento sbagliato, il server si limita a rispondere con –ERR. Il client non ha modo di sapere perchè la richiesta è stata respinta. Nel caso poi il server non riceva richieste dal client da più di 10 minuti, la connessione viene chiusa immediatamente senza passare ovviamente per la fase di aggiornamento, in modo da evitare di cancellare missive non ancora scaricate.

Il protocollo POP3 prevede svariati modi con cui il client può identificarsi. Il principale è descritto nello stesso RFC del POP3, altri possono essere trovati nell'RFC 1734. L' identificazione utilizza i comandi USER e PASS. In pratica, il client spedisce prima il comando USER seguito da un nome che il server deve conoscere e che corrisponde a una casella postale ben definita. Se il server accetta il nome allora il client spedisce il comando PASS seguito da una parola d'ordine associata al nome in questione. Nel caso viceversa il nome non sia riconosciuto, il client può ricominciare il processo di autenticazione utilizzando sempre il comando USER o un altro dei meccanismi previsti, oppure emettere il comando QUIT per chiudere la connessione. Ovviamente il comando

QUIT può essere emesso anche se il server accetta l'identificazione. Da notare che la parola d'ordine può essere una e una sola, per cui se il comando PASS è seguito da più parole, gli spazi bianchi saranno considerati parte della parola d'ordine e non separatori di più argomenti.

Una volta entrati nella fase di transazione, il server assegna a ogni messaggio un identificativo numerico a partire dal numero uno. Il client, viceversa, può iniziare a spedire richieste utilizzando una serie di comandi a sua disposizione. I due comandi più importanti sono: RETR e DELE. Il primo, permette di ricevere un messaggio e richiede come parametro il numero del messaggio; il secondo, permette di marcare un messaggio della casella postale affinché venga cancellato nella fase di aggiornamento. In generale, una buona politica sarebbe quella di far seguire a ogni comando RETR andato a buon fine un comando DELE sullo stesso messaggio, in modo da svuotare la casella postale. Questo per evitare di riempire in breve tempo la casella creando problemi di spazio disco sul server. Alcune implementazioni permettono di fissare dei limiti per le dimensioni fisiche della casella postale o sul numero di messaggi che può contenere, lasciando all'utente finale il compito di cancellare i messaggi dal server esplicitamente o utilizzando un'opzione di cancellazione a scadenze prefissate che, per esempio, rimuove automaticamente tutti i messaggi residenti da più di 10 giorni. Simili politiche sono però dannose perchè comportano il rischio di causare la perdita di vecchi messaggi non ancora letti o l'impossibilità di ricevere nuovi messaggi semplicemente perchè si è raggiunta la soglia massima. Il meccanismo più pulito consiste invece nel far cancellare al server automaticamente tutti, e solo, i messaggi scaricati con successo dalla casella postale nella fase di aggiornamento. Così facendo l'utente ne conserva comunque una copia sul sistema locale e il server ha più spazio a sua disposizione

I comandi principali sono (porta 110):

User [name]
Richiesta d'accesso alla casella postale il cui nome è fornito come parametro. Deve essere seguita dal comando PASS.

Pass [psw]
Parola d'ordine per la richiesta d'accesso. Segue sempre il comando USER.

Quit
Richiesta di fine lavoro. Se emessa durante la fase di autorizzazione, la connessione è chiusa senza che ulteriori operazioni siano effettuate. Se invece è emessa durante la transazione, il server entra nella fase d'aggiornamento, cancella tutti i messaggi marcati dal comando DELE e rilascia il blocco sulla casella postale.

Stat
Richiesta d'informazioni sullo stato della casella postale. La risposta minima è del tipo +OK num dim dove num è il numero di messaggi in casella ed dim sono le dimensioni complessive della posta in ottetti. Se lo si desidera, è tuttavia possibile aggiungere ulteriori informazioni.

List [msg]
Richiesta d'informazioni su uno o su tutti i messaggi in casella. Se è fornito un numero di messaggio, l'implementazione minima sarà +OK msg dim dove msg è l'identificativo del

messaggio e dim le dimensioni del messaggio in ottetti. Altre informazioni sono opzionali. Se invece è spedito il solo comando, il server restituisce +OK seguito da una serie di linee che riportano le informazioni suddette per ogni singolo messaggio in casella.

Retr n
Richiesta di spedire il messaggio specificato. Se non ci sono errori, il messaggio è spedito una linea alla volta dopo l'+OK. Viceversa l'intestazione del messaggio è sempre spedita per intero. Il messaggio non dovrebbe essere stato marcato per la cancellazione. Alla fine viene spedita la sequenza di chiusura (10,13).(10,13)

Dele n
Richiesta di marcare il messaggio da eliminare. La cancellazione effettiva verrà effettuata nella fase di aggiornamento.

Noop
Richiesta di emettere una risposta affermativa, cioè +OK.

Rset
Richiesta di eliminare tutti i marchi di cancellazione assegnati a seguito di comandi DELE. I messaggi in questione non saranno quindi più cancellati nella fase di aggiornamento.

IMAP - Internet Messaging Access Protocol (RFC 1064)

E' il nuovo concorrente del protocollo POP, di gran lunga più evoluto ma ancora poco diffuso in quanto la quasi totalità dei servizi Internet è basata su POP3.
Il vantaggio di IMAP è di consentire alcune manipolazioni avanzate sulla posta in entrata prima ancora di prelevarla dal server.
Oltre a quanto fatto dal POP3, il protocollo IMAP consente inoltre di:

1. rinominare la propria casella elettronica;
2. cancellare singoli messaggi senza essere costretto a prelevarli;
3. leggere le intestazioni dei messaggi senza doverli prelevare interamente;
4. addirittura prelevare solamente delle porzioni dei messaggi.

Si tratta quindi di un protocollo di posta assolutamente più sviluppato del POP3.
E' un mail server standard che ci si aspetta verrà utilizzato in modo ampio su Internet. Fornisce un magazzino messaggi per trattenere gli e-mail in arrivo fino a che l'utente effettua il log-on e li scarica sul proprio PC. IMAP4 è la versione più recente.
Imap è più sofisticato del mail server POP3 (Post Office Protocol). I messaggi possono infatti essere archiviati in cartelle, le mailbox possono essere condivise e un utente può accedere più mail server. C'è inoltre una migliore integrazione con la tecnologia MIME che viene usata per effettuare gli attachment ai messaggi. Sia IMAP che POP3 accettano messaggi formattati tramite SMTP che siano stati inviati attraverso Internet.

Pop3 vs Imap – 2 protocolli a confronto

Il protocollo POP3 permette una gestione molto semplificata della propria casella postale, o mailbox, sufficiente in genere per la posta personale, ma divenuta, con la diffusione della posta elettronica, poco adatta a quella aziendale che necessita di funzioni più sofisticate e di una maggiore flessibilità per quanto riguarda l'utilizzo dei meccanismi di autenticazione. Per rispondere a queste esigenze è stato creato l'IMAP4, cioè la quarta versione dell'Internet Message Access Protocol. Questo protocollo permette di accedere alla posta memorizzata sul server fornendo, in aggiunta, tutta una serie di funzioni che consentano di creare, rimuovere e rinominare le caselle postali, definire gerarchie di caselle postali per una migliore classificazione della posta, mantenere traccia di quali messaggi sono vecchi, quali recenti, quali sono stati già letti e quali no, associare ai messaggi uno o più marcatori per una successiva elaborazione, gestire i messaggi sia nel formato ASCII (RFC 822) sia MIME, effettuare ricerche per chiave sui vari messaggi o su un loro sottoinsieme, rimuovere permanentemente i messaggi e accedere alle caselle postali sia in lettura e scrittura sia in sola lettura. In sintesi, si tratta di un protocollo alquanto complesso e abbastanza completo per la gestione remota della posta elettronica. Ovviamente, come già il POP3, l'IMAP4 si occupa solo della gestione della posta arrivata, e non della sua spedizione o della trasmissione in Rete.
A quello ci pensa l'SMTP. Inoltre, l'IMAP4 supporta un solo server anche se è stato proposto un meccanismo, detto IMSP, per supportare più server IMAP4. L'IMAP4 può essere implementato solo su un protocollo di trasferimento dati affidabile, come, per esempio, il TCP. Non quindi sull'UDP. La porta TCP assegnata all'IMAP4 è la 143.
Il protocollo imap 4 inoltre è molto utile in due occasioni:

- Quando si vuole leggere la posta da computer diversi o comunque da posti diversi, senza doversi portare dietro il lettore di posta, infatti grazie al fatto che la posta rimane sul Server, e vi rimangono anche le informazioni relative a quali messaggi abbiamo letto, possiamo avere la situazione "posta" sotto controllo ovunque ci troviamo.

- Il secondo caso è quello più utile per gli utenti; riguarda l'intasamento della casella di posta da parte di grossi messaggi, in questi casi risulta davvero svantaggioso scaricare la posta con il normale lettore di posta; il problema si risolve collegandosi alla propria casella di posta con imap4, vedere quali sono i messaggi che la bloccano decidere se scaricarli uno per volta oppure eliminarli.

NNTP – Network News Transfer Protocol

E' giusto soffermarsi un momento anche sul protocollo NNTP, utilizzato per la gestione delle News e dei NewsGroup; non usa le Mailing List ma memorizza i "post" in directory speciali aggiornate periodicamente.
L'NNTP risponde alla porta 119 ed utilizza esclusivamente l'ASCII.

I comandi da principali sono (porta 119):

List: Lista dei gruppi di discussione.
Listgroup: lista degli articoli nel gruppo specificato come parametro del comando.
Article id: richiesta del trasferimento del messaggio identificato da "id".

Xgtitle: visualizzazione degli argomenti discussi in ciascun gruppo.
Xhdr: visualizza i soggetti dei messaggi presenti nel gruppo selezionato.
Quit: chiude la connessione.

LA SICUREZZA

La posta elettronica, al contrario di quanto si possa pensare, presenta molti aspetti critici nel capitolo sicurezza.
Una e-mail, nella sua breve vita, incontra una grande quantità di buchi; innanzitutto nei protocolli di trasmissione e di ricezione e poi nel suo passaggio per il server.

Nel protocollo SMTP,la cosa curiosa, e sicuramente per molti inaspettata, è che le specifiche originali di questo protocollo non prevedono nessuna forma di autenticazione. Ancora oggi la maggior parte dei server SMTP funzionano in questo modo e ciò significa che chiunque può utilizzarli per spedire messaggi. I dati relativi al mittente (nome, cognome, indirizzo e-mail ecc.) sono inseriti, a cura del mittente stesso, nel proprio programma di posta e quindi possono tranquillamente essere *falsi*.
Per fortuna nelle e-mail sono presenti tutta una serie di indicazioni aggiunte dai vari server attraversati dal messaggio. Analizzandole è possibile almeno capire da quale server proviene la missiva e quale indirizzo IP aveva l'utente che ha effettuato la spedizione. Analizzando l'elenco dei collegamenti presso l'ISP, si può arrivare all'utente e in molti casi anche alla linea telefonica chiamante. Queste analisi vengono però effettuate solo dalla magistratura, nel caso in cui si presenti una denuncia per qualche reato commesso via e-mail.
Inoltre, visto che la posta viaggia in chiaro, chiunque abbia accesso completo ai computer attraversati dal messaggio (e ancora di più ai nodi di rete, specie quelli più importanti), può leggere il contenuto dei pacchetti dati e riassemblarli in modo opportuno, per accedere al contenuto delle e-mail.

Nel protocollo POP3, che, al contrario del SMTP, prevede una forma di autenticazione la comunicazione avviene in chiaro, per cui i pacchetti che il server ci spedisce possono essere intercettati su uno qualsiasi dei computer attraversati. In questo modo può essere carpita perfino la stessa password.

I problemi di sicurezza nell'ambito della posta elettronica sono materialmente difficili da risolvere, si può però ridurre il problema usando connessioni protette con l'ISP tramite SMTP+SSL e POP+SSL e utilizzando le tecniche di criptazione e di sicurezza fornite dal PGP e dal S/MIME.

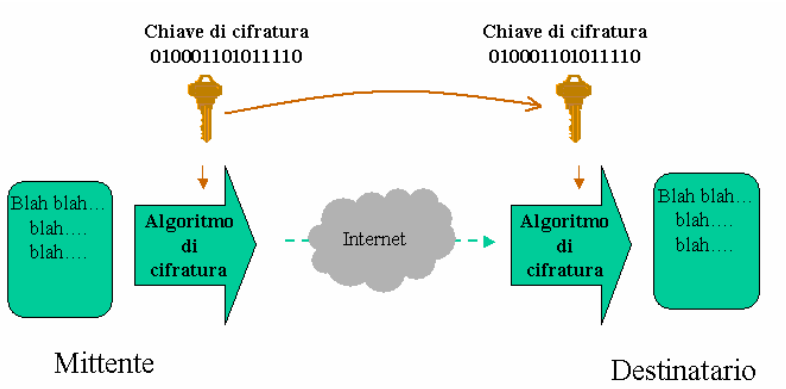
Tecniche crittografiche

Algoritmi

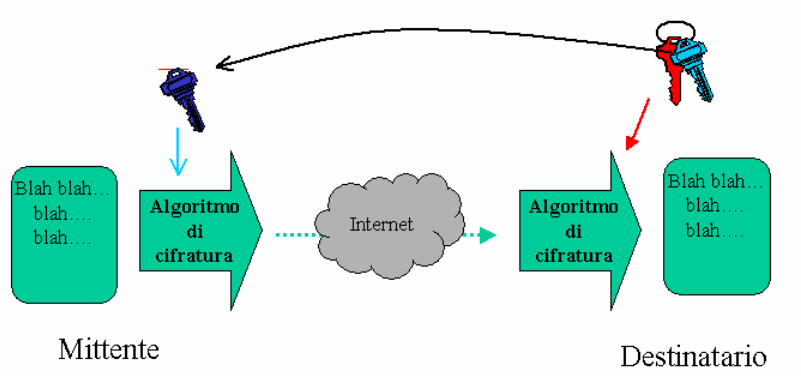
Le categorie di algoritmi di crittografazione sono fondamentalmente due: convenzionale e a chiave pubblica.

- *Crittografia convenzionale* (o *simmetrica*)
  In questo tipo di crittografia il mittente e il destinatario devono condividere una

chiave: un'informazione che permette la crittografazione e la decrittazione dei messaggi. Ovviamente il problema è come trasmettere in modo sicuro la chiave segreta. Tra gli algoritmi più comuni DES e IDEA.



- *Crittografia a chiave pubblica* (o *asimmetrica*)
  In questo caso vengono utilizzate due chiavi: una privata, che rimane nota al solo mittente e una pubblica, che viene divulgata. Un messaggio crittografato con una delle due chiavi può venire decrittato con l'altra. Ovviamente deve essere praticamente impossibile risalire dalla chiave pubblica a quella privata. Tra gli algoritmi più comuni RSA e Diffie-Hellmann.



In pratica, dato che gli algoritmi simmetrici sono circa due ordini di grandezza più veloci di quelli asimmetrici, si utilizzano entrambi i metodi contemporaneamente. Per ogni messaggio viene generata una chiave random con cui viene crittografato, con un algoritmo simmetrico, e infine questa viene a sua volta crittografata con la chiave pubblica del destinatario e spedita insieme al messaggio.

Message digest

Message digest (o *one-way hash*) sono stringhe a lunghezza fissa (piccola) calcolate a partire dal messaggio originale. La loro caratteristica è che è praticamente impossibile risalire da loro al messaggio e anche trovare due messaggi diversi con lo stesso digest. L'algoritmo più diffuso è MD5, anche se recentemente sono stati scoperti alcuni problemi di sicurezza.
Vengono utilizzati per garantire l'integrità dei messaggi ricevuti.

Firme digitali

Le firme digitali servono per attribuire con certezza il mittente di un messaggio. Si ottengono crittografando con la chiave segreta del mittente un message digest. In questo modo si ottiene la duplice sicurezza dell'identità del mittente e che il messaggio non è stato alterato durante il trasporto.

Certificati

I certificati servono per verificare l'identità dei corrispondenti: tra le altre informazioni contengono la loro chiave pubblica e sono firmati da un agente di cui le parti si fidano e che garantisce in questo modo la loro identità. Questo agente viene detto **Certificate Authority** (**CA**). In genere i certificati seguono lo standard X.509.
Una CA può emettere un certificato anche per un'altra CA, in questa maniera si possono creare delle catene gerarchiche di certificati (per PGP la cosa è diversa, dato che ogni utente riveste il ruolo della propria CA). Controllare la validità di un certificato significa quindi risalire la sua catena di autorizzazioni fino a raggiungere quella di una CA di cui il soggetto si fida. I browser in circolazione sono preconfigurati per dare fiducia ad un certo numero (modificabile, ovviamente) di CA ben note, ad es. VeriSign/RSA.
Tra le responsabilità di una CA, oltre ovviamente quella di accertarsi dell'identità dei richiedenti prima di firmare i loro certificati, c'è quella di gestire le loro scadenze, i loro rinnovi e il mantenimento di liste di certificati non più validi (*Certificate Revocation List*s o *CRL*s).

PGP - Pretty Good Privacy

Pretty Good Privacy (PGP) è un programma di crittografia a chiave pubblica sviluppata da P. Zimmerman che utilizza RSA, IDEA e MD5 per firmare e crittografare i messaggi testuali.
Le versioni reperibili, non dovendo sottostare alle restrizioni di esportazione dagli Stati Uniti, fanno uso di chiavi di lunghezza più che adeguata per qualunque esigenza di sicurezza. Gli ultimi prodotti si adeguano alla proposta di standard PGP/MIME, in concorrenza con S/MIME.
Si basa su due RFC:

- RFC 1991: *PGP Message Exchange Formats*,
- RFC 2015: *MIME Security with Pretty Good Privacy*.

Il PGP non usa una struttura gerarchica di certificati. Ogni utente mantiene la lista di chiavi pubbliche dei suoi corrispondenti (viene chiamata *keyring*), ognuna delle quali viene firmata con la propria chiave privata.
È possibile scambiarsi i *keyring*: alle chiavi importate (e quindi firmate dal proprietario) è possibile assegnare diverse gradazioni di "fiducia" che permettono di costruire il cosiddetto *web of trust* (l'equivalente della struttura gerarchica dei certificati).

Operatività

Al momento ci sono due versioni in uso: la vecchia (2.6.3i), disponibile su quasi tutte le piattaforme e la nuova (5.x) per Windows (tra poco Mac e Unix). Le chiavi generate dalla nuova versione (almeno quella reperibile in rete) non sono compatibili con quelle della vecchia, mentre è vero il vice versa.

Per garantire la massima compatibiltà è necessario utilizzare solamente la vecchia versione (2.6.3i) e rinunciare alla possibilità (quando esiste) di comporre mail MIME.

Un ipotetico scenario di utilizzo è il seguente:

- L'utente installa sulla propria piattaforma il software PGP e genera la coppia di chiavi.
- L'utente pubblicizza la propria chiave pubblica inviandola ad un key server e/o ai suoi corrispondenti.
- L'utente si procura la chiave pubblica dei suoi corrispondenti da un key server o direttamente (in entrambe i casi, sarà sua cura accertarsi che la chiave sia corretta, ad esempio telefonando al corrispondente e verificando il key fingerprint). Le chiavi vanno firmate e inserite nel proprio keyring.

S/MIME - Secure/Multipurpose Internet Mail Extensions

S/MIME (Secure/Multipurpose Internet Mail Extensions) è una proposta di standard per la crittografazione e firma dei messaggi di posta elettronica. Utilizza RSA, RC2 e MD5 per la crittografazione e firma e certificati digitali nel formato X.509. La versione 2 è descritta in due documenti:

- RFC 2311: *S/MIME Version 2 Message Specification*.
- RFC 2312: *S/MIME Version 2 Certificate Handling*.

Le implementazioni reperibili, dovendo sottostare alle restrizioni di esportazione dagli Stati Uniti, fanno uso di chiavi di lunghezza inadeguata (512 bit per RSA e 40 bit per RC2). (In un report di alcuni eminenti crittografi, Minimal Key Lenghts for Symmetric Ciphers to Provide Adequate Commercial Security, una chiave di 40 bit viene dichiarata decrittabile in una settimana facendo uso di un comune PC).

Per questo motivo non sembra probabile che la versione 2 venga accettata come standard IETF (è in preparazione la versione 3).

Operatività

Netscape Communicator (4.x) e Microsoft Outlook (fornito con Internet Explorer 4) sono entrambi in grado di generare messaggi di questo tipo (compatibili tra di loro). La gestione è molto buona: la crittazione e decrittazione viene eseguita in modo trasparente, come pure la gestione dei certificati dei destinatari, che vengono automaticamente scaricati dai messaggi ricevuti. Manca una gestione automatica delle CRL (è prevista invece, almeno per Netscape, per i certificati dei server web).
Il problema principale, prescindendo dalla bassa sicurezza, è legato alla necessità dell'esistenza di un certificato per ogni destinatario. Fintanto che tutti i possibili destinatari non saranno certificati da una qualche CA (situazione per ora ben lontana dall'essere realizzata) questo metodo si presta bene solo per la corrispondenza all'interno di una singola organizzazione, che può abbastanza facilmente organizzarsi come CA.
Un ipotetico scenario di utilizzo è il seguente:

- L'utente scarica nel proprio browser il certificato della CA.
- L'utente richiede interattivamente il proprio certificato, che gli viene spedito per e-mail. Una volta che lo ha caricato nel proprio browser è pronto per inviare mail

firmati e crittografati ad altri corrispondenti *muniti di un certificato* (non necessariamente della stessa CA: in questo caso è necessario procurarsi anche il certificato della CA).

- Lo scarico dei certificati dei corrispondenti può avvenire interattivamente o automaticamente alla ricezione del primo mail firmato dallo stesso.