PGP: beheer publieke sleutels

- maakt geen gebruik van certificaatautoriteiten
- niet zo strikt als bij S/MIME
- wie is echte eigenaar van publieke sleutel in key-ring?
- bedreigingen:
  - C zendt boodschap naar A met handtekening van B
  - A zendt boodschap naar B die C kan lezen
- mogelijkheden:
  - B heeft sleutel fysisch gekregen (floppy, …)
  - overdracht via mail en controle van fingerprint (SHA-1) via telefoon
  - sleutel verkregen via derde vertrouwde partij D
    - via PGP-certificaat = sleutel + handtekening van D
  - via CA (X509), doch niet gebruikt door PGP

PGP: “web of trust”

- in PGP is elke gebruiker CA
  - elke gebruiker kan sleutels tekenen van gebruikers die hij kent
- publieke sleutelhanger is “certificaat” database
- elke entry is een “certificaat”
- teken zelf vertrouwde sleutels
- vertrouw sleutels die anderen hebben getekend
  - indien je beschikt over een keten tot een vertrouwde persoon
- elke entry bevat 3 indicatoren voor vertrouwen
  - key legitimacy field
  - signature trust field
  - owner trust field
- indicatoren zitten vervat in *trust flag byte*

PGP: indicatoren van vertrouwen (1)

Key legitimacy field

- in hoever is dit een geldige sleutel voor die gebruiker
- hoe hoger het vertrouwensniveau, hoe sterker de binding tussen sleutel en userID
- waarden:
  - unknown or undefined trust
  - key ownership not trusted
  - marginal trust in key ownership
  - complete trust in key ownership
- wordt berekend door PGP op basis van de verzameling van *Signature trust fields*

PGP: indicatoren van vertrouwen (2)

- entry bevat 0 of meer handtekeningen van anderen
  - verzameld door de gebruiker
  - tekenen dit “certificaat”
- met elke handtekening is **Signature trust field** geassocieerd
- wordt berekend door PGP op basis van *owner trust field* van de sleutel gebruikt in de handtekening
- hoever vertrouwt eigenaar van key-ring die andere gebruiker om handtekeningen op sleutels te plaatsen
- waarden:
  - undefined trust
  - unknown user
  - usually not trusted to sign other keys
  - usually trusted to sign other keys
  - always trusted to sign other keys
  - this key is present in secret key ring (ultimate trust)

PGP: indicatoren van vertrouwen (3)

Owner trust field

- zelf bepaald door de eigenaar van de sleutelhanger
- hoe ver vertrouw je deze sleutel om andere sleutels te tekenen
- waarden:
  - undefined trust
  - usually not trusted to sign other keys
  - usually trusted to sign other keys
  - this key is present in secret key ring (ultimate trust)

PGP: bepaling van trust flag

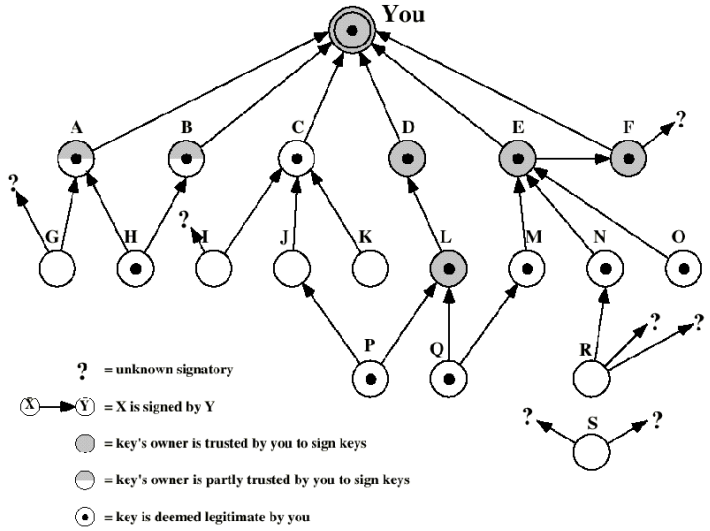
- publieke-sleutelhanger van A
- A voegt pub key toe:
  - sleutel zit ook private keyring ➔ ultimate trust
  - anders: PGP vraagt vertrouwensniveau:
    owner unknown, untrusted, marginally or completely trusted
- handtekeningen kunnen op elk moment aan entry toegevoegd worden:
  - zit eigenaar in pub keyring? ➔ SIGTRUST = OWNERTRUST
  - indien niet: ➔ unknown user
- KEYLEGIT berekend op basis van SIGTRUSTs
  - als 1 handtekening SIGTRUST==ultimate ➔ complete trust
  - anders: gewichtsfactoren opgeteld; result==1 ➔ complete trust
- PGP overloopt sleutelhanger periodiek

PGP: trust flag byte

(a) Trust Assigned to Public Key Owner (appears after key packet; user defined)	(b) Trust Assigned to Public Key/User ID Pair (appears after User ID packet; computed by PGP)	(c) Trust Assigned to Signature (appears after signature packet; cached copy of OWNERTRUST for this signator)
OWNER TRUST Field - undefined trust - usually not trusted to sign other keys - usually trusted to sign other keys - this key is present in secret key ring (ultimate trust) BUCKSTOP bit - set if this key appears in secret key ring	KEYLEGIT Field - unknown or undefined trust - key ownership not trusted - marginal trust in key ownership - complete trust in key ownership WARNONLY bit - set if user wants only to be warned when key that is not fully validated is used for encryption	SIGTRUST Field - undefined trust - unknown user - usually not trusted to sign other keys - usually trusted to sign other keys - always trusted to sign other keys - this key is present in secret key ring (ultimate trust) CONTIG bit - set if signature leads up a contiguous trusted certification path back to the ultimately trusted key ring owner

Beveiliging e-mail 35

PGP: web of trust



36

PGP: verklaring figuur

- sleutel van YOU is volledig vertrouwd (ultimate trust)
- D, E, F en L zijn volledig vertrouwd om andere sleutels te ondertekenen (OWNERTRUST=ultimate; bep dr YOU)
- elke sleutel die geheel of deels vertrouwd wordt, is getekend door gebruiker, behalve L (zou ook kunnen, zie E en F)
- sleutel H getekend door A en B (deels vertrouwd) wordt toch als legitiem bestempeld
- sleutel van N legitiem (getekend door E), maar N is niet vertrouwd om andere sleutels te tekenen
  ➔ R is geen legitieme sleutel, hoewel getekend door N
- wees-knopen kunnen ook bestaan (S)

S/MIME

- Secure/Multipurpose Internet Mail Extension
  - S/MIME wordt wellicht de industriestandaard
  - PGP voor persoonlijke e-mailbeveiliging
- voorzien in verschillende mailprogramma's
  - MS Outlook
  - Thunderbird
- gebruikt wel certificaten volgens X.509 en CA's
- S/MIME
  - beschrijft formaat boodschap
  - behandeling van certificaten
  - voegt beveiliging toe aan een MIME e-mail
  - MIME is uitbreiding op RFC-822

E-mail boodschap: RFC 822

- beschrijft formaat tekstboodschap
- bericht =
  - omslag: informatie overdracht en levering
  - inhoud: af te leveren object, boodschap (onderwerp van RFC)
- inhoud (ASCII) =
  - header-lijnen (info om de omslag te maken)
  - lege lijn
  - body
- headerlijn: *sleutelwoord: argumenten*
  (From, To, Subject, Cc, Date, Message-Id…)

RFC 822: voorbeeld boodschap

```
From - Fri May 21 12:59:36 1999
X-POP3-Rcpt: gvh087@turing.hogent.be
Received: from byron.hogent.be by turing.hogent.be with SMTP
          (1.38.193.5/16.2) id AA07260; Fri, 21 May 1999 11:36:32 +0200
Return-Path: <Paul.Vanolmen@hogent.be>
Received: from turing.hogent.be (turing.hogent.be [193.190.173.1])
          by byron.hogent.be (8.9.3/8.9.3) with SMTP id LAA30627
          for <Geert.Vanhoogenbemt@hogent.be>; Fri, 21 May 1999 11:52:36 -0400
Received: from [192.168.17.151] by turing.hogent.be with SMTP
          (1.38.193.5/16.2) id AA07257; Fri, 21 May 1999 11:36:31 +0200
Message-Id: <37452D88.9DF6BBC0@hogent.be>
Date: Fri, 21 May 1999 11:55:21 +0200
From: pvo888 <Paul.Vanolmen@hogent.be>
X-Mailer: Mozilla 4.51 [en] (Win98; I)
X-Accept-Language: en
Mime-Version: 1.0
To: Geert Van hoogenbemt <Geert.Vanhoogenbemt@hogent.be>
Subject: Thesis studenten
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-Mozilla-Status: 2001

Beste Geert,

Mij past het best Vrijdag 28/5 Tussen 9 en 10 u. Zoudt U naar het Labo
Regeltechniek / automatisering ( Practicum BME PE79B) kunnen komen.?

...
```

SMTP/822 tekorten en problemen

- kan geen binaire bestanden verzenden UUencode/UUdecode of radix64 is geen standaard
- alleen ASCII - tekens (7 bits)
- SMTP-servers kunnen mailberichten die te groot zijn afwijzen
- inconsistente vertaling tussen ASCII en EBCDIC door SMTP-gateways
- sommige SMTP-implementaties niet volgens RFC-standaard
  - veranderen, toevoegen, herschikken van CR en LF
  - verwijderen van tabs en spaties / wijzigen van tabs in spaties
  - opvullen van lijnen tot gelijke lengte

MIME

- Multipurpose Internet Mail Extensions
- compatibel met RFC-822
- verhelpen tekorten van SMTP (7 bits)
- RFC 2045 t/m 2049
- definiëren
  - 5 nieuwe headerlijnen met info over message body
  - inhoudsformaten van multimedia in e-mail
  - definitie van codeerschema's (tranfercodes)
- MIME boodschappen kunnen verzonder worden via ESMTP
  - Extended Simple Mail Transfer protocol
  - laat nu wel toe om binaire informatie te verzenden

MIME header fields

- MIME-Version: 1.0
  - boodschap is conform RFC's 2045 en 2046
- Content-Type:
  - beschrijft de gegevens in de body van de boodschap, zo dat mail-client boodschap kan tonen/verwerken
  - 7 hoofdtypes en 15 subtypes
  - text/plain, text/html, message/rfc822, multipart/mixed, …
- Content-Transfer-Encoding:
  - welke transformatie is gebruikt om body van boodschap klaar te maken voor mail transport
  - vb. base64

MIME header fields (2)

- 2 optionele headers, moeten niet ondersteund worden
- Content-ID:
  - identificeert de verschillende MIME eenheden
  - uniek ID in verschillende omgevingen
- Content-Description:
  - beschrijving van het object in body van boodschap
  - vooral zinvol indien niet leesbaar (audio, video, …)

MIME Content-Type (1)

- text
  - plain
  - enriched
  - html
- multipart
  - mixed — onafhankelijke delen, zelfde volgorde nodig bij ontvangst
  - parallel — cf.. mixed, volgorde niet belangrijk
  - alternative — verschillende versies van dezelfde informatie in volgorde van voorkeur
  - digest — cf. mixed, elk onderdeel is aparte RFC822 boodschap verzameling van mails in 1 boodschap
- *parameter van multipart*: boundary
  - bepaalt grens tussen delen, komt niet voor in delen
  - begint op nieuwe lijn en begint met twee mintekens
  - laatste grensaanduiding eindigt met twee mintekens

45

MIME Content-Type (2)

- message
  - rfc822 — body is zelf volledige boodschap conform RFC822
  - partial — deel van te grote opgesplitste boodschap assemblage is nodig
- image
  - jpeg, gif
- video
  - mpeg
- audio
  - basic
- application
  - postscript
  - extra subtypes in S/MIME

46

MIME Content-Transfer-Encoding (1)

geen echte coderingen, aanduiding voor type inhoud:

- 7bit
  - korte lijnen met ASCII-tekens
  - aanduiding, geen codering
  - veilig voor SMTP
- 8bit
  - korte lijnen met niet noodzakelijk ASCII-tekens
  - aanduiding, geen codering
  - niet veilig voor SMTP
- binary
  - zoals 8bit, maar met te lange lijnen voor SMTP-transport
  - aanduiding, geen codering
  - niet veilig voor SMTP, wel voor ESMTP

MIME Content-Transfer-Encoding (2)

2 echte coderingen gedefinieerd in MIME

- quoted printable
  - gebruikt indien meeste tekens afdrukbaar en ASCII zijn
  - codeert niet afdrukbare tekens in hexadecimale notatie
  - voegt line-breaks toe voor maximale lengte van 76 tekens
- base64
  - radix-64, armored ascii

MIME voorbeeld 1 (deel 1)

From - Mon Jan 21 13:17:57 2002
Return-Path: <jeroen.goethals@hogent.be>
Received: from byron.hogent.be (localhost.localdomain [127.0.0.1])
by casper.hogent.be (8.11.2/8.11.2) with ESMTP id g0LCKDU06357;
Mon, 21 Jan 2002 13:20:13 +0100
Message-ID: <00cf01c1a275$674b3560$f2adbec1@djer>
From: "Jeroen Goethals" <jeroen.goethals@hogent.be>
To: "Geert Van hoogenbemt" <Geert.Vanhoogenbemt@hogent.be>
Cc: "sven" <sven.ghyselinck@hogent.be>
References: <3C4BE5AF.F8245B78@hogent.be>
Subject: Re: vpn
Date: Mon, 21 Jan 2002 13:16:06 +0100
MIME-Version: 1.0

MIME voorbeeld 1 (deel 2)

Content-Type: multipart/alternative;
boundary="**----=_NextPart_000_00CC_01C1A27D.C8CF6010**"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.50.4807.1700
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4807.1700
Status:
X-Mozilla-Status: 8011
X-Mozilla-Status2: 00000000
X-UIDL: 3bf92da7000001aa

This is a multi-part message in MIME format.
Dit deel mag door client niet gelezen en getoond worden
------=_NextPart_000_00CC_01C1A27D.C8CF6010

MIME voorbeeld 1 (deel 3)

```
Content-Type: text/plain;
    charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Geert,
 …..
Jeroen
  ----- Original Message -----=20
  From: Geert Van hoogenbemt=20
  To: jeroen.goethals@hogent.be=20
  Sent: Monday, January 21, 2002 10:55 AM
  Subject: vpn

  Jeroen,
------=_NextPart_000_00CC_01C1A27D.C8CF6010
```


MIME voorbeeld 1 (deel 4)

```
Content-Type: text/html;
    charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<META http-equiv=3DContent-Type content=3D"text/html; =

……..

bereikbaar?<BR>Bedankt,<BR><BR>Geert</BLOCKQUOTE></BODY>
</HTML>

------=_NextPart_000_00CC_01C1A27D.C8CF6010--
```

MIME voorbeeld 2 (deel 1)

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A multipart example

Content-Type: multipart/mixed;
   boundary=unique-boundary-1

Deel dat een multipart message voorafgaat; mailclients die dit
formaat aankunnen, slaan dit deel over.  Indien je dit kan lezen
moet best worden uitgekeken naar een ander mailprogramma.
```


MIME voorbeeld 2 (deel 2)

```
--unique-boundary-1

   ...Some text appears here...
[Note that the preceding blank line means no header fields were
   given and this is text, with charset US ASCII. It could have been
   done with explicit typing as in the next part.]

--unique-boundary-1
Content-type: text/plain; charset=US-ASCII

This could have been part of the previous part, but illustrates
   explicit versus implicit typing of body parts.

--unique-boundary-1
```

MIME voorbeeld 2 (deel 3)

```
Content-Type: multipart/parallel; boundary=unique-boundary-2
--unique-boundary-2
Content-Type: audio/basic Content-Transfer-Encoding: base64

   ... base64-encoded 8 kHz single-channel mu-law-format audio
   data goes here ...

–-unique-boundary-2
Content-Type: image/jpeg Content-Transfer-Encoding: base64

   ... base64-encoded image data goes here....

--unique-boundary-2--

--unique-boundary-1
```


MIME voorbeeld 2 (deel 4)

```
Content-type: text/enriched

   This is <bold><italic>richtext.</italic></bold> <smaller>as
   defined in RFC 1896</smaller> Isn't it <bigger><bigger>cool?
   </bigger></bigger>

--unique-boundary-1
Content-Type: message/rfc822

From: (mailbox in US-ASCII)
To: (address in US-ASCII)
Subject: (subject in US-ASCII)
Content-Type: Text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: Quoted-printable

   ... Additional text in ISO-8859-1 goes here ...

--unique-boundary-1--
```

S/MIME

Secure Multipurpose Internet Mail Extensions

- qua functionaliteit vergelijkbaar met PGP
- alleen bruikbaar voor e-mail
- standaard ingebouwd in courante mail-clients
- beschrijft
  - formaat boodschap
  - behandeling van certificaten

PGP functies

Functie	Algoritmen	Beschrijving
digitale handtekening	DSS/SHA-MD5 RSA/SHA-MD5	Hash van boodschap met SHA-1, geëncrypteerd met DSS of RSA met private sleutel van verzender
encryptie van de boodschap	CAST-128, IDEA, RSA, AES, 3-DES / Diffie-Hellman	Boodschap gecodeerd met éénmalige sessiesleutel die geëncrypteerd wordt met DH of RSA-publieke sleutel van ontvanger en bij boodschap gevoegd.
compressie	ZIP	Boodschap kan gecomprimeerd worden
E-mail compatibiliteit	Radix-64 conversie	Geëncrypteerde boodschap kan geconverteerd worden tot ASCII-string
Segmentering		Boodschap opdelen in stukken en weer samenbrengen om maximum boodschaplengte te vermijden

S/MIME functies

- enveloped data
  - inhoud boodschap geëncrypteerd
  - samen met corresponderende encryptiesleutels voor een of meerdere ontvangers omgezet in base64
- signed data
  - digitale handtekening
  - hash van inhoud geëncrypteerd met private sleutel van ondertekenaar (alles base-64); vereist ontvanger met S/MIME
- clear-signed data
  - alleen handtekening gecodeerd in base64; ontvanger zonder S/MIME kan boodschap lezen
- signed and enveloped data
  - combinatie van de eerste twee gevallen

S/MIME algoritmen

- MOET vs ZOU MOETEN (RFC 2119)
  - zou moeten: aanbevolen, mag weggelaten als er redenen zijn
- te ondersteunen algoritmen:
  - berekenen van de message digest om te ondertekenen:
    - **SHA-1 (moet) en MD5 (ontvanger zou moeten)**
  - encrypteren van digest als handtekening:
    - **DSS (moet) en RSA (512-1024 bit sleutellengte, zou moeten)**
  - encrypteren van sessiesleutel samen met boodschap:
    - **Diffie-Hellman (moet, in praktijk ElGamal) en RSA (zou moeten)**
  - encrypteren van boodschap met éénmalige sessiesleutel:
    - **3DES (moet) en RC2 (40 bits sleutellengte, zou moeten)**
    - **AES (moet i.p.v. 3DES in RFC-draft)**

S/MIME boodschap

- S/MIME gebruikt nieuwe MIME-content (sub)types
  - (zie ook volgende dia's)
- ***application***-type draagt aanduiding *PKCS*
  - *public key cryptography standards* (uitgegeven door RSA)
- S/MIME beveiligt MIME-entiteit met
  - handtekening (signed & clear signed data)
  - encryptie (enveloped data)
  - beide
- beveiligde MIME-entiteit is
  - ofwel volledige boodschap (uitgezonderd header-lijnen)
  - ofwel één of meer onderdelen indien *multipart*-type

(zie verder)

Public-Key Cryptography Standards

PKCS #1: RSA Cryptography Standard (bevat ook #2 en #4)
PKCS #3: Diffie-Hellman Key Agreement Standard
PKCS #5: Password-Based Cryptography Standard
PKCS #6: Extended-Certificate Syntax Standard
PKCS #7: Cryptographic Message Syntax Standard
PKCS #8: Private-Key Information Syntax Standard
PKCS #9: Selected Attribute Types
PKCS #10: Certification Request Syntax Standard
PKCS #11: Cryptographic Token Interface Standard
PKCS #12: Personal Information Exchange Syntax Standard
PKCS #13: Elliptic Curve Cryptography Standard
PKCS #15: Cryptographic Token Information Format Standard

http://www.rsasecurity.com/rsalabs/PKCS

S/MIME functies

- enveloped data
  - inhoud boodschap geëncrypteerd
  - samen met corresponderende encryptiesleutels voor een of meerdere ontvangers
- signed data
  - digitale handtekening
  - hash van inhoud geëncrypteerd met private sleutel van ondertekenaar (alles base-64); vereist ontvanger met S/MIME
- clear-signed data
  - alleen handtekening gecodeerd in base64; ontvanger zonder S/MIME kan boodschap lezen
- signed and enveloped data
  - combinatie van de eerste twee gevallen

S/MIME content-types

Type	Subtype	SMIME parameter	Beschrijving
Multipart	Signed		clear-signed, 2 delen (boodschap en handtekening)
Application	pkcs7-mime	signedData	een getekende s/mime entiteit
	pkcs7-mime	envelopedData	een geëncrypteerde s/mime entiteit
	pkcs7-mime	degenerate signedData	een entiteit met uitsluitend publieke-sleutelcertificaten
	pkcs7-signature		het handtekeningdeel van een boodschap van het type multipart/signed
	pkcs10-mime		een boodschap om een certificaat te registreren

S/MIME boodschap: opbouw

- MIME-entiteit (boodschap zonder headers of deel van multipart) wordt normaal voorbereid (zoals mailclient ze zou verzenden)
- S/MIME maakt hiervan een PKCS-object
  - MIME-entiteit plus veiligheidsgegevens:
    - **identificatie algoritmen**
    - **certificaten**
- PKCS-object
  - wordt voorgesteld in de vorm BER: Basic Encoding Rules (X.209) en bestaat uit binaire gegevens
  - wordt geëncodeerd in *base64*
  - wordt behandeld als boodschap-body en ingepakt in MIME met aangepaste hoofdingen
- vaak behandeld met subtype *application/pkcs7-mime*

Opbouw Enveloped Data (1)

- doel: ecnryptie van de boodschap
- genereer random sessiesleutel K_s (voor vb. 3DES of AES)
- encrypteer de inhoud van de boodschap met K_s
- voor elke ontvanger: encrypteer K_s met publieke RSA-sleutel van de ontvanger
- voor elke ontvanger: maak een blok RecipientInfo:
  - identificatie van het algoritme voor encryptie van K_s
  - identificatie van het X.509 certificaat van ontvanger
  - geëncrypteerde K_s
- RecepientInfo-blokken + $\text{inhoud}_{\text{encrypted}}$ wordt omgezet via base64

Opbouw Enveloped Data (2)

Voorbeeld:

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

rfvbnj756tbBghyHhHUujhjhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6
7n8HHGghyHhHUujhjh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
OGhIGfHfQbnj756YT64V
```

- Ontvangst:
  - strip base64
  - decodeer sessiesleutel met private sleutel
  - decodeer bericht met sessiesleutel

Opbouw Signed Data (1)

in geval van één ondertekenaar:

- kies message digest algoritme (SHA-1 of MD5)
- bereken hashcode van de te ondertekenen inhoud
- encrypteer digest met private sleutel v ondertekenaar
- maak een blok SignerInfo:
  - X.509 certificaat van de ondertekenaar
  - identificatie van het algoritme voor de message digest
  - identificatie van het algoritme voor de encryptie van de message digest
  - geëncrypteerde message digest
- SignerInfo + te ondertekenen boodschap (+ reeks certificaten van CA tot ondertekenaar) naar base64

Opbouw Signed Data (2)

- digitale handtekening geëncrypteerd met private sleutel + inhoud
- één of meerdere ondertekenaars
- tegenpartij moet beschikken over S/MIME

Voorbeeld:

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

rfvbnj756tbBghyHhHUujhjhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6
7n8HHGghyHhHUujhjh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
OGhIGfHfQbnj756YT64V
```


Opbouw Clear signing (1)

- content type *multipart/signed*
- boodschap kan gelezen worden zonder S/MIME
- 2 delen: boodschap + aparte handtekening
- deel met handtekening zoals *signed data*
  - SignerInfo
  - leeg boodschapdeel

Opbouw Clear signing (2)

Voorbeeld:

```
Content-Type: multipart/signed;
        protocol="application/pkcs7-signature";
        micalg=shal; boundary=boundary42
--boundary42
Content-Type: text/plain

This is a clear-signed message.

--boundary42
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s
ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
4VQpfyF467GhIGfHfYT6H77n8HHGcrhyHhHUujhjh756tbB9HGTrfvbnj
n8HHGTrfvhjhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUuipfyF4
7GhIGfHfYT64VQbnj756
--boundary42--
```


Registratieaanvraag

- gebruiker of toepassing vraagt aan CA een X.509 certificaat
- aanvraag via *application/pkcs10-mime* entiteit
- certificationRequestInfo:
  - naam van te certificeren onderwerp
  - publieke sleutel
- aanvraag bevat
  - certificationRequestInfo
  - identificatie van het algoritme voor de publieke-sleutelencryptie
  - handtekening van de certificationRequestInfo mbv private sleutel van aanvrager

Boodschap met uitsluitend certificaten

- is antwoord op certificatieaanvraag
- kan CRL's bevatten
- content type: application/pkcs7-mime
  - smime-type: degenerate signed data
- zelfde stappen als signedData boodschap, maar geen boodschapinhoud en leeg signerInfo-blok

Certificatenbehandeling

- conform X.509v3
- mengen van strikte X.509-hiërarchie en PGP's *web of trust*
  - lokale mail clients moeten worden geconfigureerd met een lijst van vertrouwde sleutels en met CRL's (cfr PGP)
  - dus lokale verantwoordelijkheid om
    - **inkomende handtekeningen te verifiëren**
    - **uitgaande boodschappen te coderen en te tekenen**
  - certificaten worden getekend door CA's

Certificatenbehandeling (2)

Uit te voeren handelingen:

- generatie van de sleutels
  - Diffie-Hellman en DSS-paar moet geïmplementeerd zijn
  - RSA moet eventueel ook mogelijk zijn
- publieke sleutel moet bij een CA worden geregistreerd
- bewaren en opvragen van certificaten gebeurt lokaal of in directory voor beperkt aantal gebruikers

CA's: VeriSign, GlobalSign, www.thawte.com, ...