

# Network Security: Threats and Goals

Tuomas Aura

T-110.5240 Network security  
Aalto University, Nov-Dec 2013

# Outline

1. Network security
2. Basic network threats: sniffing and spoofing
3. Role of cryptography
4. Security and the network protocol stack
5. Case study: email security

# Network security

# What is network security

- Network security protects against **intentional bad things done to communication**
  - Protect messages (data on wire) and communication infrastructure
- Network security goals:
  - **Confidentiality** — no sniffing
  - **Authentication and integrity** — no spoofing of **data or signaling**, no man-in-the-middle attacks
  - **Access control** — no unauthorized use of network
  - **Availability** — no denial of service by preventing communication
  - **Privacy** — no traffic analysis or location tracking

# Authentication and integrity

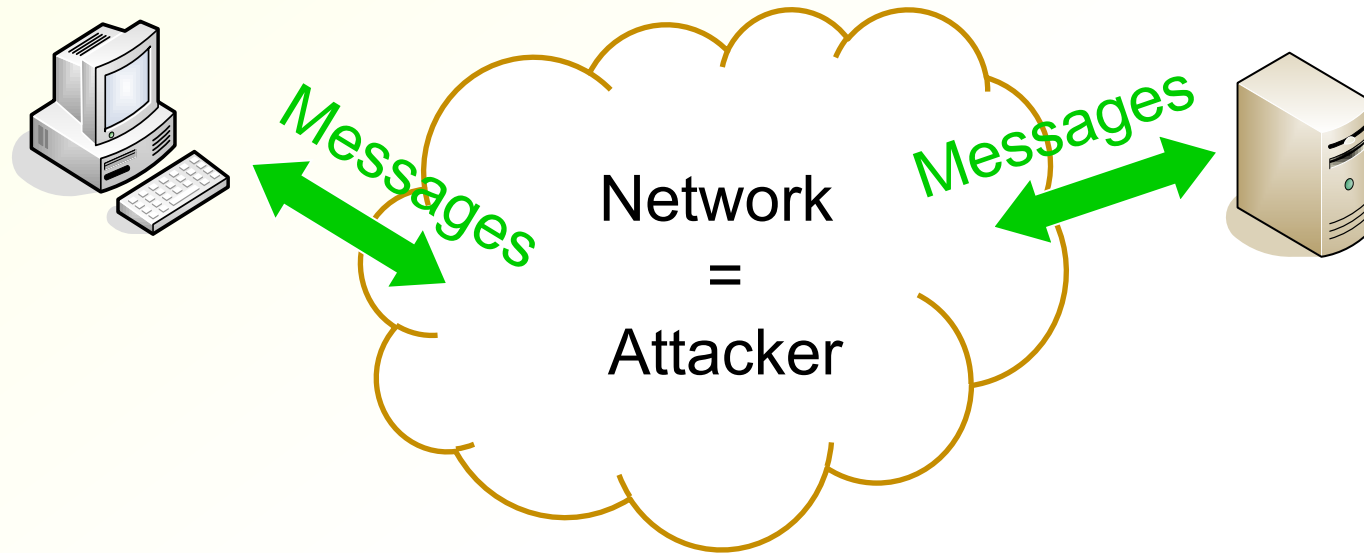
- **Peer-entity authentication** = verify the presence and identity of a person, device or service at the time; e.g. car key
- **Data origin authentication** = verify the source of data
- **Data integrity** = verify that the data was received in the original form, without malicious modifications
- In practice, **data origin authentication and integrity check** always go together
- Authentication (usually) requires an **entity name or identifier**

# Who is the attacker?

- We partition the world into the good and bad side
  - Honest parties vs. attackers
  - Different partitions lead to different perspectives on the security of the same system
- Typical attackers:
  - Curious or dishonest individuals — for personal gain
  - Hackers, crackers, script kiddies — for challenge and reputation
  - Political activists — for political pressure
  - Companies — for business intelligence and marketing
  - Organized criminals — for money
  - Security agencies — NSA, SVR, GCHQ, DGSE, etc.
  - Military SIGINT — strategic and tactical intelligence, cyber attacks
- Often, not all types of attackers matter to everyone
  - E.g. would you care if NSA/university/mom read your email?

# Basic network threats: sniffing and spoofing

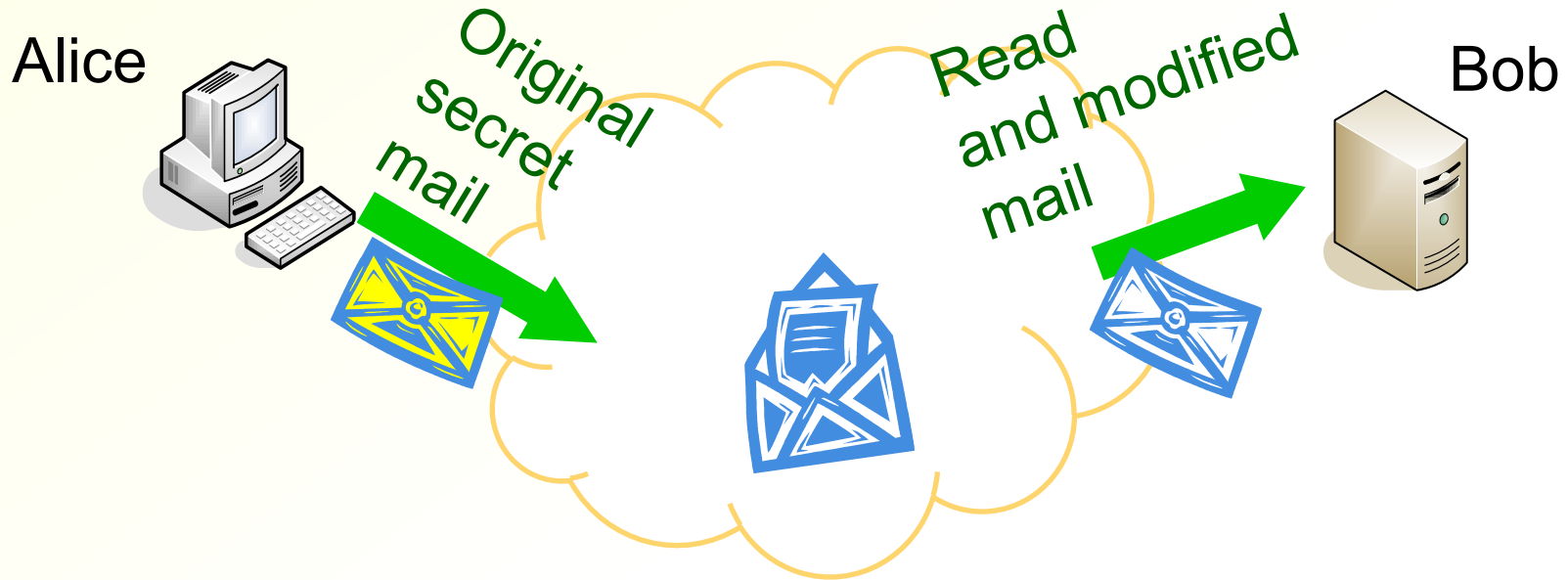
# Traditional network-security threat model



- End nodes trusted, the network is unreliable
- End nodes send messages to the network and receive messages from it
- Network will deliver some messages but it can read, delete, modify and replay them
- Metaphors: unreliable postman, notice board, rubbish basket



# Example: email



- Alice sends an email that is addressed to Bob
- The attacker may read, delete and edit the email. It may copy the email, or cut and paste pieces from one email to another. It may write a new email
- Secrets and message integrity need protection

# Basic network security threats

- Traditional major threats:
  - **Sniffing** = attacker listens to network traffic
  - **Spoofing** = attacker sends unauthentic messages
  - Data modification (**man in the middle**) = attacker **intercepts** and modifies data
- Corresponding security requirements:
  - Data **confidentiality**
  - Data-origin **authentication** and data **integrity**

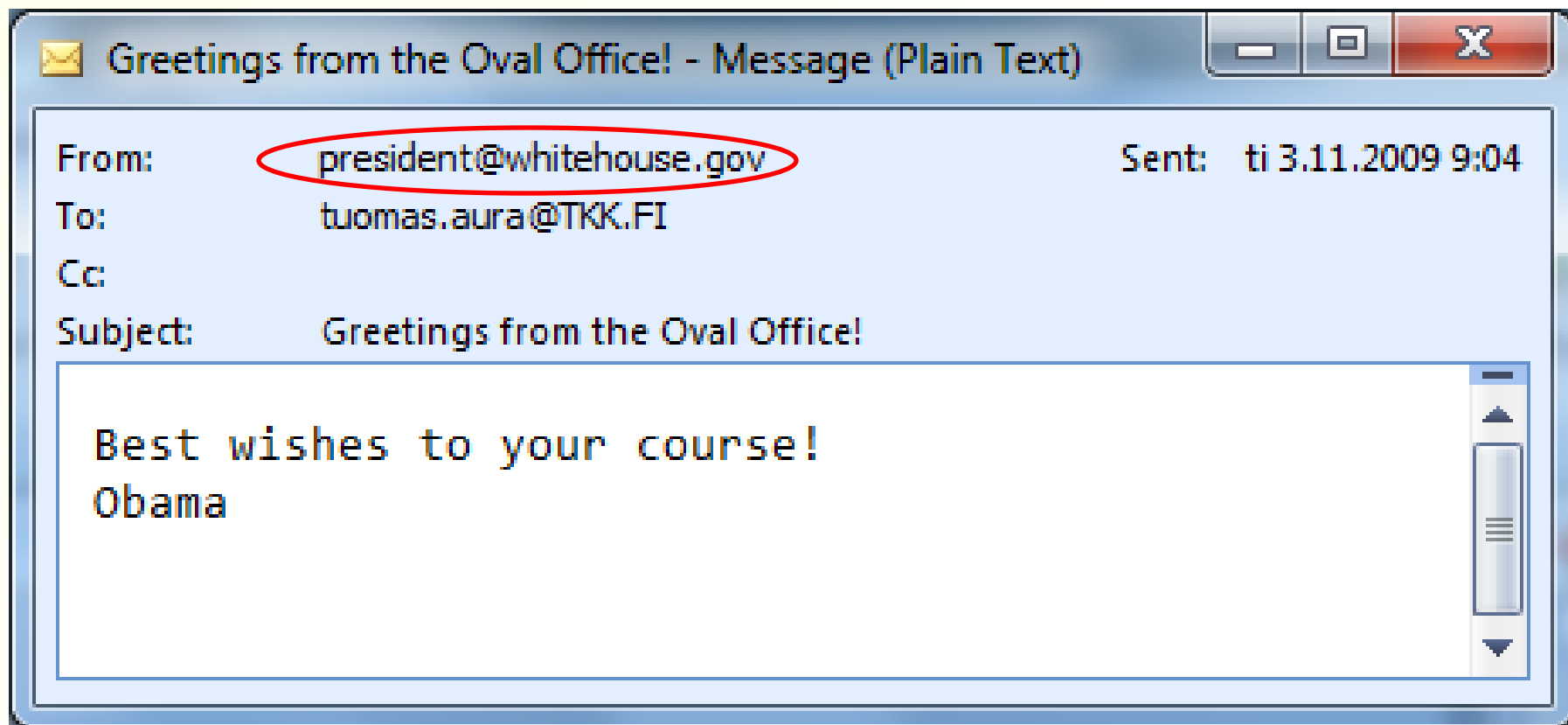
# Sniffing

- Sniffing = eavesdropping = spying = snooping = unauthorized listening = monitoring
- Sniffers:
  - Packets are sometimes broadcast on a local link  
→ all local nodes can listen
  - Sniffers listen to packets on the network and pick out interesting details, e.g. passwords
  - Hackers install sniffer software on compromised hosts; tools are available for download
  - Open wireless networks are most vulnerable – but tools exist on sniffing all types of networks
- Network admins and spies can monitor packets on routers, firewalls and proxies
  - Router security may become a serious issue in the near future

# Spoofing

- Spoofing = sending unauthentic/false/counterfeit messages = using false sender address or identifier = impersonation
- In the Internet, it is easy to send messages that appear to come from someone else
  - A modified version of the application or protocol stack is easy to write
- Examples:
  - Email spoofing: false From field
  - IP spoofing: false source IP address
  - DNS spoofing: false DNS responses
  - Mobile-IP BU spoofing: false location information

# Example: email spoofing



# Example: email spoofing

- SMTP does nothing to authenticate the sender

```
C:>telnet smtp.kolumbus.fi 25
220 emh05.mail.saunalahti.fi ESMTP Postfix
ehlo nowhere.net
250-emh05.mail.saunalahti.fi
250-PIPELINING
250-SIZE 280000000
250-8BITMIME
mail from: president@whitehouse.gov
250 2.1.0 ok
rcpt to: tuomas.aura@tkk.fi
250 2.1.5 ok
data
354 End data with <CR><LF>.<CR><LF>
From: president@whitehouse.gov
To: tuomas.aura@tkk.fi
Subject: Greetings from the oval office!

Best wishes to your course!
Obama
.
250 2.0.0 ok: queued as 9935A27D8C
```

# Example: IP spoofing

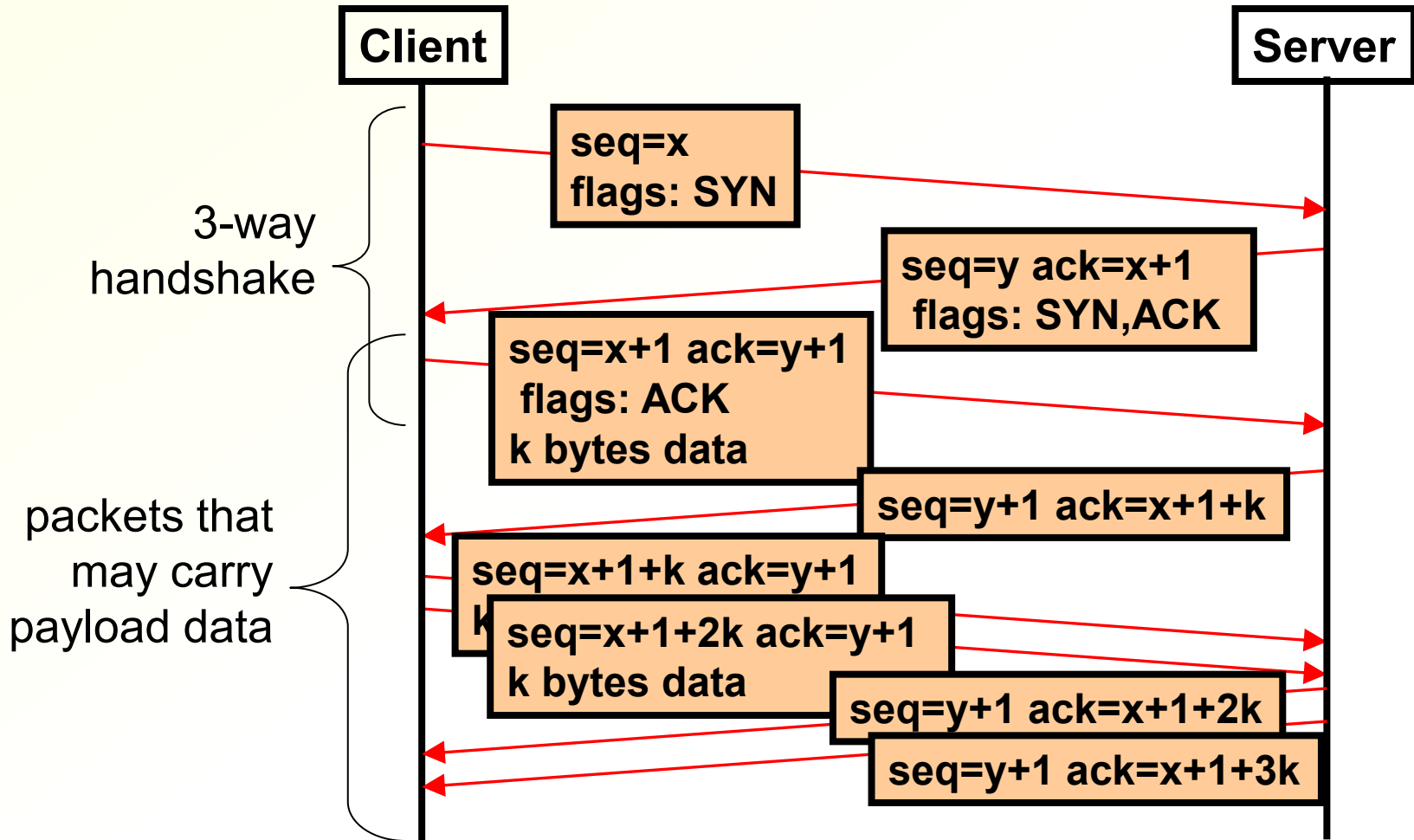
- Attacker sends IP packets with **false source address**
  - Anyone can write software to do this with raw sockets
- The destination node usually believes what it sees in the source address field
- Attacker may be anywhere on the Internet
- **Spoofing a connection** between A and B is more difficult:
  - Attacker must sniff replies from B in order to continue the conversation
  - Attacker must be on the route between A and B, or control a router on that path

# TCP sequence numbers and IP spoofing

- TCP sequence numbers are **initialized to random values** during the connection handshake
- Acknowledgment number in the third packet must be sequence number of the second packet + 1
- Sequence numbers are incremented for each byte sent. Packets must arrive in order
- Receiver rejects packets with incorrect sequence numbers and waits for the correct ones
- **TCP packets are difficult to spoof because the attacker must sniff or guess the sequence number**
- **Not secure in the traditional network security threat model, but limits attacks quite well**
- The first packet (SYN) is easy to spoof

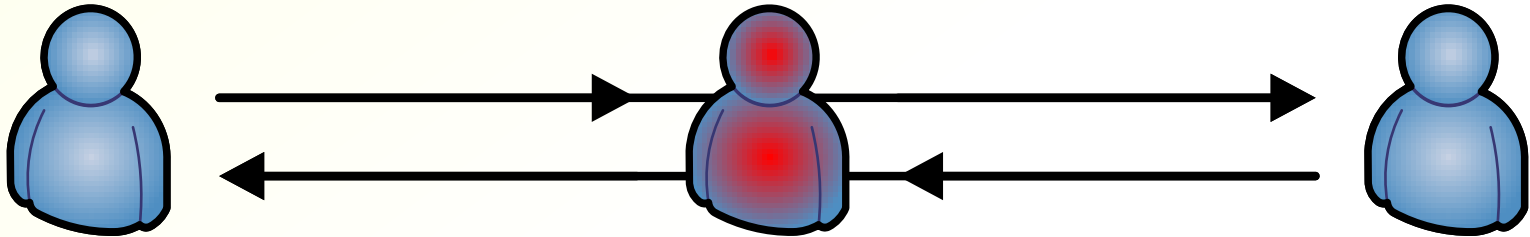


# TCP handshake



# Man in the middle (MitM)

- In the **man-in-the-middle attack**, the attacker is between the honest endpoints



- Attacker can intercept and modify data  
→ combines sniffing and spoofing
- On the Internet, a MitM attacker could
  - be at the local network of one of the end points
  - be at a link or router on the route between them, or
  - change the routing to redirect the packets via its own location
- Note: Just forwarding data between two endpoints (like a piece of wire) is not an attack. What does the attacker gain?

# Other network threats

- What other threats and security requirements are there on open networks?
- Other threats:
  - Integrity of signalling and communications metadata
  - Denial of service (DoS) (vs. availability)
  - Traffic analysis, location tracking
  - Lack of privacy
  - Software security flaws
  - Unauthorized resource use (vs. access control)
  - Liability for malicious use
- Not captured well by the traditional network-security model

# Role of cryptography

# Cryptographic primitives

- **Symmetric (shared-key) encryption** for data confidentiality
  - Block and stream ciphers, e.g. AES-CBC, RC4
- **Cryptographic hash function**
  - E.g. SHA-1, SHA256
- **Message authentication code (MAC)** for data authentication and integrity
  - E.g. HMAC-SHA-1
- **Public-key (or asymmetric) encryption**
  - E.g. RSA, ElGamal
- **Public-key signatures**
  - E.g. RSA, DSA
- **Diffie-Hellman key exchange**
- **Random number generation**

# Crypto Wars – some history

- Until '70s, encryption was military technology
  - In '70s and '80s, limited commercial applications
  - **American export restrictions** and active discouragement prevented wide commercial and private use
- Reasons to ban strong encryption:
  - Intelligence agencies (e.g. NSA) cannot spy on encrypted international communications
  - Criminals, terrorists and immoral people use encryption
- In '90s: PGP, SSL, SSH and other commercial and open-source cryptography became widely available
  - Activists argued that cryptography was a tool for freedom
  - Researchers argued that weak crypto is like no crypto
  - Encryption became necessary for business on the Internet
- Most export restrictions were lifted in 2000

# Network security mechanisms

- Cryptography is the main building block for security protocols, but not the only security mechanism
- **Strong cryptography:**
  - Encryption → confidentiality
  - Cryptographic authentication → authentication and integrity
- **Non-cryptographic** security mechanisms:
  - Perimeter defense (firewalls)
  - Routing-based semi-secure solutions
  - Over-provisioning capacity
  - Preventing attacks at source e.g. by cleaning bots
  - Proxies and pseudonyms to prevent tracking
  - Intrusion detection
  - What else?
- Non-technical solutions: **security is also a social, legal and business problem** (but that is not the topic of this course)

# Security vs. cryptography

- However:

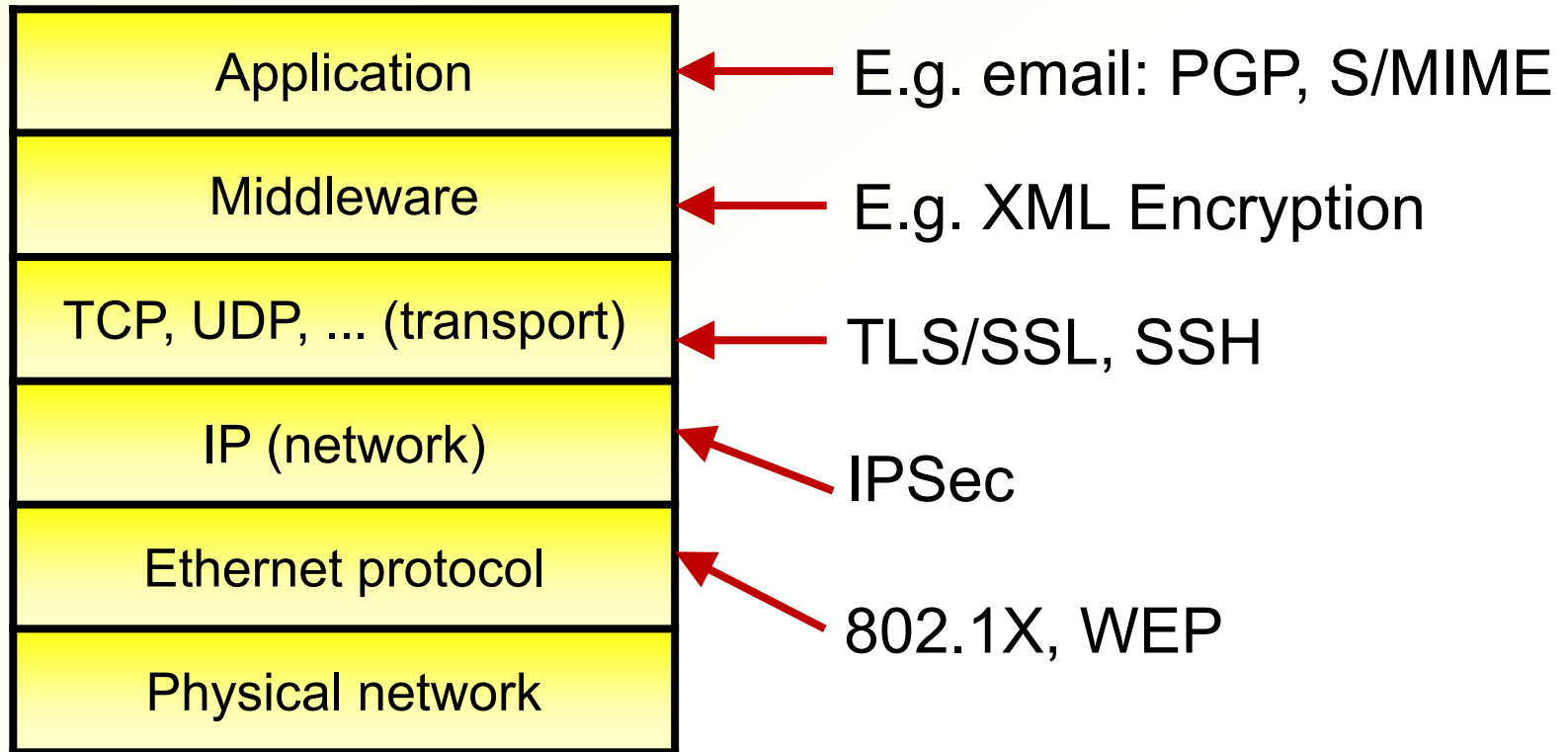
*“Whoever thinks his problem can be solved using cryptography, doesn’t understand the problem and doesn’t understand cryptography.” —*

attributed to Roger Needham and Butler Lampson



# Security and the network protocol stack

# Protocol Stack and Security



- Security solutions exist for every protocol layer
- Layers have different security and performance trade-offs, trust relations and endpoint identifiers

# Which layer security?

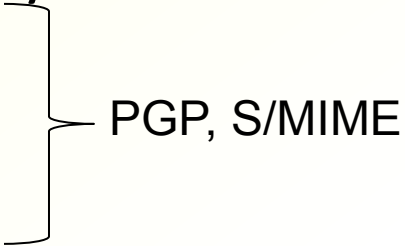
- Security mechanisms exist for all protocol layers
  - Which layer is right for encryption and authentication?
  - Which layer PDUs should a firewall filter or an IDS monitor?
- Reasons to implement cryptographic security in lower layers:
  - Security provided by physical, link or network layer is a service to all higher layers
  - Lower-layer security protects *all* higher-layer data
  - Security in lower layers is transparent to higher layers. No changes to applications needed
  - Lower-layer security protects the lower layer, too
- Reason to implement security in higher layers:
  - Security implemented in the application or middleware will fit exactly to the application requirements
  - Lower-layer identifiers may not be meaningful to higher layers
- Actually, we may need independent security in multiple network-stack layers

# End-to-end security

- Security should be implemented between the endpoints of communication. All intermediaries are part of the untrusted network
- End-to-end security only depends on the end nodes
  - Hop-by-hop (link-layer) security assumes all routers are trusted and secure
- End-to-end security protocols are independent of the network technology at intermediate links
  - Link-layer security is different for each link type
- Confidentiality and authentication are usually user or application requirements
  - Network or link layer only cannot know application-level requirements
- But link and network layer infrastructure and signalling need protection, too

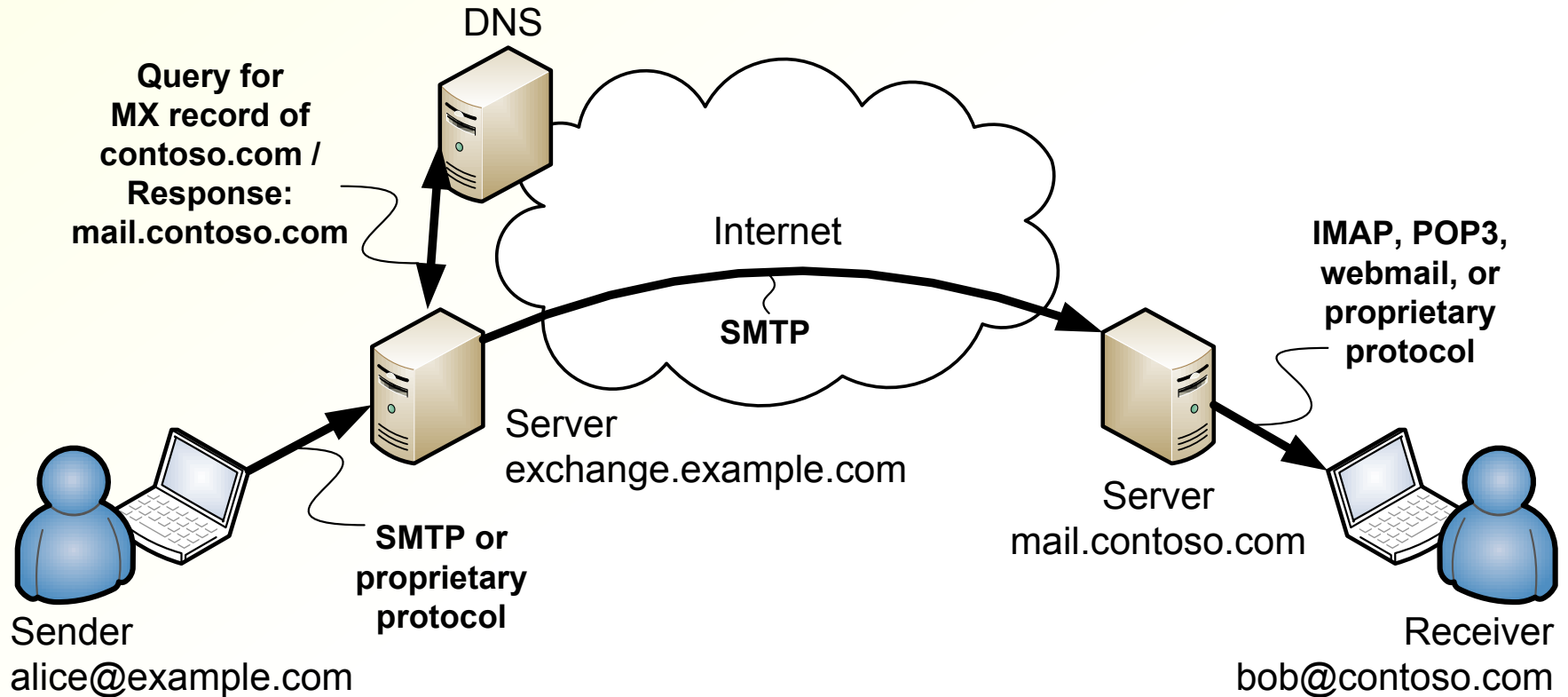
# Email security

# Security requirements

- What kind of security is needed for email?
    - Confidentiality?
    - Authentication?
    - Non-repudiation?

PGP, S/MIME
  - Time stamping?
  - Mandatory access control, DRM?
  - Spam control?
  - Phishing prevention?
  - Anonymity?
- We use email security as the first example because it is a fairly straightforward application of crypto and allows us to introduce many basic concepts
    - Crypto does not solve all email security problems

# Internet email architecture



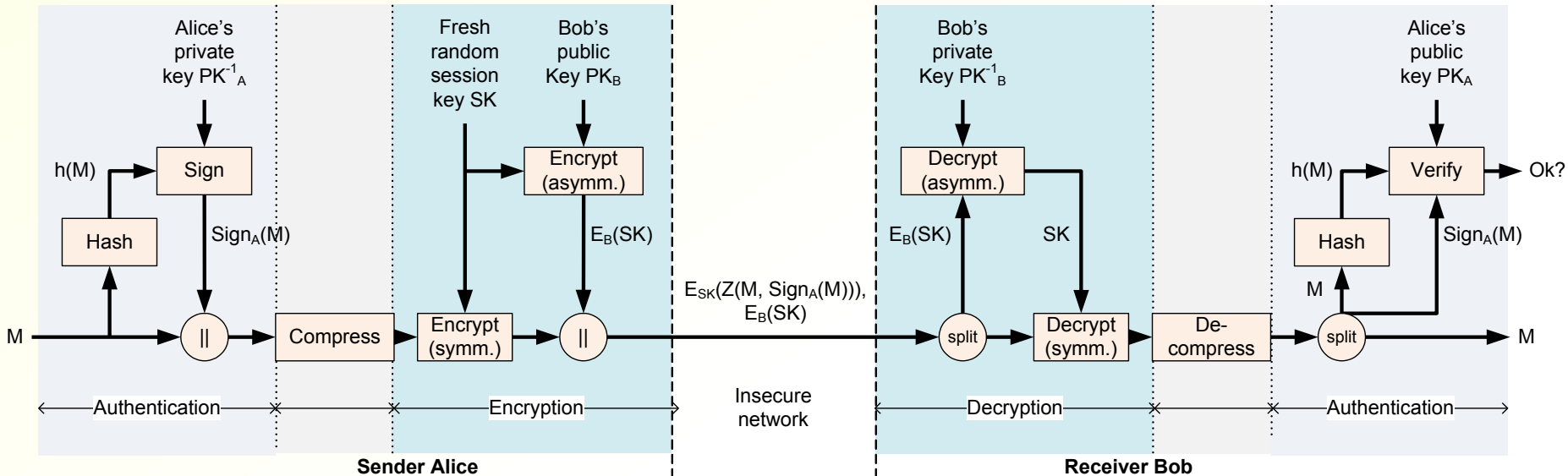
- Alice sends mail to bob@contoso.com
- Where are the security vulnerabilities?

# Pretty Good Privacy (PGP)

Extra reading material



# Sign, compress, encrypt



- Sender and receiver need to know each other's public keys
- Options to encrypt only or to sign only:
  - Possible to sign without knowing receiver's public key, or when sending to a mailing list
  - Possible to encrypt without identifying sender

# Pretty Good Privacy (PGP)

- Zimmermann 1991–
- The sign-compress-encrypt process shown earlier, instantiated with the best available algorithms of the time:
  - IDEA encryption (128-bit keys) in CBC mode (later 3DES or AES in CFB)
  - SHA-1 hash function and RSA public-key signatures
  - RSA and ElGamal public-key encryption
  - Timestamp
  - Radix-64 conversion and headers (called “ASCII armor”)
- The first strong encryption product available to the public
- OpenPGP [RFC 4880]

# Example: PGP-encrypted message

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

```
Version: GnuPG v1.4.8
```

```
Comment: Encrypted secret message.
```

```
hQEOA1e+1x6YuUMCEAQAoST1l/obnXOB6fhIhmLnGVLhuxmsksKD+Efyk7ja9gOx  
U5X98/25ZVDQz0EiOkRjW2LChuZt9Kesh1DSIRwB/llXCm3pbNX/V+ajkL4Fzxlw  
jWCCedv527SUNTUP70lhLbh4O2kHHxMdEn41zVo9TPUgtQ1BIO32k/xP2RYtPCEE  
AJDhcyp+COLaI4idibfSrDDtYcT+hVVFVveIteTicznouoS1yVyipe4mBwa380c6  
TiwImq63hOhs62c9BOQv7G9cnaqEzNg0nLiVZD+K/JeN00zILm+TzdWZxrW019nA  
+tsMwznUZ2V/kQZjS9xkPWjn7ZzPTyW6gLhjWQNlr93S0lcBT0CJy285ixFz9UrJ  
qjK2azsBdXRcVuXFdh84LW1E/8/8DwdLgSK9X/jPNv3/WGLA4Ez2xTFIUorVi5Xe  
M9dpriEQ0Jg2msnz2bjqRGZliXXo6m8ye/A=
```

```
=YWDi
```

```
-----END PGP MESSAGE-----
```

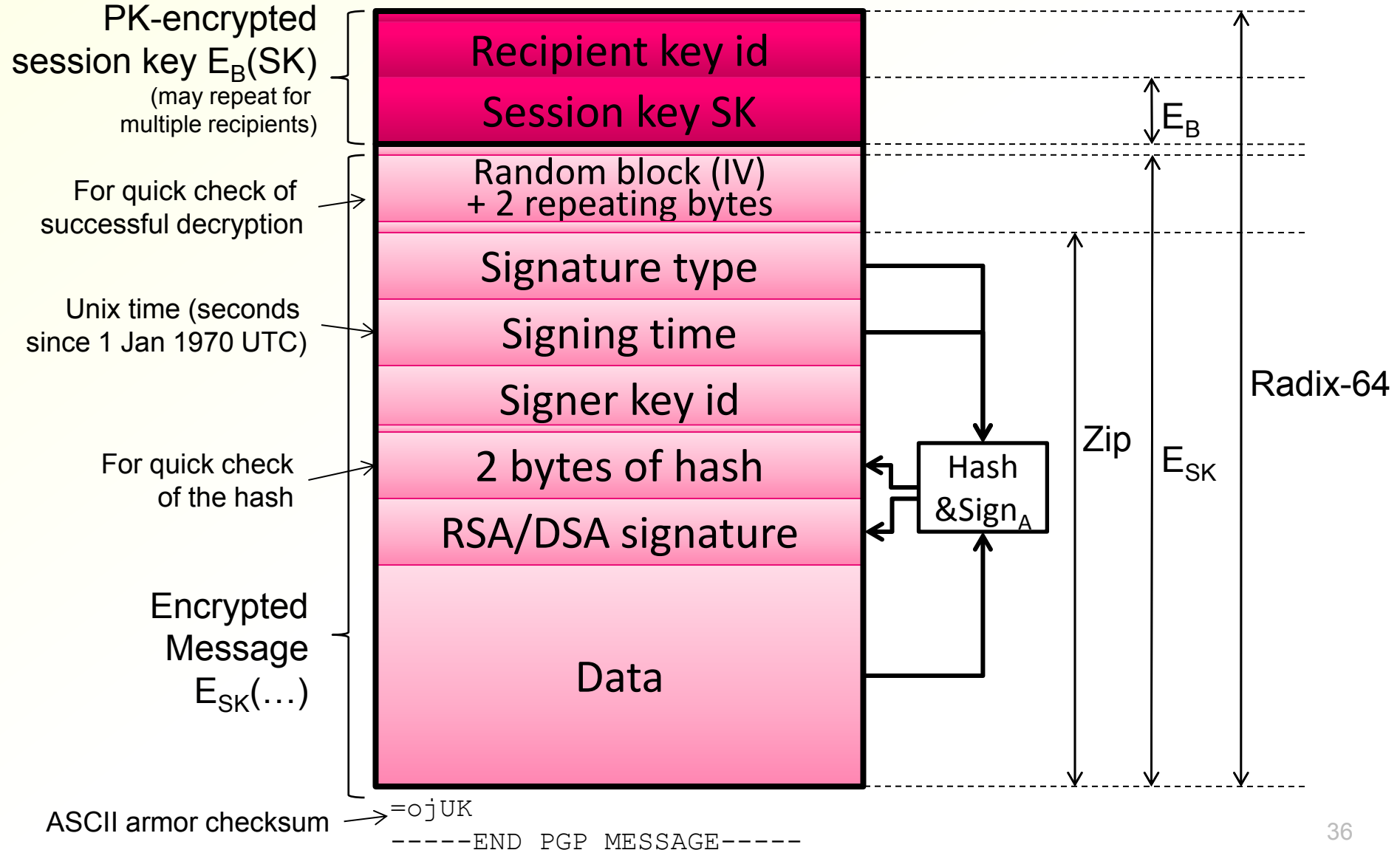
- “Meet me in the park at 6 PM.”

# Typical PGP message

ASCII armor headers  
for sending in text email

-----BEGIN PGP MESSAGE-----

Version: GnuPG v1.4.8



# Public-key distribution

- PGP public keys are usually distributed manually
    - Download from a web page or take from a received email
      - key distribution often insecure
  - Users can endorse keys of others by signing them
    - Sign: key, name, level of trust, signing, expiry time
    - Mark friends and well-known people as trusted, derive trust to others from endorsements
- PGP **web of trust**

# Email integrity problem

- Email servers modify messages:
  - Each server adds headers
  - Old email systems were not 8-bit safe
  - Servers perform character-set conversions
  - Firewalls remove or replace suspicious attachments
  - Proxies compress text and images for mobile clients→ bits change → authentication fails
- Solution: encode the signed part of the message in “safe” characters that are not modified in transit
  - **Base64** and **Radix64** encodings:
    - ~64 safe ASCII characters give 6 bits per character
- Remaining problems:
  - Signed message not human-readable text
  - 33% expansion in message size
- General lessons:
  - **Cryptographic protection makes message delivery less reliable**
  - **Authentication may make messages unreadable to human users**

# Radix-64 encoding

- Use safe ASCII characters to represent values 0..63:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/  
xyz

- Encode each 3 bytes as 4 characters:

```
+--first  octet--+-second  octet--+-third  octet--+
|7 6 5 4 3 2 1 0|7 6 5 4 3 2 1 0|7 6 5 4 3 2 1 0|
+-----+-----+-----+-----+-----+
|5 4 3 2 1 0|5 4 3 2 1 0|5 4 3 2 1 0|5 4 3 2 1 0|
+--1.char---+--2.char---+--3.cahr---+--4.char---
```

- If the data length is not divisible by 3, pad with one or two = characters to indicate actual length

# S/MIME

- PGP is mainly used by private persons and academia
- S/MIME is a similar standard used primarily by enterprises, e.g. Outlook
- Message structure based on the MIME standards
  - Envelopes and signatures are new **MIME types**
  - **Base64** encoding



# Non-repudiation

- Proof of authenticity to third parties
  - Email sender cannot later deny sending the message (i.e. cannot repudiate the message)
  - Third party, such as a judge, is needed to make the decision
  - The public key must be somehow registered to bind it to the person signing
- Uses:
  - Accountability for sent messages
  - Contract signing
- Questions:
  - Does the sender of an email want to go do extra work in order to be accountable for the emails he sends?
    - Little motivation to sign messages
  - Are business contracts signed using secure email?

# How good is email security?

- Is it secure?
  - PGP public keys are rarely distributed through secure channels
  - Absolute security not necessary for privacy or to prevent large-scale monitoring by governments
- Is email security needed?
  - Many people sign email when sending but few verify the signatures or take action if a signature is invalid
  - Email sniffing may take place but most users have not experienced negative consequences
  - Mandatory email authentication would help to prevent spam, phishing and malware, but what if authentication is optional?
  - Email users rarely want non-repudiation

# Related reading

- William Stallings, Network security essentials: applications and standards, 3rd ed.: chapter 1
- William Stallings, Cryptography and Network Security, 4th ed.: chapter 1
- Dieter Gollmann, Computer Security, 2nd ed. chapter 13; 3rd ed. chapters 16.1, 17.1
- Ross Anderson, Security Engineering, 2nd ed.: chapter 6

# Exercises

- What security goals do TCP sequence numbers have, and what are their weaknesses?
- Design a more spoofing-resistant acknowledgement scheme to replace TCP sequence numbers. Hint: use random numbers (and maybe hashes) to ensure that acknowledgements can only be sent by someone who has really seen the packets
- Which applications of hash functions in network protocols require strong collision resistance? Which do not?
- Why is link-layer security needed e.g. in WLAN or cellular networks, or is it?
- To what extent are the identifiers in each protocol layer of the TCP/IP stack unique? Does one layer in the protocol stack know the identifiers of other layers?
- How do the properties of these identifiers differ from each other: URL, IP address, DNS name, email address, person's name, company name, national identity number (HETU)?
- Spoof some emails to yourself
- How to prevent SMTP spoofing without end-to-end cryptography? What can be filtered at SMTP servers and what cannot?
- New technology often fails if the early adopters do not receive immediate benefits. Could this explain the limited use of email security?
- Does signing of emails help spam control?
- Install an OpenPGP implementation (e.g. GPG). How do you check that the binary or source code has not been tampered with? Would you use PGP itself to verify the signature or fingerprint on the installation package?