Segurança de Sistemas de Correio Electrónico

Sistema PGP

Segurança do E-Mail

- Discussão e motivação:
  - Um sistema E-mail é um sistema crítico do ponto de vista da segurança ?

    - Deverão as mensagens de correio electrónico serem consideradas como dados críticos ?
      - Uso institucional e empresarial ?
      - Uso pessoal ?

  - Porquê ?
  - Exemplos em que essa criticidade se manifesta ?

Segurança do E-Mail

- Quais são os principais e seus identificadores num sistema E-Mail ?
- Que propriedades de segurança?

  - Autenticação
  - Integridade
  - Confidencialidade
  - Não Repudiação

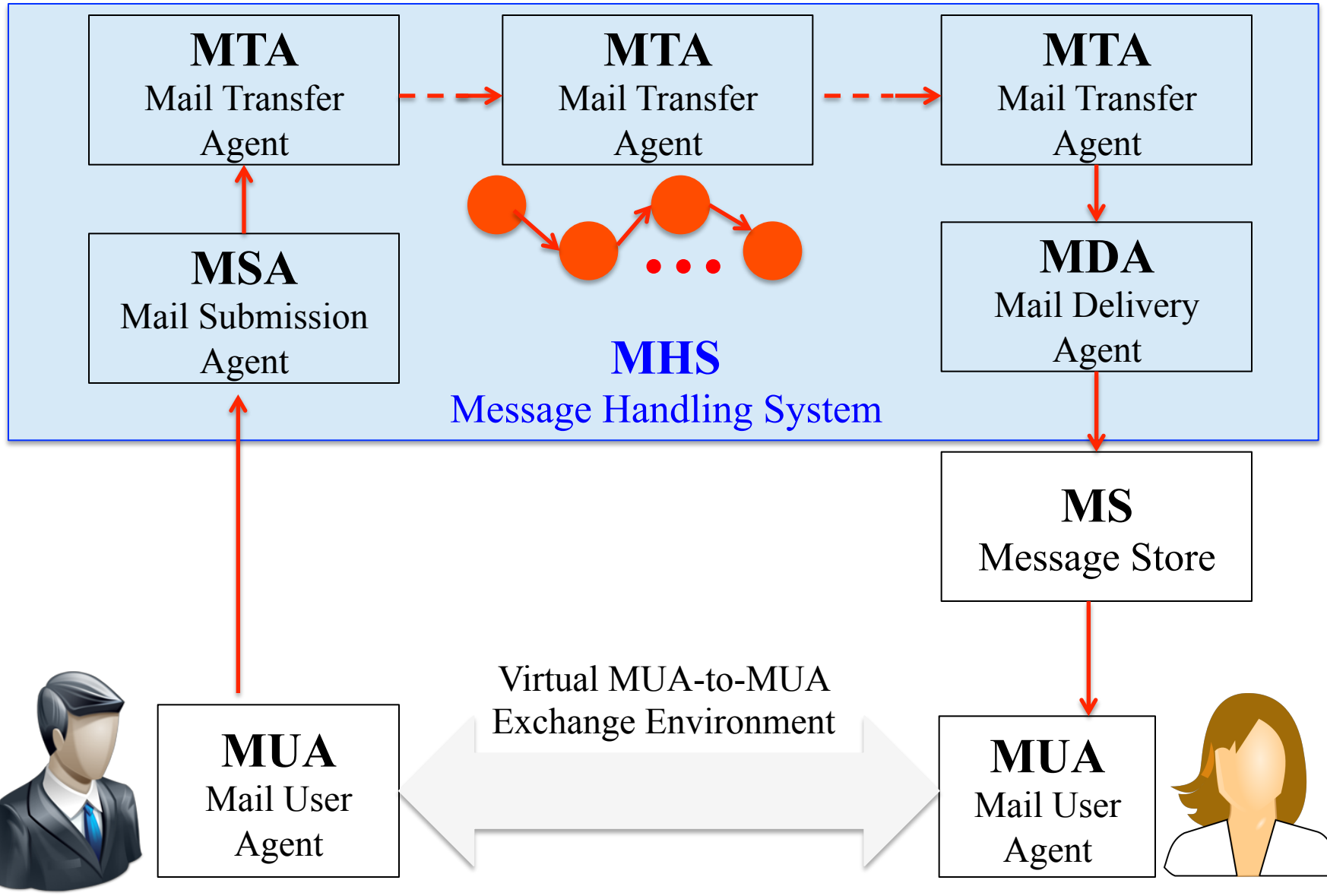
Para garantir:

No Spamming
No Phishing
No Mail Forging/Replaying/Tampering
Trusted Mail Message Exchanges

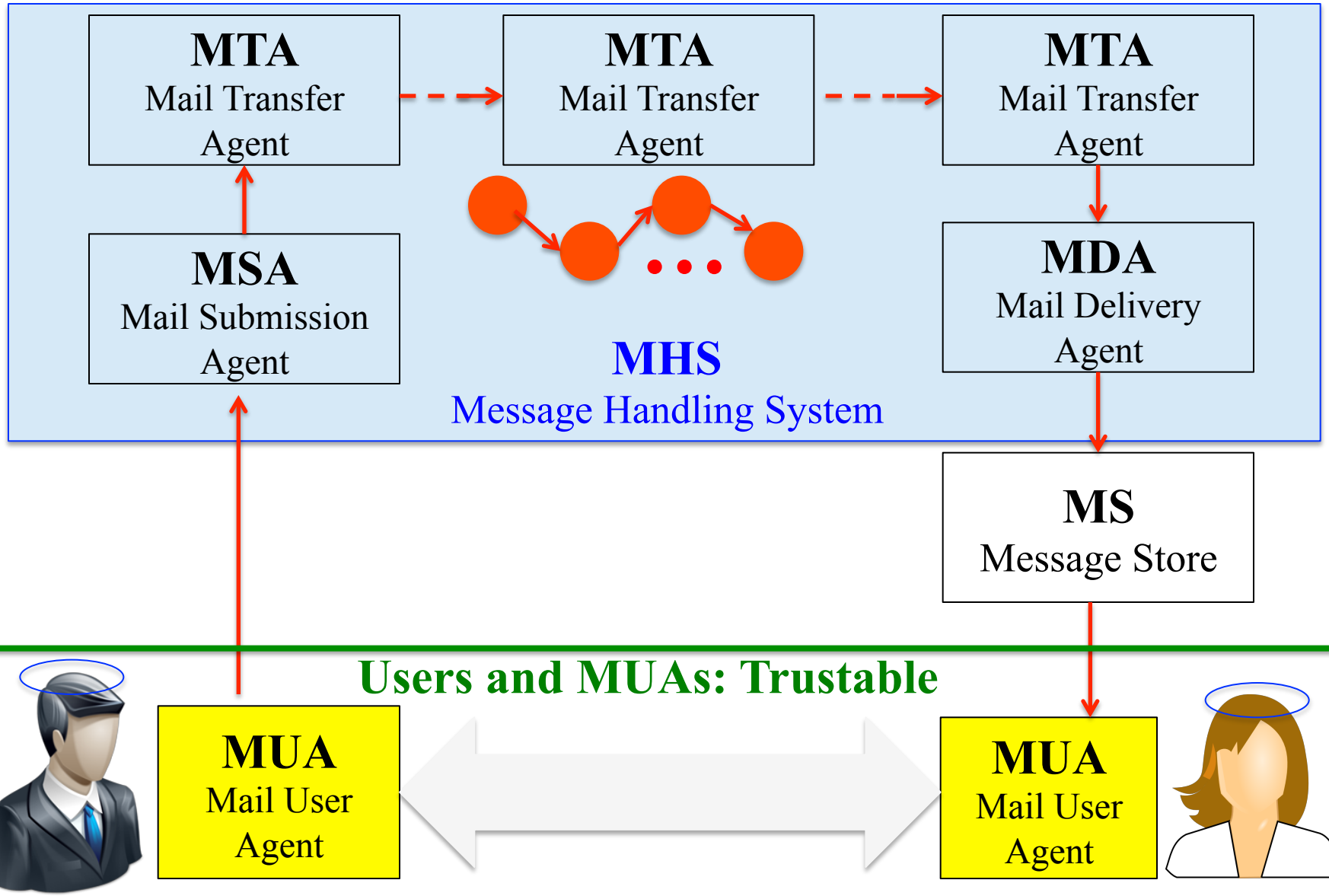
- Propriedades de segurança devem ser estabelecidas em condições "extremo-a-extremo" no E-Mail.
- **Complicado ? Porquê ?** (***Internet Mail Architecture***)

Internet Mail Architecture

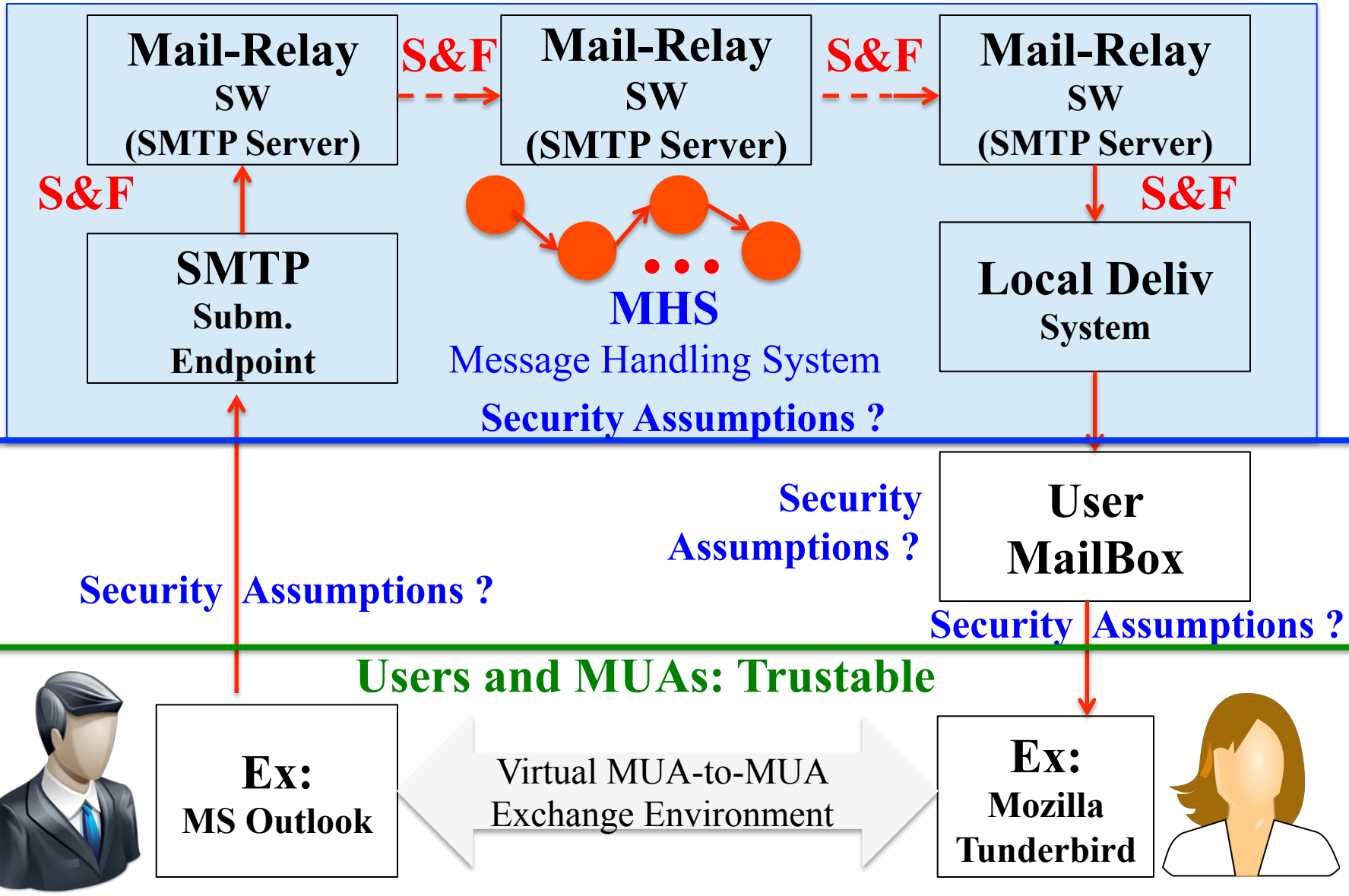
Internet Mail Architecture



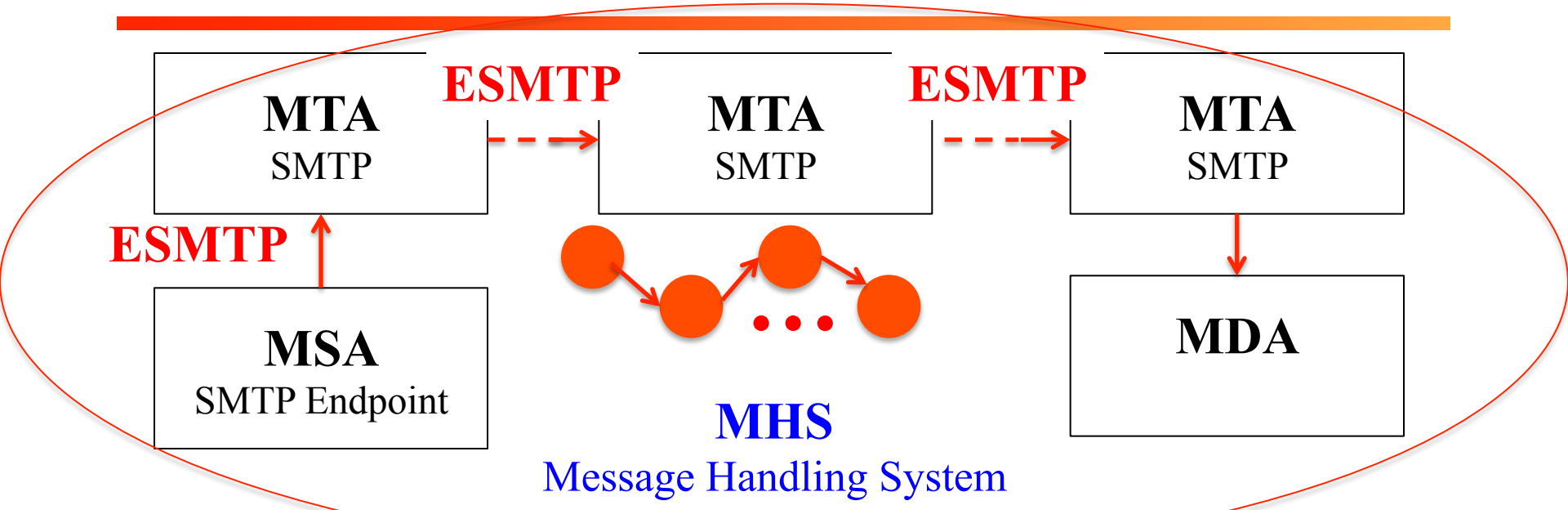
Internet Mail Architecture



MUA – MHS Interaction: Security Assumptions ?



Standards and Protocols



MHS:

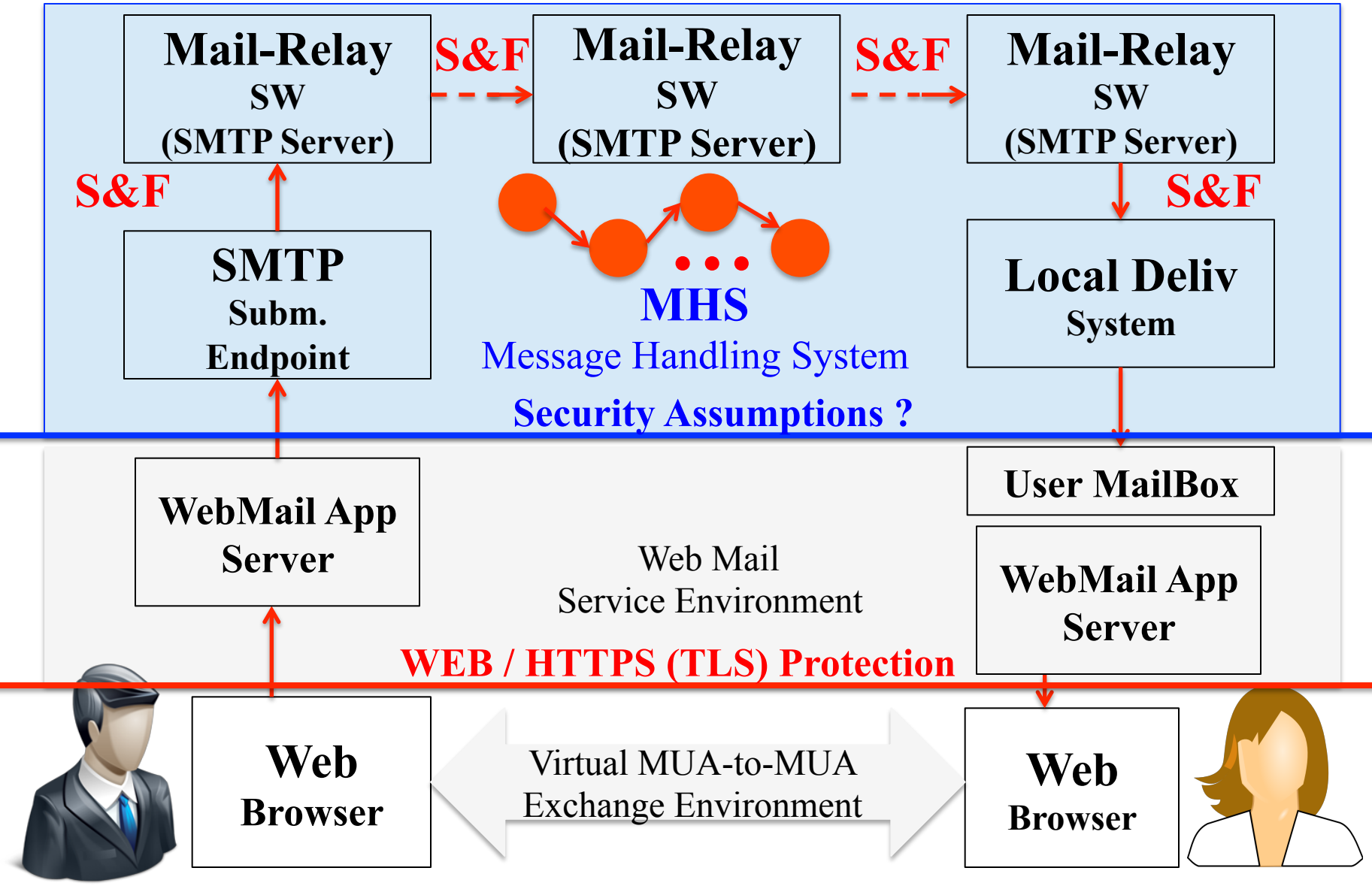
No proofs of authenticity/integrity

In the S&F delivering of Mail Messages

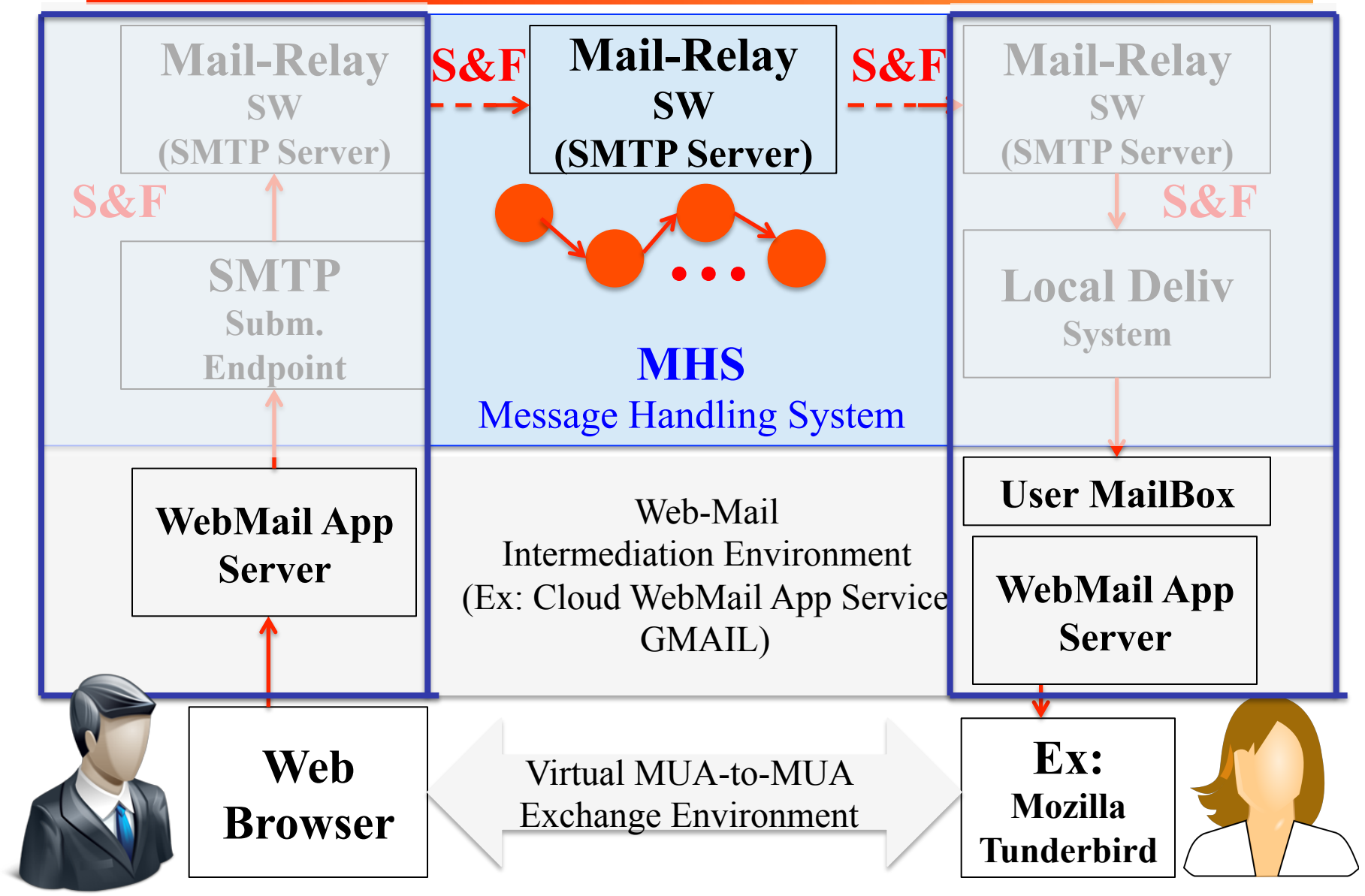
- Vulnerabilities (Possible Threats)

Spamming, Spoofing and Phishing Dissemination

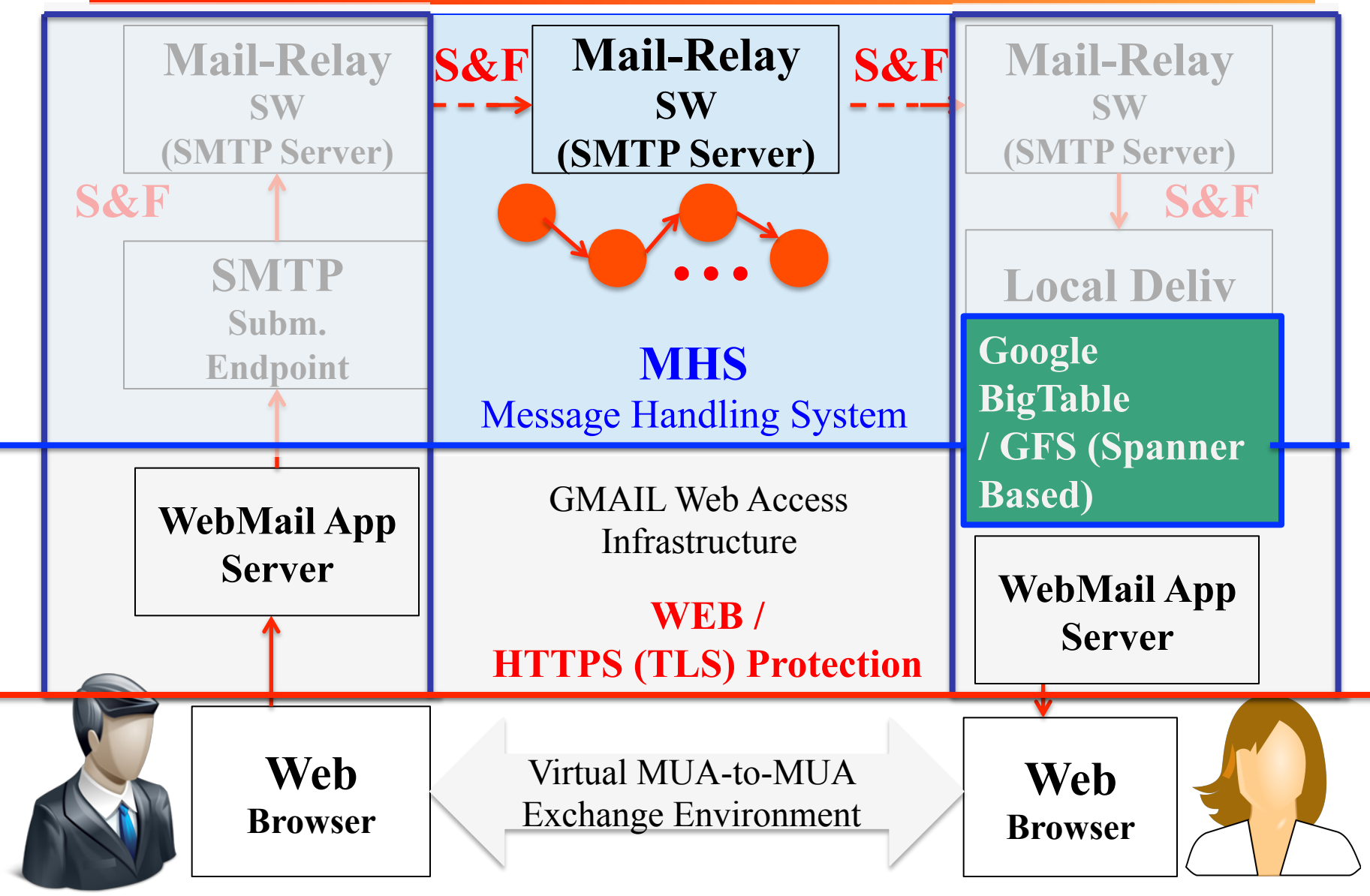
WebMail – MHS Interaction: Security Assumptions ?



WebMail Service Providers

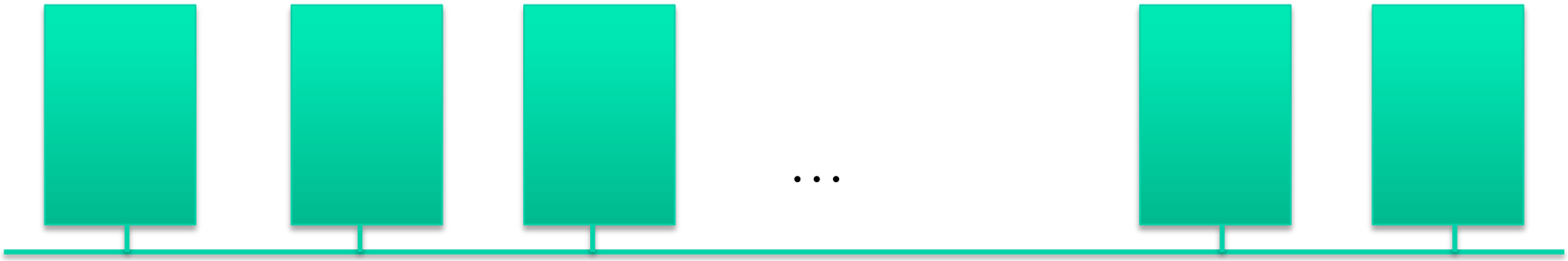


Ex: Gmail vs. Internet Mail Architecture



Ex: Gmail vs. Internet Mail Architecture

Spanner-*Based* Solution (Geo / DC Replication)



Scalable, multi-version, globally-distributed, and synchronously-replicated database.

Google BigTable / GFS (Spanner Based)

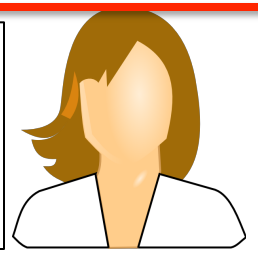
Security and Trust Assumptions ?

Outsourced: See:

https://cloud.google.com/security/security-design/resources/google_infrastructure_whitepaper_fa.pdf

Web Browser

Web Browser

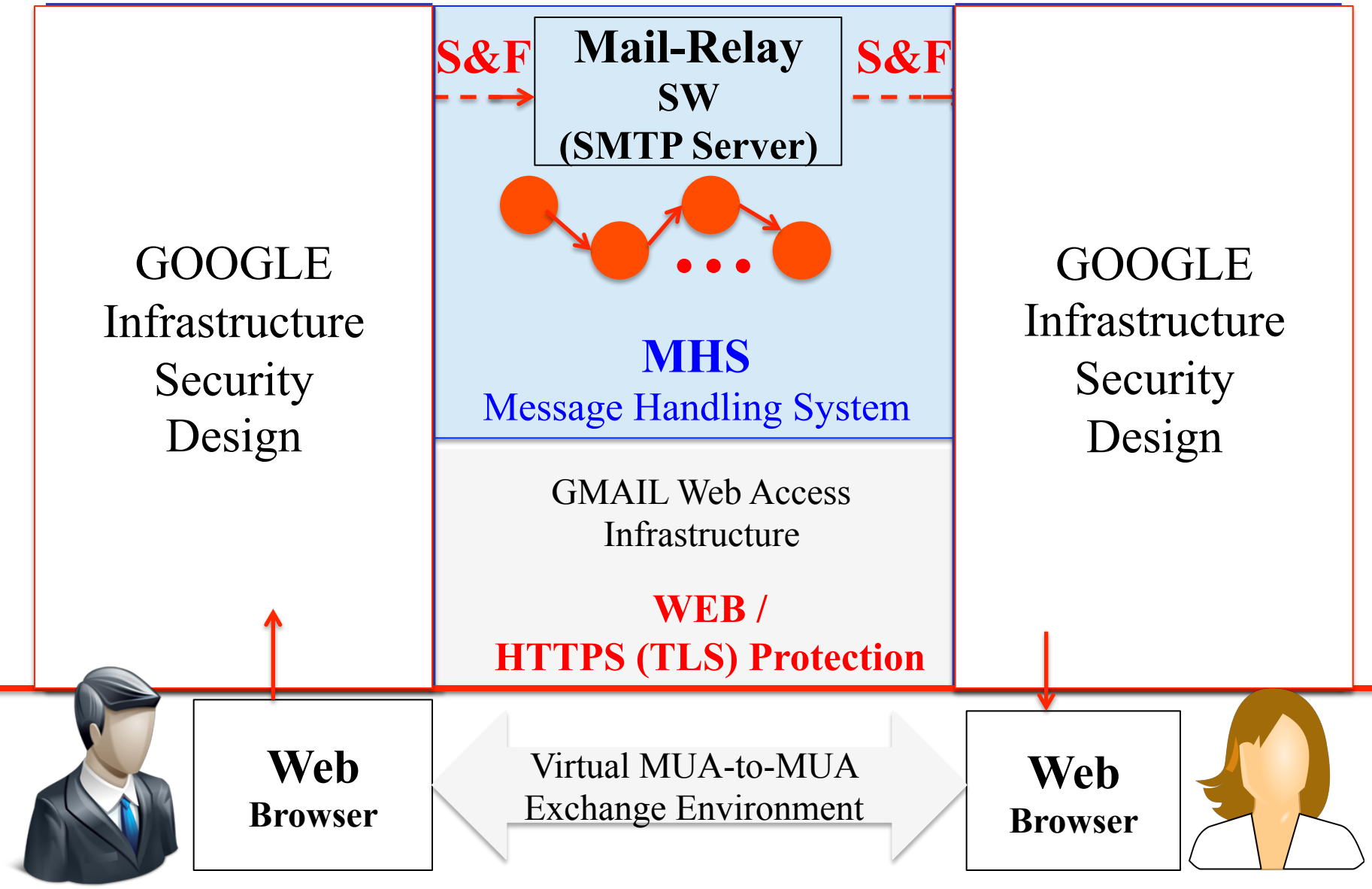


Spanner (Paper)

- https://research.google.com/archive/spanner.html
- OSDI 2012, Tenth Symposium on Operating System Design and Implementation, Hollywood, CA, October, 2012

- **Spanner is Google's scalable, multi-version, globally-distributed, and synchronously-replicated database.**
- It is the first system to distribute data at global scale and support externally-consistent distributed transactions. This paper describes how Spanner is structured, its feature set, the rationale underlying various design decisions, and a novel time API that exposes clock uncertainty. This API and its implementation are critical to supporting external consistency and a variety of powerful features: non-blocking reads in the past, lock-free read-only transactions, and atomic schema changes, across all of Spanner.

Ex: Gmail vs. Internet Mail Architecture



Ex: Gmail vs. Internet Mail Architecture

GOOGLE
Infrastructure
Security
Design

GMAIL In the Internet Mail Architecture … Security and Privacy, as well as, Trust Computing Bases Control are outsourced to Google

Web
Browser

Ex: Gmail vs. Internet Mail Architecture

See: Google Infrastructure Security Design Overview
https://cloud.google.com/security/security-design/resources/google_infrastructure_whitepaper_fa.pdf

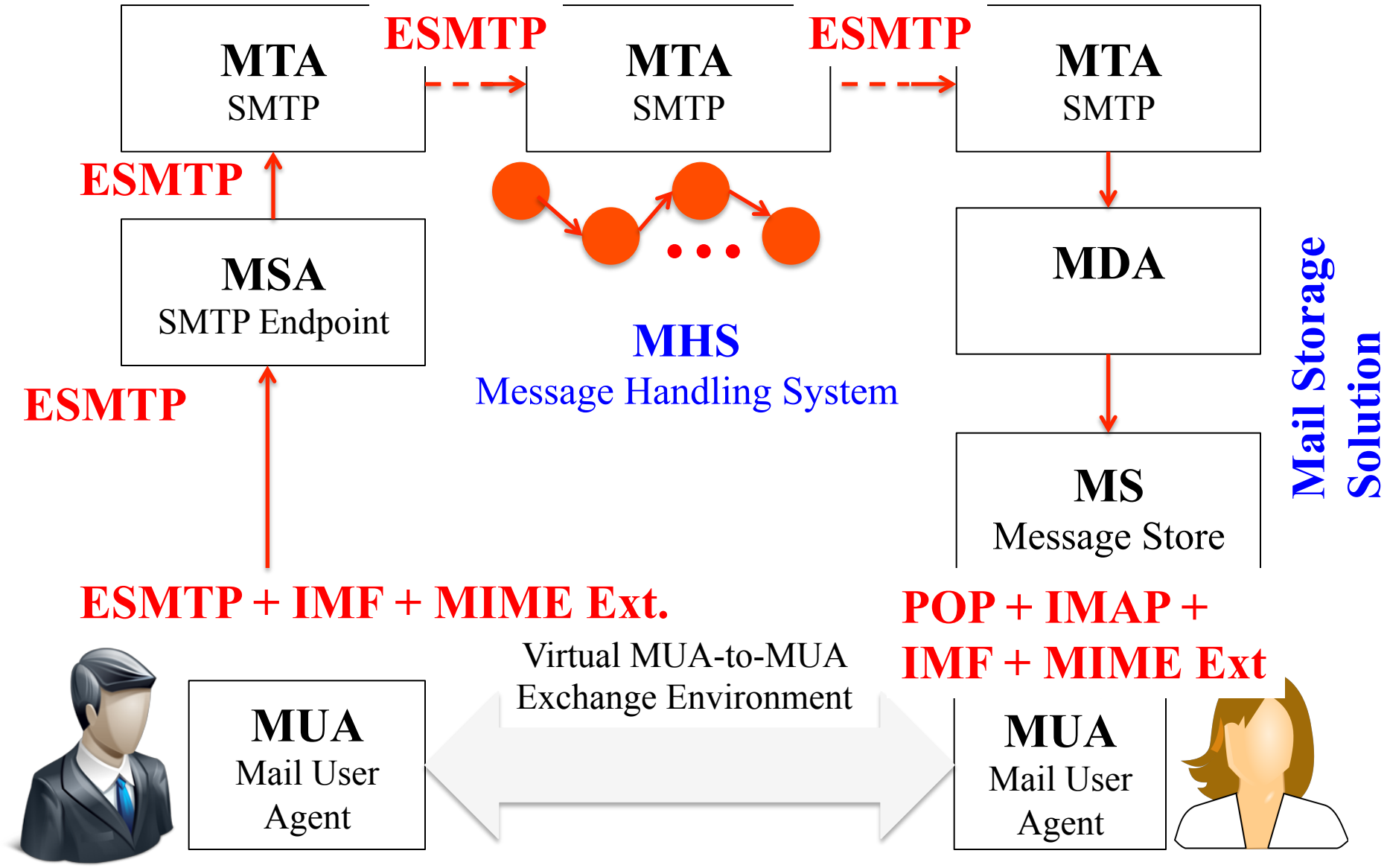
See: Data Privacy Guaratees:
https://privacy.google.com/your-security.html

See: about encryption
https://www.google.com/transparencyreport/saferemail/faq/

See also ….
https://support.google.com/googlecloud/answer/6056693?hl=en

But Data-Encryption is provided in a REST-Based Encryption Solution:
he "data at rest" on the Google servers is most likely encrypted, but encrypted with keys that Google owns and manages. This means that if the hardware is stolen, the data will not be decryptable. However, any authorized person or program at Google with access to the keys for that service can read the data. Remember that the GMail ads are tailored to you because the algorithm "reads" your mail. Google Now also generates alerts based on "reading" your mail.

Standards and Protocols



Resumo: protocolos relevantes

Sobre estes protocolos e sua evolução: ver anexo nos slides

- SMTP / ESMTP
  - Mensagens re-encaminhadas (S&F) sem controlo de autenticidade e integridade entre os pares envolvidos
- POP3, IMAP
- IMF and Ext. + Formatos MIME
  - Problema de autenticação dos utilizadores que enviam / recebem mensagens
  - Mensagens e conteúdos (multi-partes) passadas em claro
  - Fácil quebrar autenticidade, integridade e confidencialidade
  - Perigo: injeção de Spamming, Spoofing e Physhing

Tópicos cobertos

- Insegurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP

Tópicos cobertos

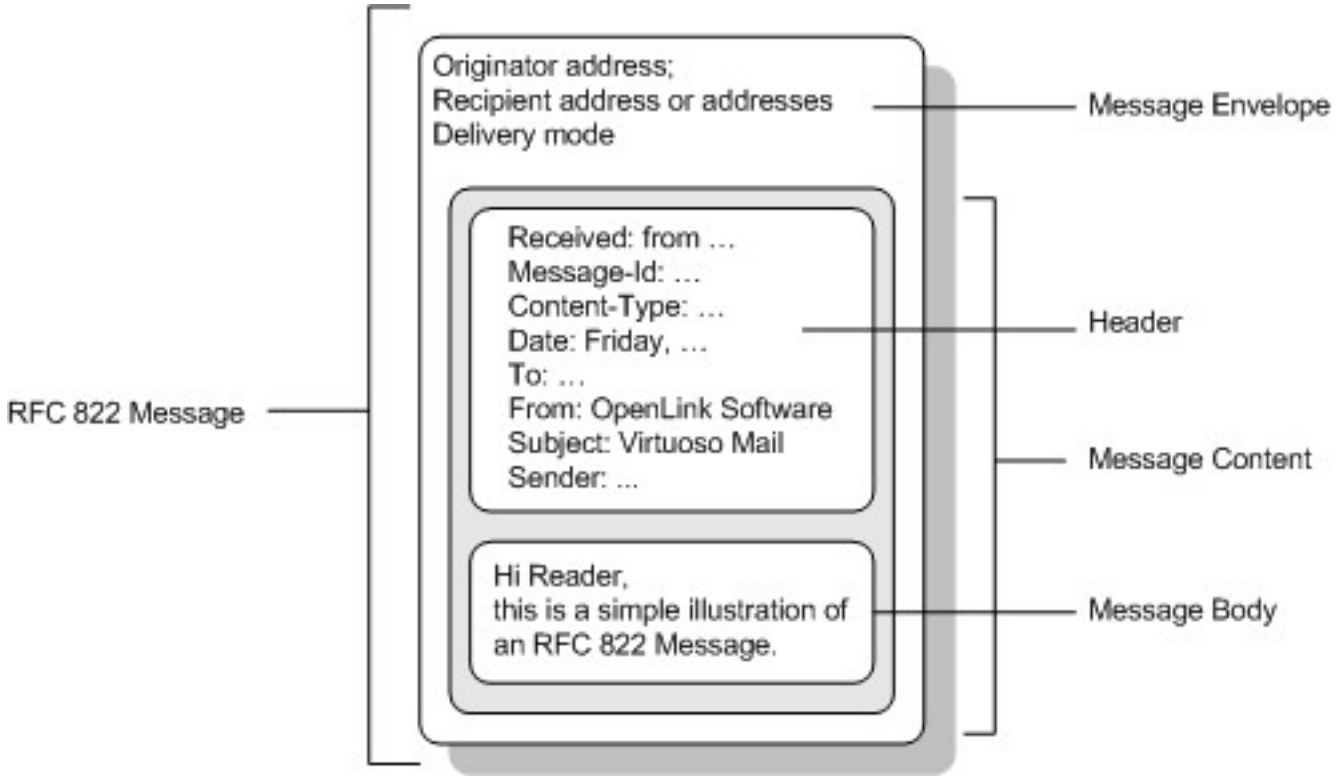
- Insegurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP

Normalização E-Mail: Formatos

- Formatos de mensagens de correio electrónico na INTERNET (IMF - Internet Mail Format)
  - Inicialmente RFC 822 ... hoje 5322
    - *Mail Headers, Body, text format*
    - **Conteúdo "em claro"**
  - Normalização MIME (RFC 2045-2049)
    - *Multipurpose Internet Mail Extensions*
    - Normalização de conteúdos de extensão (multi-partes numa mensagem, anexos)
    - **Formatos de codificação de informação (multimedia) mas com controlo e conteúdos "em claro"**
      - **MIME Types**

    Segurança da normalização MIME só mais tarde, com a normalização **S/MIME (Secure MIME)**

RFC 822 Mail Message Format



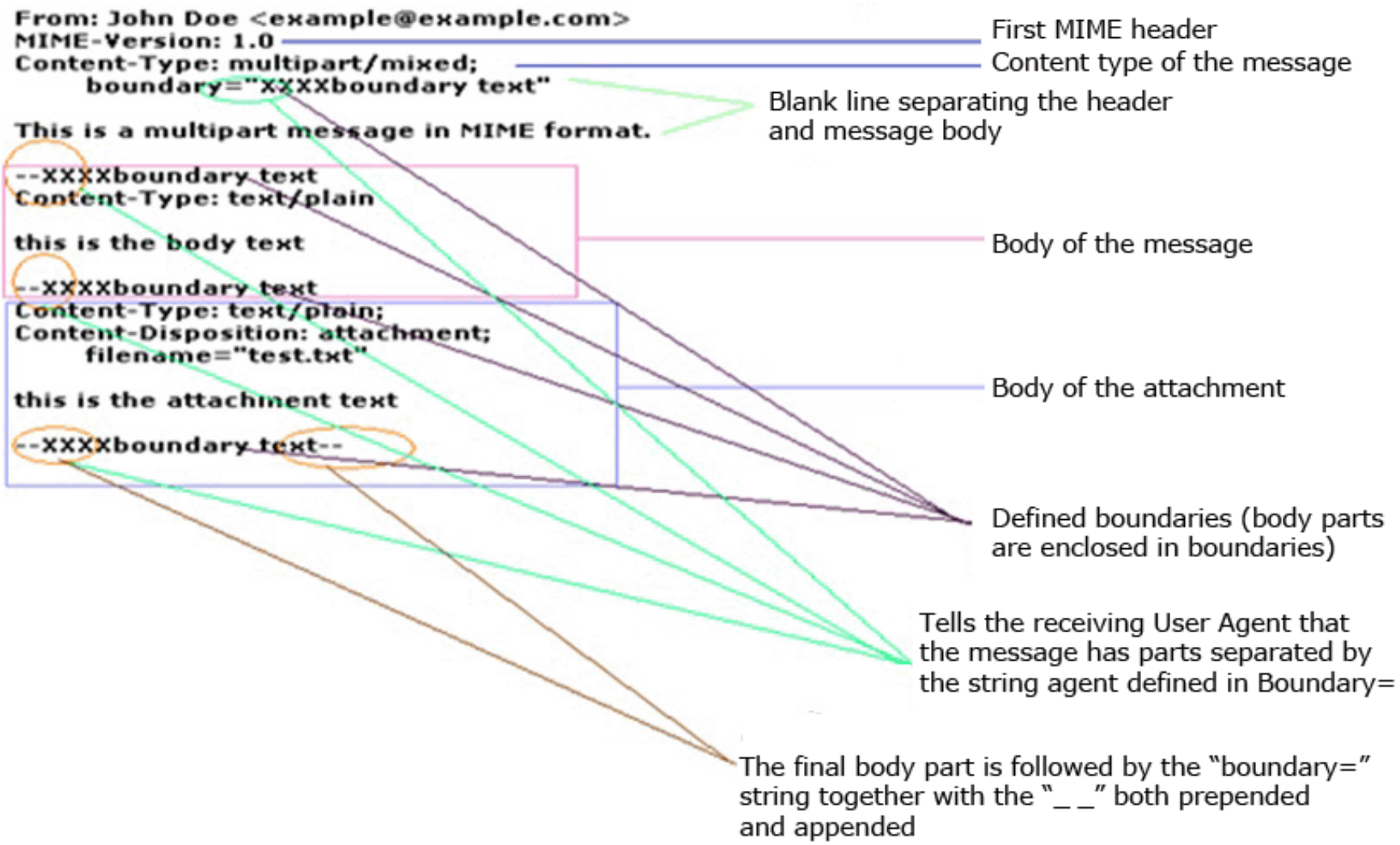
Conteúdo em claro …

MIME Formats

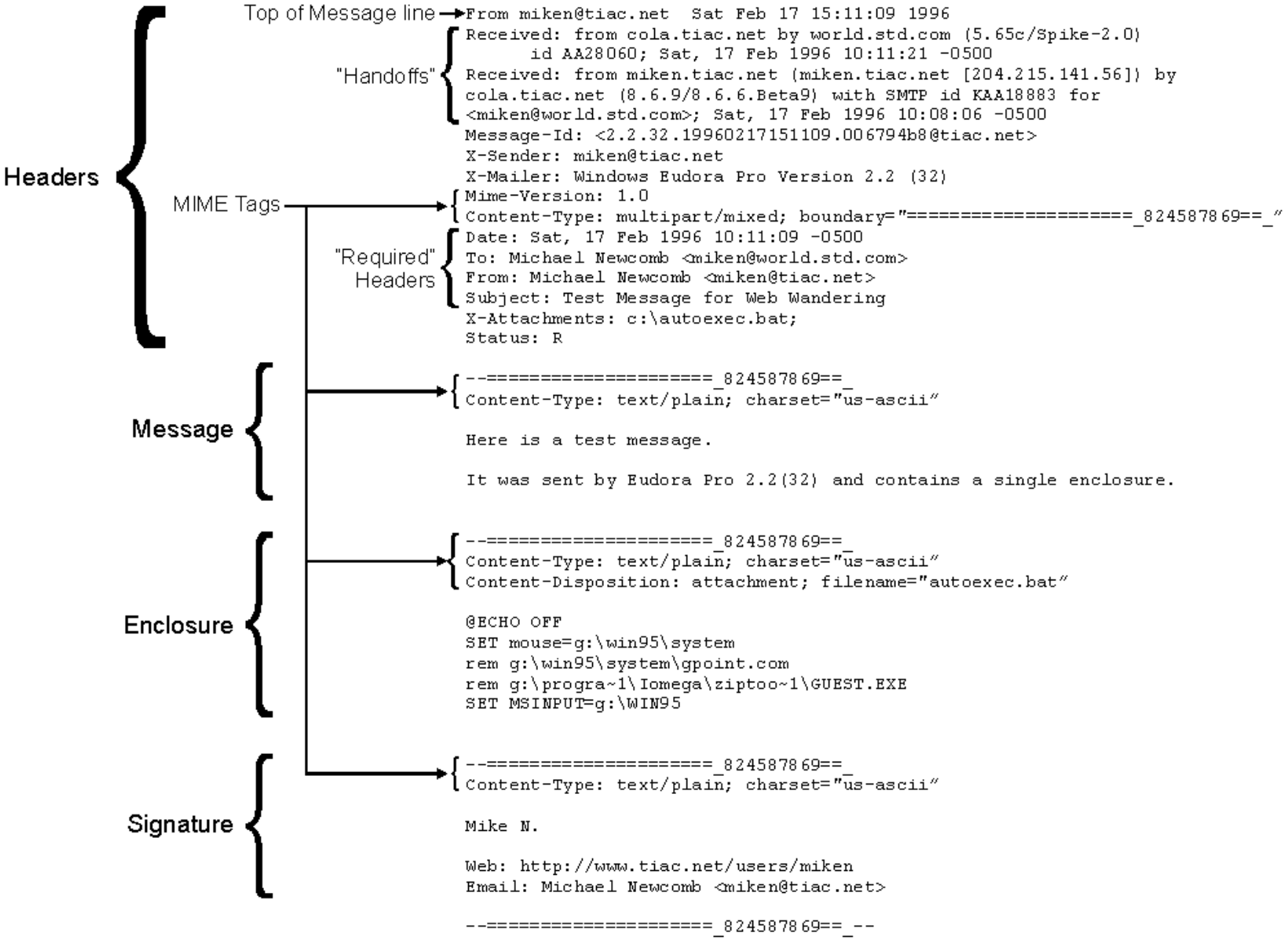
Conteúdo em claro …

```
From: <joseph.neubauer@hp.com>
To: <jones@demo.com>
Subject: Report Status
Date: Tue, 5 Nov 2002 08:45:41 -0500
MIME-Version: 1.0
Content-Type: multipart/alternative;
      boundary="----=_NextPart_000_0000_01C284A7.B8F453A0"

This is a multi-part message in MIME format.
Your email client may not be able to understand MIME.

------=_NextPart_000_0000_01C284A7.B8F453A0
Content-Type: text/plain;
      charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

Update Status Report
-----
neubauerNeubauer, Joe
masonMason, Kevin
jonesJones, Mosby
havensHavens, Mary
There was 1 error
------=_NextPart_000_0000_01C284A7.B8F453A0
Content-Type: text/html;
      charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<h2>Update Status Report</h2>
<hr><p>
<table border>
<tr><td>neubauer</td><td>Neubauer, Joe</td><td><img =
src=3Dhttp://router/images/up.gif></td></tr>
<tr><td>mason</td><td>Mason, Kevin</td><td><img =
src=3Dhttp://router/images/up.gif></td></tr>
<tr><td>jones</td><td>Jones, Mosby</td><td><img =
src=3Dhttp://router/images/up.gif></td></tr>
<tr><td><b>havens</b></td><td><b>Havens, Mary</b></td><td><img =
src=3Dhttp://router/images/dn.gif></td></tr>
</table>
<p>
<font color=3Dblue size=3D5><b>There was 1 error</b></font>

------=_NextPart_000_0000_01C284A7.B8F453A0--
```

MIME Formats



MIME Message Format



Normalização E-Mail: SMTP / ESMTP

- **SMTP (Send Mail Transfer Protocol)**

  Envio de mensagens:

  - Entre MUAs (Mail User Agents) e MTAs (Mail Transfer Agents)
  - Entre MTAs
    - End-to-End Message Delivering
    - RFCs 2821 (SMTP) e RFC 5321 (ESMTP)
    - Porto 25, sobre transporte TCP

  - SMTP não possui segurança para proteção das mensagens
    - Processamento necessita da informação dos cabeçalhos das mensagens
    - Transmissão "*store & forward*" dos conteúdos das mensagens em claro, as mesmas são enviadas em claro.

POP, IMAP

- Recuperação de mensagens de repositórios E-Mail:
  - **POP (POP 3, POP 4 ou IMAP 4)**
    - POP3 (porto 110 tcp)
      - RFCs: 1081 (1988), 1939 (1996)
        - » User/Pwd LOGON, descarregamento das mensagens sem proteção
      - POP3 with APOP (RFC 1460): Autenticação de utilizadores com Username + Hash(TS-Server-Generated || Pwd)
      - POP 3 RFC 1734 with AUTH Extension
        - » Ex., AUTH KERBEROS_V4
    - IMAP4 service (port 143 tcp), RFC 3501, 4466(2006), 4469(2006), 4551(2006), 5032(2007), 5182(2008), 5738(2010), 6186(2011), 6858(2013)
- Canais de transporte das mensagens não protegidos

Problemas a ter em conta ...

- Mensagens (e seus conteúdos) depositadas em claro (sem confidencialidade, autenticidade ou integridade) nas caixas de correio dos utilizadores, no servidor de EMail onde são mantidas

- Envio por SMTP revela as mensagens na interação entre MUAs, MTAs e entre estes, durante o seu reencaminhaento até ao servidor destino

- Descarregamento (POP, IMAP) por parte do MUA final a partir do servidor EMail que armazena as mensagens revela as mensagens no canal
  - Apenas fornece mecanismos de autenticação e controlo de acesso às mailboxes através de esquemas de autenticação de utilizadores

Segurança Extremo-a-Extremo

- Providencia a proteção necessária para que se assegurem propriedades de segurança das mensagens, durante a sua transferência entre o emissor e o receptor,

- Proteção das mensagens das vulnerabilidades anteriores ou para além das proteções anteriores

- Isto é, garantia de propriedades de segurança
  - entre o emissor
    - (antes de as enviar, possivelmente via SMTP)
  - e o receptor
    - (depois de as obter, possivelmente por qualquer um dos protocolos anteriormente referidos nas variantes POP ou IMAP

Que propriedades de segurança ?

- **Autenticação das mensagens** enviadas pelo emissor
  - Assinatura digital da mensagem, verificável pelo receptor
    - Criptografia assimétrica, com uso da chave privada do emissor
- **Integridade das mensagens**
  - Síntese de segurança verificável pelo receptor
- **Confidencialidade das mensagens**
  - A mensagem é transmitida cifrada (no caso de mensagens MIME, isso incluirá as múltiplas partes ou anexos)
  - Com uso de criptografia simétrica

- **Não-Repudiação**
  - Através de mensagens de resposta assinadas com notificação de ACK ou AVISO DE RECEPÇÃO

Confidencialidade com chaves OTK

- **Confidencialidade com base em chaves do tipo "One-Time-Key" (o OTK)**

- Cada mensagem a enviar será cifrada por uma chave única (não reutilizável), gerada pelo emissor para efeitos desse envi

- Esta chave OTK será transmitida (distribuída ao destinatário) por um envelope seguro, usando criptografia assimétrica
  - O envelope é cifrado com a chave pública do receptor

Contra-medidas

- Adicionar autenticação, confidencialidade, integridade e princípios de não-repudiação nas mensagens enviadas/recebidas (MUA-MSA-MTA)
  - END-to-END Arguments
  - Controlo dos utilizadores (Senders/Receivers)
- Dados passados sempre cifrados ao longo da cadeia de Store & Forward do MHS (MTA-MTA)

Soluções: PGP, S/MIME

Contra-medidas

- Proteção de dados (mensagens) em trânsito com base em canais seguros para suportar SMTP
  - Interações: MUA – MSA e MSA – MTA
  - Interações MTS - MTS

- Ex: Pode ser adotado TLS ou outro transporte seguro com os mesmos serviços de segurança

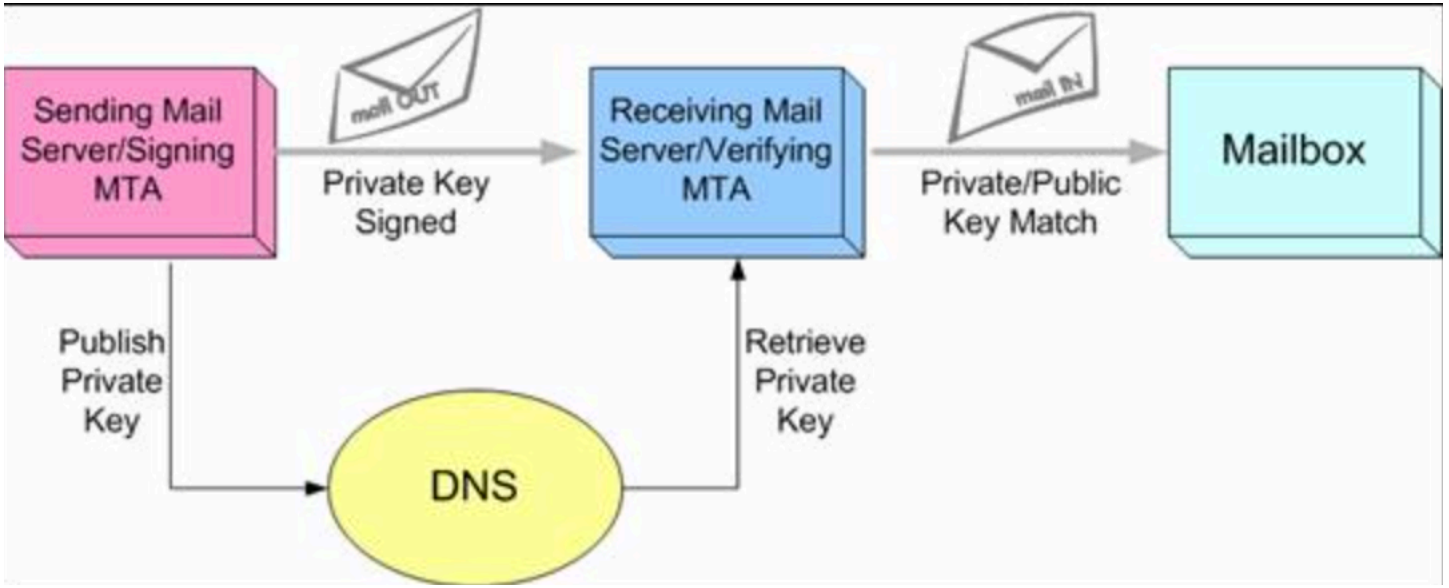
Soluções: ESMTP / TLS

Contra-medidas

- Garantir autenticação e controlo de acesso enter MTAs no encaminhamento de mensagens de correio electrónico

Solução: DKIM

DKIM


Sending Mail Server/Signing MTA
mail OUT
Private Key Signed
Receiving Mail Server/Verifying MTA
mail IN
Private/Public Key Match
Mailbox
Publish Private Key
DNS
Retrieve Private Key

DKIM – Domain Keys Management

- Email Authentication method
- It allows the receiver to check that an email claimed to have come from a specific domain was indeed authorized by the owner of that domain
- It is intended to prevent forged sender addresses in emails
  - In technical terms, DKIM lets a domain associate its name with an email message by affixing a digital signature
  - Verification via the public key of the sender (signer's) domain, published in DNS

Example of DKIM Signature

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
        d=gmail.com; s=20120113;
        h=mime-version:in-reply-to:references:date:message-id:subject:from:to
         :content-type;
        bh=r8R+NS5CU0Cgx5OUcPnHGiw5R1AzjGtdFQoju9HIxCo=;
        b=zXKVrJAN2Md8bLPFOzF9Tte87J3wBULQvv+7wRlPja7n/1pPGYZiaGyHOn/t61yfbQ
         x21E+17D0is7XhGzmz2cu1Xunpst2+wcC20wZoXO+VvX1AwqXo01wC+CsFENqW74kjbc
         +QdYg86dFV3w/qDBBjqWiuW5xGTIsdH0RuRexlnd4RGVQjoGQGpyyMG+LZozUiQEjhkd
         lRpF4y19/sBMDshqXuCBzGtnf6DVUNa0Q2KvJVDGGxmGaSGtgVGtk2PZoarJLBNfYm0i
         cTe2kFDdB4APmQbqg5dOUFPDz4b0wVac+9wBW+YZuhxyNvcwlKLyEMGMKZj6/q8djdIP
         F7Fw==
```

DNS Query: Chave Pública DKIM

```
$ host -t txt 20120113._domainkey.gmail.com
20120113._domainkey.gmail.com descriptive text "k=rsa\; p=MIIBIjANBgkqhki-
G9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1Kd87/UeJjenpabgbFwh+eBCsSTrqmwIYYvywlbhbqoo2Dymnd-
FkbjOVIPIldNs/m40KF+yzMn1skyoxcTUGCQs8g3FgD2Ap3ZB5DekAo5wMmk4wimDO+U8QzI3S-
D0" "7y2+07wlNWwIt8svnxgdxGkVbbhzY8i+RQ9DpSVpPbF7ykQxtKXkv/ahW3KjViiAH+ghvv-
Ihkx4xYSIc9oSwVmAl5OctMEeWUwg8Istjqz8BZeTWbf41fbNhte7Y+YqZOwq1Sd0DbvYAD9NOZK9vl-
fuac0598HY+vtSBczUiKERHv1yRbcaQtZFh5wtiRrN04BLUTD21MycBX5jYchHjPY/wIDAQAB"
```

Ou ..

dig TXT 20120113._domainkey.gmail.com

Envio e encaminhamento de mensagens

- Enquanto são transmitidas via SMTP, as mensagens ficam vulneráveis a ataques às comunicações
- Mensagens temporárias armazenadas durante o processo de reencaminhamento ficam vulneráveis nesse armazenamento
- No armazenamento final, as mensagens ficam também vulneráveis
  - Ex., Adversário do tipo "Honesto mas Curioso", pode ter vasculhado as mensagens
    - Ex., Administradores de sistemas com abuso de privilégio

Contra-medidas

- Garantir autenticação de utilizadores no acesso a mailboxes e no descarregamento de mensagens por POP e IMAP

Soluções:
POP3 c/ APOP (RFC 1939)
POP3 c/ KPOP (Kerberized POP), RFC
POP3 e IMAP c/ STARTTLS Extension
RFC 2595
POP3 SASL: RFC 5034

Descarregamento das mensagens

- MUA <> Servidor do Repositório (Mailboxes)
  - **POP3S Endpoints**
    - **RFC 5034, POP3 SASL** (*Protected User Authentication and Message Retrieve Protection*)
      - Comandos CAPA
    - **POP3 / SSL (STARTTLS Extension, STLS command)**
      - RFC 2249, porto 995 tcp (também porto 110 tcp)
  - **IMAP4 secure endpoints**
    - IMAP SASL (Protected User *Authentication and Message Retrieve Protection*)
    - IMAP / SSL

Alguns exemplos

Ex: APOP (POP 3), RFC 1939

```
S: <wait for connection on TCP port 110>
C: <open connection>
S:    +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C:    APOP mrose c4c9334bac560ecc979e58001b3e22fb
S:    +OK mrose's maildrop has 2 messages (320 octets)
C:    STAT
S:    +OK 2 320
C:    LIST
S:    +OK 2 messages (320 octets)
S:    1 120
S:    2 200
S:    .
C:    RETR 1
S:    +OK 120 octets
S:    <the POP3 server sends message 1>
S:    .
C:    DELE 1
S:    +OK message 1 deleted
C:    RETR 2
S:    +OK 200 octets
S:    <the POP3 server sends message 2>
S:    .
C:    DELE 2
S:    +OK message 2 deleted
C:    QUIT
S:    +OK dewey POP3 server signing off (maildrop empty)
C: <close connection>
S: <wait for next connection>
```

Hash(PWD)

… Sem APOP

```
C:    USER mrose
S:    +OK User accepted
C:    PASS tanstaaf
S:    +OK Pass accepted
```

Ex: IMAP com TLS

```
Example:    C: a001 CAPABILITY
            S: * CAPABILITY IMAP4rev1 STARTTLS LOGINDISABLED
            S: a001 OK CAPABILITY completed
            C: a002 STARTTLS
            S: a002 OK Begin TLS negotiation now
TLS ──►     <TLS negotiation, further commands are under TLS layer>
            C: a003 CAPABILITY
TLS         S: * CAPABILITY IMAP4rev1 AUTH=EXTERNAL
Protection  S: a003 OK CAPABILITY completed
            C: a004 LOGIN joe password
            S: a004 OK LOGIN completed
```

Ex: Outro exemplo, com SASL - TLS

```
Example:    S: * ACAP (SASL "CRAM-MD5") (STARTTLS)
        →   C: a001 AUTHENTICATE "CRAM-MD5"
            S: + "<1896.697170952@postoffice.reston.mci.net>"
        →   C: "tim b913a602c7eda7a495b4e6e7334d3890"
        →   S: a001 NO (TRANSITION-NEEDED)
               "Please change your password, or use TLS to login"
TLS     →   C: a002 STARTTLS
            S: a002 OK "Begin TLS negotiation now"
            <TLS negotiation, further commands are under TLS layer>
TLS         S: * ACAP (SASL "CRAM-MD5" "PLAIN" "EXTERNAL")   ← SASL
Protection  C: a003 AUTHENTICATE "PLAIN" {21+}
            C: <NUL>tim<NUL>tanstaaftanstaaf
            S: a003 OK CRAM-MD5 password initialized
```

Exemplo com POP3, STARTTLS

EX: openssl s_client -connect pop.gmail.com:995

……LOGIN na sessão POP3

USER <username>
PASS <password>

….. e a seguir podem dar-se comandos POP3:

STAT
LIST
RETR
DELE
RSET
TOP
QUIT

Exemplo com SMTP e TLS

openssl s_client -starttls smtp -connect smtp.gmail.com:25

…

// Fazer na shell ...

echo -ne '\0admin@example.com\0password' | base64

…

Obtém outpit na forma …

AGFkbWluQGV4YW1wbGUuY29tAHBhc3N3b3Jk

// Fazer então cut & paste para:

AUTH PLAIN AGFkbWluQGV4YW1wbGUuY29tAHBhc3N3b3Jk

A seguir fazer …

ehlo example.com

mail from: admin@example.com

rcpt to: admin@other.com

data

quit

Tópicos cobertos

- Segurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP

Tópicos cobertos

- Segurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP
  - Contexto e criação do sistema PGP
  - Segurança e Mecanismos criptográficos
  - Outros componentes do processamento PGP
  - Processamento e formato de mensagens
  - Gestão de chaves públicas e modelo de confiança
  - Geração de chaves para encriptação de mensagens

Tópicos cobertos

- Segurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP
  - Contexto e criação do sistema PGP
  - Segurança e Mecanismos criptográficos
  - Outros componentes do processamento PGP
  - Processamento e formato de mensagens
  - Gestão de chaves públicas e modelo de confiança
  - Geração de chaves para encriptação de mensagens

PGP – *Pretty Good Privacy*

- O sistema PGP foi criado inicialmente por Phil Zimmermman
  - Alvo de prisão e processo de acusação de ilegalidade nos EUA, durante cerca de três anos
  - http://www.philzimmermann.com/EN/background/index.html
  - https://philzimmermann.com/EN/audiovideo/index.html
  - http://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html

Aura: "anti-establishment endeavor",
mais tarde normalizado (RFC 3156)

PGP – *Pretty Good Privacy* (by L. Fish 1993)

The G-men all are cryin'
And tearin' out their hair,
'Cause there's a new cryptography
That's shown up everywhere.
Nobody can break it,
However good they be.
Everybody's PC got the PGP.

....

https://w2.eff.org/Net_culture/Folklore/Humor/pgp.song

Agenda na criação do sistema PGP (1)

- **Seleção das melhores práticas e técnicas criptográficas de domínio público e algoritmos criptográficos publicados e amplamente estudados para proteção de mensagens EMail**
  - Uso como mecanismos de elevada reputação e aceitação, pela comunidade de segurança, ambiente académico e comunidade de desenvolvimento e utilização de software livre
  - Adopção do algoritmo criptográfico IDEA
  - Embora opcionalmente também proposto com uso do algoritmo DES (e posteriormente 3DES)
- **Um modelo de TCB descentralizada para disseminação e gestão P2P de chaves poúblicas**
- **Integração numa ferramenta (*open-source*)**
  - Completamente auditável (código aberto)

Agenda na criação do sistema PGP (2)

- **Pacote com o software (open-source) e documentação publicados para uso livre**
  - O que motivou os problemas legais nos EUA e a prisão
  - E um amplo movimento de solidariedade

Agenda da criação do sistema PGP (3)

- Acordo com uma empresa (Viacrypt)
  - Para disponibilização de uma ferramenta inteiramente compatível com a versão Open Source, com condições de licenciamento de baixo custo, mas com garantias de desenvolvimento em base industrial
  - Assegurando o controlo do processo de desenvolvimento e evolução do software, APIs de integração e suporte de manutenção

- O sistema visou estabelecer uma "norma aberta e de facto" para troca segura e privada de mensagens de correio electrónico na INTERNET
  - Mas também para proteção de ficheiros em sistemas de ficheiros

PGP – *Pretty Good Privacy*

- Proteção de **autenticação, integridade e confidencialidade** de mensagens numa perspectiva **extremo-a-extremo** (entre emissor e receptor)
  - **Formatação/representação das mensagens cifradas**
  - **Suporte de não repudiação: possível confirmação autenticada de entrega e recepção**
- **Geração e gestão de chaves**
  - **One Time KG por cada mensagem**
  - **Gestão de chaveiros dos pares de chaves públicas/ privadas**
- **Modelo descentralizado de gestão de confiança das chaves públicas: Web-of-Trust Model**

Mensagem assinada em PGP

```
---BEGIN PGP SIGNED MESSAGE--

Hash: SHA1

Bob:My husband is out of town
   tonight.Passionately yours, Alice

---BEGIN PGP SIGNATURE---
Version: PGP 5.0
Charset: noconv
yhHJRHhGJGhgg/12EpJ
   +lo8gE4vB3mqJhFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

PGP Example: Confidentiality

PGP Example

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.5 (FreeBSD)
Comment: This is what your message looks like after encryption.

hQEOA1e+1x6YuUMCEAP/bWfPplkWHOtrEMxHcmcv+tP9jhg7Vcf/I0k5ee8jdsB9
mAS3smJ3F6xENYbTi+26SchP29jYR7xcIxjXi7LchNVDXYgxa3H2FiPxZrC2Hyfj
hA+nsVnEXGlgXsP91W520HZbQmCIEK1vaacc3TidOyTYBKrM1WUit4bhYsCGpAkD
+wROOFqGSfIErqLj1T4vIrlFhJaXC7f4uxfIGmSHzhRGi12Tz8zZEVmXKJqtnzw0
9zAF4+Qns/lhayI4C8JchNjiOhwiwpdomBLhAcV4kpWFu9zKgLMerIQ6R1FyJvVD
LQC7BFH1BUXru+wzJU5gxmOtvAvKiQTPmhizAFg7AzCZ0j8BTLGmboZLfcKR/Oqm
Y0bCsGdfGwQkAP+IzEOuu+ZzVBkZ5YwzPB5d0AGWnHcc2DDtsdy20wmD7Iv3dPL5
NKA=
=HUMH
-----END PGP MESSAGE-----
```

Descentralização da cadeia de confiança

Tópicos cobertos

- Segurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP
  - Contexto e criação do sistema PGP
  - Segurança e Mecanismos criptográficos
  - Outros componentes do processamento PGP
  - Processamento e formato de mensagens
  - Gestão de chaves públicas e modelo de confiança
  - Geração de chaves para encriptação de mensagens

Criptografia usada no sistema PGP e Mecanismos compementares

Autenticação + Integridade

Confidencialidade

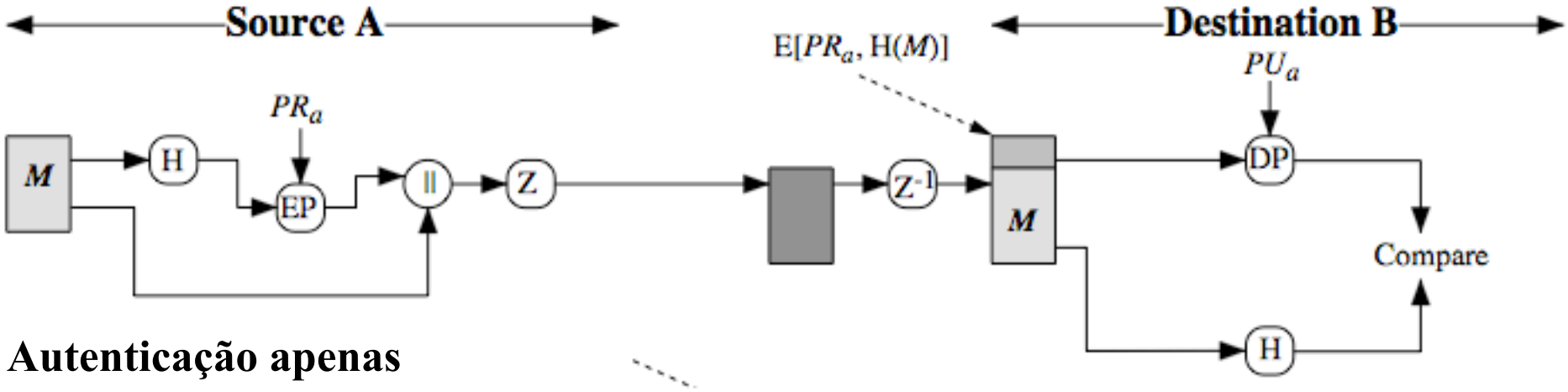
Compressão

Compatibilidade

Fragmentação ou segmentação

Function	Algorithms Used	Description
Digital signature	**DSS/SHA RSA/SHA**	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	**CAST, IDEA, TDES + DH or RSA**	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	**ZIP**	A message may be compressed, for storage or transmission, using ZIP.
Email compatibility	**RADIX 64**	To provide transparency for email applications, an encrypted message may be converted to an ASCII string using radix 64 conversion.
Segmentation	**Segmentation standardized**	To accommodate maximum message size limitations, PGP performs segmentation and reassembly.

Envio de mensagem com autenticidade

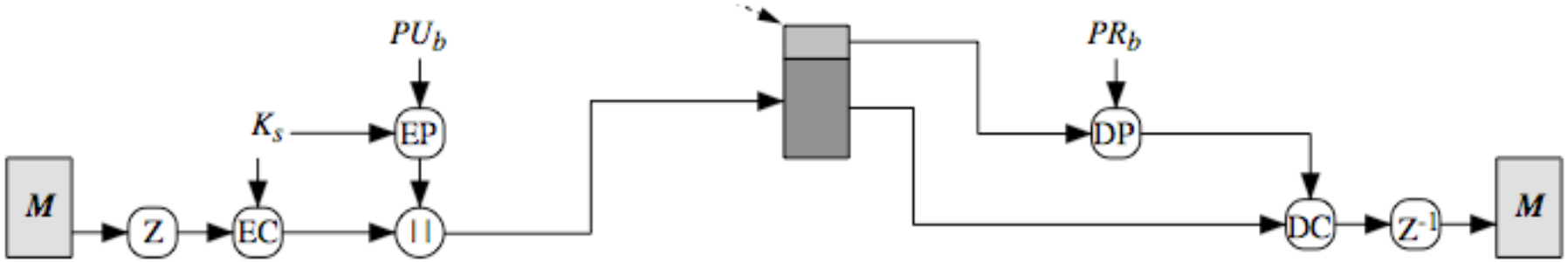


Autenticação apenas

Nota: COMPRESSÃO SEMPRE DEPOIS DA ASSINATURA !

PORQUÊ ?

Envio de uma mensagem confidencial

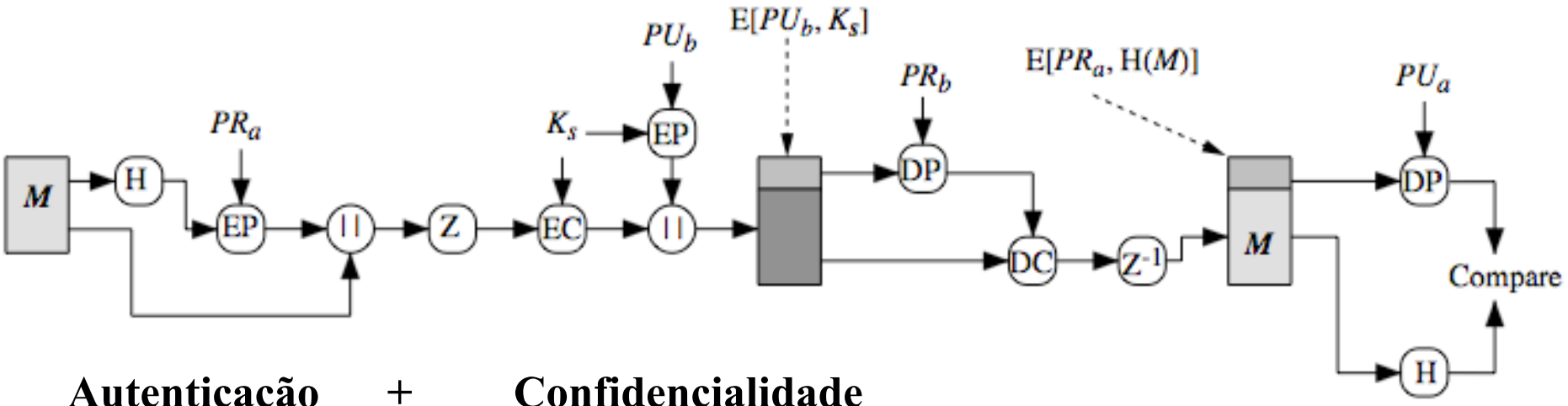


Confidencialidade apenas

Nota: COMPRESSÃO SEMPRE ANTES DA CIFRA DA MENSAGEM !

PORQUÊ ?

Confidencialidade e Autenticidade



Autenticação + Confidencialidade

Nota:
COMPRESSÃO
DEPOIS DA ASSINATURA E ANTES DA CIFRA DA MENSAGEM !

Tópicos cobertos

- Segurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP
  - Contexto e criação do sistema PGP
  - Segurança e Mecanismos criptográficos
  - Outros componentes do processamento PGP
  - Processamento e formato de mensagens
  - Gestão de chaves públicas e modelo de confiança
  - Geração de chaves para encriptação de mensagens

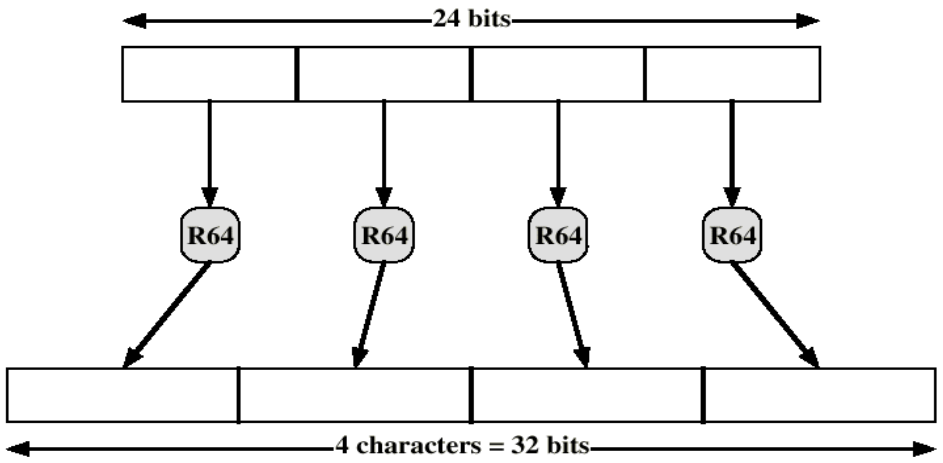
Compatibilidade com RADIX-64

- Incluido para garantir neutralidade face a limitações SMTP que só adoptem processamento de mensagens ASCII 7 bit
- Conversão e codificação Radix-64

Fragmentos de 6-bits (Representação Numérica: 0 a 63)

Codificação ASCII de Acordo com o valor numérico ASCII (8 bit, com BMS = 0), Com base numa tabela de conversão

3 bytes = 24 bits



4 bytes = 32 bits

Tabela de conversão

6-bit value	character encoding	6-bit value	character encoding	6-bit value	character encoding	6-bit value	character encoding
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

Ex:

00100011 01011100 10010001 => 01001001 00110001 01111001 01010010

235C91 (hex) => I1yR (ascii) = 49317952 (hex)

Compensação: RADIX64 vs. Compressão

- A Conversão RADIX expande a mensagem em cerca de 33%
- Mas a compressão tende a comprimir dessa mesma ordem os conteúdos texto

- Vantagens da compressão

  A compressão tem assim vários benefícios na aplicação no contexto do PGP:

  - Atenua o efeito da conversão RADIX 64 nos conteúdos de TEXTO
  - O pré-processamento da compressão antes da cifra, também é benéfico para efeitos de segurança, pois provoca quebra de regularidades de padrões de conteúdos de texto, antes da cifra (dificultando ataques por criptanálise de amostras PLAINTTEXT-CIPHERTEXT

Segmentação de mensagens

- Porquê ?
- Porque inicialmente (na altura da criação do sistema) existiam limites relativamente apertados nos tamanhos das mensagens (impostos pelos servidores SMTP durante o encaminhamento)
  - Exemplo: limitações a 50 K Bytes
  - Assim, o PGP segmentava as mensagens de modo a adaptá-las aos mínimos tamanhos vulgares noemalizando o respetivo processo de fragmentação na origem e reassemblagem no destino

Tópicos cobertos

- Segurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP
  - Contexto e criação do sistema PGP
  - Segurança e Mecanismos criptográficos
  - Outros componentes do processamento PGP
  - Processamento e formato de mensagens
  - Gestão de chaves públicas e modelo de confiança
  - Geração de chaves para encriptação de mensagens

Processamento PGP

Note-se que tudo o que se referiu para processamento de mensagens de correio electrónico, pode ser aplicado para efeitos de construção de uma ferramenta de proteção de ficheiros que usa o processamento normalizado PGP

- Para enviar (ou a disponibilizar) a destinatários, por algum protocolo do tipo FILE TRANSFER ou FILE DOWNLOAD

- Ou apenas para proteger localmente os ficheiros (por exemplo proteções de ficheiros em disco, pen-drives ou qualquer dispositivo do tipo File-Storage)

Proteção de mensagens (ou ficheiros) na origem

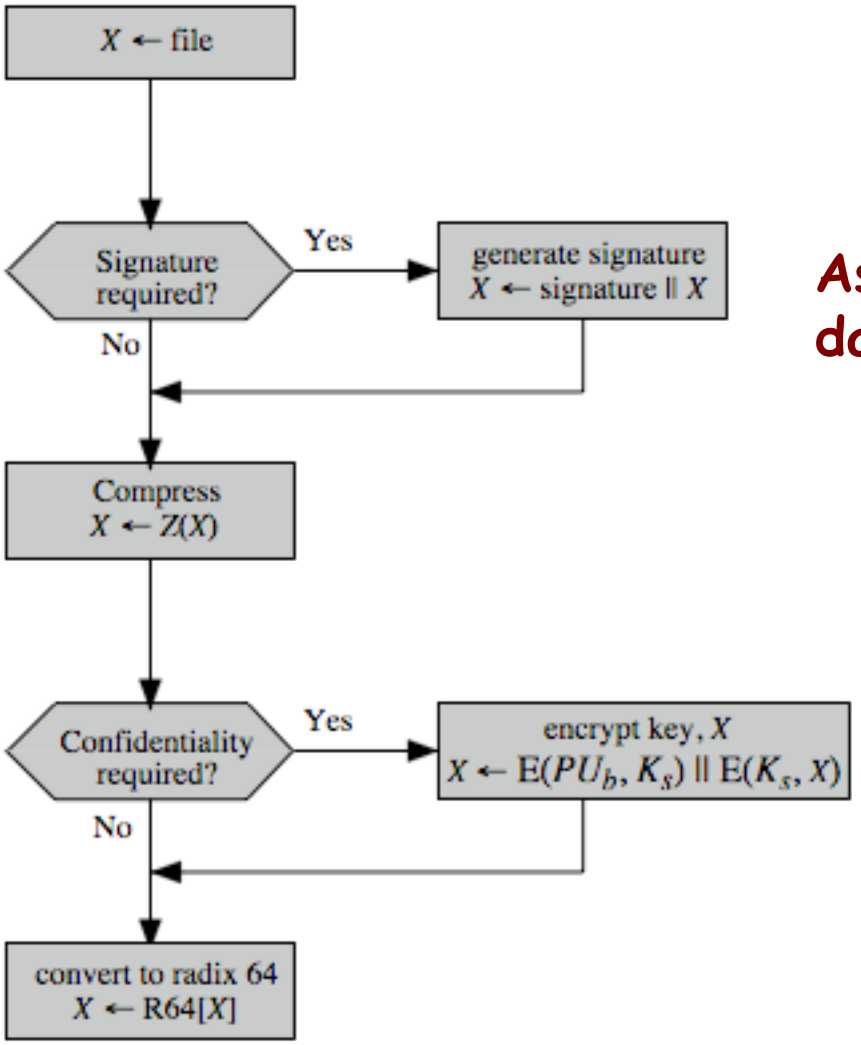
Mensagem (ou ficheiro)

Geração da Assinatura

Compressão

Encriptação

Conversão RADIX



Assinatura antes da Compressão

Proteção de mensagens (ou ficheiros) no destino

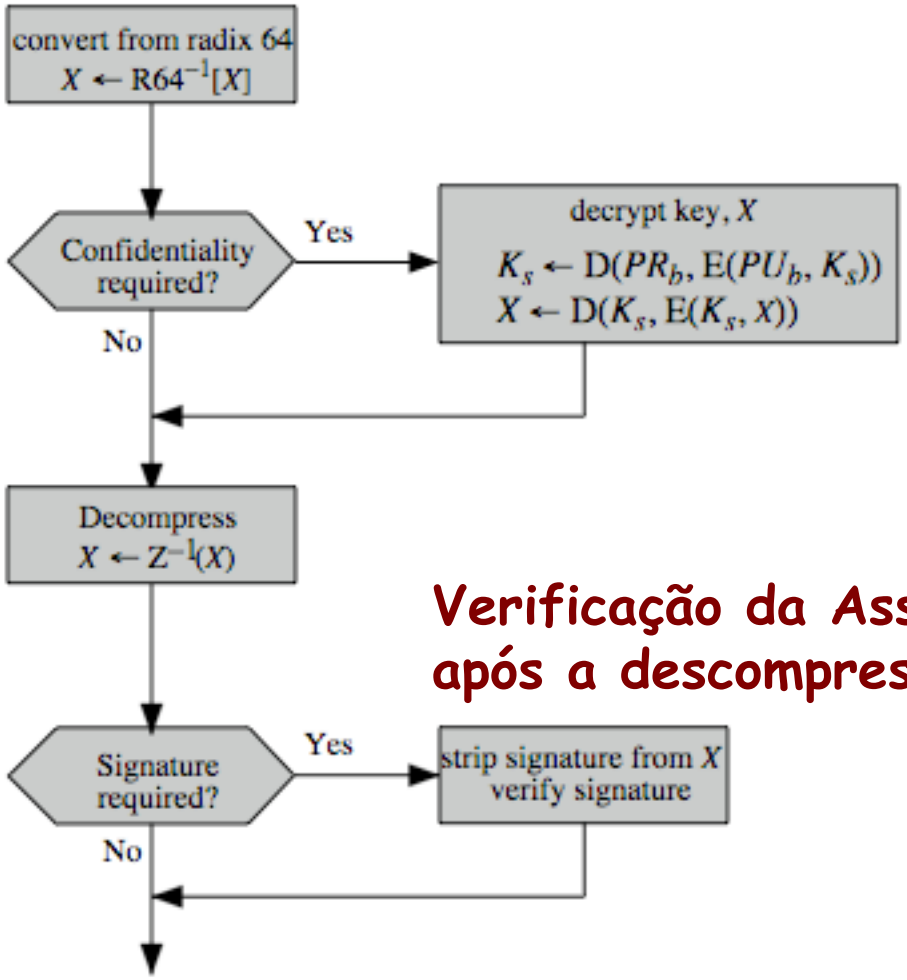
Mensagem (ou ficheiro)

Desencriptação

descompressão

Verificação da Assinatura

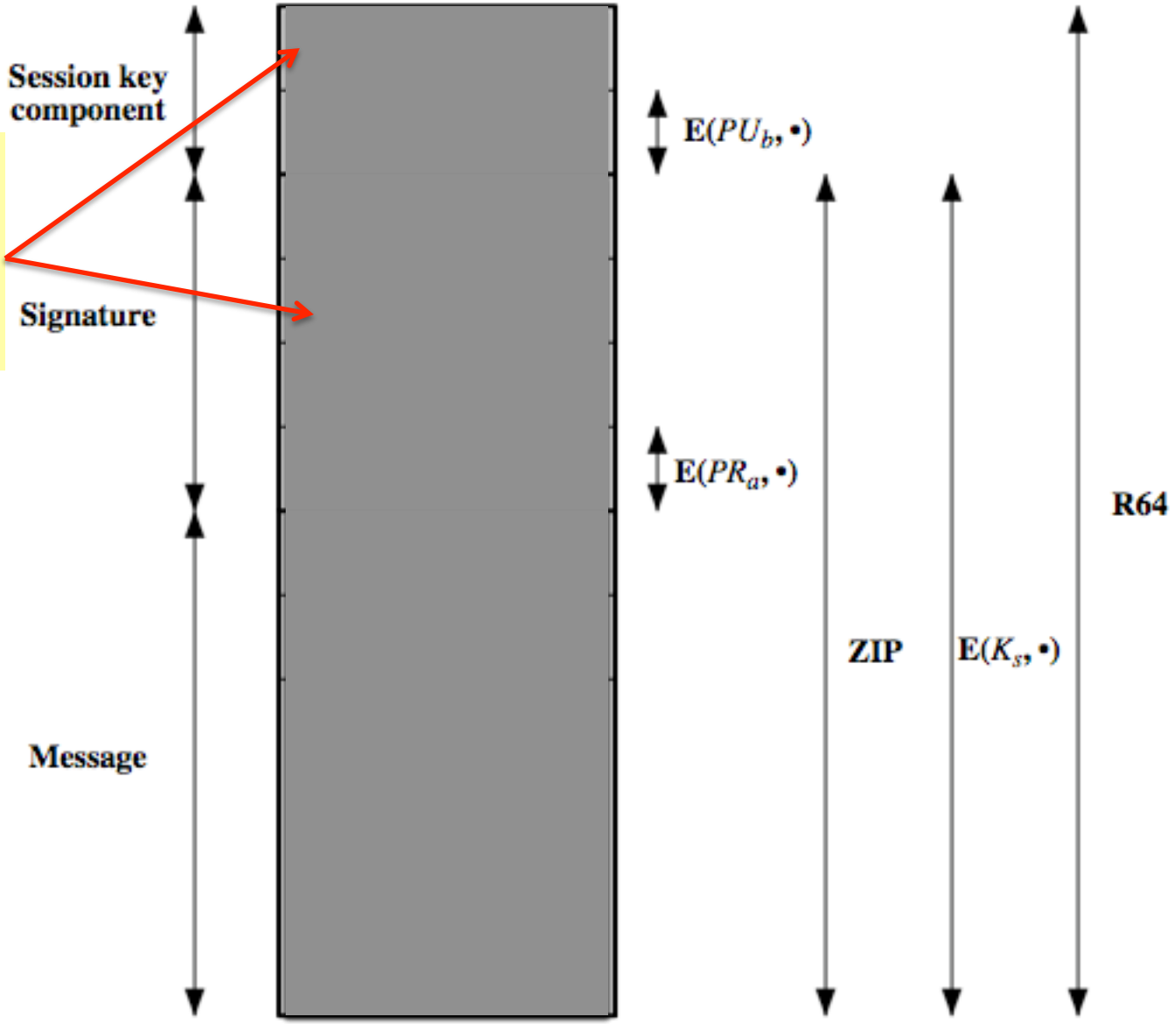
Recuperação da mensagem (ou ficheiro)



Verificação da Assinatura após a descompressão

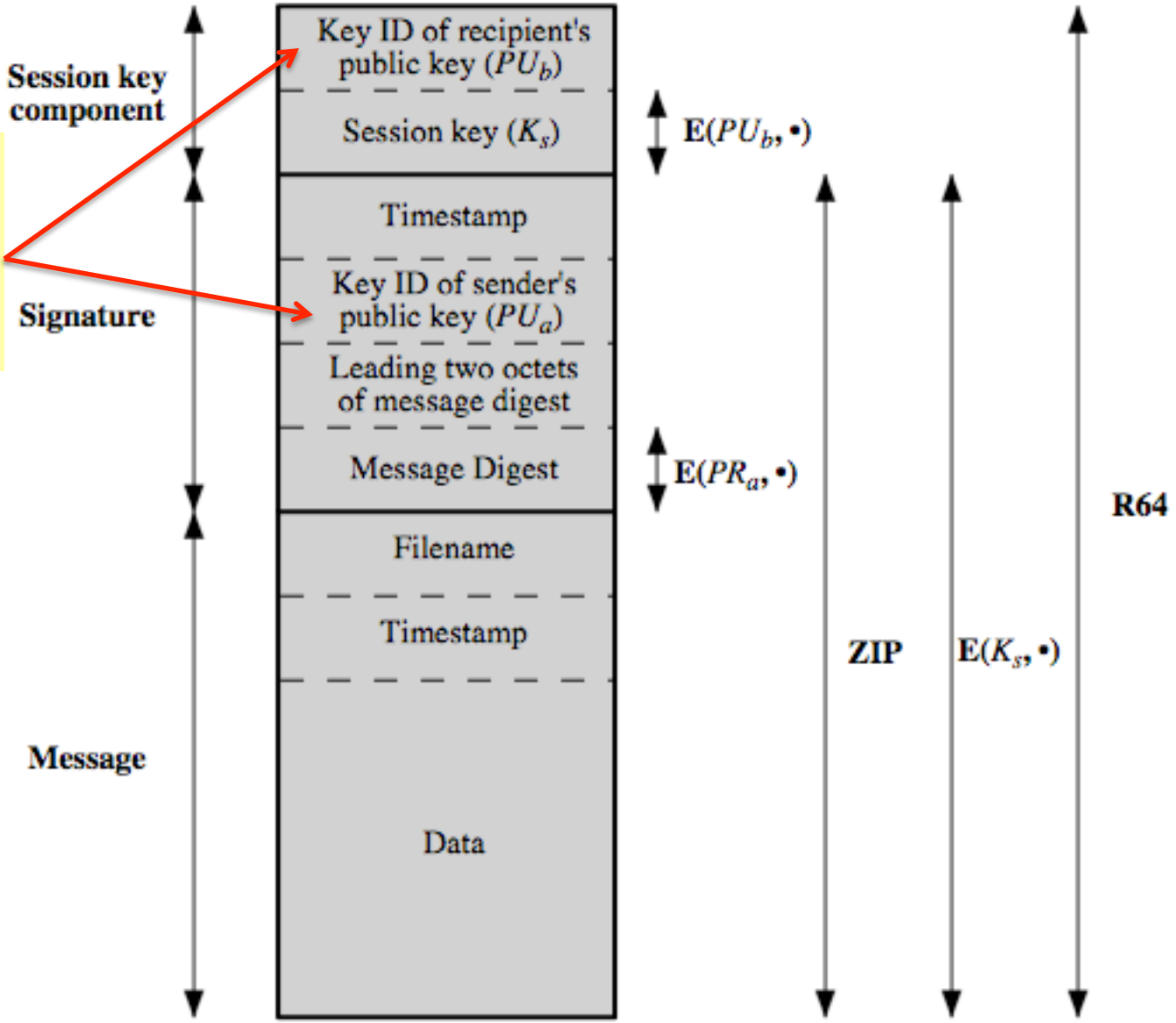
Formato da mensagem PGP

KeyIDs:
Least significant
64 bits of Pub Keys.
PU_a mod 2^{64}



Formato da mensagem PGP

KeyIDs:
Least significant
64 bits of Pub Keys.
PU_a mod 2^{64}



Tópicos cobertos

- Segurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP
  - Contexto e criação do sistema PGP
  - Segurança e Mecanismos criptográficos
  - Outros componentes do processamento PGP
  - Processamento e formato de mensagens
  - Gestão de chaves públicas e modelo de confiança
  - Geração de chaves para encriptação de mensagens

Gestão de chaves

Modelo *Web-of-Trust*

- O programa PGP vulgarizou um método de distribuição de chaves baseado num modelo conhecido como Web of Trust
- Consiste em cada utilizador adquirir e guardar localmente as chaves públicos dos interlocutores, por uma de duas vias:
  - através de outros utilizadores em quem tem confiança
    - Podendo as chaves públicas serem transmitidas em mensagens de correio electrónico autenticadas por estes
  - ou através de servidores públicos (utilizados como diretórios de chaves públicas de comunidades de utilizadores)

Gestão de chaves

- Cada utilizador gere dois chaveiros:
  - **PUBLIC KEY RING**
    - chaveiro de chaves públicas onde estão guardadas as chaves públcas de outros utilizadores
    - As chaves vão sendo guardadas à medida que o utilizador as vai obtendo de acordo com o modelo Web-of-Trust
  - **PRIVATE KEY RING**

    O chaveiro das suas chaves privadas

- Um utilizador pode ter mais do que um par de chaves (privada, pública), usando-as como entender no envio de mensagens
- Na recepção, deve ser usada a chave públcia adequada que foi usada pelo emissor

Gestão de chaves

- O sistema PGP prevê um método de assinatura digital de chaves públics para que cada principal possa assinar as chaves que envia a quem confia nele (no pressuposto que o destinatário já conhece e confia na(s) sua(s) chave(s) públoica(s)
- Para tal, estabelece-se uma métrica de confiança para diversos fins:
  - Confiança para recohecimento de autenticidade de uma mensagem normal
  - Confiança para reconhecimento de autenticidade de uma mensagem em que se anuncia a chave públcia de outro principal

- Bootstrap da Web of trust+: utilizam-se métodos administrativos de gestão das métricas de confiança, nomeadamente:
  - chaves iniciais depositadas no servidor de chaves públicas. Ou instaladas manualmente no chaveiro local de chaves públicas com a métrica de confiança a considerar como SETUP

Chaveiro de chaves privadas

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
• • •	• • •	• • •	• • •	• • •
T_i	PU_i mod 2^{64}	PU_i	E(H(P_i), PR_i)	User i
• • •	• • •	• • •	• • •	• • •

Campos usados para indexar a tabela

Chave sprivadas são mantidas
Protegidas (cifradas) com um
Esquema do tipo PBE encryption

$\{Kpriv\}_K$
K=H(passprhase)
Usando:
Cast-128, 3DES or IDEA

Chaveiro de chaves públicas

e processamento de confiança das entradas

KeyID or UserID Fields usados para indexar a tabela

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
T_i	$PU_i \bmod 2^{64}$	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

Trust level of the owner of each public key: How much this public key owner is trustable to sign other public keys ?

Trust level of key legitimacy: The public key belongs really to this UserID ?

Chaveiro de chaves públicas

e processamento de confiança das entradas

KeyID or UserID Fields used to index the table

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
T_i	$PU_i \bmod 2^{64}$	PU_i	trust_flag_i	User i	trust_flag_i		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

Trust level of the owner of each public key: How much this public key owner is trustable to sign other public keys ?

Trust level of key legitimacy: The public key belongs really to this UserID ?

Trust of each signatures

Gestão de métricas de confiança

Ownertrust
Trust assigned to
public key owners

OWNERTRUST 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)

BUCKSTOP bit
- set if this key appears in secret key ring

Keylegit
Trust assigned to
public key / user id pairs

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
Trust assigned Signature

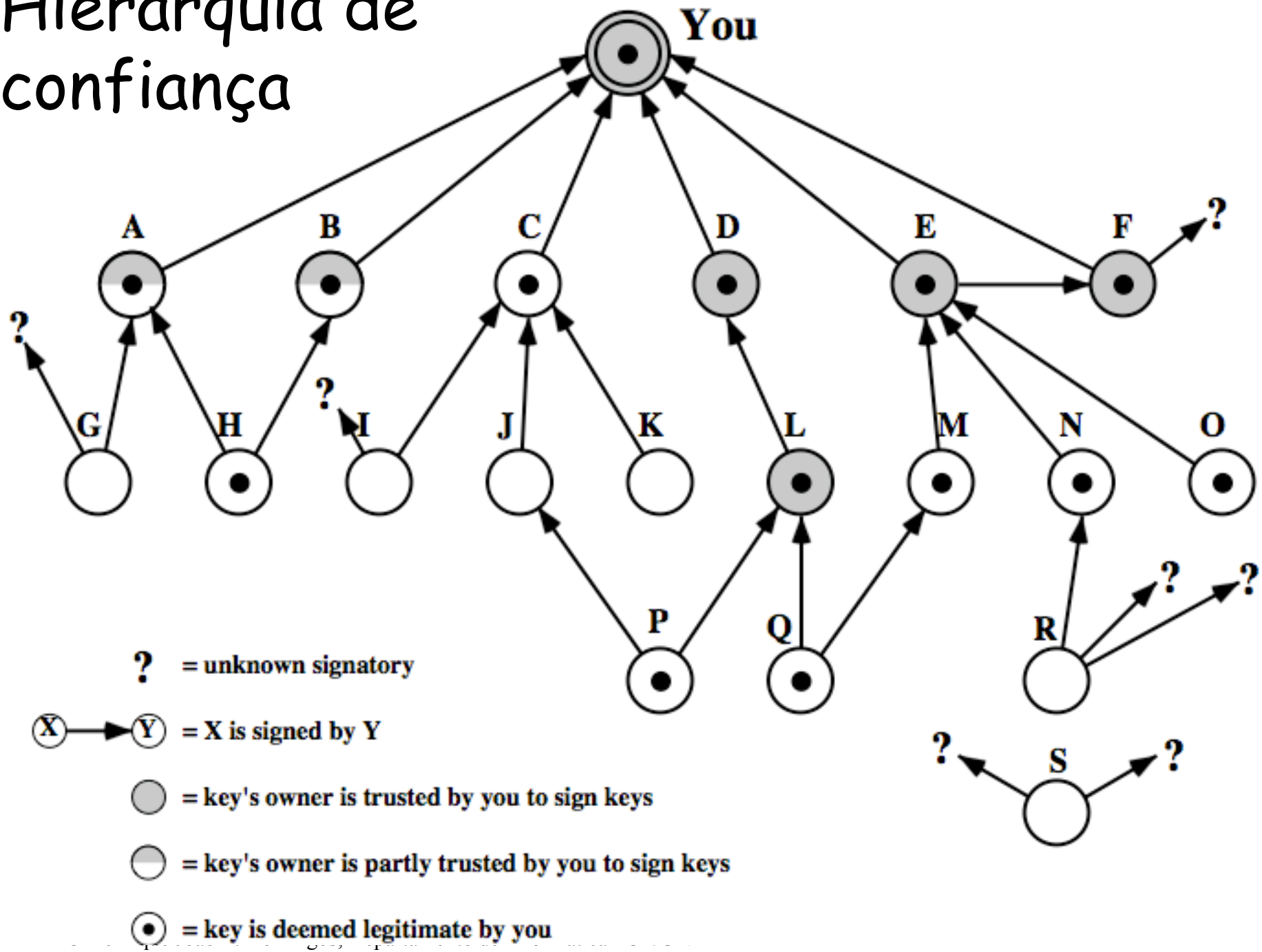
SIGTRUST Field
- undefined trust
- unknown user
- usually not trusted to sign other keys
- usually trusted to sign other keys
- always trusted to sign other keys
- this key is present in secret key ring (ultimate trust)

CONTIG bit
- set if signature leads up a contiguous trusted certification path back to the ultimately trusted key ring owner

***Ownertrust** field: trust metrics assigned to public-key owners*
***Keylegit** field: trust metrics assigned to PublicKey/UserID pairs*
***Sigtrust** field: trust metrics assigned to signatures*

Key legitimacy field = 1/X + 1/Y
Need X always trusted signatures + Y usually trusted

Hierarquia de confiança



Como e quando revogar as chaves públicas ?

- Revogar significa retirar as chaves públicas do chaveiro de chaves públcias

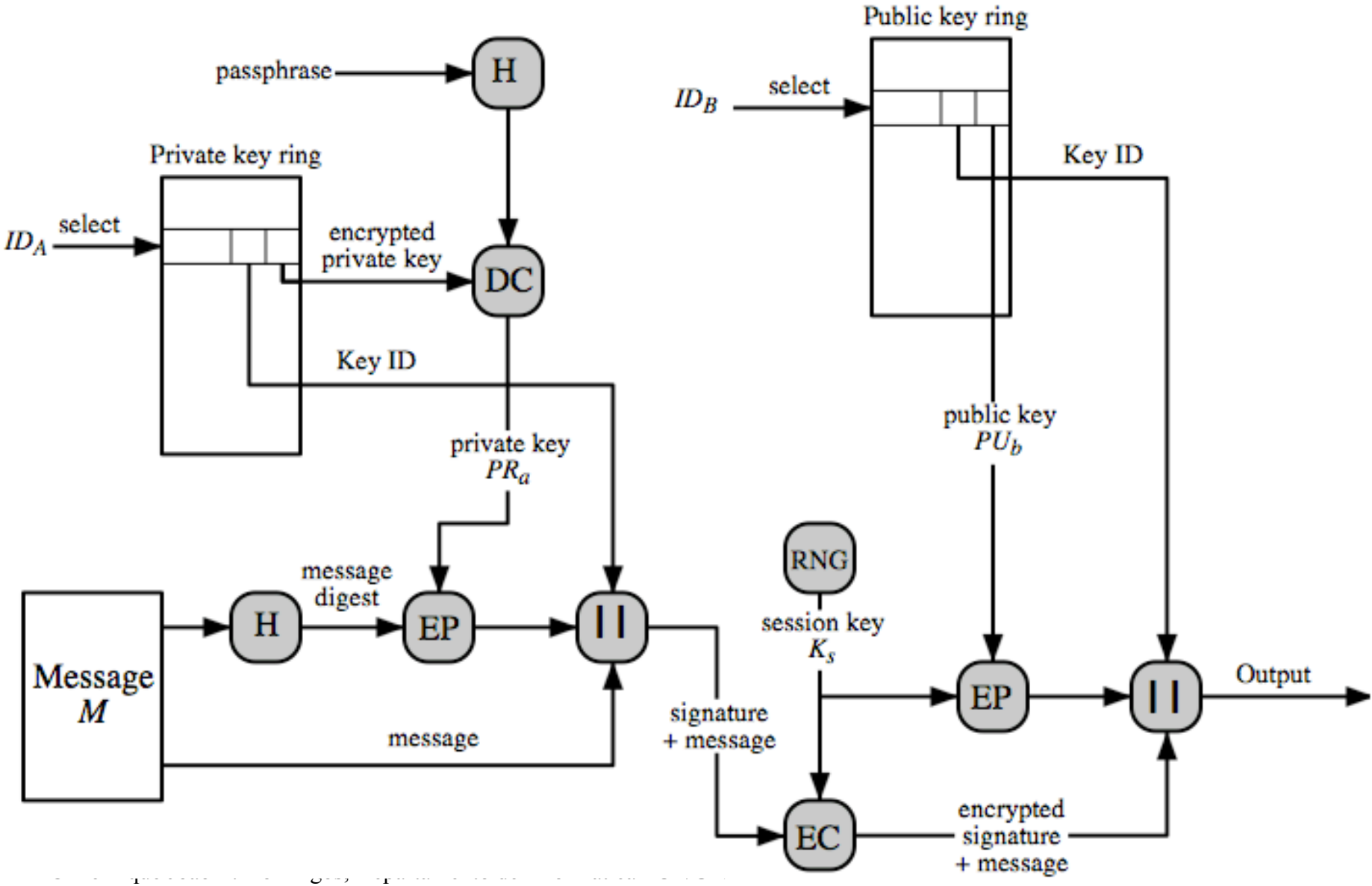
- Deve fazer-se quando:
  - O dono da chave públcia envia uma mensagem com um pedido de revogação (como um certificado de revogação)
  - Pode ser enviado numa mensagem enviada assinada pelo dono da chave públcia (assinado pela respetiva chave privada).

  - Notar que perante uma mensagem destas, temos mesmo que revogar a chave pública …

Porquê ?

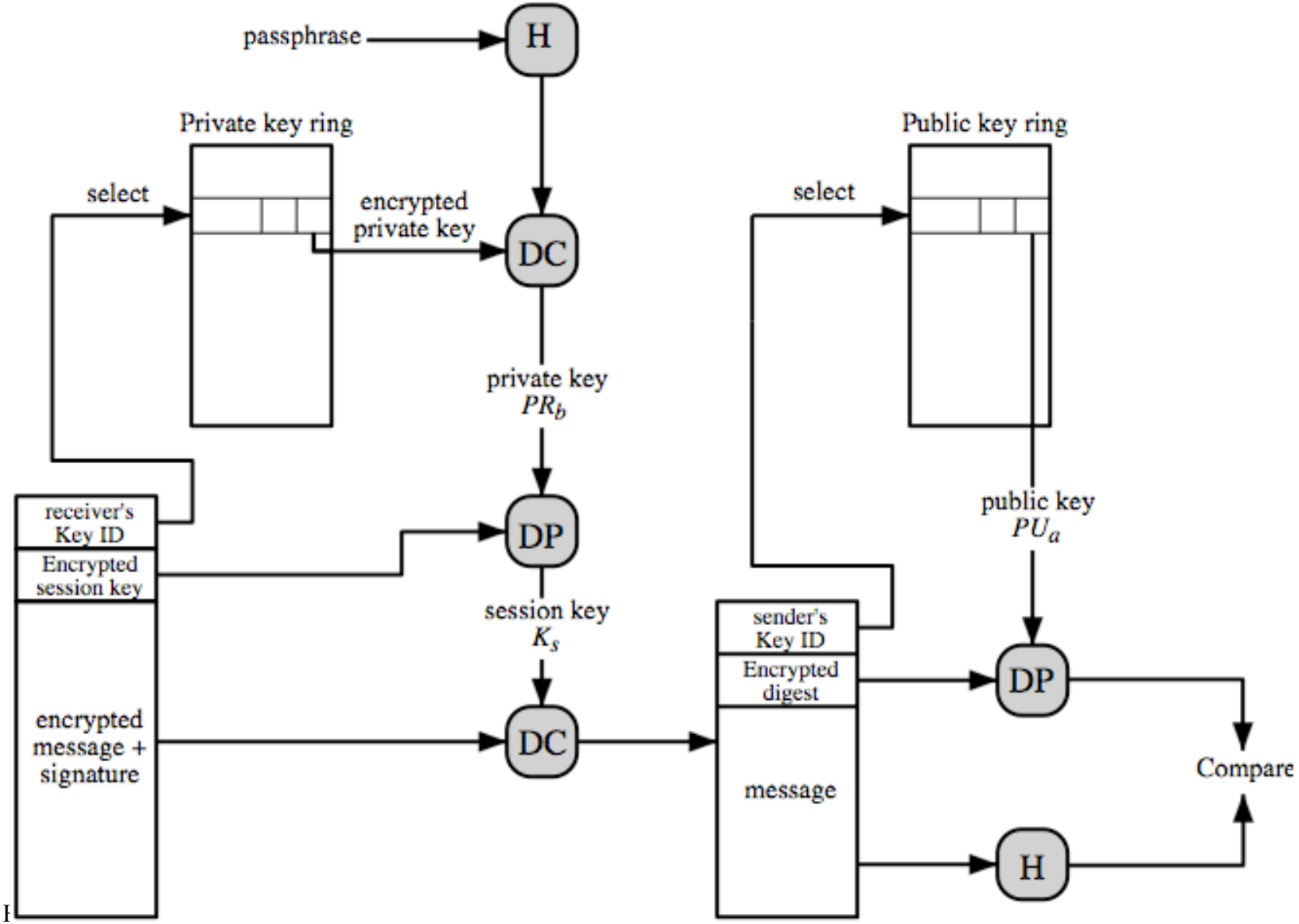
Processamento com os chaveiros

Cenário: A enviando mensagem a B



Processamento com os chaveiros

Cenário: B ao receber mensagem de A



Tópicos cobertos

- Segurança em sistemas E-Mail
- Segurança no envio e recepção de mensagens
- Segurança no sistema PGP
  - Contexto e criação do sistema PGP
  - Segurança e Mecanismos criptográficos
  - Outros componentes do processamento PGP
  - Processamento e formato de mensagens
  - Gestão de chaves públicas e modelo de confiança
  - Geração de chaves para encriptação de mensagens

Geração de chaves (*one-time-key*)

- Como anteriormente se viu, para suportar confidencialidade cada mensagem é enviada cifrada (usando criptografia simétrica), por uma chave gerada para essa mensagem
  - Chave usada uma única vez por cada mensagem

- Geração baseia-se num processo "*passphrase-key-generation*"

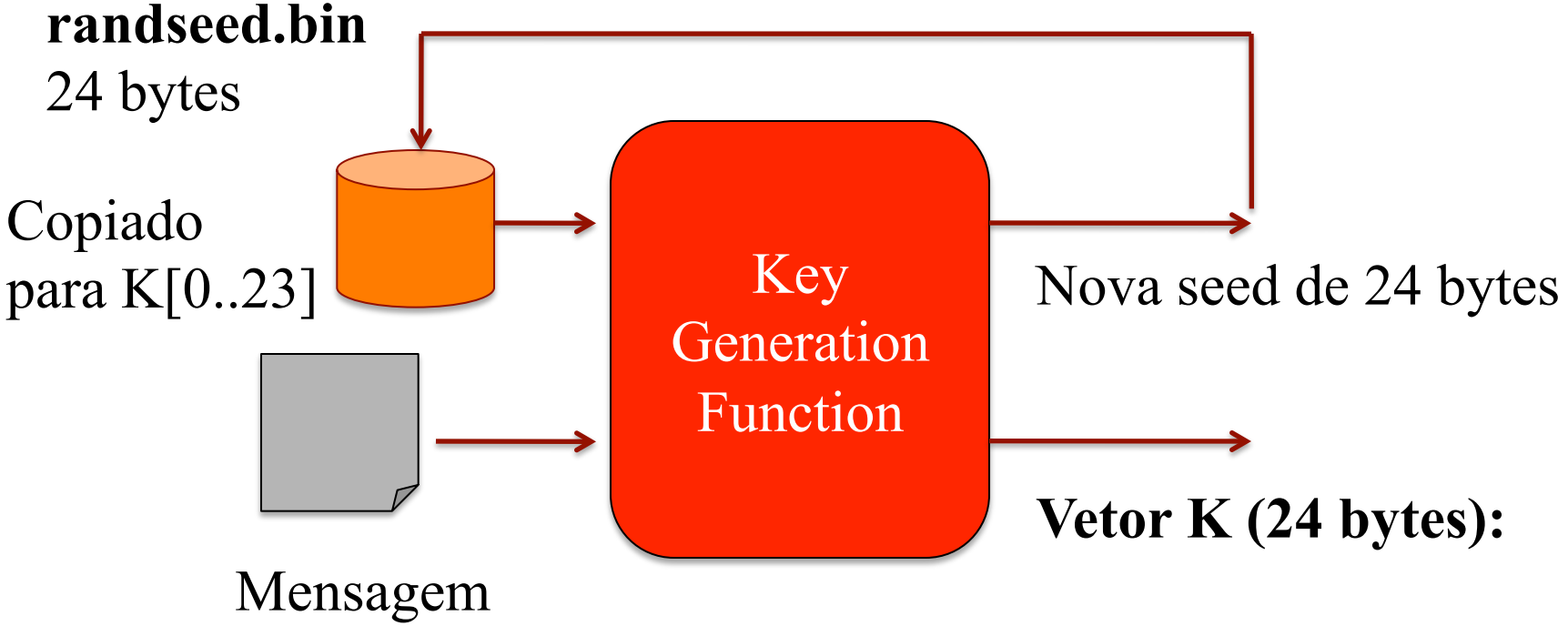
- Com base num processo que passa por uma geração PSEUDORANDOM com base num esquema normalizado: ANSI X12.17

Geração de chaves (*one-time-key*)

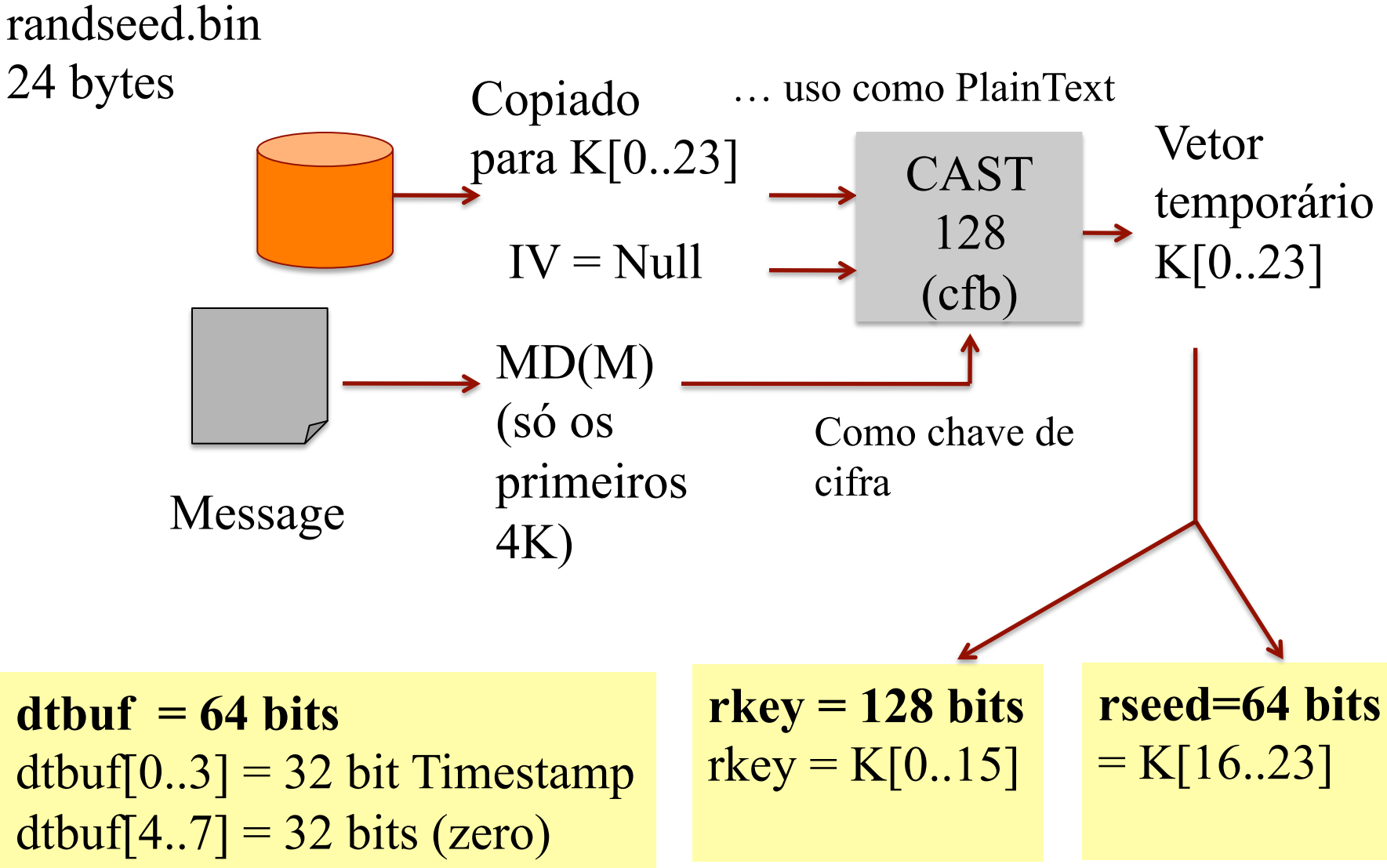
- Princípio: Usam-se *random inputs* a partir de mensagens enviadas antes e de um processo de sensoriamento do tipo "keystroke timing", na edição da mensagem em causa
- Para geração futura:

  - keystroke timming e a chave actual é usada como semente de um processo de geração aleatória, baseado em sucessivas iterações, de onde resultará a chave que irá ser usada para cifrar uma próxima mensagem (detalhes no livro – vefr informação disponível nas páginas da disciplina)

PGP: Processo de geração da chave



PGP: Processo de geração da chave

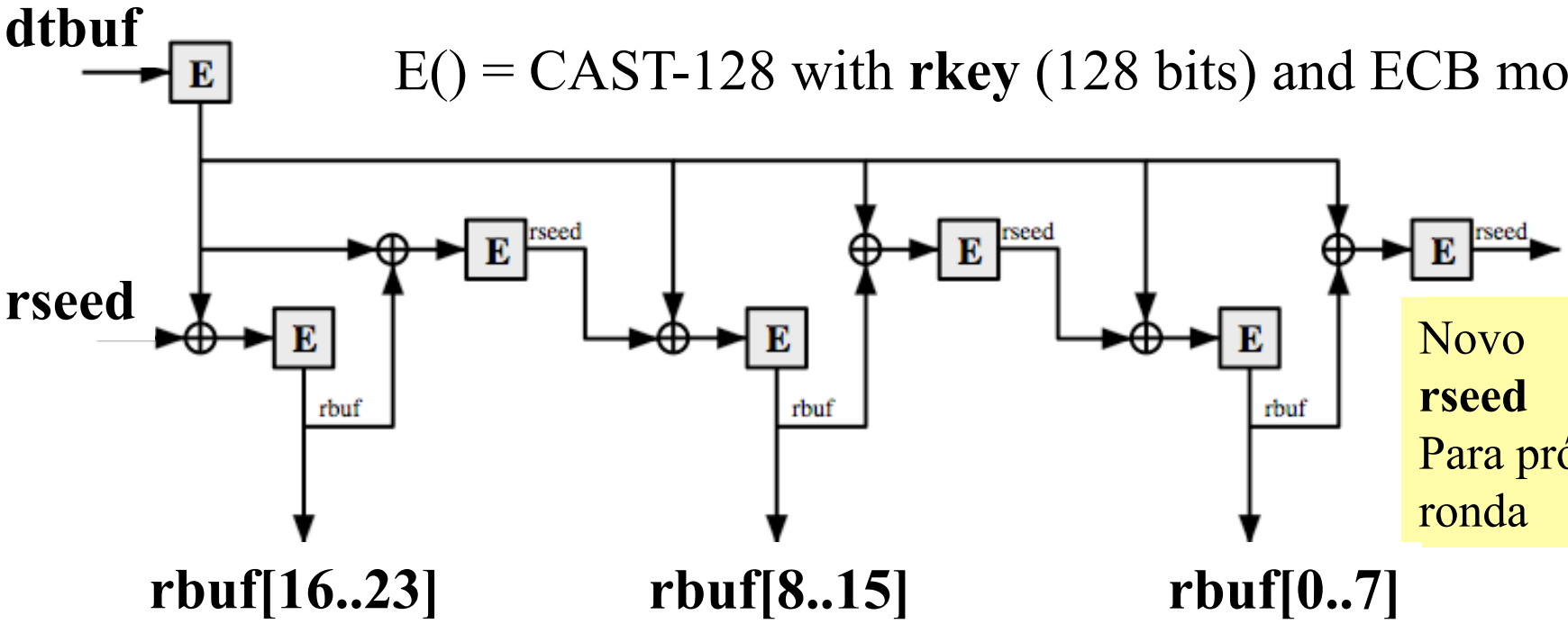


Função de geração da chave

dtbuf = 64 bits
dtbuf[0..3] = 32 bit Timestamp
dtbuf[4..7] = 32 bits (zero)

rkey = 128 bits
rkey = K[0..15]

rseed=64 bits
= K[16..23]



Novo **rseed** Para próxima ronda

Novo **rkey** + novo **IV** para próx. ronda

Novo **dtbuf = {dtbuf}rkey** para próx. ronda

Geração de random B e 1 byte (ronda i)
Transferido de **rbuf** para **K**: K[..] = B xor rbuf[…]

Postwash seed + return result

Geração de random B
e 1 byte (ronda i)
Transferido de **rbuf** para
K: K[..] = B xor rbuf[…]

- Do novo K[0—23]:
  - K[0..15] será a chave para cifrar a mensagem
  - K[16..12] será o vetor de incialização para modo CBC

O Processo é repetido 24 vezes, de modo a gerar mais 24 bytes para gerar uma chave K' (que será depositada no ficheiro randseed.bin)

novo valor de randseed.bin = $\{K'\}_K$ = 24 bytes

…. E tudo volta ao princípio quando se enviar a proxima mensagem

Uso do PGP

- Várias possibilidades ...
  - Distribuições Licenciáveis
  - Free com Dist. Source code

- Minhas favoritas:
  - GPG /Gnu PG – Gnu Privacy Guard
    - https://www.gnupg.org
  - GPG
    - https://gpgtools.org

Ex., GPG on MAC OS X

- Install from: https://gpgtools.org
- Keypair generation + Passphrase protection
- gpg –h
- Integrated with the Desktop environment
  - Right Button / Services … Done
- User Guides:

  https://gnupg.org/documentation/guides.html

Leituras sugeridas para estudo

Bibliografia

- W. Stallings, Network Security Essentials
  - Electronic Mail Security
    - PGP
      - Operational Description
      - Cryptographic Keys and Key Rings
      - Public Key Management (Web of Trust model)

COMPLEMENTOS / ANEXOS

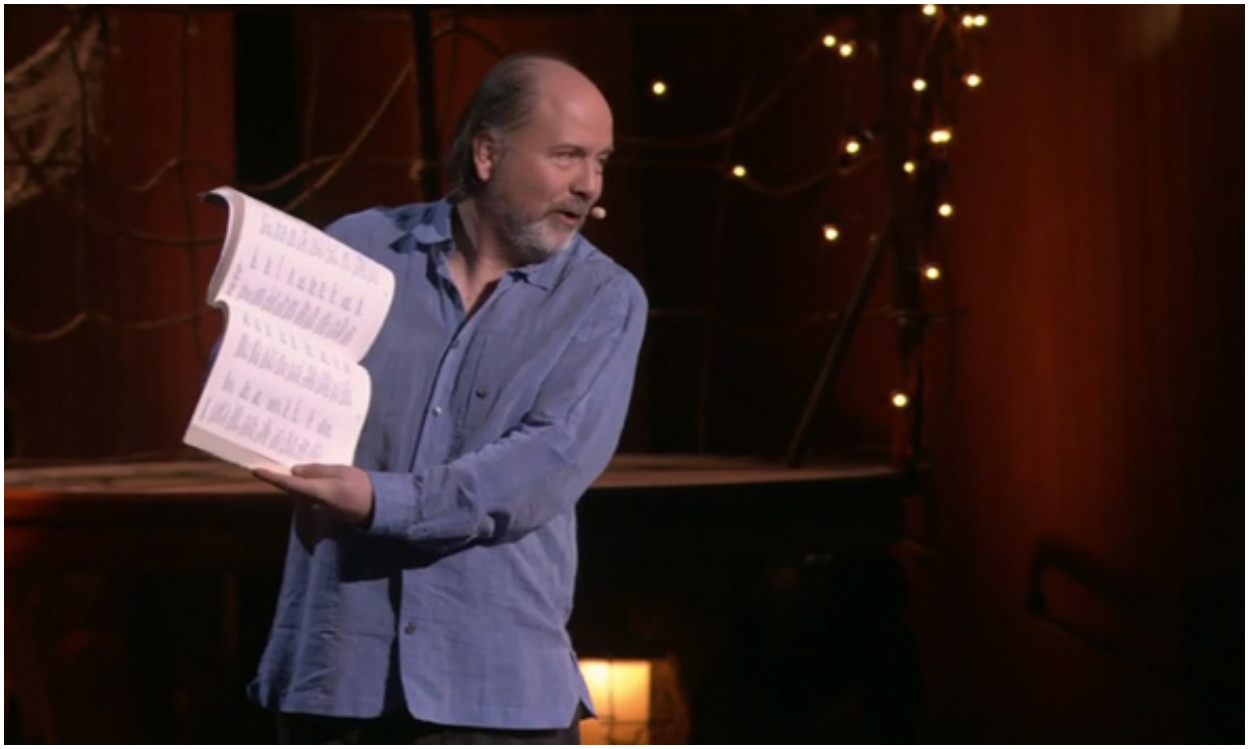
Correio Electrónico na Internet: Protocolos e sua história

"The Good Old Times of the Internet of "Well-Known/Trusted/Naïve/ Correct and Very Good People …"

Protocolos inicialmente desenhados, normalizados, implementados e operados com base em pressupostos de que a Internet era CONFIÁVEL !

Desenvolvimento: LEGACY TRUST ASSUMPTIONS ?!

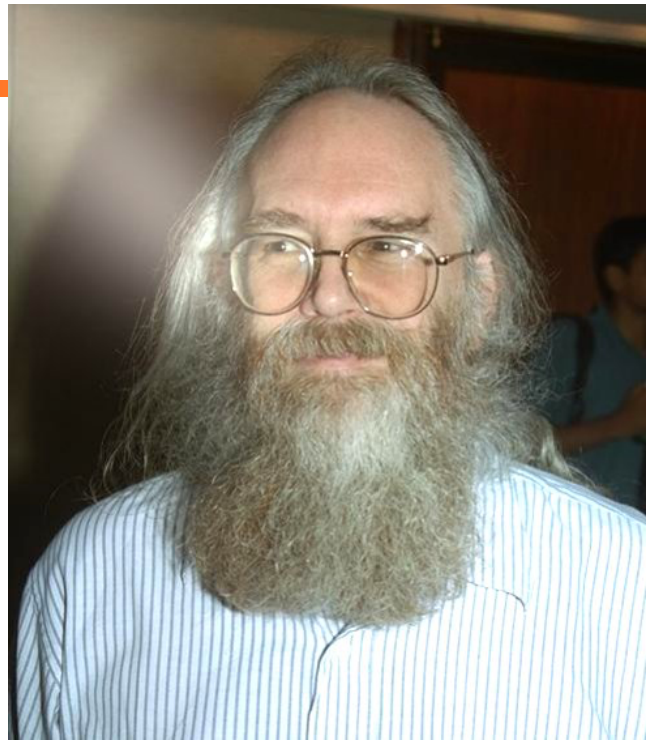
… What the internet looked like in 1982: A closer look at Danny Hillis' vintage directory of users …



http://blog.ted.com/what-the-internet-looked-like-in-1982-a-closer-look-at-danny-hillis-vintage-directory-of-users/

SMTP

- Originally, designed by J. Postel
  - RFCs: 780 (5/81), 788 (11/81), 821 (8/82)



… The objective of Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently. SMTP is independent of the particular transmission subsystem and requires only a reliable ordered data stream channel.

SMTP evol. RFCs: 974>1123>1425>1651>1869>5321

IMF

- David Crocker,
  - RFCs 561, 680, 724
  - RFC 822, Aug 1982

This standard specifies a syntax for text messages that are sent among computer users, within the framework of "electronic mail". The standard supersedes the one specified in ARPANET Request for Comments #733, "Standard for the Format of ARPA Net- work Text Messages". In this context, messages are viewed as having an envelope and contents.

IMF evol. RFCs, 2822 Apr 01 > 4021 , Mar 05 > 5322, Oct 08

MIME

- Multipurpose Internet Mail Extensions
- Extension to RFC 5322
  - Old RFCs: 1521, 1522, 1590
  - Established w/ RFC 2045 to 2049
- Updates:
  - RFCs 2184 > 2231
    - MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations
  - RFC 5335 > 6532
    - Internationalized Email Headers (convergence)

ESMTP, IMF and MIME

- ESMTP (RFCs 5321 > … 5386 > … 6531)
  - **This document specifies the Internet Message Format (IMF), a syntax for text messages** that are sent between computer users, within the framework of "electronic mail" messages. This specification is a revision of Request For Comments (RFC) 2822, which itself superseded Request For Comments (RFC) 822, "Standard for the Format of ARPA Internet Text Messages"
- IMF: Internet Message Format:
  - RFC 5322
- Internationalization:
  - RFCs 5336, 9/08) > 6531, 2/12

ESMTP

- Protocol used to transport internet mail.
- Used as both:
  - MTA-MTA: Inter-server transport protocol and (with restricted behavior enforced)
  - MUA-MSA-MTA: Mail submission protocol.

ESMTP (example)

- Protocol used to transport internet mail.
- Used as both:
  - MTA-MTA: Inter-server transport protocol and (with restricted behavior enforced)
  - MUA-MSA-MTA: Mail submission protocol.

Example: SMTP Dialogue (1) …

S: 220 smtp.server.com Simple Mail Transfer Service Ready

C: HELO client.example.com

S: 250 Hello client.example.com

C: MAIL FROM:<mail@samlogic.com>

S: 250 OK

C: RCPT TO:<john@mail.com>

S: 250 OK

C: DATA

S: 354 Send message content; end with <CRLF>.<CRLF>

C: *<The message data (body text, subject, e-mail header, attachments etc) is sent>*

C: .

S: 250 OK, message accepted for delivery: queued as 12345

C: QUIT

S: 221 Bye

Example ESMTP Dialogue (2) …

S: 220 smtp.server.com Simple Mail Transfer Service Ready
C: EHLO client.example.com
S: 250-smtp.server.com Hello client.example.com
S: 250-SIZE 1000000
S: 250 AUTH LOGIN PLAIN CRAM-MD5
C: AUTH LOGIN
S: 334 VXNlcm5hbWU6
C: adlxdkej
S: 334 UGFzc3dvcmQ6
C: lkujsefxlj
S: 235 2.7.0 Authentication successful

ESMTP + STARTTLS Dialogue …

- S: 220 smtp.server.com Simple Mail Transfer Service Ready
- C: EHLO client.example.com
- S: 250-smtp.server.com Hello client.example.com
- S: 250-SIZE 1000000
- S: 250-AUTH LOGIN PLAIN CRAM-MD5
- S: 250-STARTTLS
- S: 250 HELP
- C: STARTTLS
- S: 220 TLS go ahead
- C: EHLO client.example.com *
- S: 250-smtp.server.com Hello client.example.com
- S: 250-SIZE 1000000
- S: 250-AUTH LOGIN PLAIN CRAM-MD5
- S: 250 HELP
- C: AUTH LOGIN

…

ESMTP + STARTTLS Dialogue …

S: 334 VXNlcm5hbWU6
C: adlxdkej
S: 334 UGFzc3dvcmQ6
C: lkujsefxlj
S: 235 2.7.0 Authentication successful
C: MAIL FROM:<mail@samlogic.com>
S: 250 OK
C: RCPT TO:<john@mail.com>
S: 250 OK
C: DATA
S: 354 Send message, end with a "." on a line by itself
C: *<The message data (body text, subject, e-mail header, attachments etc) is sent>*
S .
S: 250 OK, message accepted for delivery: queued as 12345
C: QUIT
S: 221 Bye

POP (Post-Office Protocol)

- POP 3

The Post Office Protocol - Version 3 (POP3) is intended to permit a workstation to dynamically access a maildrop on a server host in a useful fashion. Usually, this means that the POP3 protocol is used to allow a workstation to retrieve mail that the server is holding for it.

  - RFCs 1081 > 1225 > 1460 > 1725 > 1939
  - Updates/Extensions:
    - RFCs **1957, 2449, 5034, 6186**

POP3 Dialogue

telnet pop.eo.lu 110
Trying 194.177.33.81...
Connected to unicorn.europeonline.net.
Escape character is '^]'.
+OK Europe Online Mail Server.
USER rewt
+OK rewt selected
PASS XXXXXXXX
+OK Congratulations!
QUIT
+OK blah blah

Other commands:

POP3 + APOP Dialogue (1)

S: <wait for connection on TCP port 110>
C: <open connection>
S: +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C: APOP mrose c4c9334bac560ecc979e58001b3e22fb
S: +OK mrose's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S: <the POP3 server sends message 1>

POP3 + APOP Dialogue (2)

S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S: <the POP3 server sends message 2>
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <close connection>
S: <wait for next connection>

IMAP (Internet Message Access Protocol)

- Allows a client to access and manipulate electronic mail messages on a server. IMAP4rev1 permits manipulation of mailboxes (remote message folders) in a way that is functionally equivalent to local folders. IMAP4rev1 also provides the capability for an offline client to resynchronize with the server.
- RFCs 2060 > **3501** (Ma5/03)
  - **INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1**
  - **Ext. RFC 4466 > ..... > 6858**
  - RFC 7817:

    Updated Transport Layer Security (TLS) Server Identity Check Procedure for Email-Related Protocols

IMAP Example (1)

C: <open connection>
S: * OK IMAP4rev1 Service Ready
C: a001 login mrc secret
S: a001 OK LOGIN completed
C: a002 select inbox
S: * 18 EXISTS
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * 2 RECENT
S: * OK [UNSEEN 17] Message 17 is the first unseen message
S: * OK [UIDVALIDITY 3857529045] UIDs valid
S: a002 OK [READ-WRITE] SELECT completed

IMAP Example (2)

C: a003 fetch 12 full

S: * 12 FETCH (FLAGS (\Seen) INTERNALDATE "17-Jul-1996 02:44:25 -0700"
RFC822.SIZE 4286 ENVELOPE ("Wed, 17 Jul 1996 02:23:25 -0700 (PDT)"
"IMAP4rev1 WG mtg summary and minutes"
(("Terry Gray" NIL "gray" "cac.washington.edu"))
(("Terry Gray" NIL "gray" "cac.washington.edu"))
(("Terry Gray" NIL "gray" "cac.washington.edu"))
((NIL NIL "imap" "cac.washington.edu"))
((NIL NIL "minutes" "CNRI.Reston.VA.US")
("John Klensin" NIL "KLENSIN" "MIT.EDU")) NIL NIL
"<B27397-0100000@cac.washington.edu>")
BODY ("TEXT" "PLAIN" ("CHARSET" "US-ASCII") NIL NIL "7BIT" 3028
92))

IMAP Example (3)

S: a003 OK FETCH completed
C: a004 fetch 12 body[header]
S: * 12 FETCH (BODY[HEADER] {342}
S: Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
S: From: Terry Gray <gray@cac.washington.edu>
S: Subject: IMAP4rev1 WG mtg summary and minutes
S: To: imap@cac.washington.edu
S: cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
S: Message-Id: <B27397-0100000@cac.washington.edu>
S: MIME-Version: 1.0
S: Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
S:
S:)
S: a004 OK FETCH completed

IMAP Example (4)

C a005 store 12 +flags \deleted
S: * 12 FETCH (FLAGS (\Seen \Deleted))
S: a005 OK +FLAGS completed
C: a006 logout
S: * BYE IMAP4rev1 server terminating connection
S: a006 OK LOGOUT completed

IMAP c/ STARTTLS

openssl s_client -crlf -debug -connect imap.gmail.com:993

....

tag username password

...

openssl s_client -showcerts -connect mail.example.com:993

...