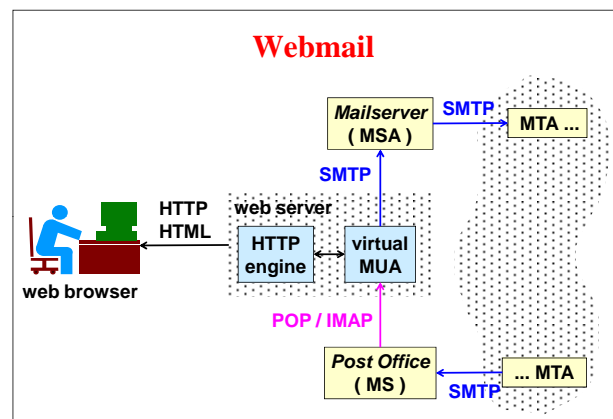
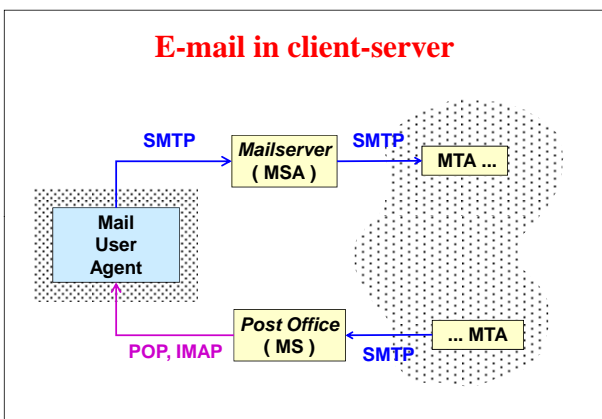
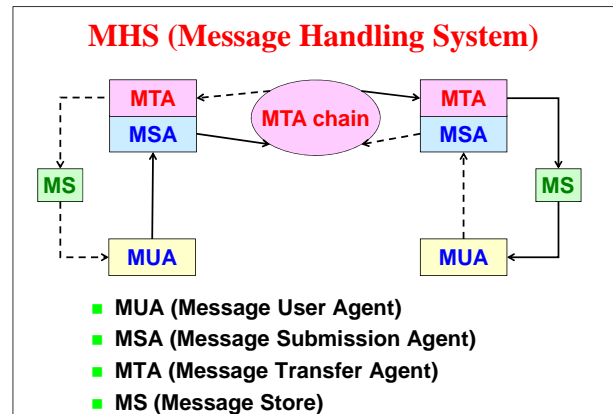


Sicurezza della posta elettronica

Antonio Lioy
<lioy @ polito.it>

Politecnico di Torino
Dip. Automatica e Informatica



- Protocolli e porte**
- SMTP (Simple Mail Transfer Protocol)
 - 25/tcp (MTA)
 - 587/tcp (MSA)
 - POP (Post Office Protocol)
 - 110/tcp
 - IMAP (Internet Message Access Protocol)
 - 143/tcp

Un esempio SMTP / RFC-822

```
telnet duke.colorado.edu 25
Trying .....
Connected to duke.colorado.edu
Escape character is '^]'
220 duke.colorado.edu ...
HELO leonardo.polito.it
250 Hello leonardo.polito.it ... Nice to meet you!
MAIL FROM: cat
250 cat ... Sender ok
RCPT TO: franz
250 franz ... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
```

From: cat@athena.polito.it (Antonio Lioy)
 To: franz@duke.colorado.edu
 Subject: vacanze
 Ciao Francesco,
 ti rinnovo l'invito a venirmi a trovare nelle tue
 prossime vacanze in Italia. Fammi sapere
 quando arrivi.
 Antonio
 .
 250 Ok
 QUIT
 221 duke.colorado.edu closing connection
 connection closed by foreign host

Problematiche

- sistema connectionless (store-and-forward, anche solo per via dei record MX)
- MTA non fidati
- sicurezza del MS
- mailing-list
- compatibilità con l'installato
- soluzioni concorrenti:
 - Internet = PGP, PEM, MOSS, S/MIME
 - OSI = X.400

ESMTP

- Extended SMTP, definito in RFC-1869 e quindi incorporato (con SMTP) in RFC-2821
- non cambia il protocollo base ed il canale
- i client ESMTP devono presentarsi con: **EHLO hostname**
- se il server ricevente parla ESMTP, deve dichiarare le estensioni che supporta, una per riga, nella sua risposta all'EHLO

SMTP-Auth

- estensione di ESMTP definita in RFC-4954
- comando AUTH + opzioni di MAIL FROM
- per autenticare un client ...
- ... prima di accettarne i messaggi!!!
- utile contro lo spamming:
 - dopo il comando EHLO il server invia i meccanismi di autenticazione supportati
 - il client ne sceglie uno
 - viene eseguito il protocollo di autenticazione
 - se l'autenticazione fallisce, il canale viene chiuso

Esempio AUTH negativo

- il mailer non conosce (o non accetta) la modalità di autenticazione proposta dal client:

```
220 example.polito.it - SMTP service ready
EHLO mailer.x.com
250-example.polito.it
250 AUTH LOGIN CRAM-MD5 DIGEST-MD5
AUTH PLAIN
504 Unrecognized authentication type
```

AUTH: metodo LOGIN

```
220 example.polito.it - SMTP service ready
EHLO mailer.x.com
250-example.polito.it
250 AUTH LOGIN CRAM-MD5 DIGEST-MD5
AUTH LOGIN
334 VXNlcm5hbWU6 ----- Username:
bGlveQ== ----- lioy
334 UGFzc3dvcmQ6 ----- Password:
YW50b25pbw== ----- antonio
235 authenticated
```

AUTH: metodo PLAIN

- sintassi (RFC-2595):
AUTH PLAIN *id_pwd*_{BASE64}
- *id_pwd* è definito come:
[*authorize_id*] \0 *authentication_id* \0 *pwd*

```
220 example.polito.it - SMTP service ready
EHLO mailer.x.com
250-example.polito.it
250 AUTH LOGIN PLAIN
AUTH PLAIN bGlveQBsaW95AGFudG9uaW8=
235 authenticated
```

lioy \0 lioy \0 antonio

Protezione di SMTP con TLS

- RFC-2487 "SMTP Service Extension for Secure SMTP over TLS"
- STARTTLS = opzione di EHLO e comando
- se la negoziazione ha successo, si resetta lo stato del protocollo (si riparte da EHLO e le estensioni supportate possono essere diverse)
- se il livello di sicurezza negoziato è insufficiente:
 - il client invia subito QUIT ed esce
 - il server risponde ad ogni comando col codice 554 (refused due to low security)

Protezione di SMTP con TLS: esempio

```
220 example.polito.it - SMTP service ready
EHLO mailer.x.com
250-example.polito.it
250-8BITMIME
250-STARTTLS
250 DSN
STARTTLS
220 Go ahead
... TLS negotiation is started between client and server
```

Servizi di sicurezza per messaggi di e-mail

- **integrità (senza comunicazione diretta):**
 - il messaggio non può essere modificato
- **autenticazione**
 - identifica il mittente
- **non ripudio**
 - il mittente non può negare di aver spedito il mail
- **riservatezza (opzionale):**
 - messaggi non leggibili sia in transito sia nella casella postale

Sicurezza dell'e-mail - idee guida (I)

- **nessuna modifica agli attuali MTA**
 - messaggi codificati per evitare problemi nell'attraversare i gateway (es. Internet-Notes) oppure gli MTA non 8BITMIME
- **nessuna modifica agli attuali UA**
 - interfaccia utente scomoda
- **con modifica agli attuali UA**
 - interfaccia utente migliore

Sicurezza dell'e-mail - idee guida (II)

- **algoritmi simmetrici**
 - per la crittografia dei messaggi
 - con chiave di messaggio
- **algoritmi asimmetrici**
 - per crittografare e scambiare la chiave simmetrica
 - per la firma digitale
- **usare certificati a chiave pubblica (es. X.509) per il non-ripudio**
- **la sicurezza del messaggio si basa solo sulla sicurezza dell'UA del destinatario, non su quella degli MTA (non fidati)**

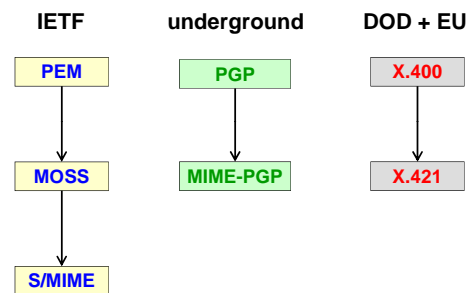
Tipi di messaggi sicuri

- **clear-signed**
 - msg in chiaro (perché tutti possano leggerlo) + firma
 - solo chi ha MUA sicuro può verificare la firma
- **signed**
 - msg + firma codificati (es. base64, uuencode)
 - solo chi ha MUA sicuro (o fa uno sforzo manuale) può decodificarli e verificare la firma
- **encrypted / enveloped**
 - msg cifrato + chiavi cifrate, codificato
 - solo chi ha MUA sicuro (e chiavi!) può decifrarlo
- **signed and enveloped**

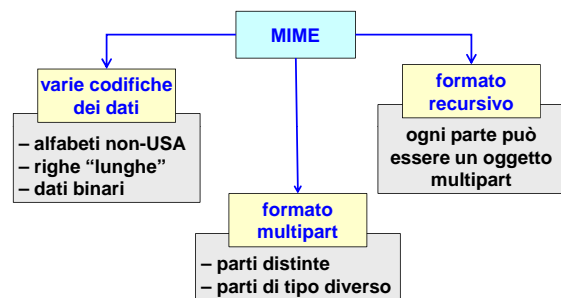
Messaggi sicuri: creazione

- **canonicalizzazione**
 - formato standard, indipendente da OS / host / net
- **MIC (Message Integrity Code)**
 - integrità ed autenticità
 - tipicamente: msg + { h(msg) } Kpri_sender
- **cifratura**
 - riservatezza
 - tipicamente: { msg } K_M + { K_M } Kpub_receiver
- **codifica**
 - per evitare alterazioni da parte degli MTA
 - tipicamente: base64, uuencode, binhex

Formati di posta elettronica sicura



MIME (Multipurpose Internet Mail Extensions)



Posta elettronica multimediale sicura (MOSS o S/MIME)

- firma digitale / cifratura con certificati X.509v3
- protegge messaggi MIME



RFC-1847

- estensioni MIME per la sicurezza dei messaggi
- per la firma digitale:


```
Content-Type: multipart/signed;
protocol="TYPE/STYPE";
micalg="...";
boundary="..."
```
- con due body part:
 - quella da proteggere (content-type: ...)
 - la firma (content-type: TYPE/STYPE)
- rischioso se un gateway altera il messaggio

S/MIME



- sicurezza di messaggi MIME
- promosso da RSA
- v2 pubblicato come serie di informational RFC:
 - RFC-2311 "S/MIME v2 message specification"
 - RFC-2312 "S/MIME v2 certificate handling"
 - RFC-2313 "PKCS-1: RSA encryption v.1-5"
 - RFC-2314 "PKCS-10: certification request syntax v.1-5"
 - RFC-2315 "PKCS-7: cryptographic message syntax v.1-5"

S/MIMEv3

- proposed standard IETF
- RFC-2633 "S/MIME v3 message specification"
- RFC-2632 "S/MIME v3 certificate handling"
- RFC-2634 "Enhanced Security Services for S/MIME"
- RFC-2314 "PKCS-10: certification request syntax v.1-5"
- RFC-2630 "CMS (Cryptographic Message Syntax)"

S/MIME: algoritmi

- message digest:
 - SHA-1 (preferito), MD5
- firma digitale:
 - DSS (obbligatorio)
 - digest + RSA
- scambio chiavi:
 - Diffie-Hellmann (obbligatorio)
 - chiave cifrata con RSA
- cifratura del messaggio:
 - 3DES con 3 chiavi
 - RC2/40

MIME type

- application/pkcs7-mime, usato per:
 - msg. cifrati (PKCS-7 envelopedData)
 - msg. firmati binari (PKCS-7 signedData) destinati solo ad utenti S/MIME perché il messaggio è all'interno della busta PKCS-7
 - msg. che contengono solo una chiave pubblica (= certificato; PKCS-7 signedData con dati nulli)
 - estensione standard: .p7m

MIME type

- multipart/signed
 - messaggi firmati destinati anche ad utenti non S/MIME (clear-signed)
 - il messaggio resta in chiaro
 - l'ultima parte MIME è la firma
 - la parte di firma ha estensione standard .p7s
- application/pkcs10
 - utilizzato per inviare una richiesta di certificazione ad una CA

S/MIME: esempio di firma

```
Content-Type: multipart/signed;
protocol="application/pkcs7-signature";
micalg=sha1;
boundary="-----aaaaa"

-----aaaaa
Content-Type: text/plain
Content-Transfer-Encoding: 7bit

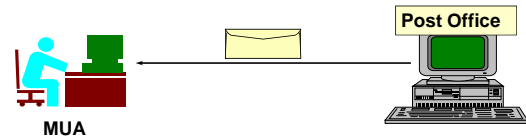
Ciao!
-----aaaaa
Content-Type: application/pkcs7-signature
Content-Transfer-Encoding: base64

MIIN2QasDDsDwe/625dBxgdhdsf76rHfrJe65a4f
fvVSW2Q1eD+SfDs543Sdwe6+25dBxfdeR0eDsrs5
-----aaaaa-
```

Naming in S/MIME

- per:
 - selezionare il certificato
 - verificare l'indirizzo del mittente
- in S/MIMEv2 consigliato **Email=** o **E=** nel DN nel certificato X.509, ma possibile usare l'estensione **subjectAltName** con codifica **rfc822**
- in S/MIMEv3 obbligatorio usare l'estensione **subjectAltName** con codifica **rfc822**

Protocolli di accesso al MS



- autenticazione dell'utente
- autenticazione del server
- riservatezza/integrità della posta
 - sul server
 - durante il trasferimento

Protocolli di accesso al MS

- **POP (Post-Office Protocol)**
 - POP-2 (RFC-937), POP-3 (RFC-1939)
autenticazione dell'utente mediante password in chiaro (!!!)
 - APOP
autenticazione dell'utente mediante sfida
 - K-POP
mutua autenticazione grazie ai ticket
- **IMAP (Internet Mail Access Protocol)**
 - username e password in chiaro
 - può usare OTP, Kerberos o GSS-API

Esempio POP-3

```
telnet pop.polito.it 110
+OK POP3 server ready <7831.84549@pop.polito.it>
USER lioy
+OK password required for lioy
PASS antonio
+OK lioy mailbox locked and ready
STAT
+OK 2 320
.....
QUIT
+OK POP3 server signing off
```

RFC-2595 (TLS per POP / IMAP)

- RFC-2595
"Using TLS with IMAP, POP3 and ACAP"
- prima si apre il canale e poi si negozia la sicurezza tramite un apposito comando:
 - STARTTLS per IMAP e ACAP
 - STLS per POP3
- client e server devono poter essere configurati per rifiutare *user* e *password*
- client confronta identità del certificato con l'identità del server

Porte separate per SSL/TLS?

- sconsigliate da IETF per i seguenti motivi:
 - implicano URL diverse (es. http e https)
 - implicano un modello sicuro / insicuro non corretto (es. è sicuro SSL a 40 bit? è non sicura un'applicazione senza SSL ma con SASL?)
 - non facile implementare "usa SSL se disponibile"
 - raddoppia il numero di porte necessarie
- ... ma presentano alcuni vantaggi:
 - semplicità di filtraggio sui firewall packet-filter
 - SSL con client-authentication permette di non esporre le applicazioni ad attacchi