Beveiliging van e-mail

- e-mail
  - een van de meest gebruikte toepassingen
  - onafhankelijk van platform en architectuur
- inhoud boodschap is niet beveiligd
  - inspectie in transit
  - inspectie bij ontvangst door andere gebruiker
  - namaak via SMTP

Verbeteringen e-mailbeveiliging

Mogelijkheden:

- confidentialiteit
  - bescherming tegen het openbaar maken
- authenticatie van de afzender
- integriteit van de boodschap
  - beveiliging tegen wijziging
- niet kunnen ontkennen van de oorsprong
  - non-repudiation
  - afzender kan niet ontkennen dat hij de boodschap heeft verzonden

Beveiliging van e-mail: toepassingen

- PGP
- S/MIME
- beide gebruiken Radix-64 of Base64 conversie

Base64 – Radix64 – Armoured ASCII

- binaire tekens → afdrukbare tekens
- tekenset 6 bits ➔ 64 tekens
- 3 x 8 bits ➔ 4 x 6 bits
- bitpatroon aangevuld met 0 tot 6 bits
- string aangevuld met = tot 4 tekens

6 bits	code
0..25	A..Z
26..51	a..z
52..61	0..9
62	+
63	/

- **ABC D ➔ QUJD RA==**
  - hint: A = $(41)_{16}$
- **ABC DE ➔ QUJD REU=**

Gebruik van radix-64

- SMTP voorziet alleen in doorvoer van ASCII-tekens
- bericht met “speciale” tekens omgezet naar radix-64
- nadeel: grootte neemt toe met 33%

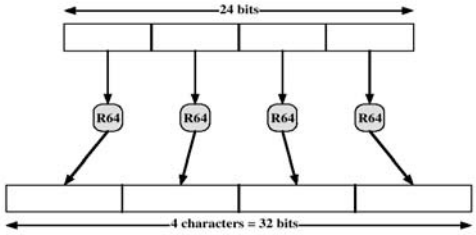


Figure 5.11 Printable Encoding of Binary Data into Radix-64 Format

Pretty Good Privacy - PGP

- de facto standaard voor e-mailbeveiliging
- wijd verbreid
- ontworpen door Phil Zimmermann (1991)
- selecteerde de beste beschikbare cryptografe algoritmen
- integreerde alles in 1 enkel programma
- beschikbaar op Unix, PC, Macintosh, …
- origineel freeware
- nu ook commerciële versies beschikbaar (Viacrypt, PGPcorporation: www.pgp.com)
- www.pgpi.org, www.openpgp.org, philzimmermann.com
- RFC2440 sinds 1998

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

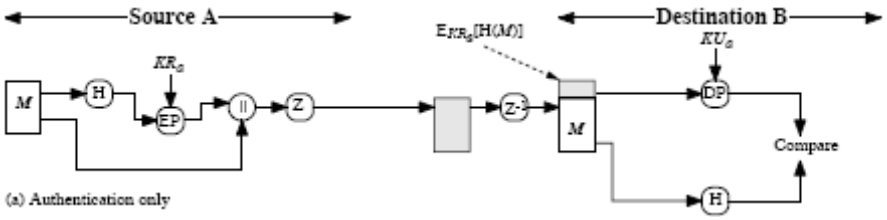
Beveiliging e-mail 7

PGP: authenticatie

- afzender A creëert boodschap
- SHA-1 om 160-bit hash code te genereren
- hash code wordt geëncrypteerd (RSA of DSS) met de private sleutel van A; resultaat bij boodschap gevoegd
- ontvanger B decrypteert hash code (RSA of DSS) met de publieke sleutel van afzender A om hash code te bekomen
- B genereert nieuwe hash code van de boodschap en vergelijkt ze met de gedecrypteerde hash code
- bij overeenkomst wordt de boodschap als authentiek aanvaard

Beveiliging e-mail 8

PGP: authenticatie (RSA)



PGP: clear signing

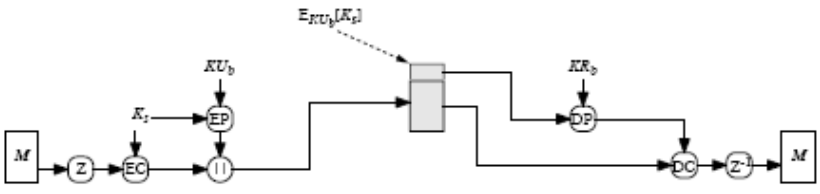
- detached signature
- handtekening apart opslaan en verzenden
- Radix-64
- wanneer noodzakelijk?
  - indien verschillende personen ondertekenen
    - **handtekening alleen op document, niet op de bijgevoegde handtekeningen**
  - indien uitvoerbaar bestand geïnfecteerd door virus
    - **handtekening ook besmet**

PGP: confidentialiteit

- gegenereerde boodschap verzonden of lokaal opgeslagen
- A kiest uit CAST-128, IDEA, 3DES of AES
- A genereert éénmalige sessiesleutel alleen voor deze boodschap (random, 128 of 256 (AES) bits)
- boodschap gecodeerd met sessiesleutel in CFB-mode
- sessiesleutel wordt
  - geëncrypteerd met de public key van ontvanger
  - bij boodschap bewaard
- ontvanger gebruikt RSA private key om sessiesleutel te decoderen
- sessiesleutel wordt gebruikt om boodschap te decoderen

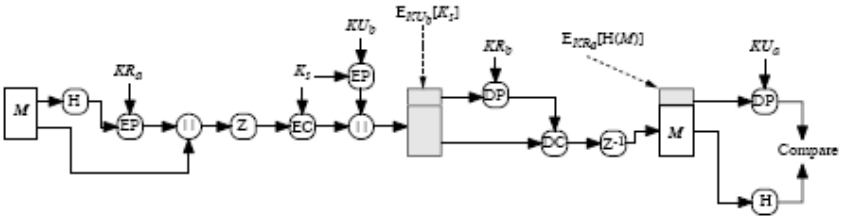
PGP: confidentialiteit (2)

- PGP voorziet een alternatief voor RSA sleutelcodering / uitwisseling:
  - gebaseerd op Diffie-Hellman: ElGamal



PGP: confidentialiteit + authenticatie

- beide worden toegepast op dezelfde boodschap
  - A creëert handtekening en voegt ze bij boodschap
  - A encrypteert boodschap en handtekening
  - A voegt sessiesleutel toe
- volgorde laat toe om
  - handtekening en plaintext apart op te slaan
  - handtekening kan geverifieerd worden los van sessiesleutel



PGP: compressie - zip

- compressie gebeurt standaard
- plaats is kritisch
- compressie **na** plaatsen van handtekening
  - redenen:
    - verschillende compressiealgoritmen genereren verschillende zip-files ➔ verschillende handtekening
    - indien handtekening na compressie: ook gecomprimeerde versie moet bewaard worden voor handtekeningcontrole
- encryptie **na** compressie
  - reden:
    - veiliger want zip-file bevat minder redundante informatie
    - betere bescherming tegen cryptanalyse

PGP: e-mail compatibiliteit

- PGP bericht bevat zeker tekens met code >127
- gegevens worden verzonden in radix-64
- bericht wordt groter: + 33%
- gecompenseerd door compressie: - 50%
- resultaat:
  - M x 0,5 x 1,33 = 0,66 M
  - handtekening en sessiesleutel zijn redelijk compact
- normaal ascii –bericht ook in radix-64 ➔
  - geheel bericht is niet direct leesbaar
- radix-64 kan ook alleen toegepast worden op handtekening
  - geen PGP nodig om bericht te lezen

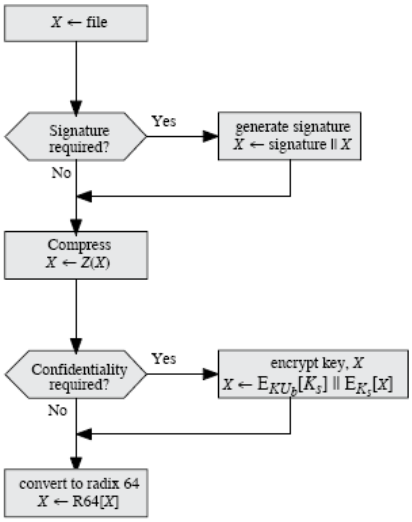
Beveiliging e-mail 15

PGP: e-mail compatibiliteit (2)

- berichten worden vaak beperkt in lengte (50KB)
- PGP kan bericht opdelen in segmenten
- segmentering gebeurt na volledige afhandeling, ook na de radix-64 conversie
- bij ontvangst:
  - alle headers van de verschillende berichten worden weggenomen
  - body van alle berichten worden samengevoegd

Beveiliging e-mail 16

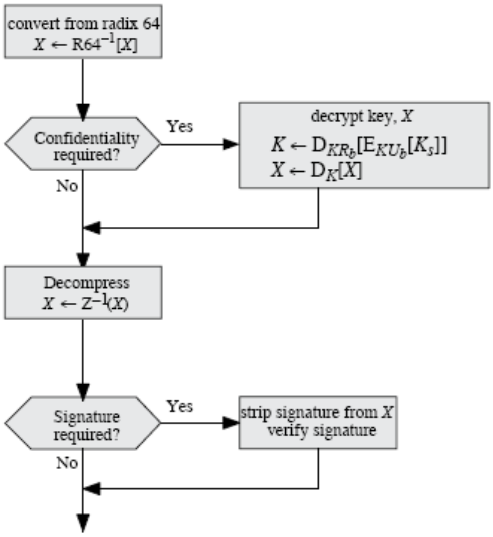
PGP: overzicht verzending



(a) Generic Transmission Diagram (from A)

17

PGP: overzicht ontvangst



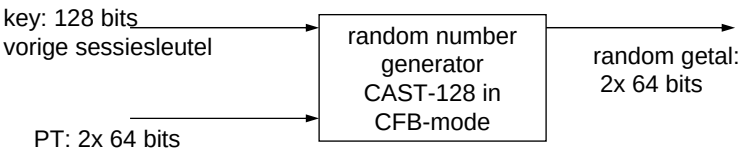
18

PGP: sleutels

- 4 soorten sleutels:
  - éénmalige sessiesleutels uniek voor elk bericht
  - publieke sleutels
  - private sleutels
  - op pass-phrase gebaseerde sleutels
- vereisten:
  - generatie van niet-voorspelbare sessiesleutels
  - gebruiker moet over meerdere publiekesleutelparen kunnen beschikken
    - sleutels af en toe wijzigen, is ontvanger op hoogte?
    - verschillende sleutels met verschillende groepen correspondenten
    - ➔ geen 1-1 relatie tussen gebruiker en publieke sleutel
  - elke entiteit moet eigen sleutelparen en publieke sleutels van anderen kunnen beheren

PGP: generatie van sessiesleutels

- sleutellengte afhankelijk van gebruikt algoritme
- voorbeeld: 128 bits voor cast-128, IDEA en AES
- sleutel wordt gegenereerd dmv van algoritme zelf



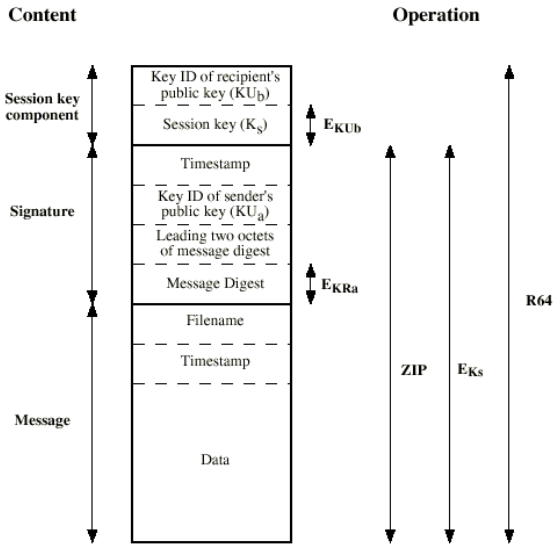
- PT afgeleid van toetsaanslagen door gebruiker
  - timing en toetscode bepalen 2 x 64 bits blokken
- resultaat is willekeurige stroom van 128 bits

PGP: sleutelidentificatie

- vereiste: gebruiker moet over meerdere publieke-sleutelparen kunnen beschikken
- hoe weet de ontvanger welke van zijn publieke sleutels is gebruikt bij coderen van sessiesleutel
  - publieke sleutel mee versturen met boodschap?
    - **plaatsverspilling (honderden cijfers), veilig?**
  - key-id uniek per gebruiker
    - **extra overhead en management (opslag van sleutel en id)**
  - oplossing PGP:
    - **key-id voor elke publieke sleutel, met hoge graad van waarschijnlijkheid uniek voor de gebruiker**
    - **K_U mod 64**

Beveiliging e-mail 21

PGP: samenstelling boodschap



22

PGP: sleutelhangers

- 2 key-rings per gebruiker
- laten toe alle sleutels te beheren
- private-sleutelhanger en publieke-sleutelhanger
- tabel met entry's, elke rij bevat info over sleutelpaar

PGP: private-sleutelhanger

- elke rij bevat info over sleutelpaar (privaat-publiek)
  - timestamp: moment van creatie
  - key-id
  - publieke sleutel
  - private sleutel (geëncrypteerd)
  - user-id: e-mailadres of andere naam; let op voor duplicaten!

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
. . .	. . .	. . .	. . .	. . .
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{H(P_i)}[KR_i]$	User i
. . .	. . .	. . .	. . .	. . .

PGP: beveiliging private sleutel

- private sleutelhanger op eigen machine, toch beveiligd
- acties:
  - kies een wachtwoordzin (pass phrase)
  - systeem berekent SHA-1 hashcode van zin
  - systeem vergeet de wachtwoordzin
  - systeem genereert sleutelpaar
  - private sleutel geëncrypteerd via CAST-128 met als sleutel de hashcode
  - hashcode wordt *niet* opgeslagen, geëncrypteerde sleutel *wel*
- opvragen sleutel
  - typ wachtwoordzin en bereken hashcode ervan
  - decrypteer opgeslagen sleutel met CAST-128 en hashcode
  - hoe controleren of resultaat correct is?
  - key-ID moet overeenkomen met opgeslagen key-ID

25

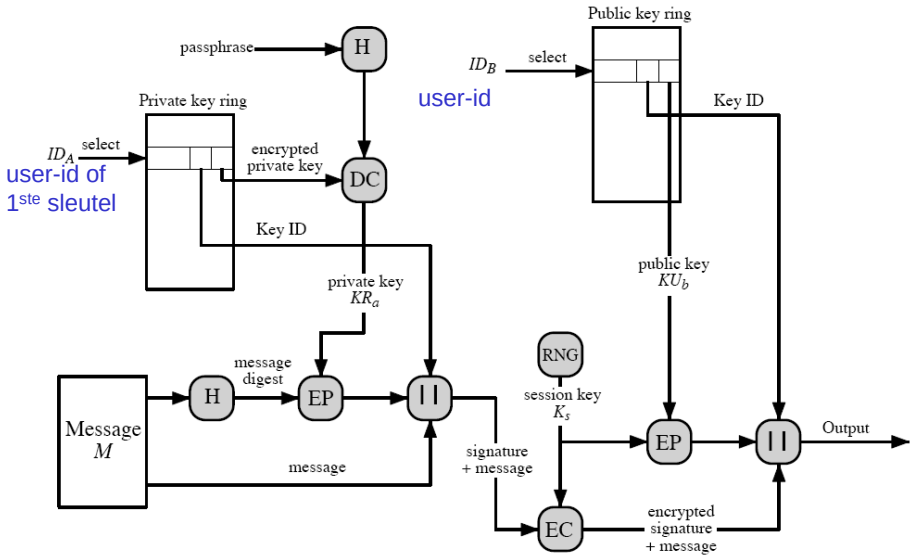
PGP: publieke-sleutelhanger

- elke rij bevat info over publieke sleutels van correspondenten
  - timestamp: moment van creatie
  - key-id
  - publieke sleutel
  - user-id: e-mailadres of andere naam; let op voor duplicaten!

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
T_1	$KU_i \bmod 2^{64}$	Ku_i	trust_flag_i	User i	trust_flag_i		
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .

Beveiliging e-mail 26

PGP: verzenden boodschap



PGP: ontvangen van boodschap