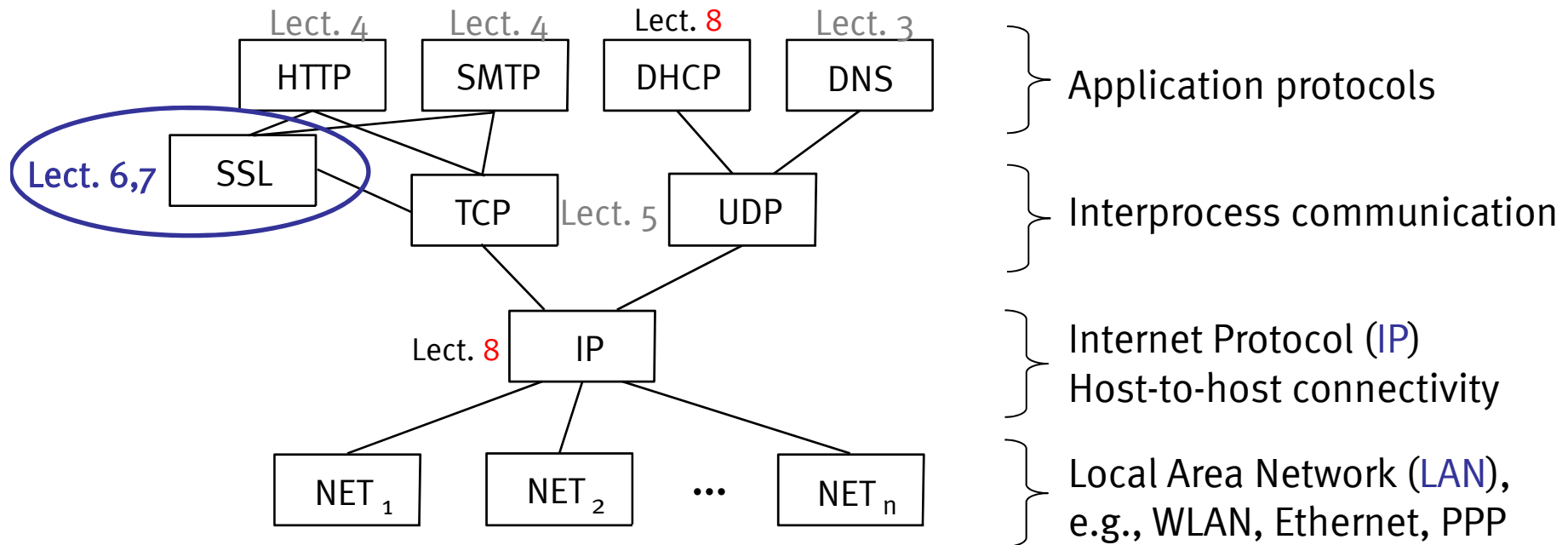




WESTFÄLISCHE  
WILHELMS-UNIVERSITÄT  
MÜNSTER

# Computer Networks, Winter Term 2009/2010

## Secure Channels: SSL/TLS

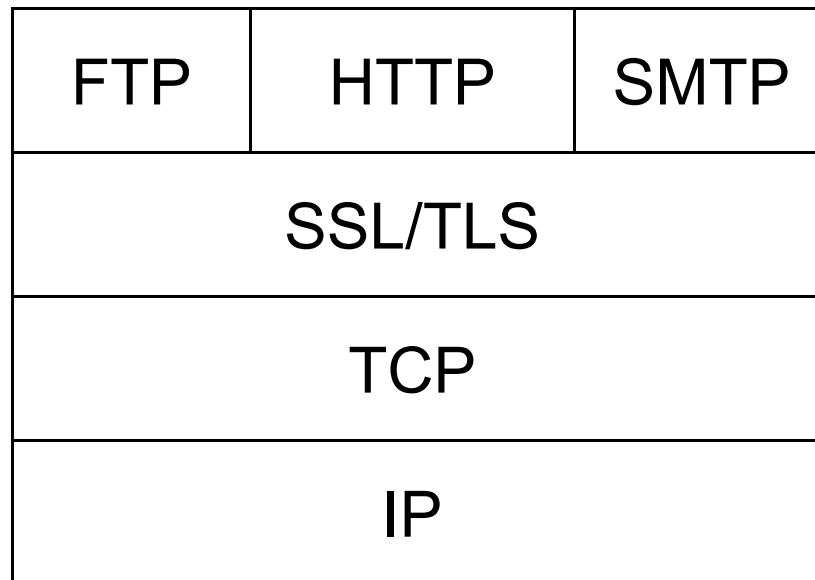


- How to establish identity?
- How to verify integrity?
- How to ensure confidentiality?
- How do SSL and TLS work?

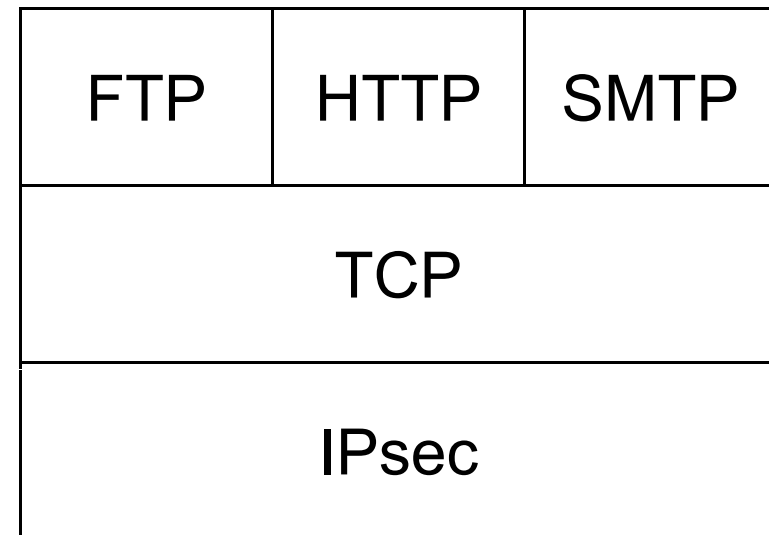
- Basics
- Cryptography
  - Hash Functions
  - Symmetric Encryption
  - Asymmetric Encryption
- Authentication
- Message Integrity and Signatures
- Certificates
- SSL/TLS

- Protection against deliberate attacks/threats
  - Confidentiality
  - Integrity
  - Availability
  - (Anonymity, non-repudiation, ...)
- Security via management and design processes
  - See seminar talks

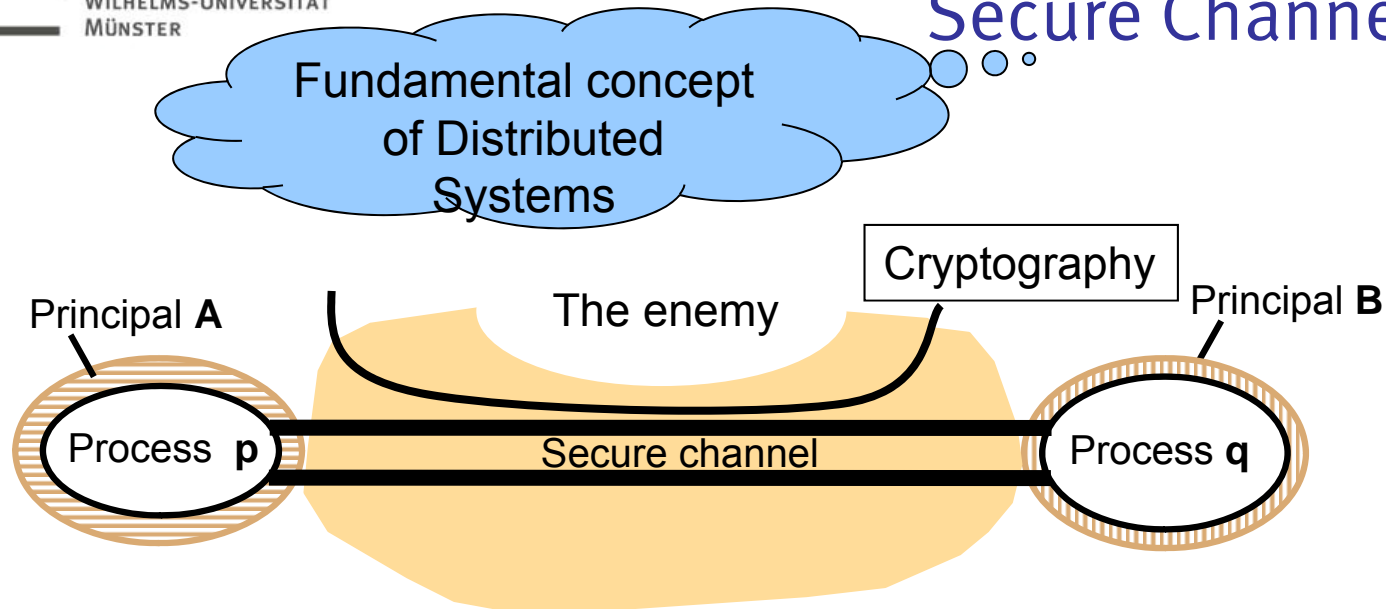
- Sniffing on share media networks
  - Violation of confidentiality
- Routing via store-and-forward
  - Routers may sniff and manipulate
  - Violation of confidentiality and integrity
- IP (ARP, BGP, DNS) spoofing
  - Violation of integrity



(a) Transport layer



(b) Network layer



## • Properties

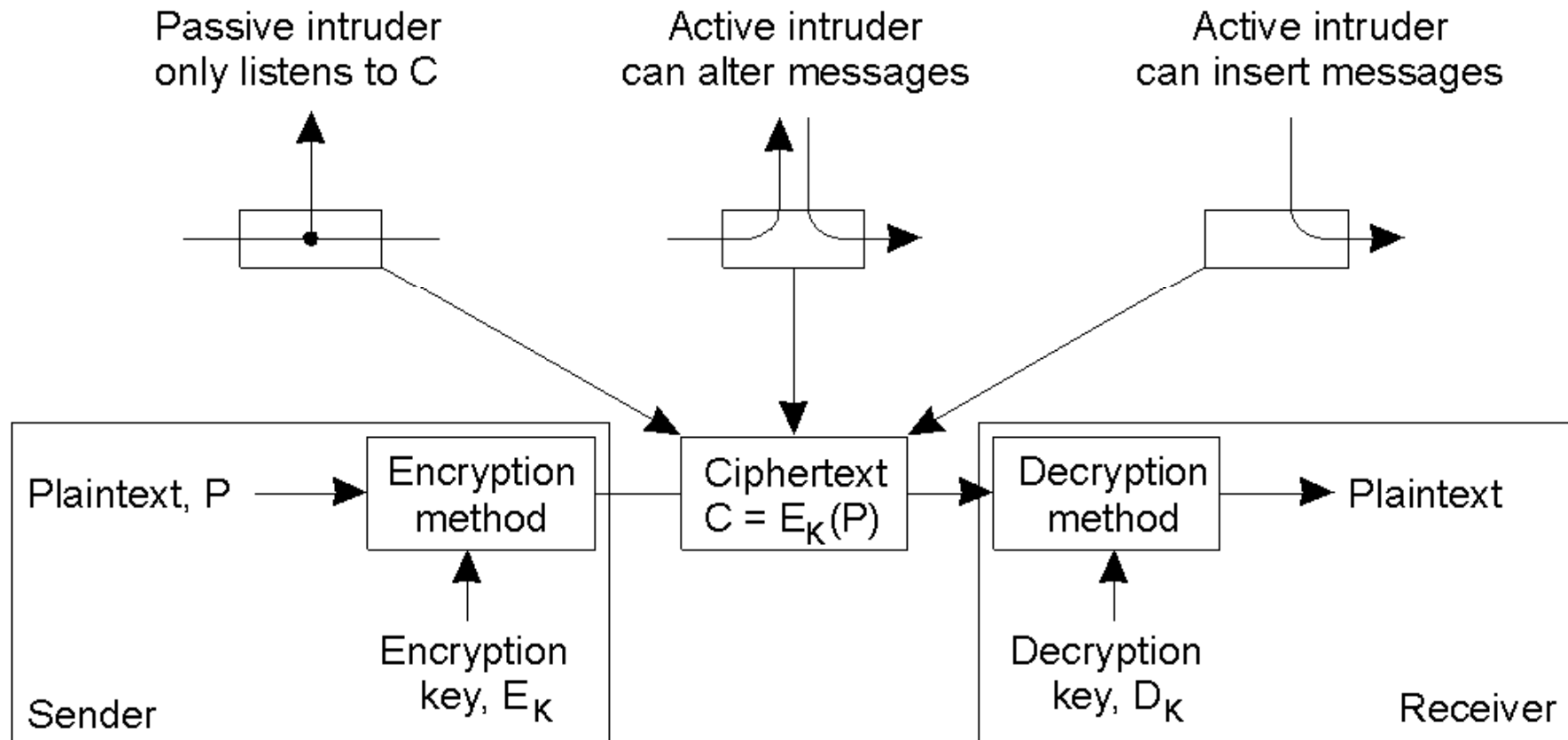
- Each process is sure of the **identity** of the other
- Data is **confidential** and **integrity** protected
- Protection against repetition and reordering of data

## • Employs **cryptography**

- Secrecy based on cryptographic concealment
- Authentication based on proof of ownership of secrets

Slide based on Instructor's Guide for Coulouris, Dollimore and Kindberg:

Distributed Systems - Concepts and Design (3<sup>rd</sup> ed.), © Addison-Wesley Publishers 2000



- Denial of service attacks
- Trojan horses and viruses
  - Software installed by users
  - Mobile code
  - Accidental execution
  - Defences
    - Code authentication (signed code)
    - Code validation (type checking, proof)
    - Sandboxing
- Protection tied to channel, not content
  - Must know/protect all channels
  - Recall dissemination pattern

- Basics
- **Cryptography**
  - Hash Functions
  - Symmetric Encryption
  - Asymmetric Encryption
- Authentication
- Message Integrity and Signatures
- Certificates
- SSL/TLS

- Art of “secret writing”
- Set of mathematical functions
  - Hash functions
  - Classes of encryption algorithms
    - Symmetric/secret key: shared secret key
    - Asymmetric/public key: pairs of secret and public keys
    - Hybrid: asymmetric initialization, symmetric encryption
  - Basis for various security mechanisms
- Performance
  - Hashing › Symmetric Enc. › Asymmetric Enc.

- Fundamental Tenet of Cryptography (Kaufman, Perlman, Speciner)
  - If lots of smart people have failed to solve a problem, then it probably won't be solved (soon).
- Kerckhoffs Principle
  - Security of crypto systems should not depend upon secrecy of en- and decryption functions
    - Not respected in national security/military/intelligence settings
    - Why?
  - Opposite: Security through obscurity



- Alice, Bob: Ordinary participants
  - <http://www.johngordonsweb.co.uk/concept/alicebob.html>
- Eve, Mallory: Eavesdropper, Malicious Attacker
  
- $K_A$ : Alice's secret key
- $K_B$ : Bob's secret key
- $K_{AB}$ : Secret key shared between Alice and Bob
- $K_A^-$ : Alice's private key
- $K_A^+$ : Alice's public key
- $K(M)$ : Message  $M$  encrypted with key  $K$
- $[M]_K$ : Message  $M$  signed with key  $K$

- Basics
- Cryptography
  - Hash Functions
  - Symmetric Encryption
  - Asymmetric Encryption
- Authentication
- Message Integrity and Signatures
- Certificates
- SSL/TLS

- Hash (or message digest): **One way** function
  - Input: Message  $M$  (bit string of arbitrary length)
  - Output: Hash value  $h$  (bit string of fixed length)
  - **Collision**: Different messages mapped to same hash value
- Hash value  $\approx$  digital fingerprint
  - Collision resistant
    - (Different fingerprints for different messages)
- Applications
  - Integrity tests
  - Digital signatures

- MD4: 128 Bit – outdated, **broken**
- MD5: 128 Bit – widely used, but **broken**
  - <http://en.wikipedia.org/wiki/MD5>
- SHA-1: 160 Bit – US Standard
  - Bruce Schneier: Algorithms from the NSA are considered a sort of alien technology: they come from a superior race with no explanations
  - RSA Conference 2005: **Collisions** for  $2^{69}$  messages (instead of  $2^{80}$ )
  - One **MD5-attack** may work for SHA as well:
    - <http://eprint.iacr.org/2006/105>
- RIPEMD-160: 160 Bit – European Standard
  - Some **MD5-attacks** may work as well
- Hash Futures Panel, August 2006,  
<http://www.proper.com/lookit/hash-futures-panel-notes.html>
  - “Joux says that we do not understand what we are doing and that we do not really know what we want; there is agreement from all the panelists.”
- Since 2007: SHA-3 competition
  - <http://csrc.nist.gov/groups/ST/hash/sha-3/index.html>
  - Selection of winner scheduled for 2012

- Basics
- Cryptography
  - Hash Functions
  - **Symmetric Encryption**
  - Asymmetric Encryption
- Authentication
- Message Integrity and Signatures
- Certificates
- SSL/TLS

- En- and decryption algorithms E and D
  - Parameters: Secret key K, message M
- $E(K, M) = K(M)$ 
  - Ciphertext
- $D(K, K(M)) = M$ 
  - Plaintext
- **Challenge:** Secret exchange of K

- Principles

- Confusion: Combination of plaintext and key via non-destructive functions such as XOR and Shift
- Diffusion: Elimination of repetitions and redundancy within plaintext via permutations

- Some common algorithms

- IDEA (128 Bit, patented, used in PGP—not in GPG)
- DES, Triple-DES (56 Bit, outdated, superseded by AES)
- AES (Rijndael, 256 Bit)
- RC6, Twofish (AES candidates)
- Blowfish (448 Bit)

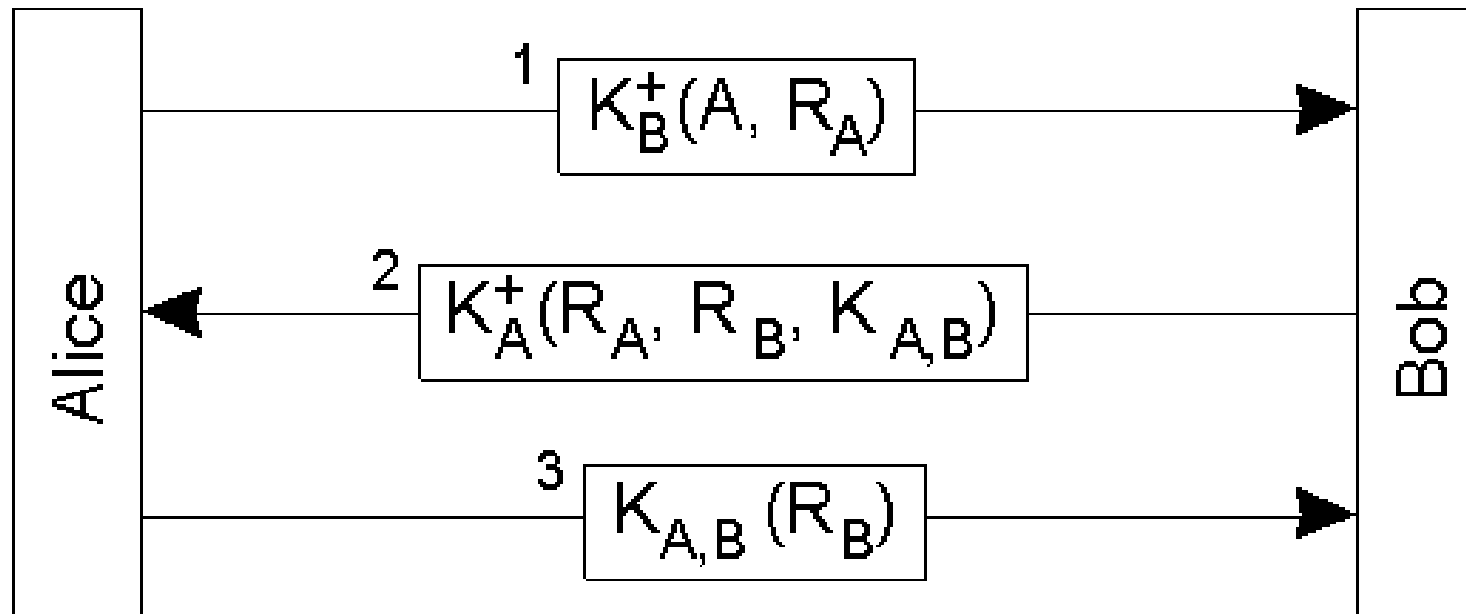
- Basics
- Cryptography
  - Hash Functions
  - Symmetric Encryption
  - **Asymmetric Encryption**
- Authentication
- Message Integrity and Signatures
- Certificates
- SSL/TLS

- Participants have key **pairs**
  - Private key, e.g.,  $K_B^-$ : secret
  - Public key, e.g.,  $K_B^+$ : published
- Encryption with Bob's public key
  - $E(K_B^+, M) = K_B^+(M)$
  - **Notice**: No secret key exchange necessary
- Decryption with Bob's secret key
  - $D(K_B^-, K_B^+(M)) = M$
  - **Notice**: Only Bob can do this
- **Challenge**: Reliable distribution of public keys  
(Public Key Infrastructure, PKI)

- Diffie-Hellman Key Exchange (1976)
  - Used, e.g., in IPsec, SSL, Tor
- RSA (Rivest, Shamir, Adleman 1978)
  - Most famous, PGP
- ElGamal (1984)
  - Based on Diffie-Hellman
  - GnuPG and newer PGP variants
- Elliptic curves
  - Newest class, shorter keys, patents

- Basics
- Cryptography
  - Hash Functions
  - Symmetric Encryption
  - Asymmetric Encryption
- **Authentication**
- Message Integrity and Signatures
- Certificates
- SSL/TLS

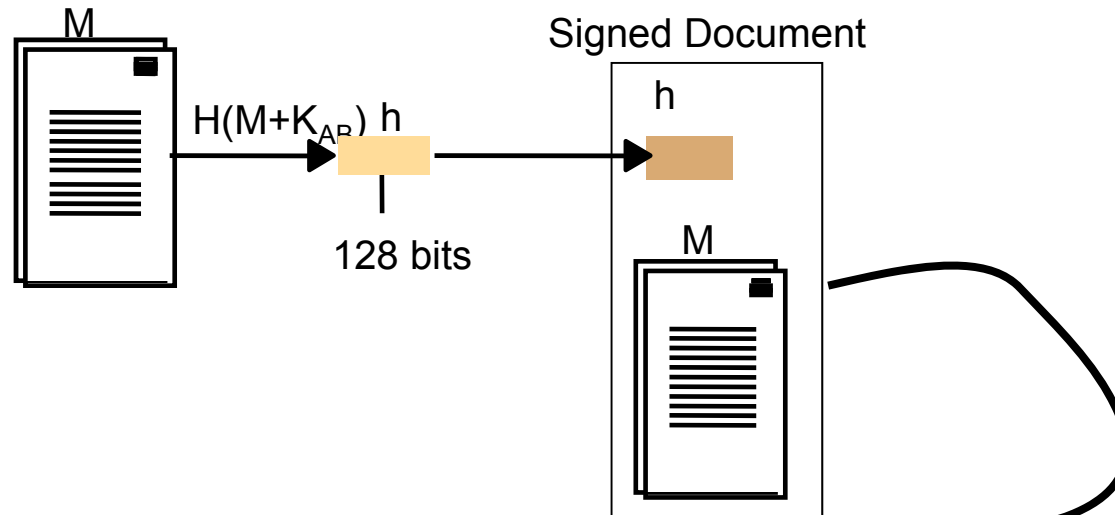
- Proof of identity
- Here: **Challenges**
- Notice: Need to **combine** authentication with message integrity
  - E.g., symmetric encryption with secret session key after authentication
    - Integrity
    - Confidentiality



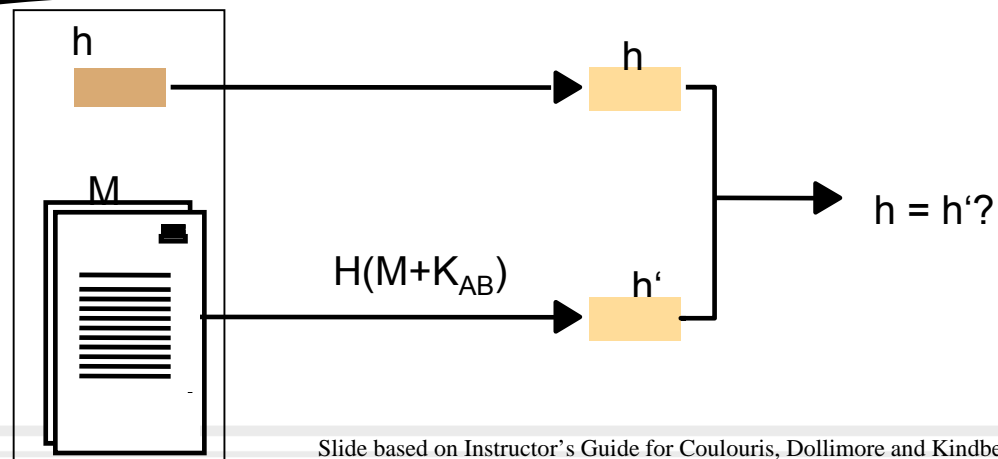
- Basics
- Cryptography
  - Hash Functions
  - Symmetric Encryption
  - Asymmetric Encryption
- Authentication
- **Message Integrity and Signatures**
- Certificates
- SSL/TLS

- MAC = cryptographic checksum
- E.g., Keyed MD5
  - Concatenate message  $M$  and shared secret  $K_{AB}$ :  $M+K_{AB}$
  - Compute hash value as checksum:  $MD5(M+K_{AB})$
- Notice: No “real” signature
  - Everyone knowing  $K_{AB}$  can produce checksum

“Signing” by Alice



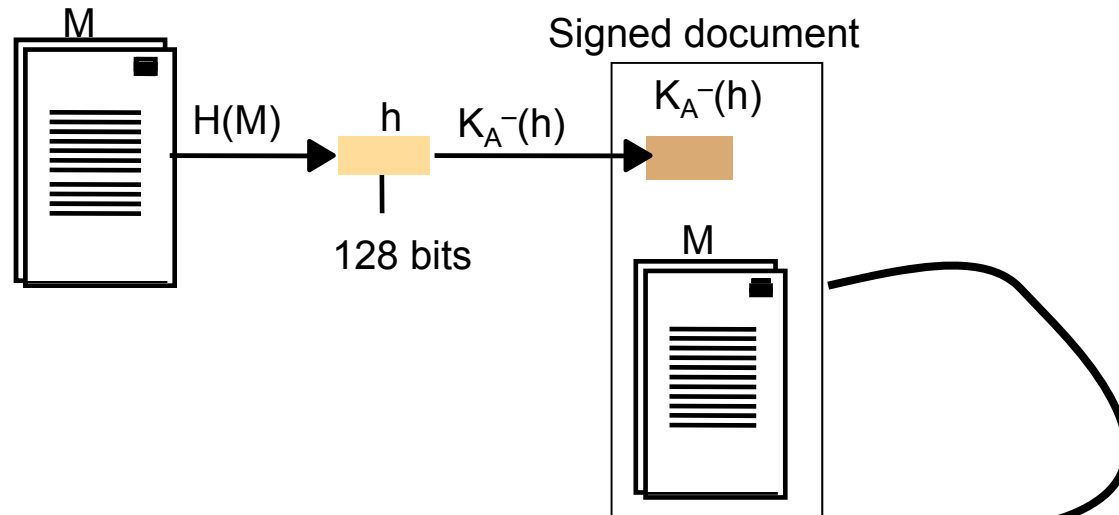
Verification by Bob



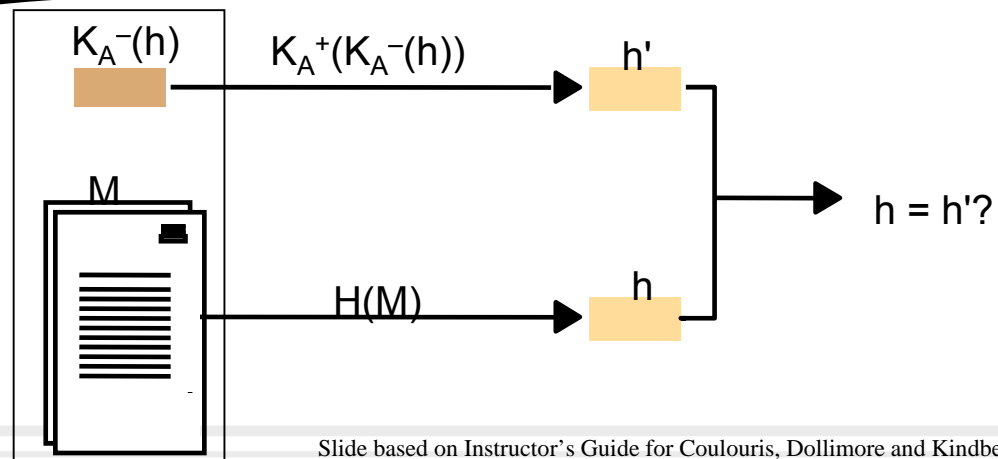
Slide based on Instructor's Guide for Coulouris, Dollimore and Kindberg:  
Distributed Systems - Concepts and Design (3<sup>rd</sup> ed.), © Addison-Wesley Publishers 2000

- Based on asymmetric cryptography
  - En- and decryption reversed
- Basic idea for signatures
  - Signature obtained by encryption with private key
    - $K_B^-(M)$
    - Only Bob can create this!
  - Verification via decryption with public key
    - $D(K_B^+, K_B^-(M))$
    - Everyone can do this!
  - Practice: Encrypt hash value of M

Signing by Alice



Verification by Bob



Slide based on Instructor's Guide for Coulouris, Dollimore and Kindberg:  
Distributed Systems - Concepts and Design (3<sup>rd</sup> ed.), © Addison-Wesley Publishers 2000

- Alice wants to publish a document for a restricted set of recipients. Every recipient should be able to detect whether the document has been manipulated.
- What are the security goals?
- What security mechanisms are applicable?



- Basics
- Cryptography
  - Hash Functions
  - Symmetric Encryption
  - Asymmetric Encryption
- Authentication
- Message Integrity and Signatures
- **Certificates**
- SSL/TLS

- Certificate = digitally signed document
  - *“I certify that the **public key** contained in this document belongs to the **named person/entity**, signed X.”*
  - Name of certified entity
  - Public key of certified entity
  - Name of certifying authority
  - Period of validity
  - Digital signature
    - Hash value encrypted with private key of certifying authority

- Goal: Reliable distribution of public keys
- Components
  - Certificates
  - Database for distribution of certificates
  - Method to revoke certificates
  - Verification method, starting from known public keys (“trust” anchors)

- Meaning?

- <http://www.lafkon.net/tc/>
- Mutual understanding
- Local property → **Not applicable** on Internet

- PKI uses “trust” anchors

- Certificates that are known/asserted to be authentic
- Browser
  - Vendor includes “trust” anchors
  - All of us use them all the time!

- Sample CA failures

- <http://www.win.tue.nl/hashclash/rogue-ca/>
- <http://support.microsoft.com/kb/293817/>
- <http://isc.sans.org/diary.html?storyid=1118>

- Certifying Authority (CA)
  - Administrative unit issuing certificate
  - Useful if CA's public key already known
    - Knowledge via "trust" anchor
- Chain of Trust
  - C certifies:  $K_B^+$  belongs to B
  - B certifies:  $K_A^+$  belongs to A
  - Generates chain from C via B to A
  - Verification of A's public key
    - Possession of C's public key ("trust" anchor)
    - Follow chain
- Certificate Revocation List (CRL)
  - List of certificates that should not be used any more
  - Published by CA (also online services, OCSP)

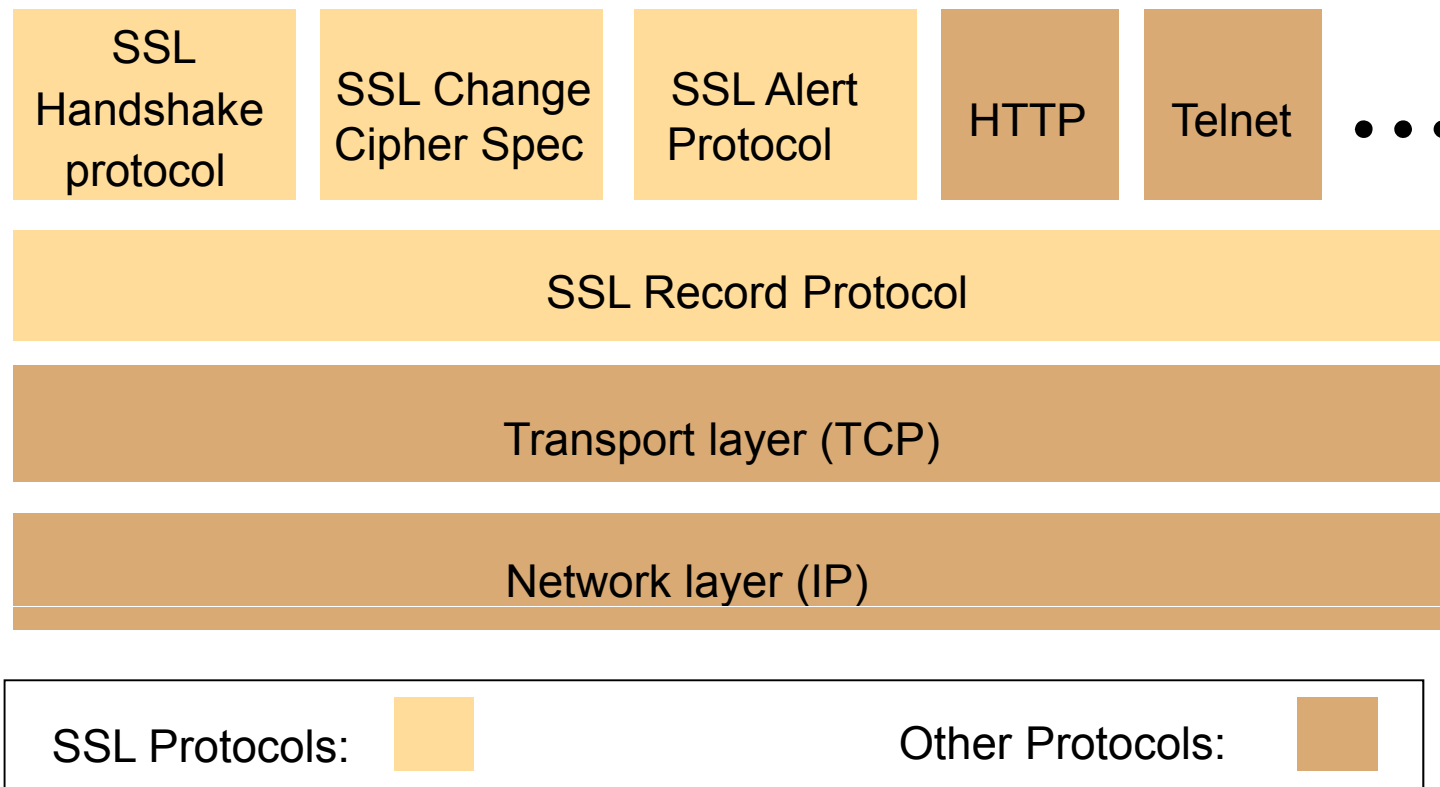
$K_A^+$  belongs to [www.example.com](http://www.example.com), signed by C

1. (C did not make a mistake)
2. Recipient has  $K_C^+$
3. Valid signature by C
4. Certificate presented by [www.example.com](http://www.example.com)
5. Period of validity OK
6. Certificate not revoked

- Basics
- Cryptography
  - Hash Functions
  - Symmetric Encryption
  - Asymmetric Encryption
- Authentication
- Message Integrity and Signatures
- Certificates
- **SSL/TLS**

- Secure Sockets Layer (SSL)
  - Originally by Netscape (1994); first implementation known as Version 2.0
    - Outdated, MITM attacks
  - 1996 [SSL Version 3.0](#)
- SSL development under new term [Transport Layer Security \(TLS\)](#)
  - Allows [Perfect Forward Secrecy](#) with Diffie-Hellman
  - 2008 [TLS Version 1.2 \(RFC5246\)](#)
    - Security improvements, more flexibility
- Secure channels for Internet commerce (<https://www...>, IMAPS, POPS, etc.)
  - Certificates, [hybrid](#) cryptography
    - Server presents certificate ([asymmetric crypto](#)), client may present certificate
    - Encryption with session keys ([symmetric crypto](#))
    - Recall: Symmetric encryption more efficient than asymmetric one
  - Mutual authentication, integrity, confidentiality

- Recent news: Protocol weakness
  - <http://www.links.org/?p=780>
- Browser bugs
  - <http://research.microsoft.com/apps/pubs/default.aspx?id=79323>
- Quality of random numbers/keys
  - <http://www.debian.org/security/2008/dsa-1571>
  - <http://eprint.iacr.org/2007/419.pdf>
- Questionable usability
- Above trust issues

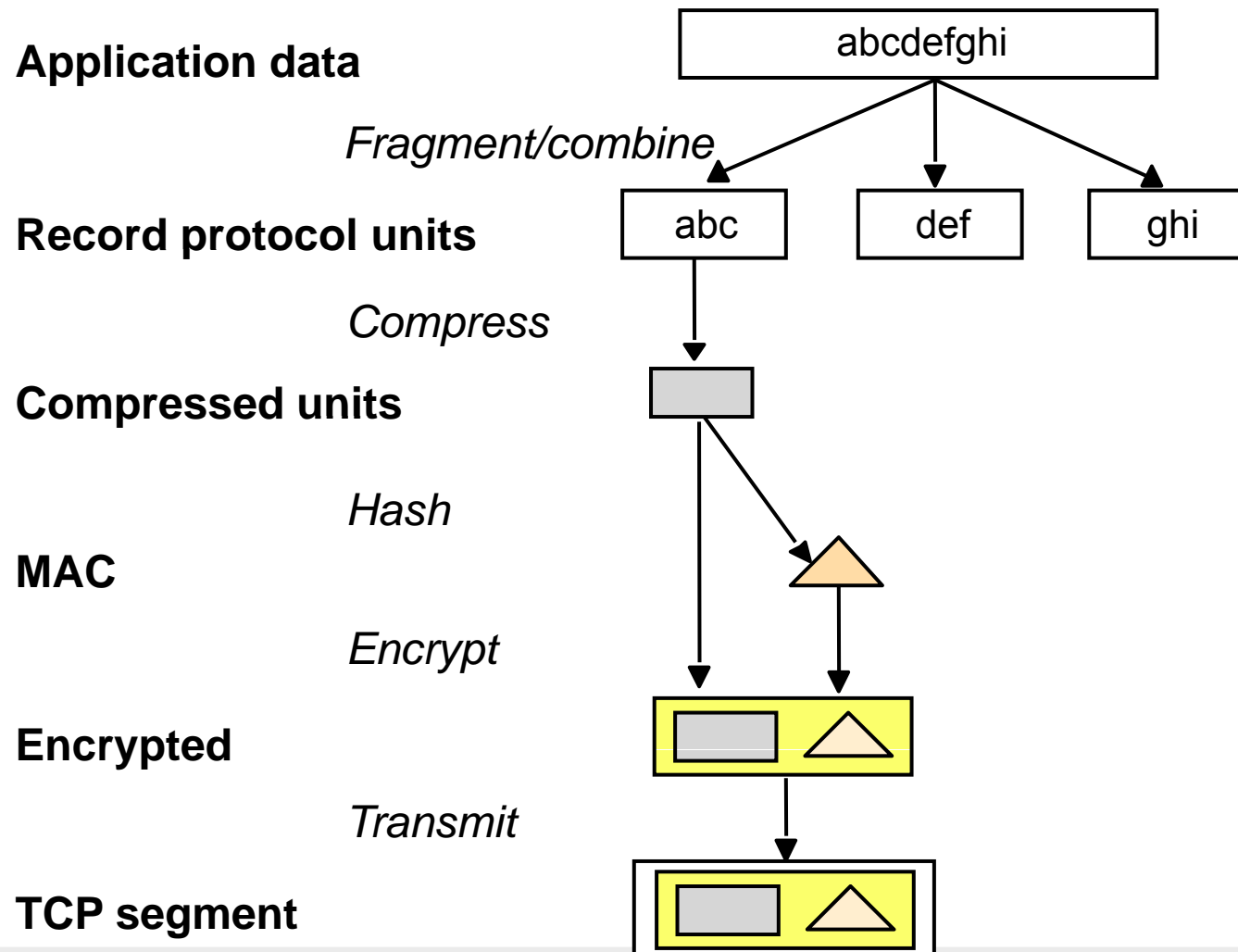


- Handshake layer
  - Management of parameters for secure channel
  - Cipher suite: Chosen crypto algorithms

## Cipher suite

<i>Component</i>	<i>Description</i>	<i>Example</i>
Key exchange method	the method to be used for exchange of a session key	RSA with public-key certificates
Cipher for data transfer	the block or stream cipher to be used for data	IDEA
Message digest function	for creating message authentication codes (MACs)	SHA

- SSL Record Protocol
  - Implements secure channel



Slide based on Instructor's Guide for Coulouris, Dollimore and Kindberg:  
Distributed Systems - Concepts and Design (3<sup>rd</sup> ed.), © Addison-Wesley Publishers 2000

- Secure channels form building block for secure network applications
  - Digital signatures (with certificates) for identity
  - (Keyed) Hashing for integrity
  - Symmetric encryption for confidentiality
- SSL/TLS most important standard for secure channels
  - Hybrid
  - Certificates, PKI
  - Between application and transport layer

- Discuss goals and limitations of secure channels
- Discuss alternatives for confidentiality and integrity of messages
- Explain use and verification of certificates