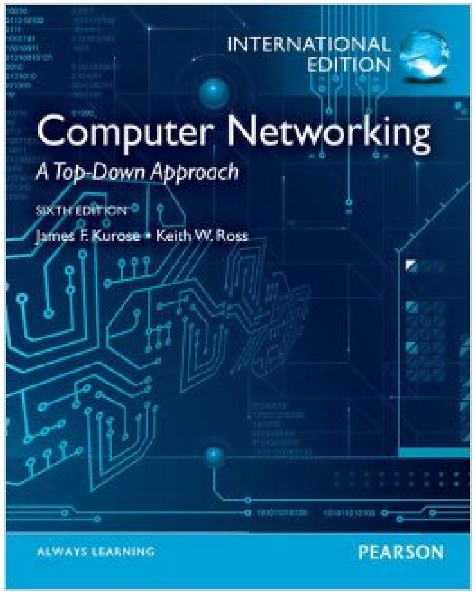
Chapter 8
Network Security



Slides adapted from the book and Tomas Olovsson

Roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity

Security protocols and measures:

- Securing TCP connections: SSL
- Network layer security: IPsec
- Firewalls

What is security? CIA!

Confidentiality: only sender, intended receiver should "understand" message contents

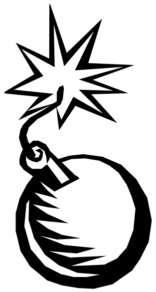
- sender encrypts message
- receiver decrypts message

Integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

Availability: services must be accessible and available to users

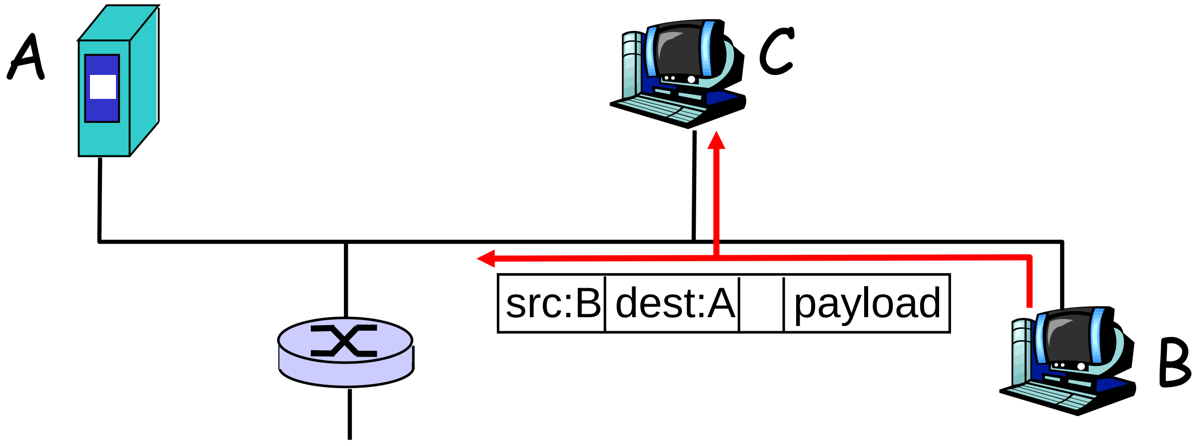
> The book also includes **Authentication**: it is normally seen as a mechanism to implement the services above

Internet security threats



Packet sniffing:

- broadcast media
- promiscuous NIC reads all packets passing by
- can read all unencrypted data (e.g. passwords)
- e.g.: C sniffs B's packets

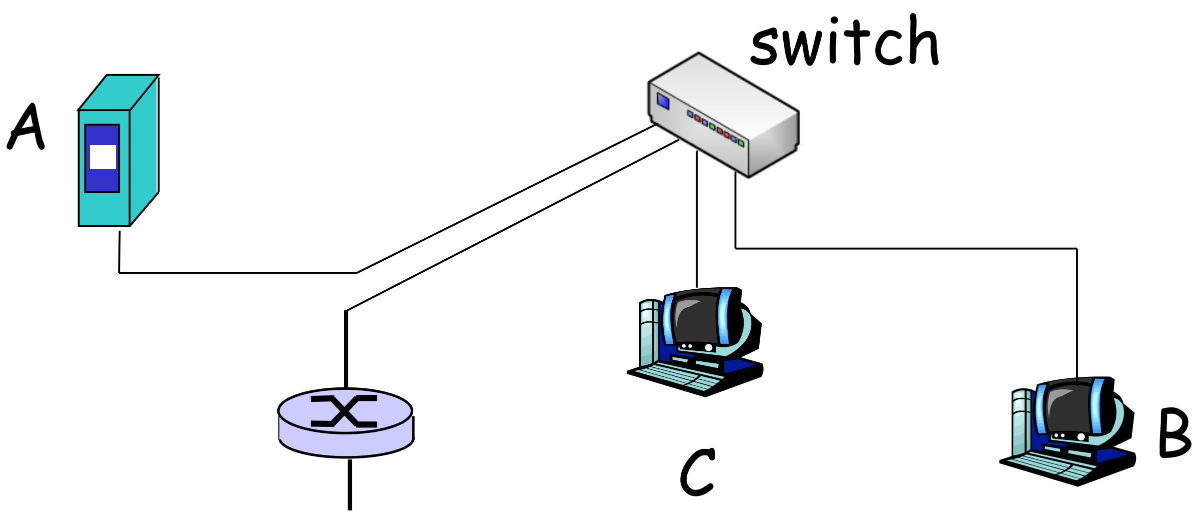


Countermeasures?

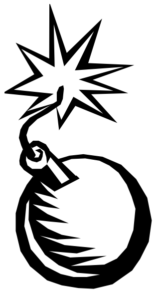
Internet security threats

Packet sniffing: countermeasures

- One host per segment of broadcast media
  - Use switches (not hubs)
- Segment network
  - Use routers
- Encryption

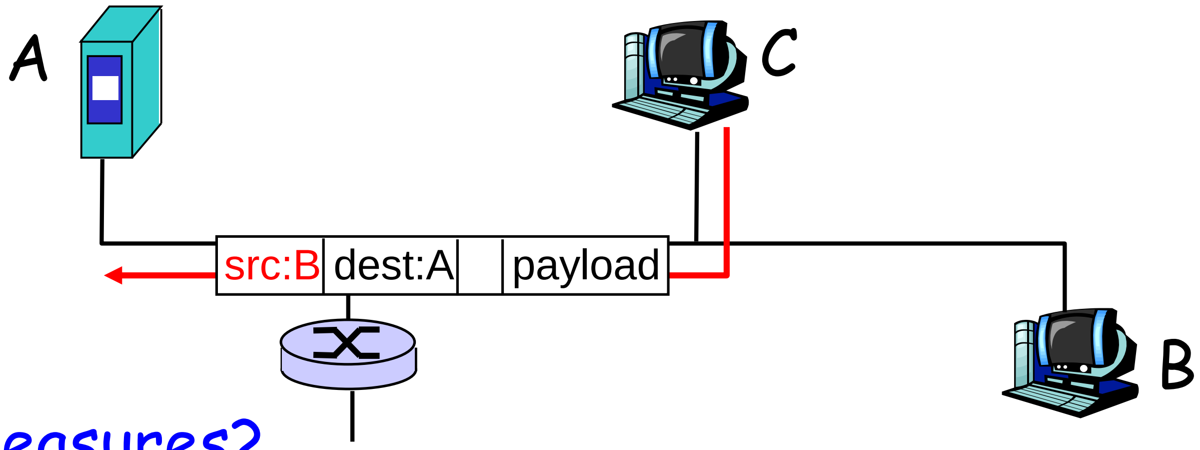


Internet security threats



IP Spoofing:

- can generate "raw" IP packets directly from application, putting any value into IP source address field
- receiver can't tell if source is spoofed, e.g.: C pretends to be B

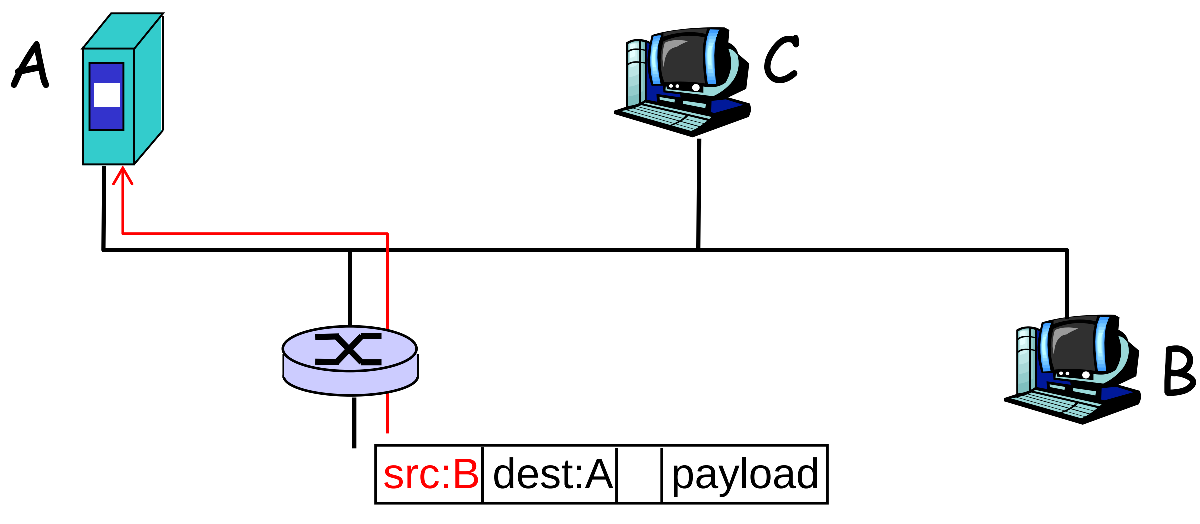


Countermeasures?

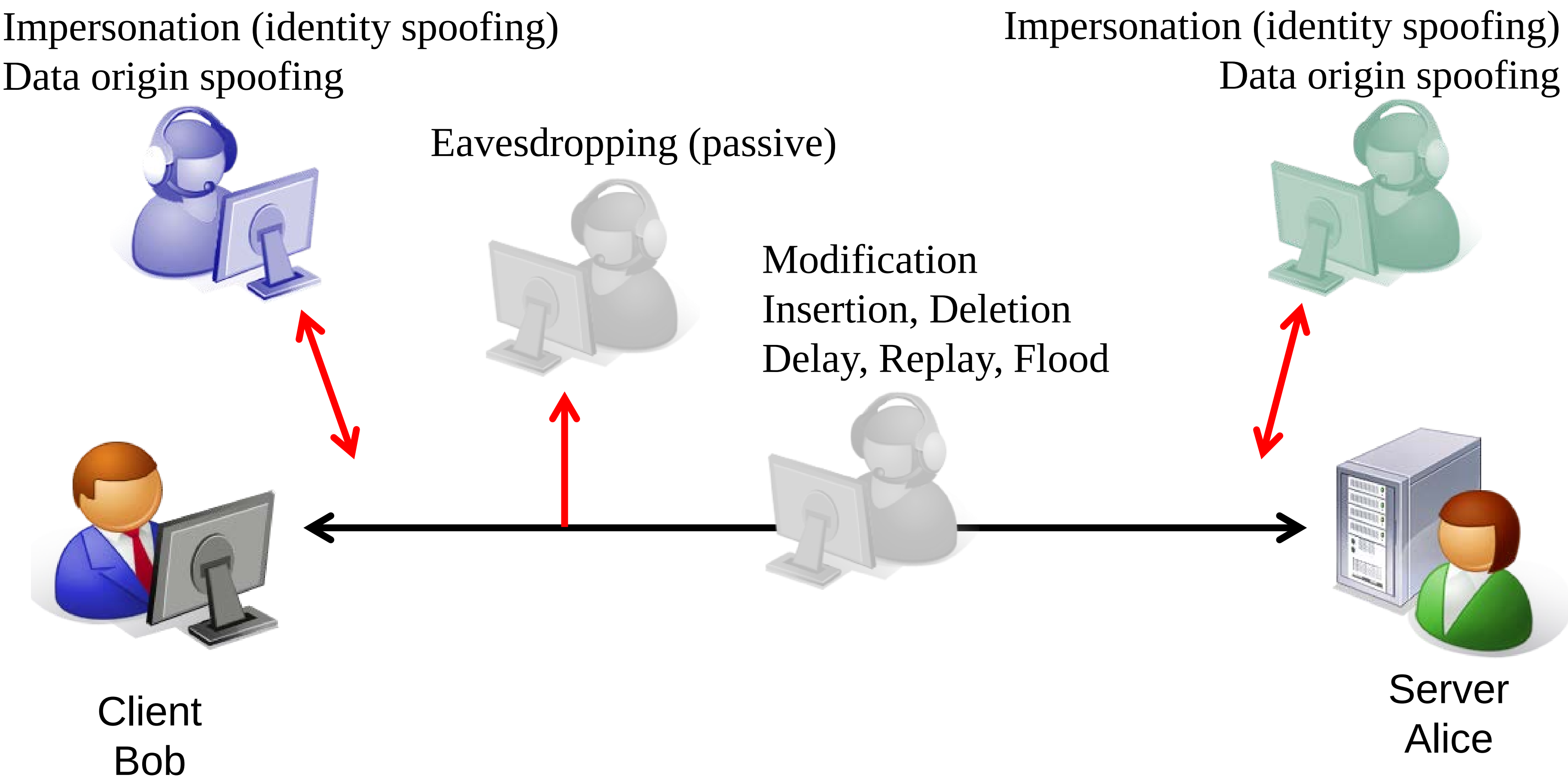
Internet security threats

IP Spoofing: ingress filtering

- routers should not forward incoming and outgoing packets with invalid addresses
  - Outgoing datagram source address not in router's network (egress filtering)
  - Incoming datagram has internal address as source address (ingress filtering)



Communication threats - Summary



Roadmap

8.1 What is network security?

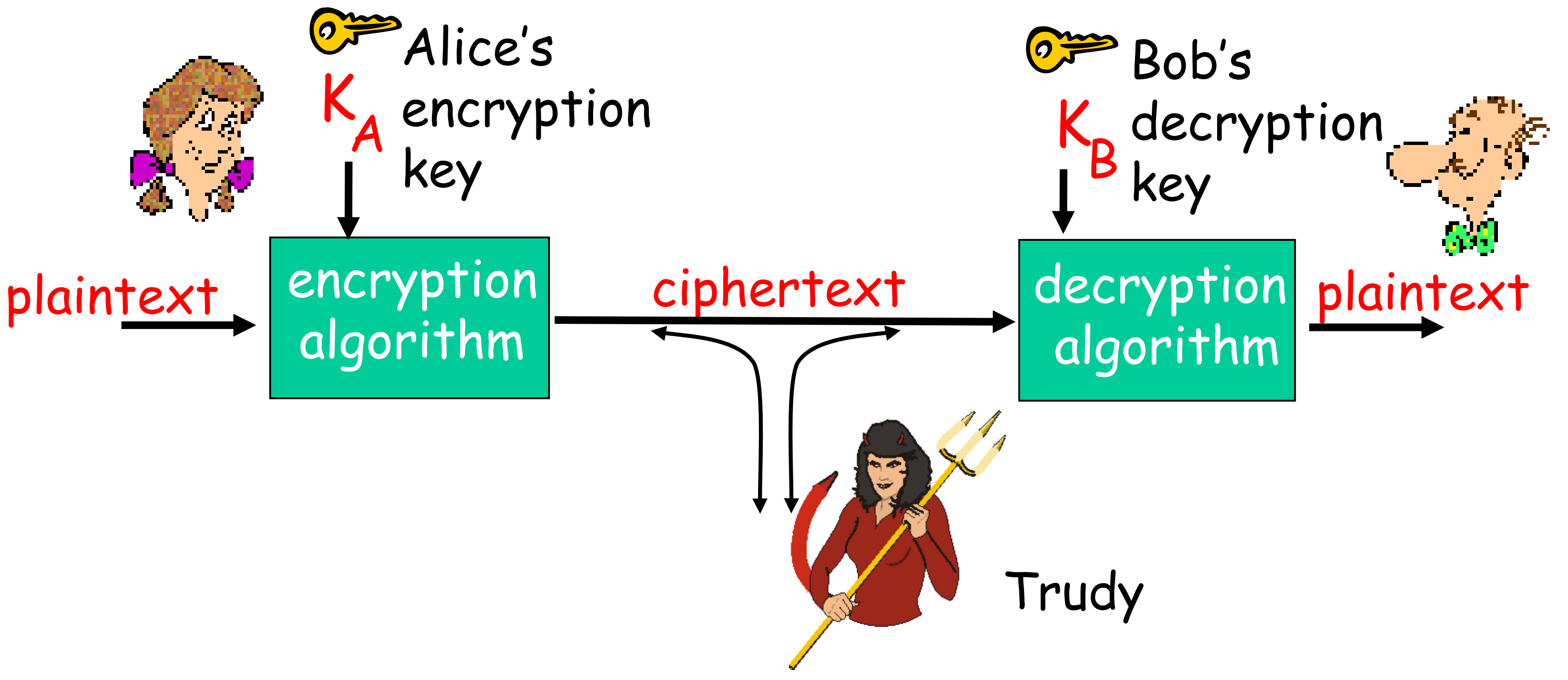
8.2 Principles of cryptography

8.3 Message integrity

Security protocols and measures:

- Securing TCP connections: SSL
- Network layer security: IPsec
- Firewalls

The language of cryptography



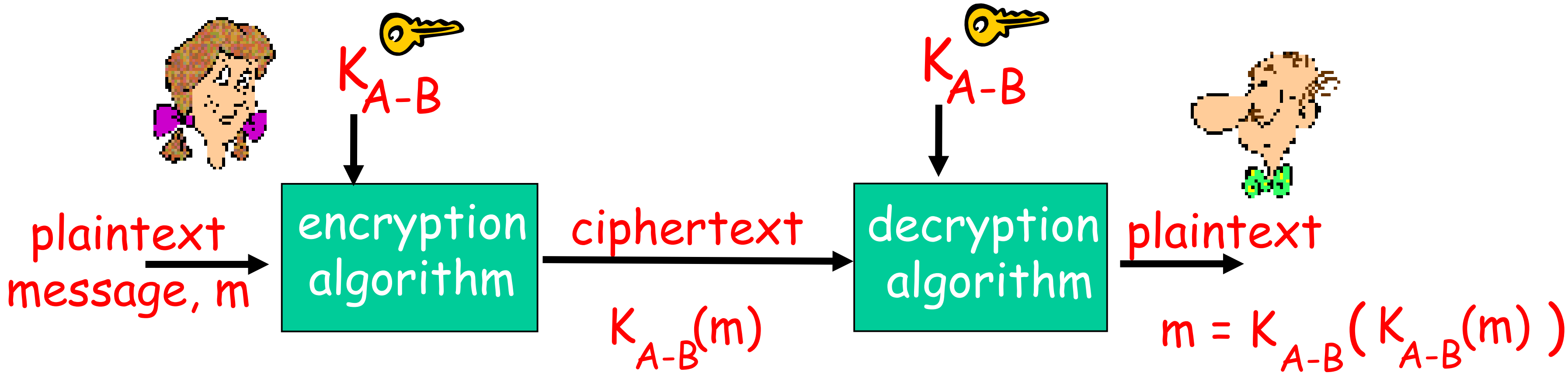
Symmetric key crypto: sender & receiver keys *identical*

Asymmetric key crypto (or Public-key crypto):

One key for encryption, another for decryption.

One of the keys can be *public*, the other *private*.

Symmetric key cryptography



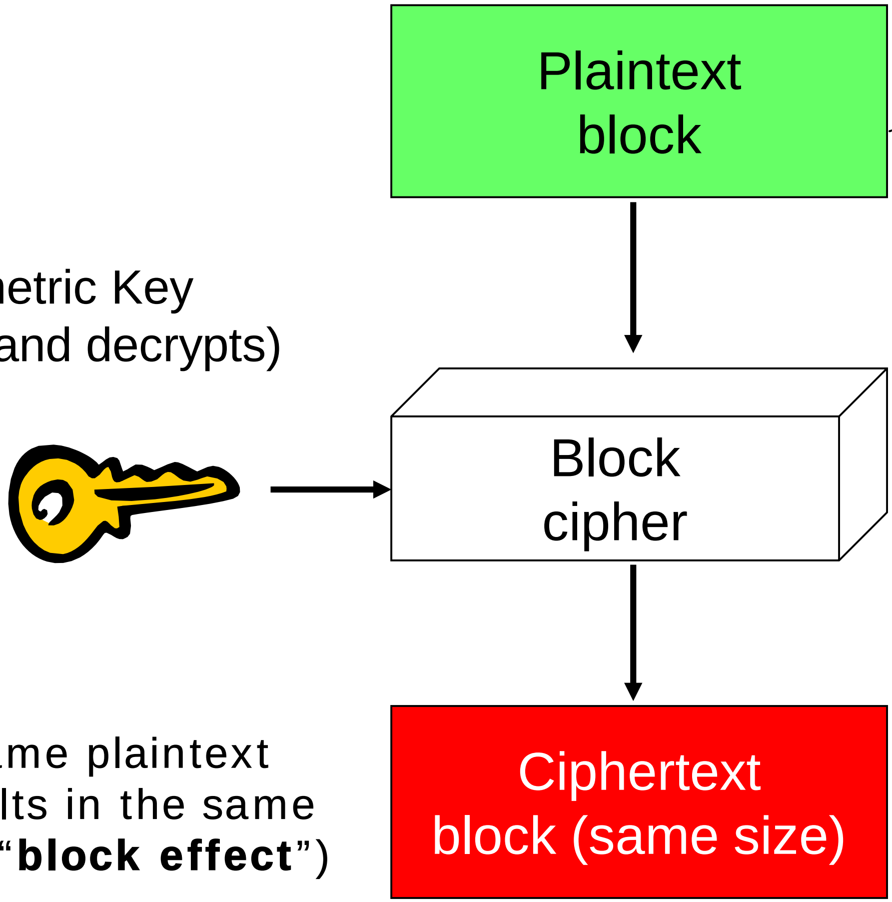
symmetric key crypto: Bob and Alice share the same (symmetric) key: K_{A-B}

Q: how do Bob and Alice agree on key value?

Block Encryption (ECB mode)

Block size depends on cipher:
DES=64 bits, AES=128 bits, …
Independent of key length.

Symmetric Key
(encrypts and decrypts)

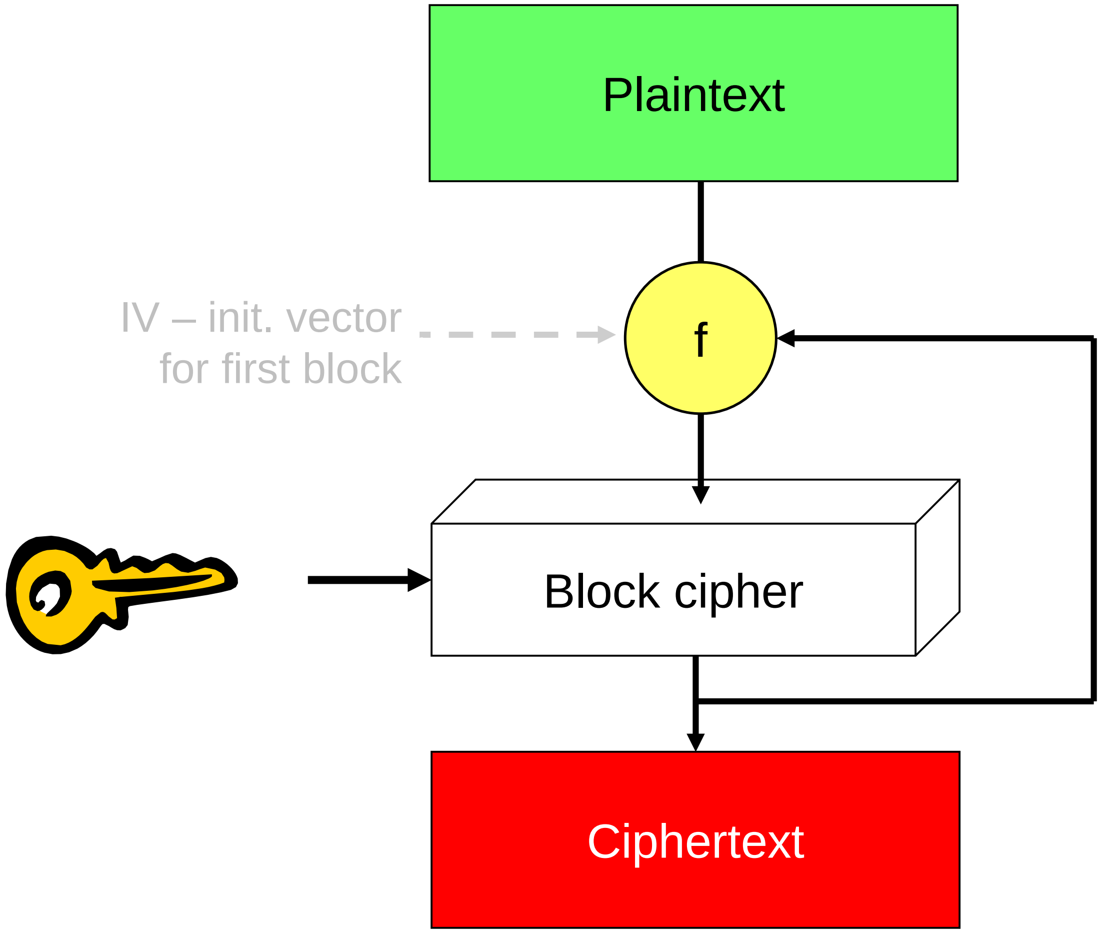


The algorithm is publicly known!

Problem: same plaintext always results in the same ciphertext ("**block effect**")

This mode is called electronic codebook mode (**ECB**)

CBC - Cipher block chaining mode

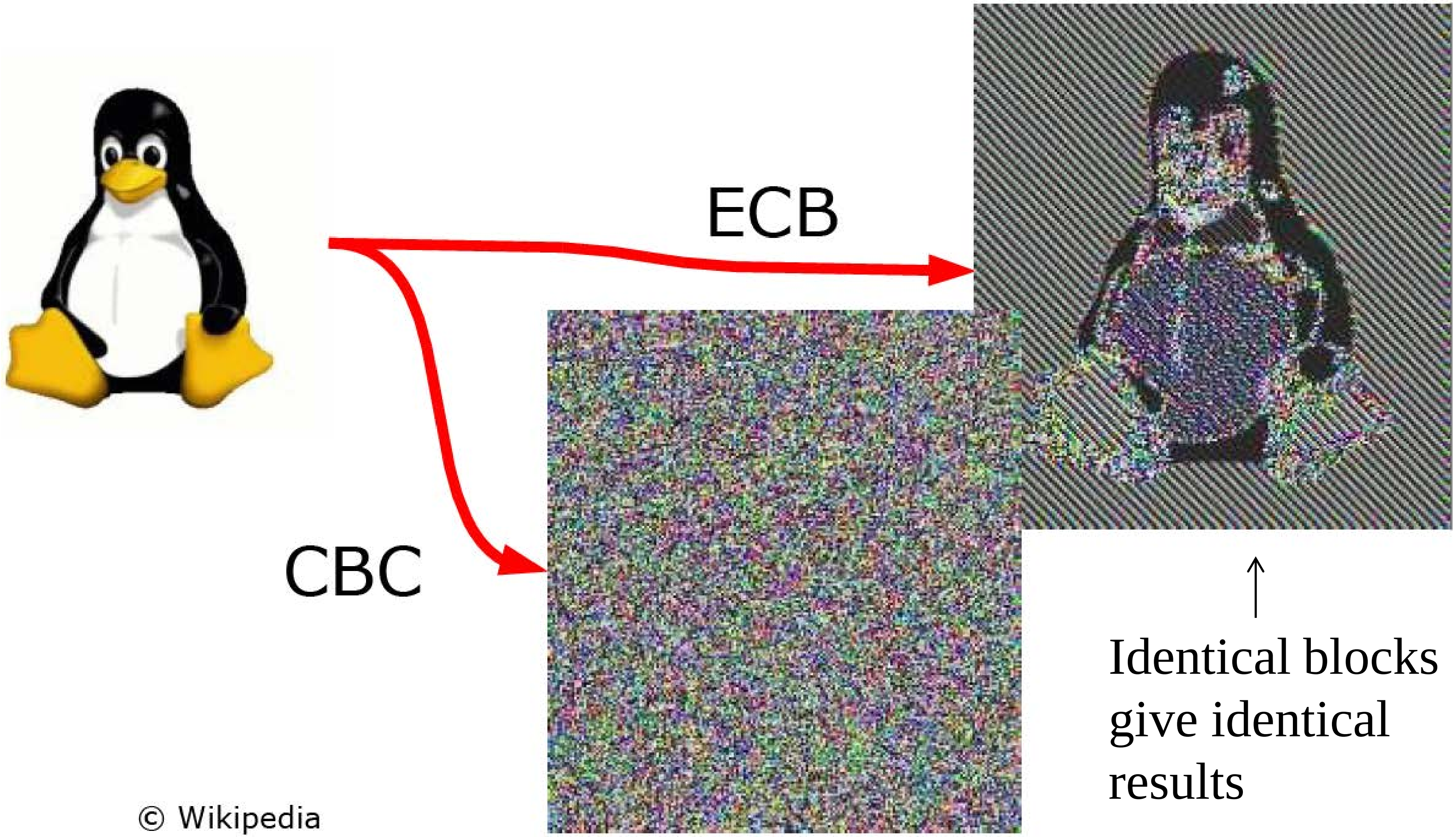


Identical blocks now encrypted differently.

May not always be practical, for example for hard disk encryption.

Note that there is no protection against replays and alteration!

ECB vs. CBC



Identical blocks give identical results

Symmetric Key Ciphers

- **DES** (Data Encryption Standard)
  - Designed by IBM 1975, Adopted by NIST* 1977
  - Criticized for key length (64→ 56) and mysterious "S-boxes"
  - Turned out to have protection against differential cryptanalysis (found 1990)
  - Probably more effort is spent on cracking DES than on all other ciphers together
  - Today key length is a major problem: 56-bit keys can be cracked

EFF DES cracker.
Jan 19, 1999: 22h15m

- **3-DES** (repeating DES three times with different keys)
  - 3-DES probably secure today but too computational intensive
- **AES** (Advanced Encryption Standard)
  - Replaces DES as of 2001
  - Result of an official competition
  - Key lengths: 128, 192 or 256 bits
  - Brute force decryption: if DES takes 1 second, AES-128 takes 149 trillion years, AES-256 would take 10^{52} years
- **RC4, RC5, RC6**
  - RC4 is considered weak but it is fast
- ...

*NIST = National Institute of Standards and Technology, US, formerly NBS

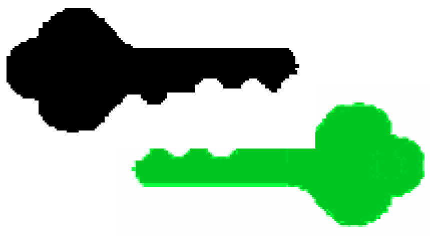
Key Length and Number of Possible Keys

Key Length in Bits	Number of Possible Keys
1	2
2	4
40	1,099,511,627,776
56	72,057,594,037,927,900
112	5,192,296,858,534,830,000,000,000,000,000,000
168	3.74144E+50
256	1.15792E+77
512	1.3408E+154

WEAK

Strong

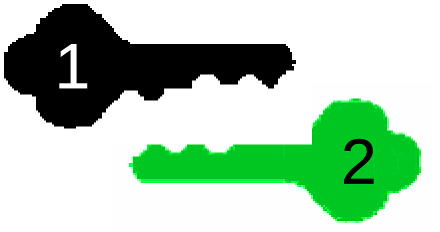
Asymmetric key encryption



- One key is used to encrypt, the other to decrypt
- One key can be public – the other kept secret
- Based on mathematically hard problems
  - Factorization of very large primes (RSA)
- Slow because of the large numbers involved
  - 1024 bits and up (RSA), 384 bits (ECC)
  - $2^{1024} = 10^{308}$ which means >300 digit numbers
- Ciphers:
  - RSA – Rivest, Shamir, Adleman (Patent expired 2000)
  - ECC – Elliptic Curve Cryptosystem
- 768-bit RSA was reported cracked Jan 2010:
  - They generated a five-terabyte decryption table. It would have taken around 1,500 years using a single AMD Opteron-based PC (they used a cluster)
- 1024-bit RSA is too short to protect against extremely large organizations
  - Use 2048-bit RSA keys in sensitive applications

"*the overall effort [as] sufficiently low that even for short-term protection of data of little value, 768-bit RSA moduli can no longer be recommended.*"

Asymmetric key encryption



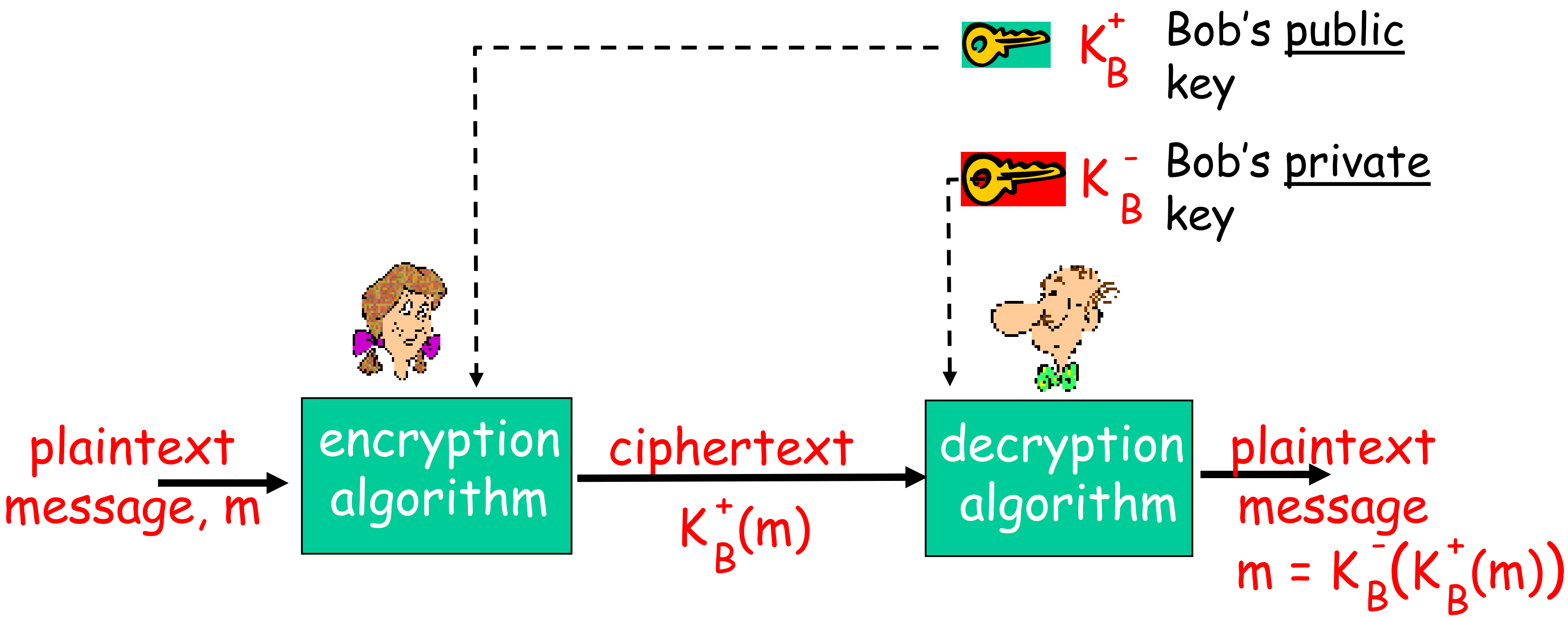
- One key is normally made public ("Public key encryption")

- You decide whether it is the encryption or decryption key that is public:

1. Encryption key public: everyone can send encrypted messages to owner of the private key
2. Decryption key public: only one can encrypt, everyone can verify that the secret key has been used.
   - Can be used to sign documents and data.

Useful?

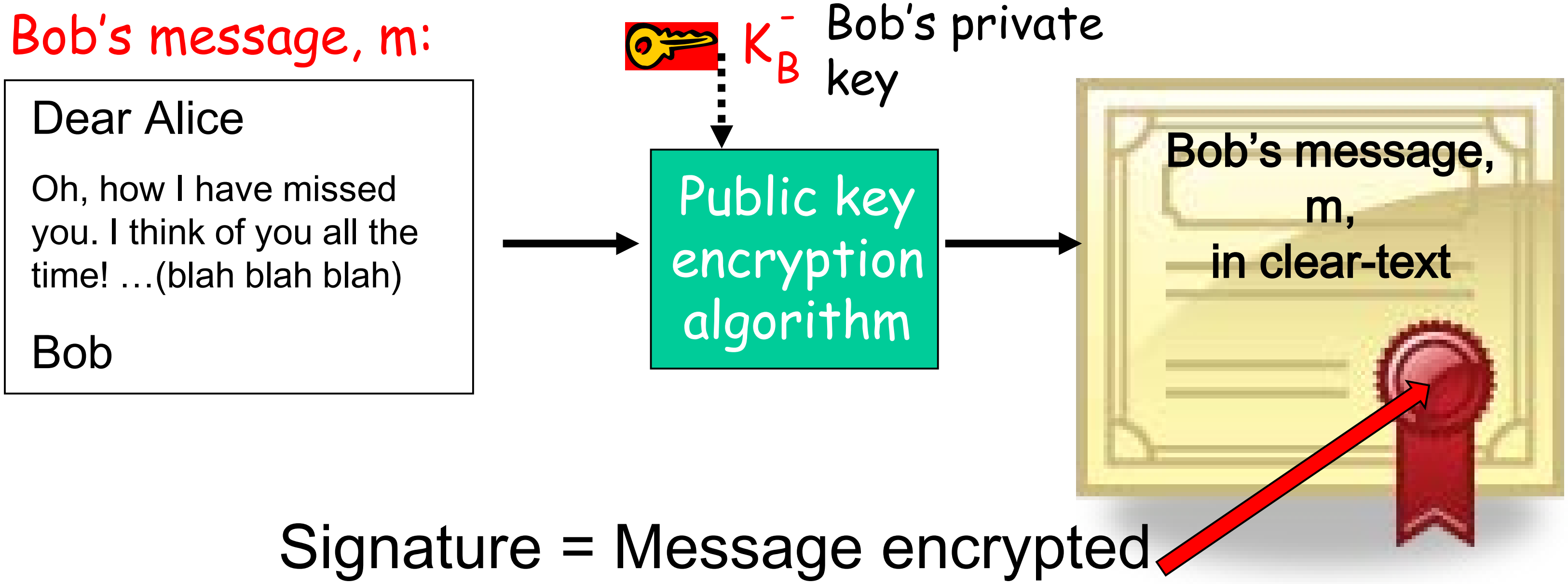
Example 1: Public Key Encryption



Example 2: Digital Signatures

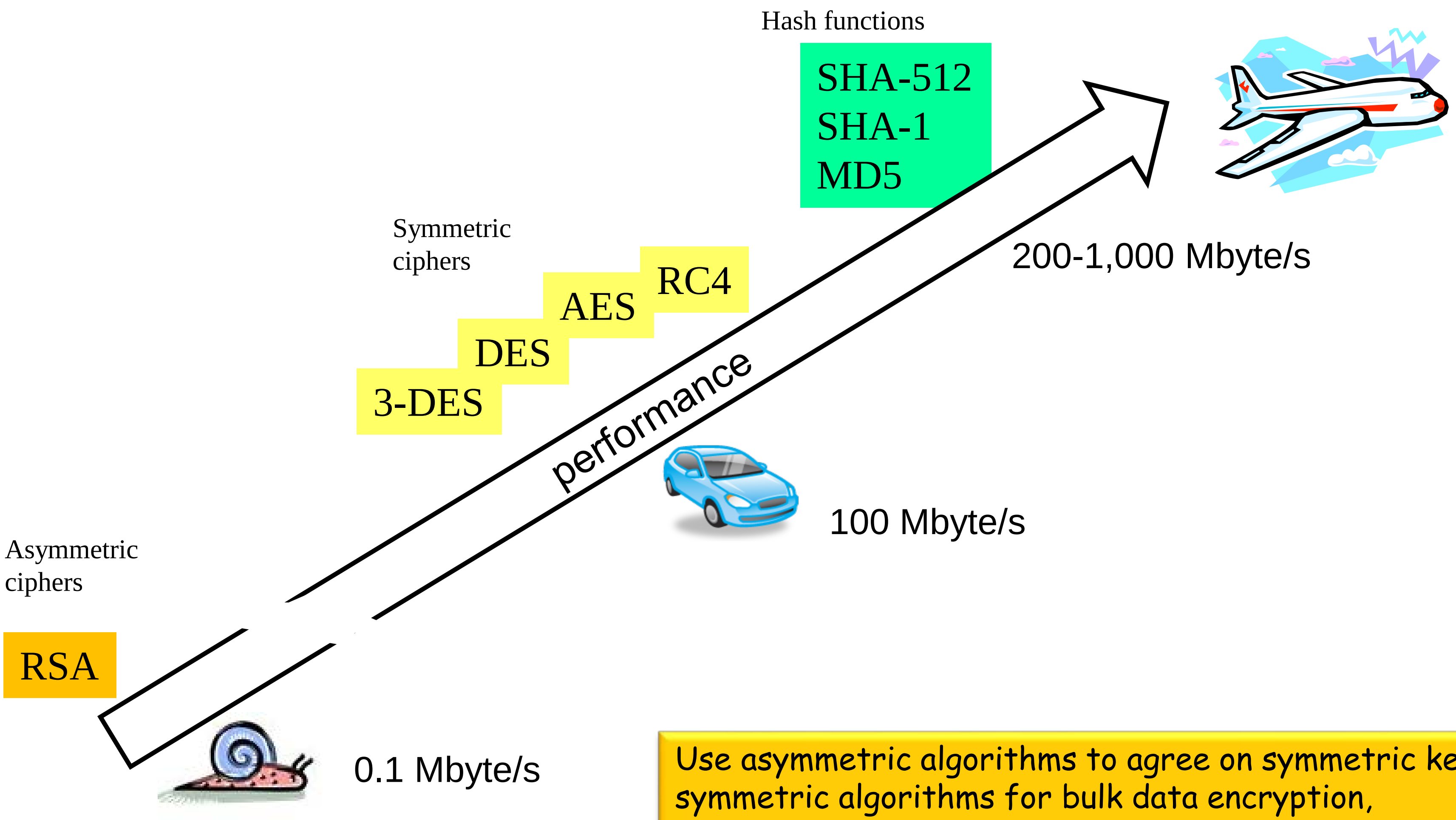
Simple digital signature for message m:

- Bob signs m by encrypting with his private key K_B^-, creating "signed" message, $K_B^-(m)$



Signature = Message encrypted with Bob's private key $K_B^-(m)$

Relative performance



Use asymmetric algorithms to agree on symmetric keys, symmetric algorithms for bulk data encryption, and hash functions for integrity protection if encryption is not needed

Roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity

Security protocols and measures:

- Securing TCP connections: SSL
- Network layer security: IPsec
- Firewalls

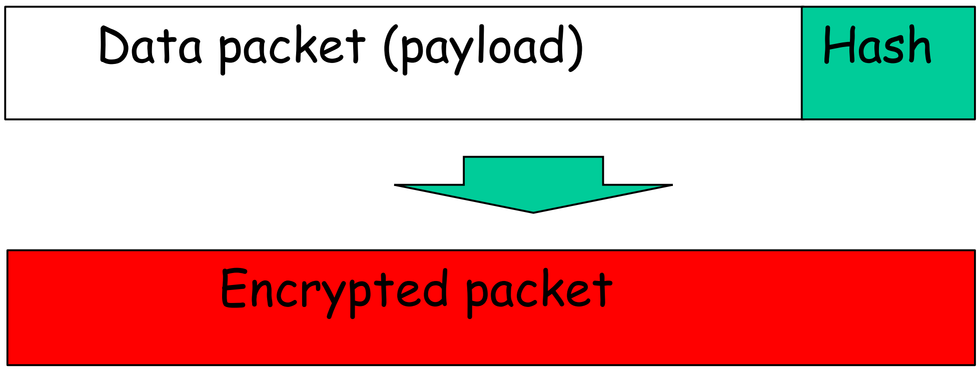
Message Integrity

Bob receives msg from Alice, wants to ensure:

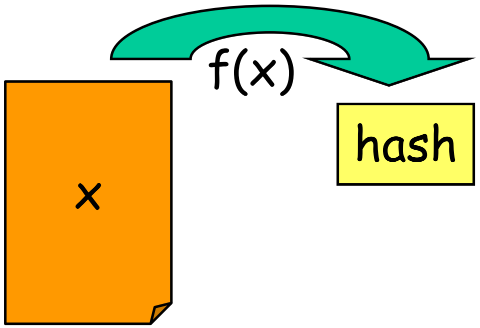
- message originally came from Alice
- message not changed since sent by Alice

Just encryption is not enough!

- Contents can be changed even if it is encrypted
- Solution: add some kind of checksum (hash) to the message before it is encrypted:



(Cryptographic) hash functions

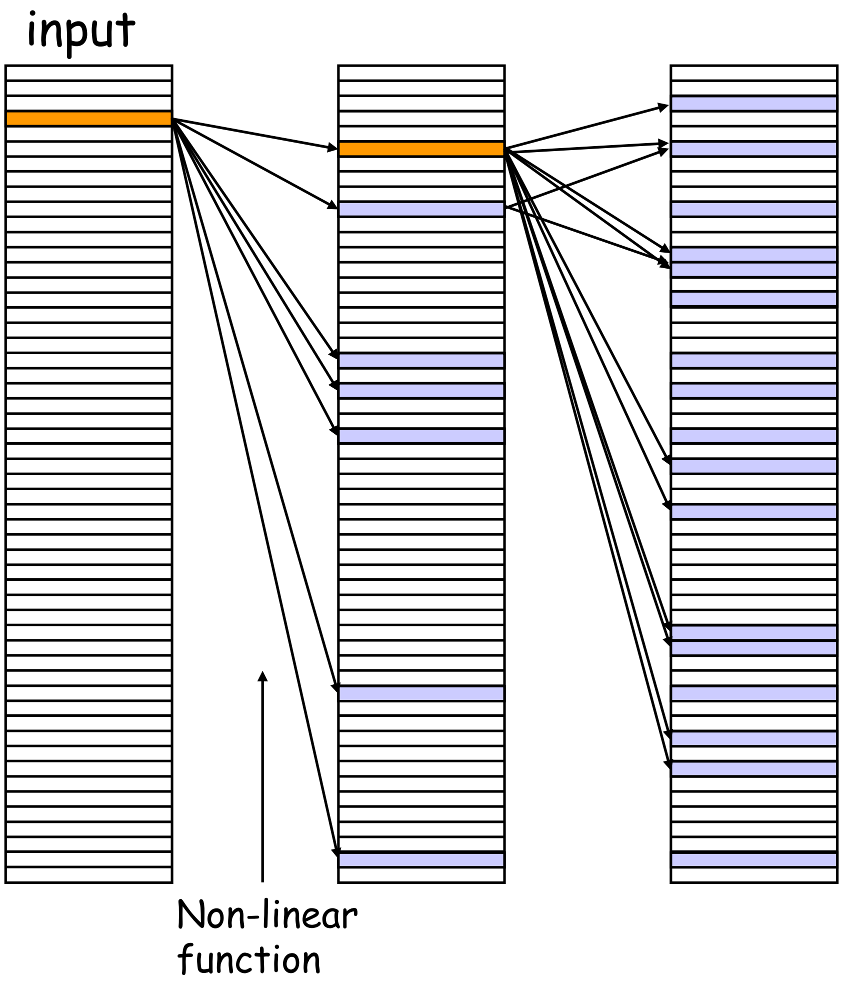


- Input: arbitrary length bit-string
  Output: fixed length bit-string
  - Not a one-to-one mapping,
    output space typically 128 bits
- Requirements:
  - Computationally efficient: Typically >10 times faster than symmetric ciphers
  - Must be repeatable (same input → same output)
  - Impossible to reverse the computation (preimage resistant)
  - Infeasible to find an input X with a given hash
  - Infeasible to find two inputs resulting in the same hash (pseudo-randomness)
- Today's hash functions are not based on mathematical foundations – may lead to problems

"SSL broken! Hackers create rogue CA certificate using MD5 collisions" [www.zdnet.com]

Hash functions

Even a single bit change should give a completely different result → avalanche effect



SHA-512 has 80 rounds

Hash functions

- Even just one changed bit gives a completely different result:
  - `md5("hello") = 5d41402abc4b2a76b9719d911017c592`
  - `md5("Hello") = 8b1a9953c4611296a827abf8c47804d7`

- **MD5** – Message Digest 5 (RFC 1321, 1992)
  - 128-bit message digest → 10^{38} different hashes
  - Avoid in new implementations - weak

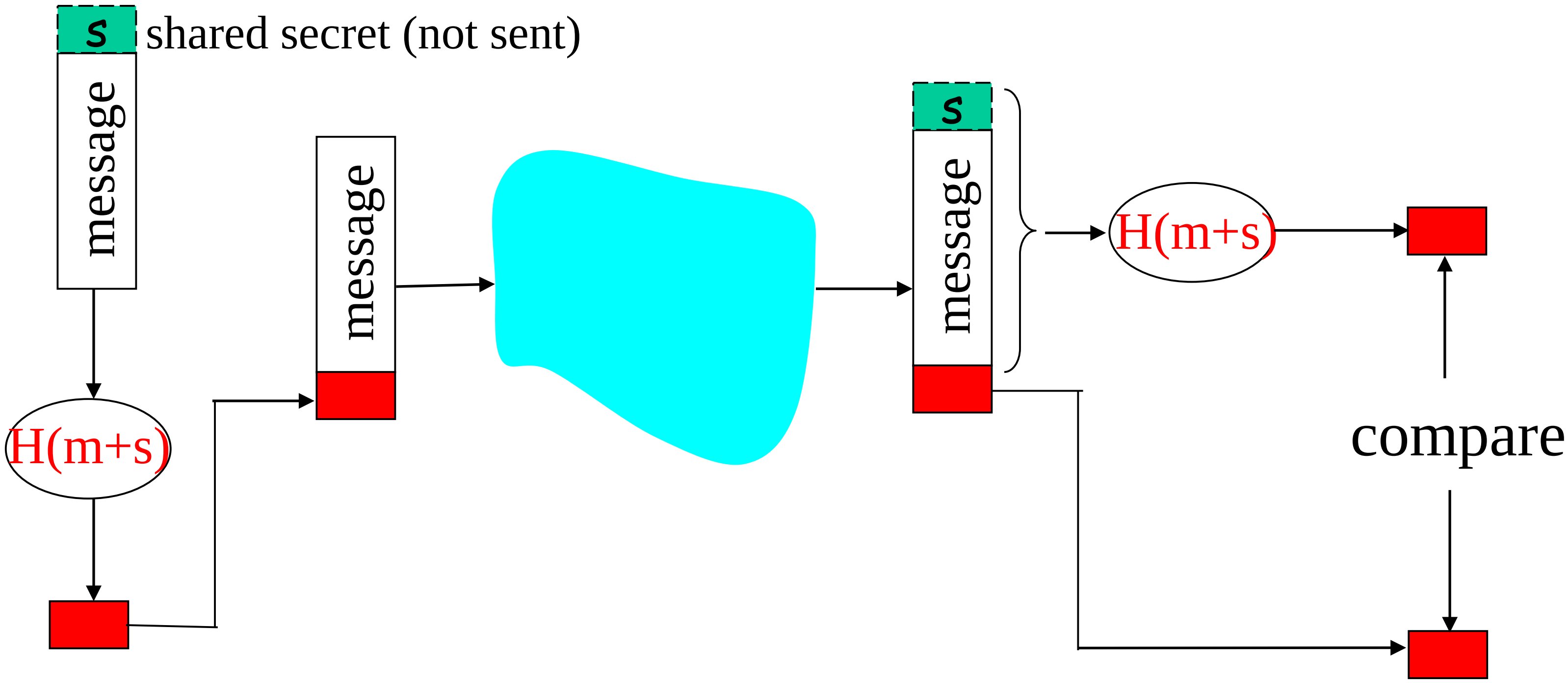
- **SHA-1** – Secure Hash Algorithm
  - Designed by NSA, became NIST standard 1995: FIPS-180-2
  - 160-bit message digest → 10^{48} different hashes
  - Avoid if collisions may cause problems in application, otherwise ok

"As of 2012, an estimated cost of $2.77M to break a single hash value by renting CPU power from cloud servers."
- SHA-1, Wikipedia

- **SHA-2** (family name for SHA-224, SHA-256, SHA-384 and SHA-512)
  - Similar design as SHA-1, but at least today SHA-1 attacks not applicable

- SHA-3 – next generation hash functions
  - Keccak - winner of open competition (NIST draft 2014)
  - Arbitrary digest size (standard proposes 224, 256, 384 and 512 bit digests)

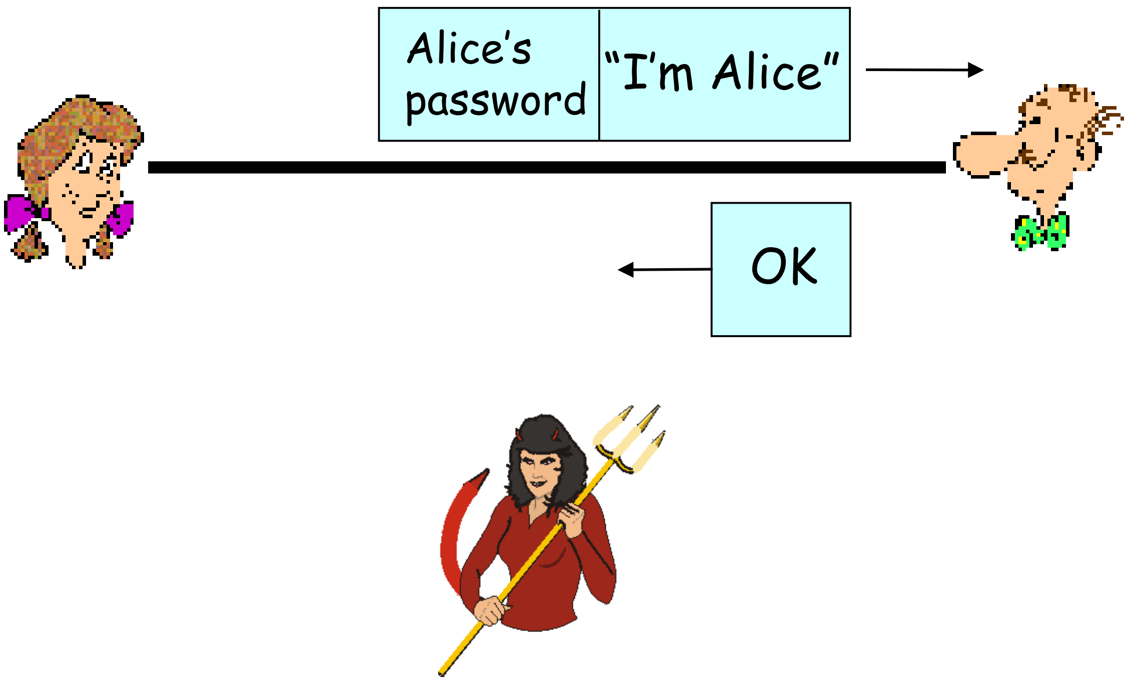
Keyed Hash - No need to encrypt message



- *Authenticates sender*
- *Verifies message integrity*
- No encryption !
- Example: HMAC (Key-Hashing for Message Authentiction)

End point (User) Authentication

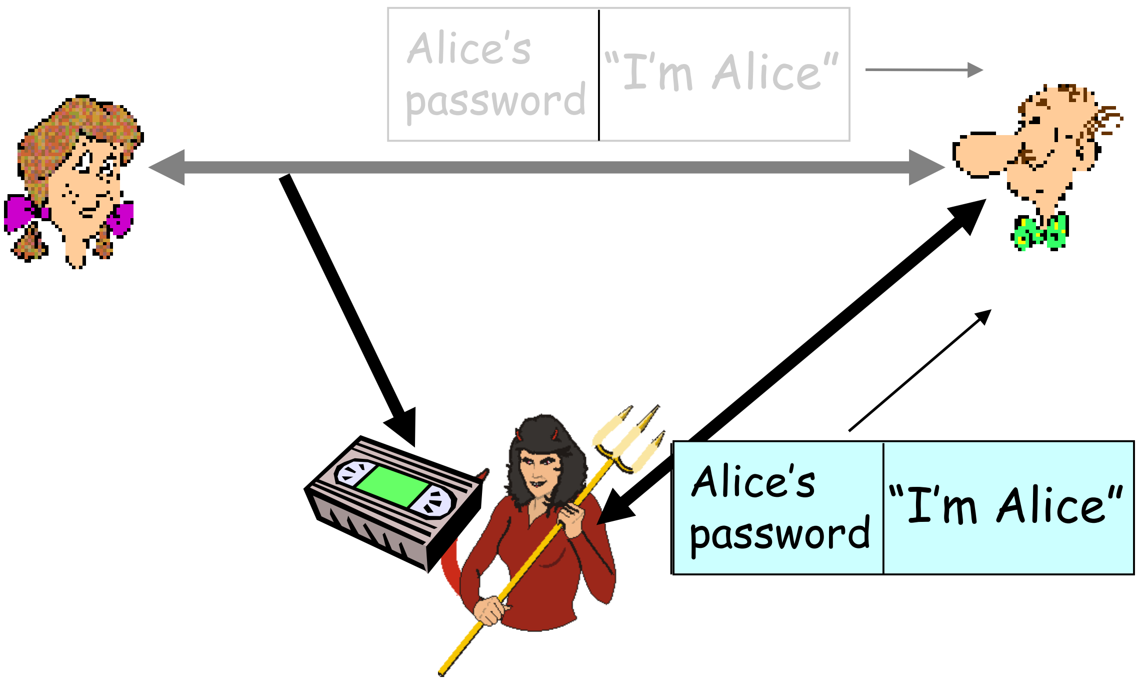
Alice says "I am Alice" and sends her secret password to "prove" it.
(Just like the FTP protocol)



Failure scenario??

End point (User) Authentication

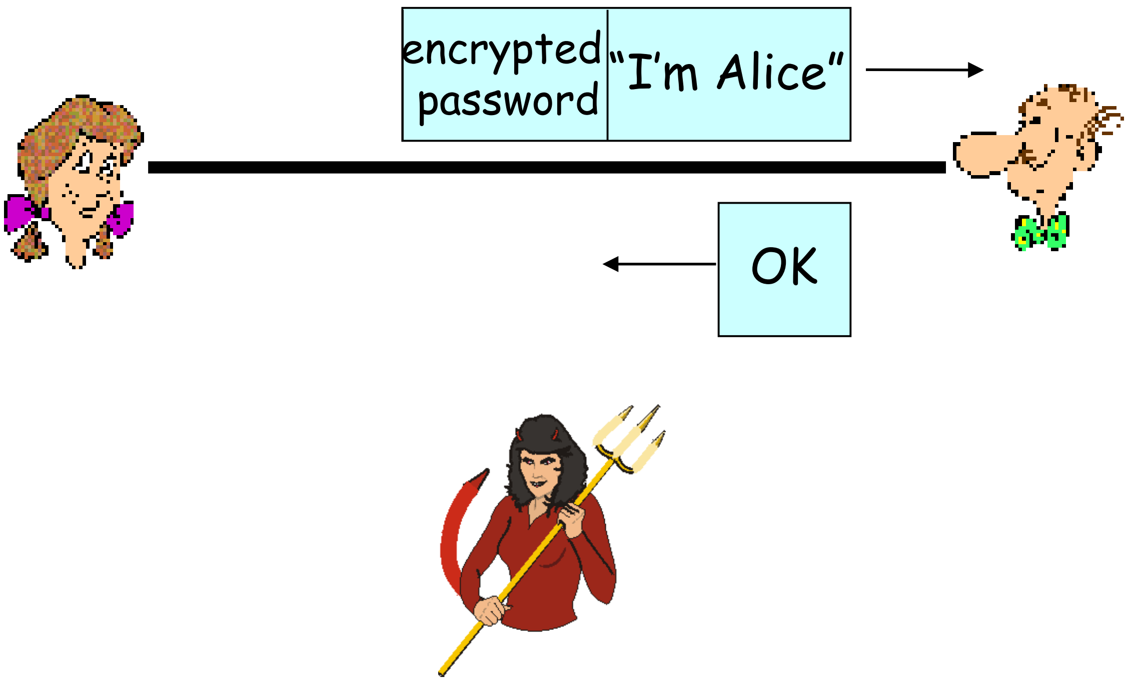
Alice says "I am Alice" and sends her secret password to "prove" it.



playback attack: Trudy records Alice's packet and later plays it back to Bob

Authentication: another try

Another attempt: Alice says “I am Alice” and sends her *encrypted* secret password to “prove” it.



Failure scenario??

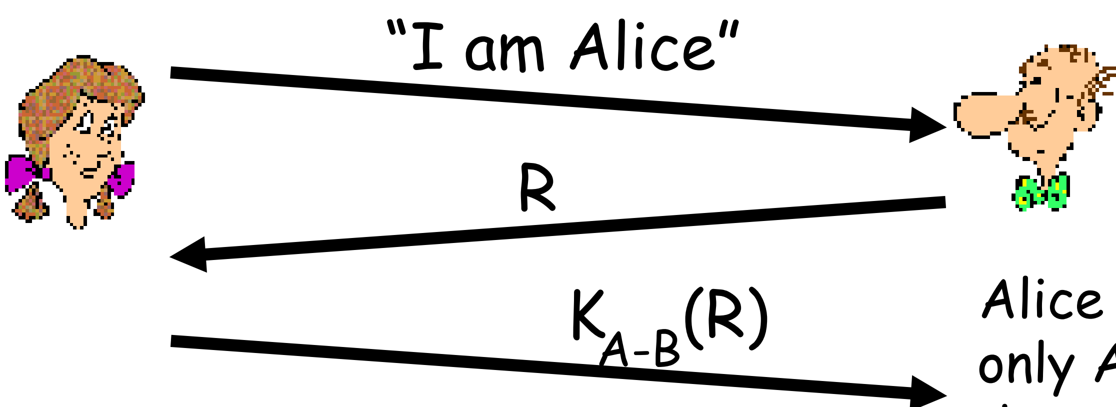
record
and
playback
still works!

Authentication: *Challenge response*

Goal: avoid playback attack

Nonce: number (R) used only *once–in-a-lifetime*

To prove Alice is "live", Bob sends Alice nonce, R.
Alice must return R, encrypted with shared secret key



Alice is live, and only Alice knows key to encrypt nonce, so it must be Alice!

Failures, drawbacks?

Summary

- Encryption for confidentiality
- Hashes for data integrity
- Sequence numbers for replay protection
- Authentication (mutual) for identity protection

- Symmetric encryption for bulk data
- Asymmetric encryption for key negotiation

Roadmap

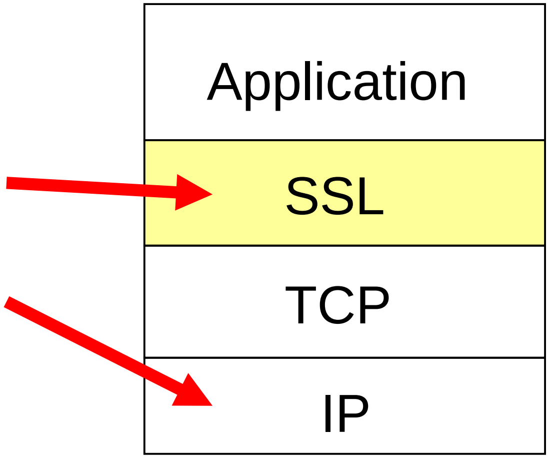
8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity

Security protocols and measures:

- Securing TCP connections: SSL
- Network layer security: IPsec
- Firewalls

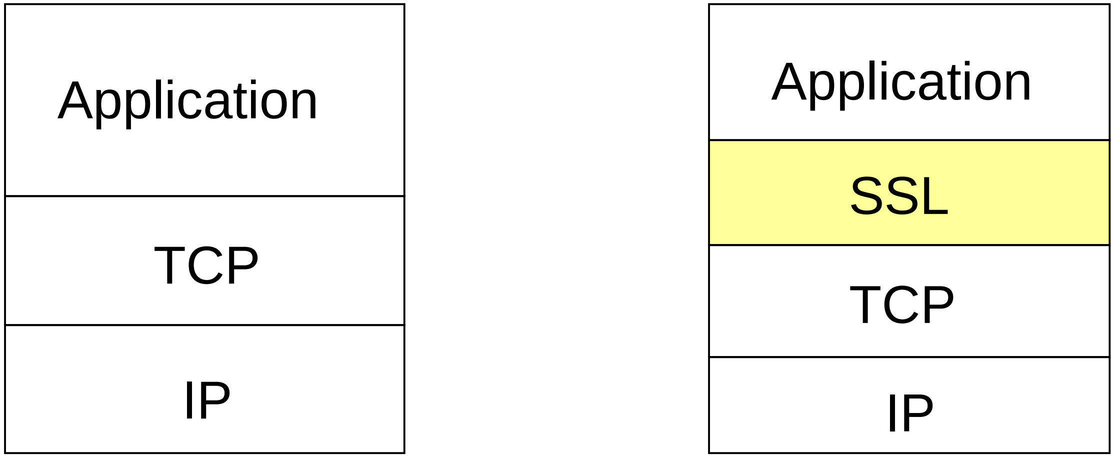


SSL: Secure Sockets Layer

- widely deployed security protocol
  - supported by almost all browsers, web servers
  - https
  - billions $/year over SSL
- mechanisms: [Woo 1994], implementation: Netscape
- variation -TLS: transport layer security, RFC 2246
- provides
  - *confidentiality*
  - *integrity*
  - *authentication*

- original goals:
  - Web e-commerce transactions
  - encryption (especially credit-card numbers)
  - Web-server authentication
  - optional client authentication
  - minimum hassle in doing business with new merchant
- available to all TCP applications
  - secure socket interface

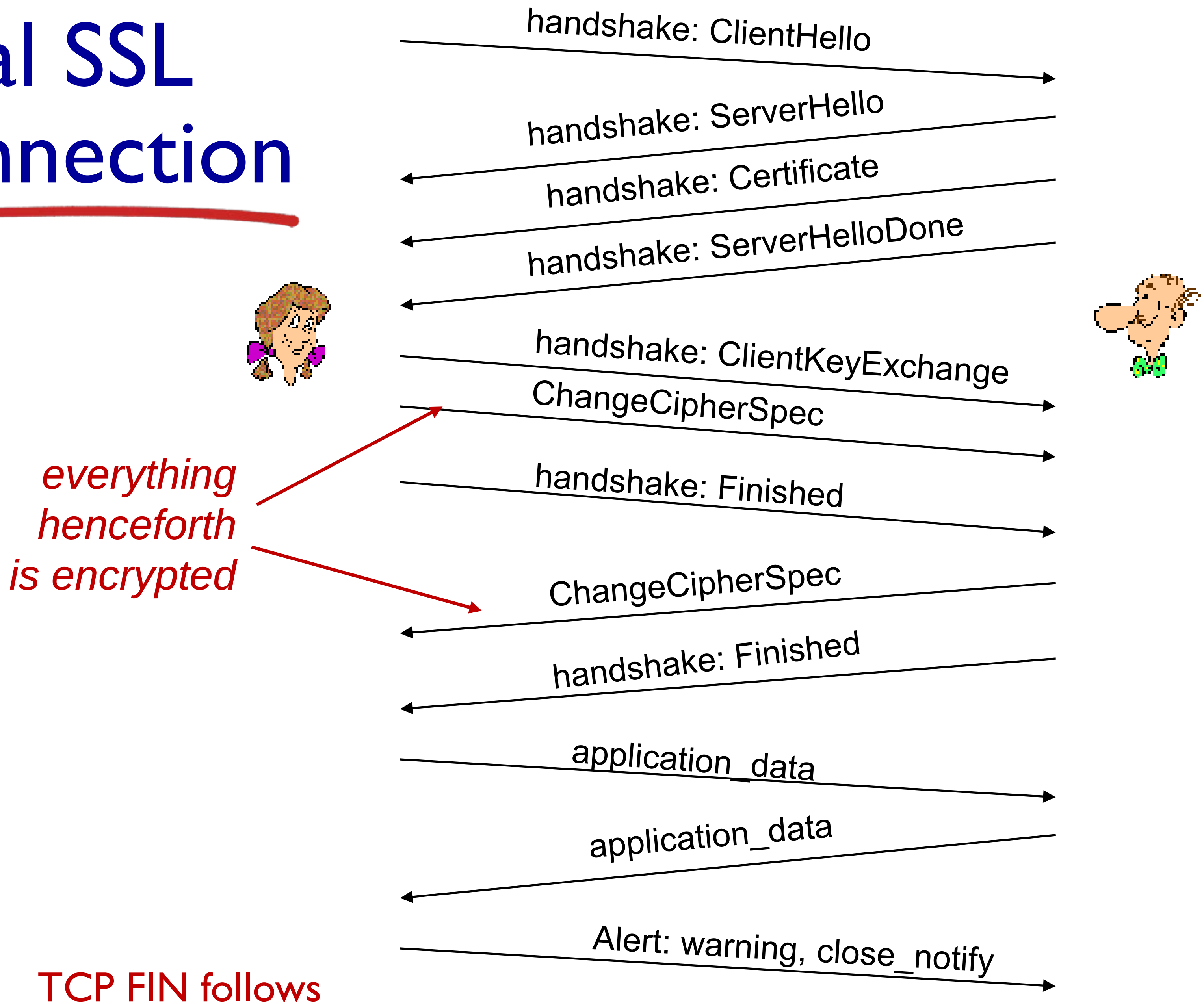
SSL and TCP/IP



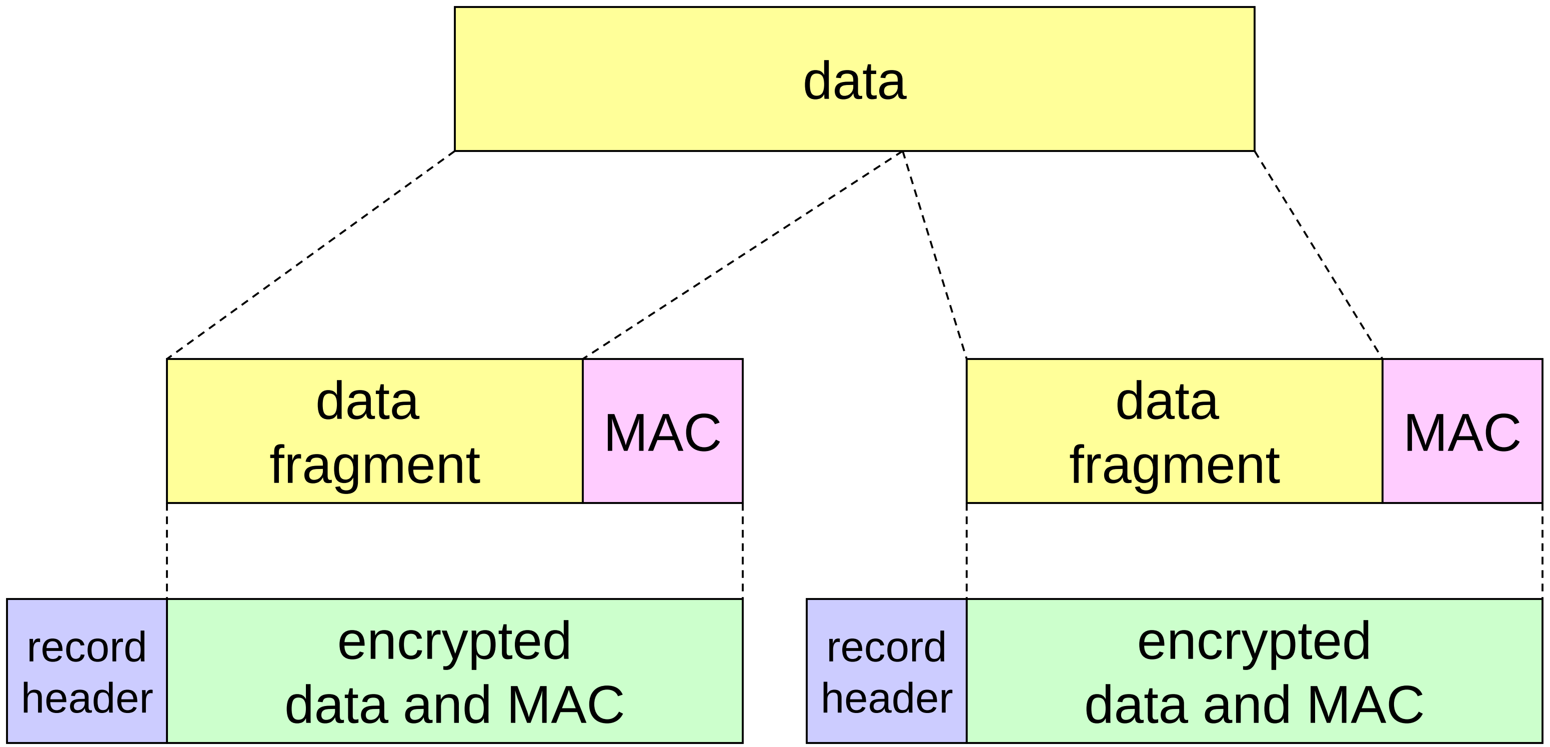
normal application   *application with SSL*

- SSL provides application programming interface (API) to applications
- C and Java SSL libraries/classes readily available

Real SSL connection



SSL record protocol



record header: content type; version; length

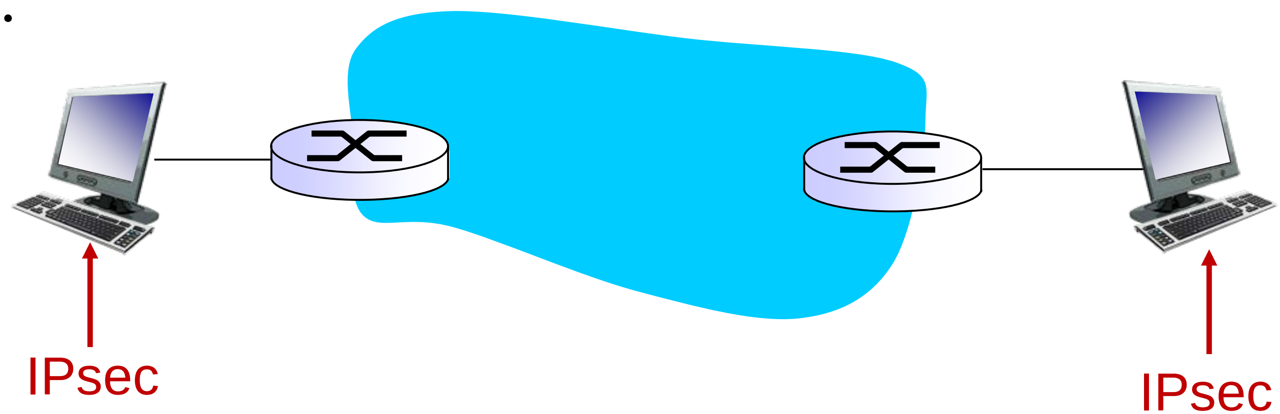
MAC: includes sequence number, MAC key M_x

fragment: each SSL fragment 2^{14} bytes (~16 Kbytes)

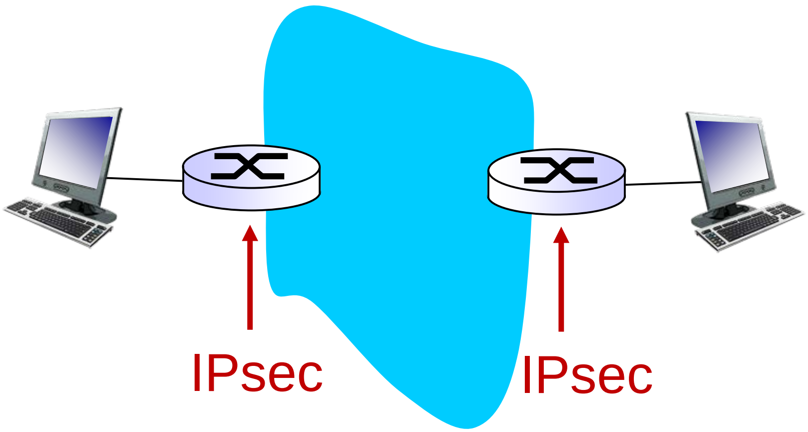
What is network-layer confidentiality ?

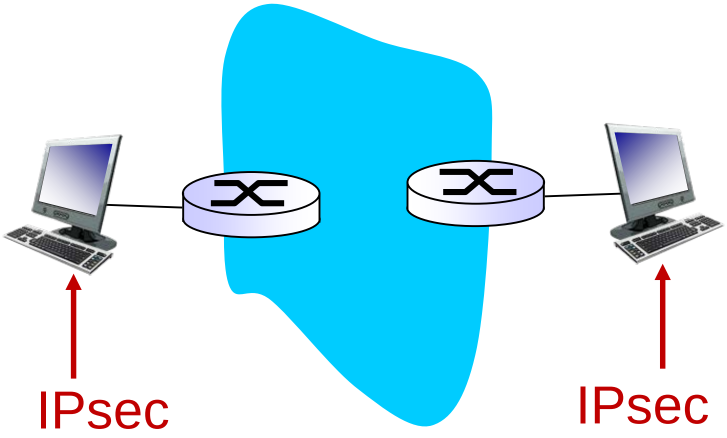
between two network entities:

- sending entity encrypts datagram payload, payload could be:
  - TCP or UDP segment, ICMP message, OSPF message ….
- all data sent from one entity to other would be hidden:
  - web pages, e-mail, P2P file transfers, TCP SYN packets …



The two modes of IPSec





- Tunnel mode
  - edge routers IPsec-aware
  - protects communication gw-to-gw (over Internet)
  - Virtual Private Network (VPN)
- Transport mode
  - hosts IPsec-aware
  - protects communication all the way from end-to-end

IPsec services

- data integrity
- confidentiality
- origin authentication
- replay attack prevention

two protocols providing different service models:

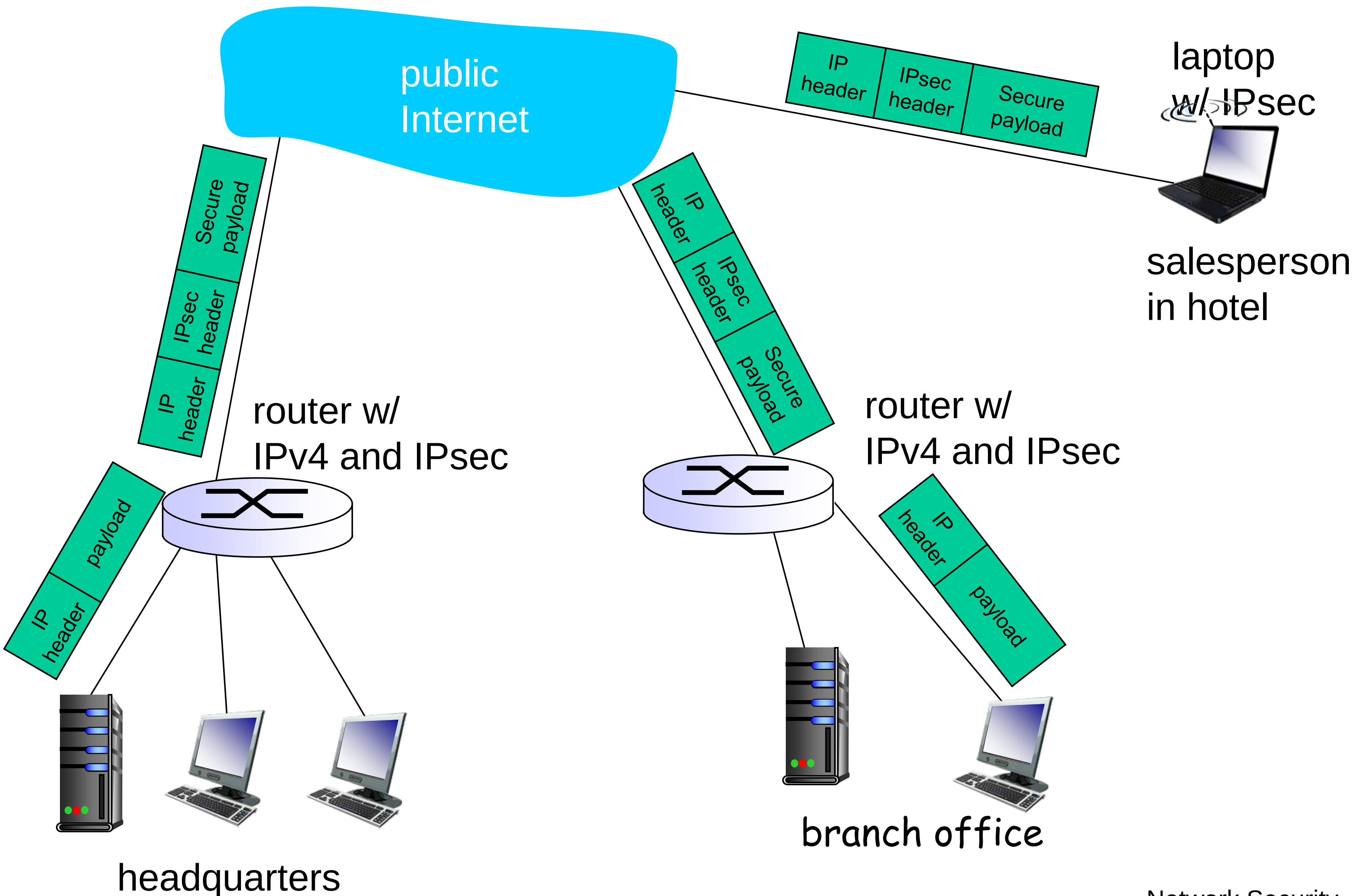
- Authentication Header (AH) protocol
  - provides source authentication & data integrity but *not* confidentiality
- Encapsulation Security Protocol (ESP)
  - provides source authentication, data integrity, *and confidentiality*
  - more widely used than AH

Virtual Private Networks (VPNs)

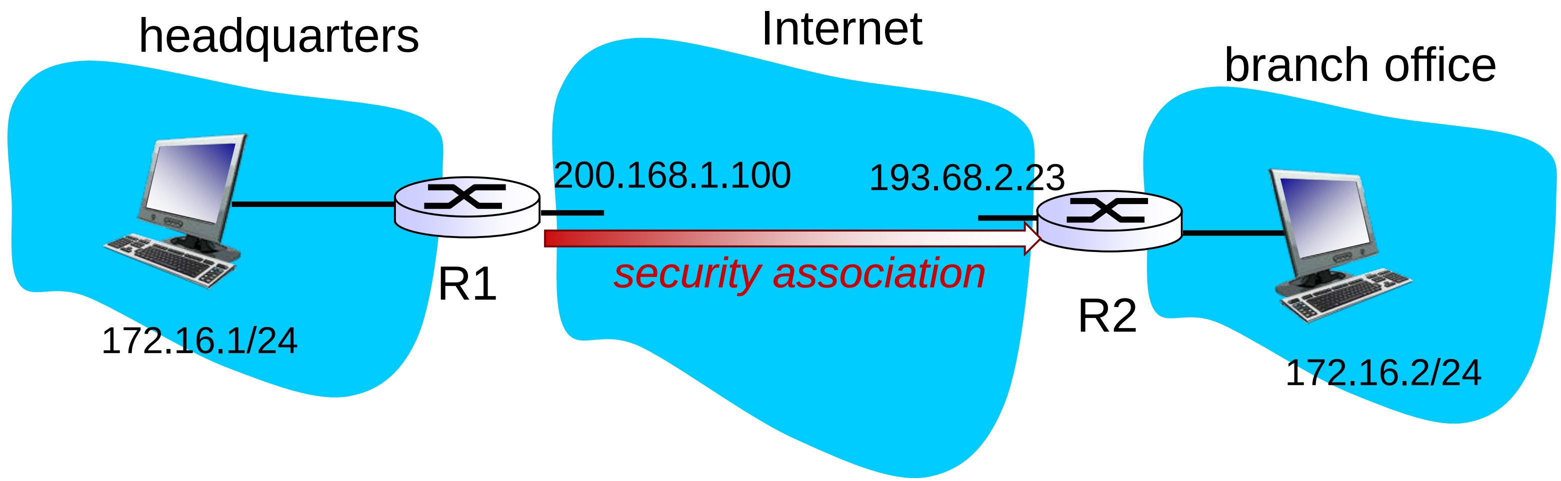
motivation:

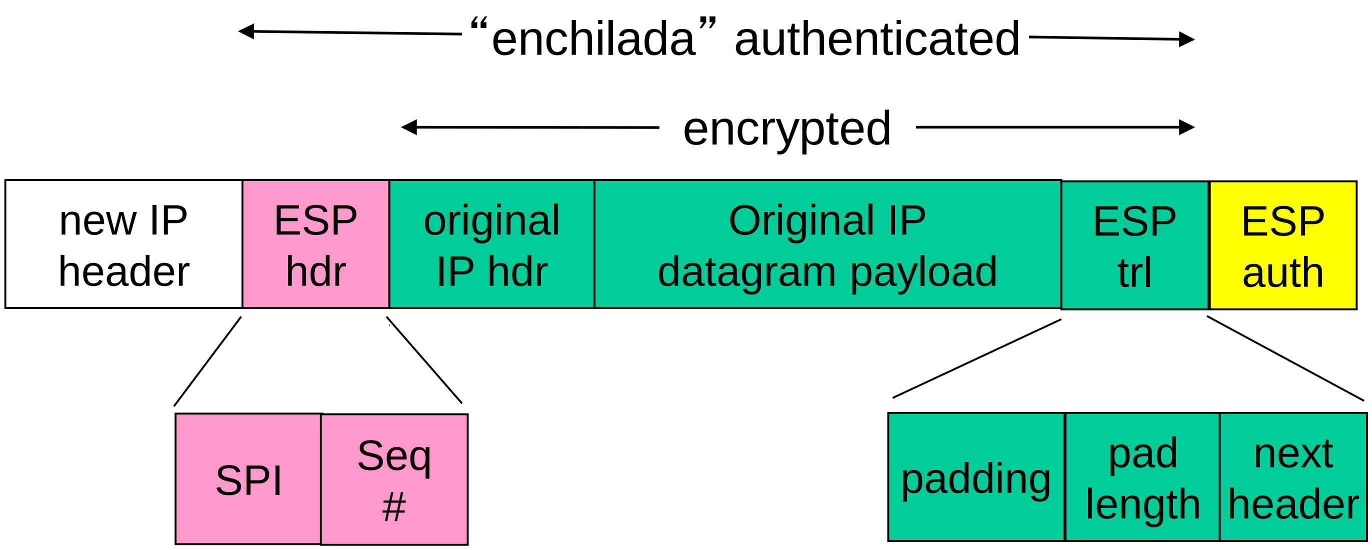
- institutions often want private networks for security.
  - costly: separate routers, links, DNS infrastructure.
- VPN: institution's inter-office traffic is sent over public Internet instead
  - encrypted before entering public Internet
  - logically separate from other traffic

Virtual Private Networks (VPNs)



What happens?



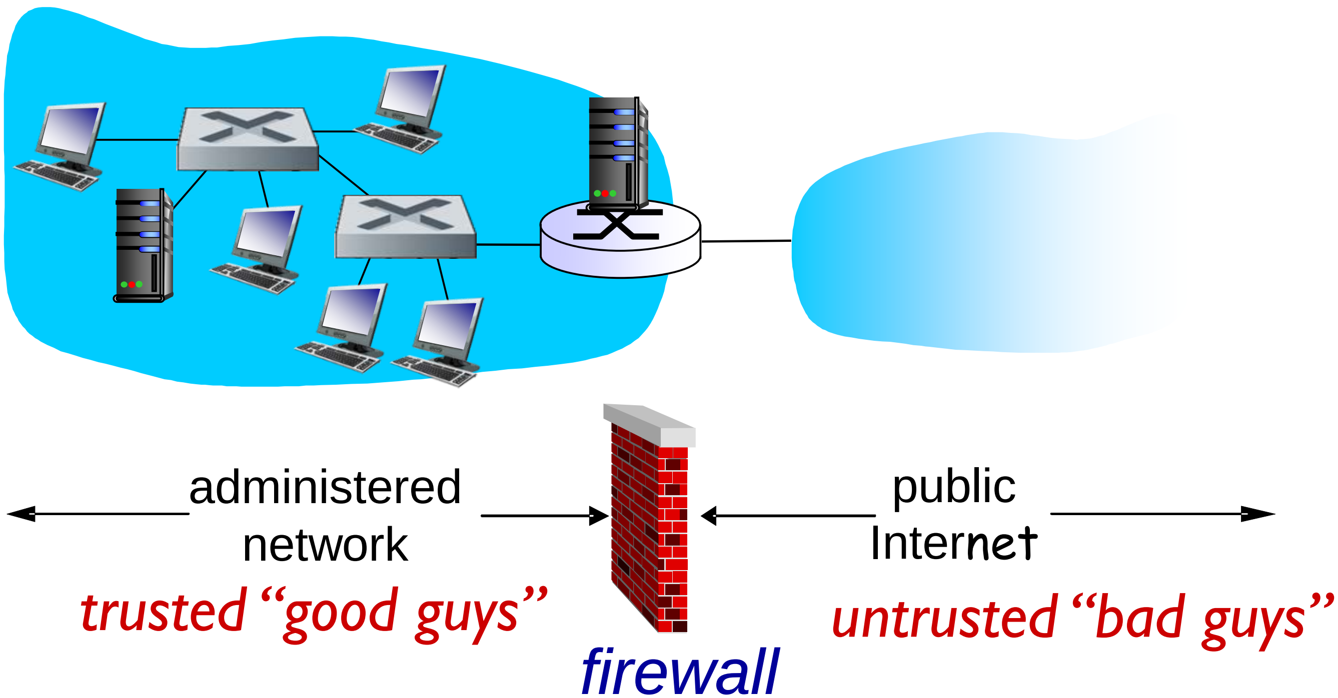


before sending data, "security association (SA)" established from sending to receiving entity

Firewalls

firewall

isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others



Firewalls: why

prevent denial of service attacks:

- SYN flooding: attacker establishes many bogus TCP connections, no resources left for "real" connections

prevent illegal modification/access of internal data

- e.g., attacker replaces CIA's homepage with something else

allow only authorized access to inside network

- set of authenticated users/hosts

three types of firewalls:

- stateless packet filters
- stateful packet filters
- application gateways

Säkerhetskurser på Chalmers

- Datasäkerhet EDA 263
- Nätverkssäkerhet EDA 491
- Kryptografi TDA 351
- Språkbaserad säkerhet TDA 602
- Feltoleranta datorsystem EDA 122