# Internetwork Security

- Why Network Security Layers?
- Fundamentals of Encryption
- Network Security Layer Overview
  - PGP
  - SSL/TLS
  - Lower Layers
- Security on Internet Layer
  - IPSec
  - IPv6-GCAs



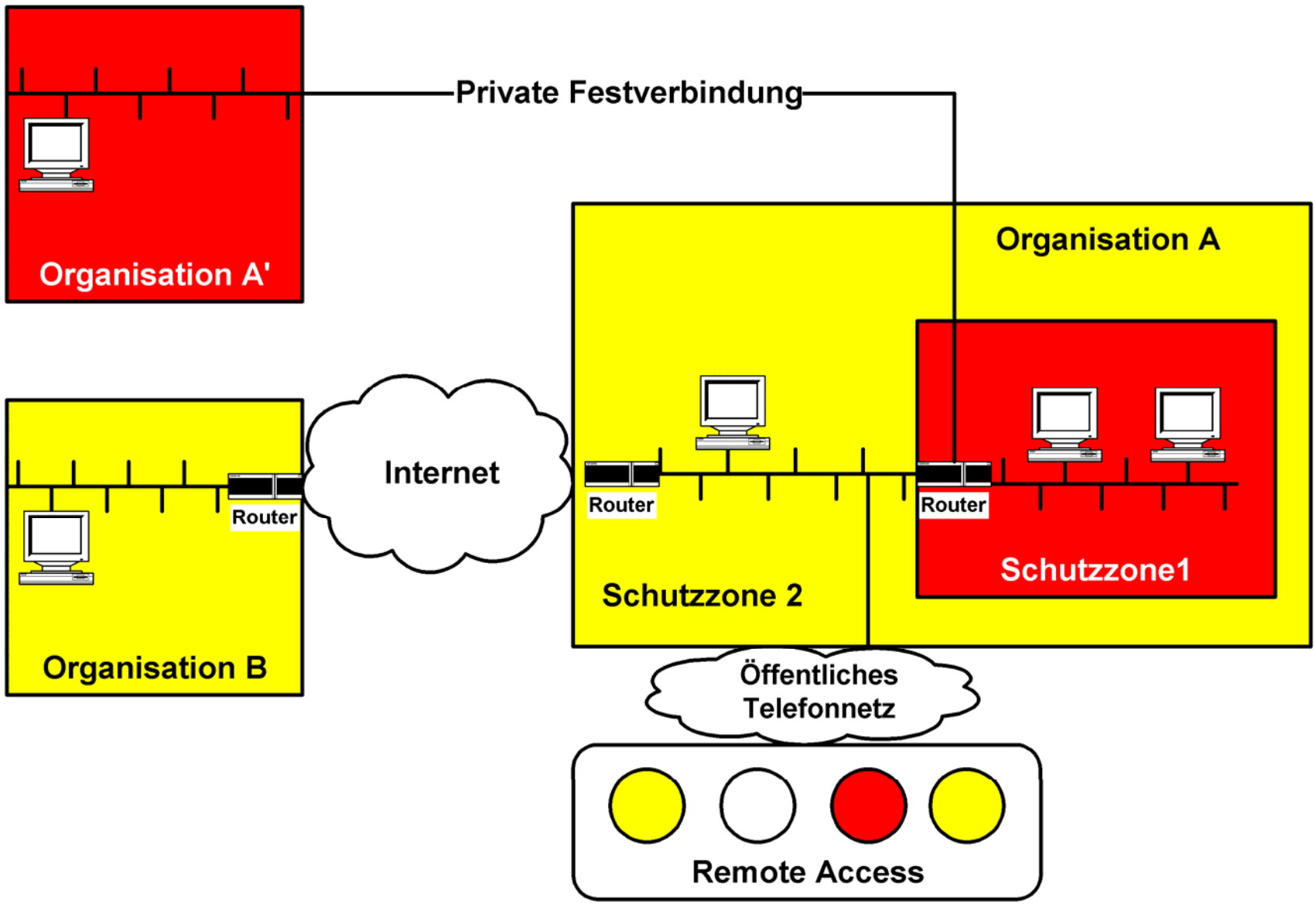
Hochschule für Angewandte Wissenschaften Hamburg
*Hamburg University of Applied Sciences*

# Security Threats in the Network

- Spying out your data
- Manipulating your data
- Computer and system sabotage
- Analysis of communication profiles
- ...

Problem: To gain physical control of networks is expensive and often unreachable

# Wide-Area Scenarios



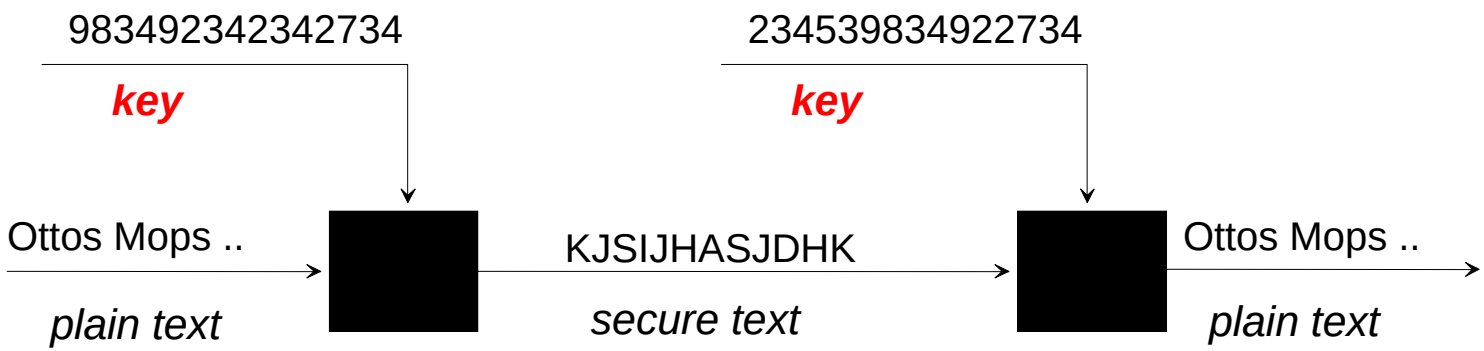
# Objectives of Security Layers

Assure:

- Secrecy of information
- Secrecy of communication relations
- Verification of information integrity
- Verification of (sender-) authenticity
- Protection of infrastructure
- ...

# Basis: Encryption

Gain security objectives in public networks by encryption



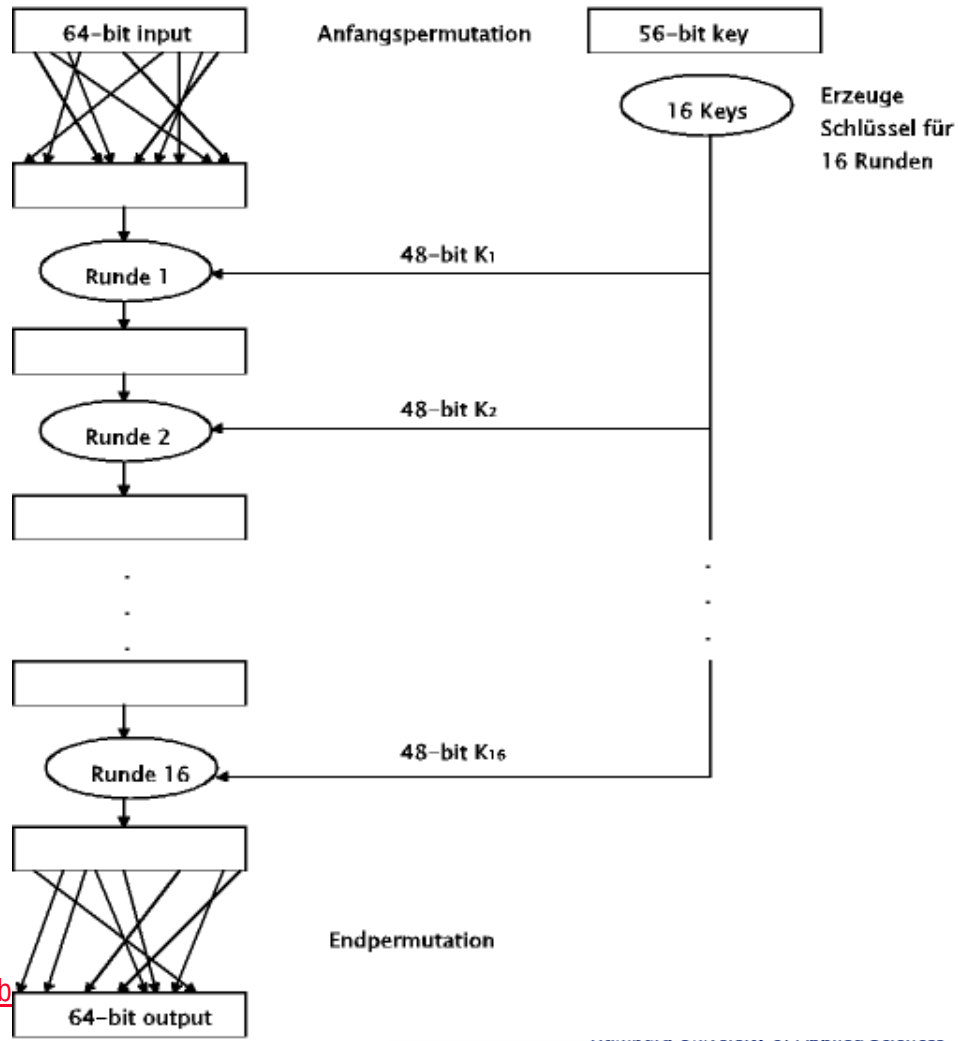

Public Key: public execution of key exchange
- asymmetric method can exchange in the clear

Private Key: secret key needs 'out of band' installation
- symmetric method needs pre-shared secret

# Symmetric Encryption

- Private key method
- Classical, high performance
- Key exchange at runtime
- Needed: exchange of initial seed (out of band)
- Problem:
  No method for signature
- Authentication:
  Challenge-Response-Scheme

Example: DES



# Asymmetric Encryption

- Public key method (Diffie/Hellman 1976)
- Calculations numerically complex (long keys!)
- Separate key generation
- Public key exchange
- External key certification by Certification Authorities (CAs)
- Permits sender authentication

## RSA-Algorithm

**p, q large prime number, n= p * q**

**let e, d and k with**

$$e*d = k*(p-1)*(q-1)+1$$

**Number Theory: for every m**

$$(m^{**}e)^{**}d \bmod n = m$$

**m: message to send**
**e: Encryptor (public key)**
**d: Decryptor (private key)**

# Key Agreement: Diffie-Hellmann

Problem: Two mutually unknown parties (A & B) want to exchange an encryption key via a public data channel

Approach: Use public key cryptography to spontaneously establish a shared secret key.

Method: Diffie-Hellmann “New Directions in Cryptography” (1976)

Shortcoming: Mutual authentication left open - to public key infrastructure or off-channel solution

# Diffie-Hellmann Algorithm

Let $p$ be a sufficiently large prime,

$$g : g^n \bmod p = p \text{ for some } n,$$

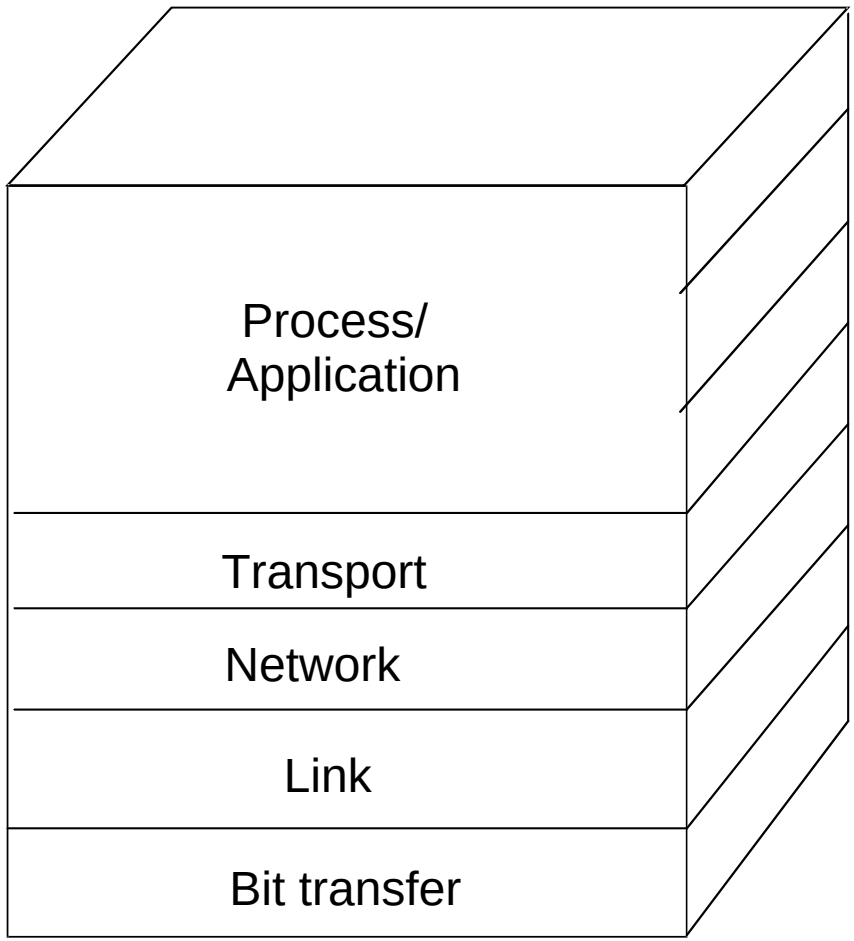
$p$ and $g$ publicly available.

Then:

1. A chooses $0 \leq a \leq p-2$ at random and sends $c := g^a$ to B
2. B chooses $0 \leq b \leq p-2$ at random and sends $d := g^b$ to A
3. A computes the shared key $k = d^a = (g^b)^a$
4. B computes the shared key $k = c^b = (g^a)^b$

The strength of the algorithm relies on the secrets $a$ and $b$, which are discrete logarithms $\bmod p$

# Layers of Encryption




Layer 7: Application encryption

Layer 4+: Socket layer security

Layer 3: Network encryption

Layer 2: Logic tunnelling

Layer 1: Line encryption

# Application Layer

Example: Pretty Good Privacy (Mail)

Advantage:

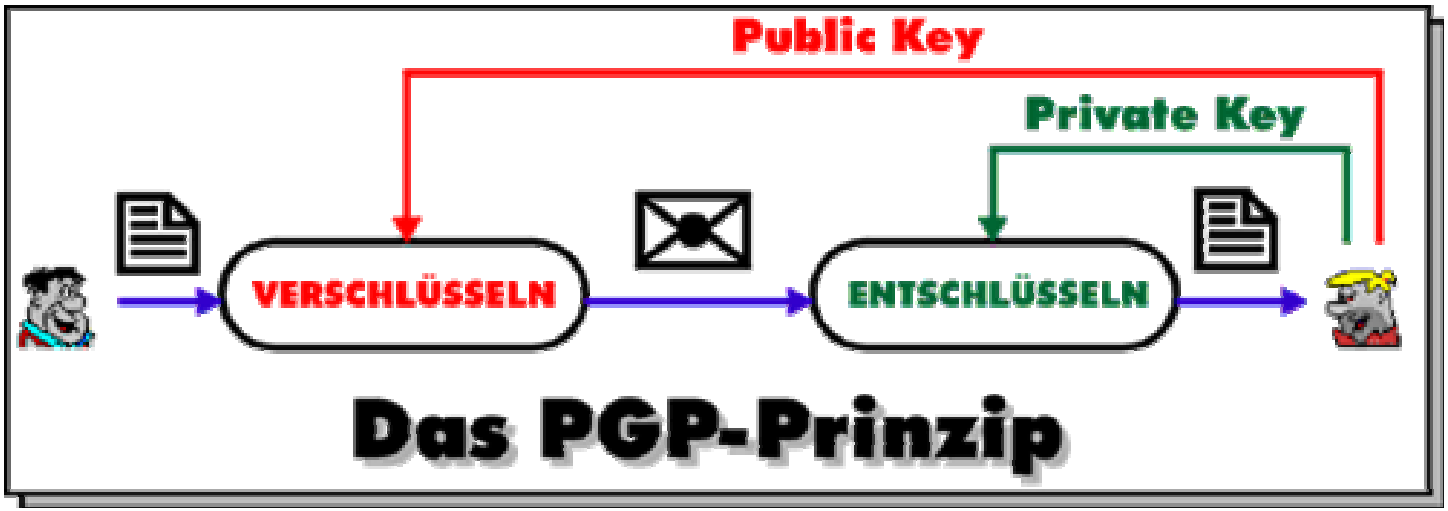
- serves all purposes
- Inter-application security model
- application specifically optimized

Disadvantage:

- Communication profiles remain visible on application layer
- Needs application programs for provisioning



# Example: Pretty Good Privacy




- Public key based:
  Fred encrypts his message with the public key of Barney.
- For authentication Fred appends a ‚signature' at his mail.
- Only Barney can decrypt the content of this mail.
- Barney decrypts the signature with the public key of Fred.

# Socket Layer (4+)

Example: Secure Socket Layer (SSL/TLS)

Advantage:

- end-to-end security model
- transparent w.r.t application data
- easy to integrate (secure socket library)

Disadvantage:

- Communication profiles remain visible on the transport layer (incl. appl. protocol)
- Needs incorporation into application programs

# Example: SSL/TLS

- Transport Layer Security: RFC 2246, 3546
- Protocol for encrypted transfer between unknown clients and known servers (approved by certification).
- Public key based session-initiation:
  on request server sends public key to a client.
- Client generates a pre-shared secret (private key) and sends this with the received public key encrypted to the server.
- Communication afterwards will be encrypted symmetrically .

# Line Encryption (L 1)

Example: Transmission-Scrambling, WEP

Advantage:

- complete information encryption
- completely transparent

Disadvantage:

- bound to line, not end-to-end
- normally requires hardware support

# Example: WEP

- Protocol for encrypting wireless transmission between Access Point and Stations.
- Private key based: AP & STA hold pre-shared secret.
  - Fixed length: 40 or 104 bits
  - Static: no key exchange, except by reconfiguration
- Authentication: Challenge (AP) – Response (STA) scheme.
- Encryption: RC4 encryption (XOR with pseudorandom stream) with (insufficiently changed) Initialisation Vectors (IV).
- Improvement: WPA – the upgrade to Temporal Key Integrity Protocol (TKIP) – a deficit healing by improved IV selection and re-keying.

# Layer 2: MAC Protection + Tunnels

Examples:

- MAC Protection: ACLs, 802.1x port authentication
- Tunnels: PPP/PPTP, L2TP (+encryption), ...

Advantage:

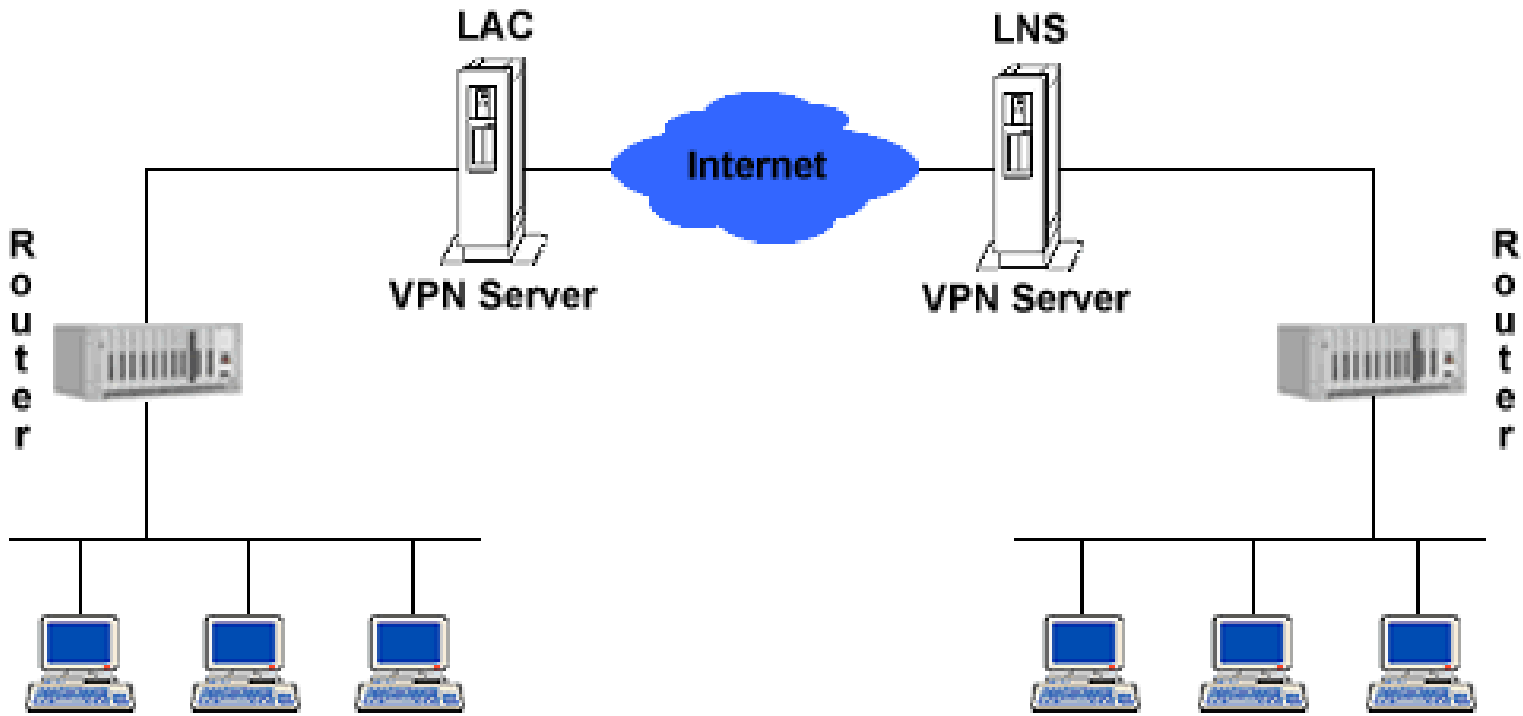
- prevents ARP spoofing + network intrusion
- transparent to network layer (only tunnel visible)

Disadvantage:

- needs server / provider support
- limited scaling / performance

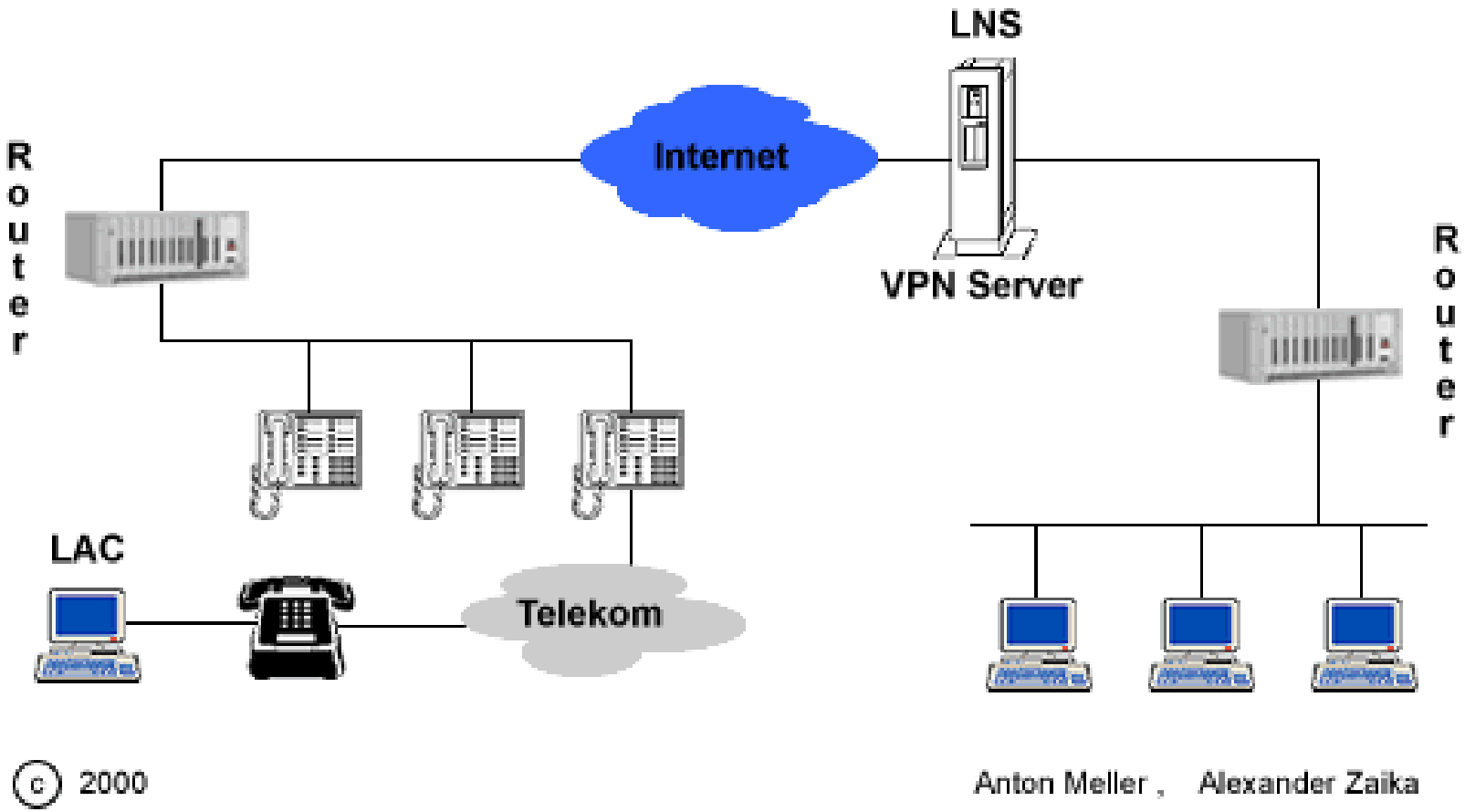
# Compulsory Tunnel (Carrier / ISP Model)

IP(Message)
PPP(IP(Message))
PPP(IP'(L2TP(××××××××××)))



# Voluntary Tunnel (Client Model)

IP(Message)
PPP(IP(Message))
PPP(IP'(L2TP(××××××××××)))



# Internet Layer (IP)

Example: packet encryption, address authentification

Advantage:

- transport transparent
- efficient & wide-area routable

Disadvantage:

- communication profile visible on IP layer

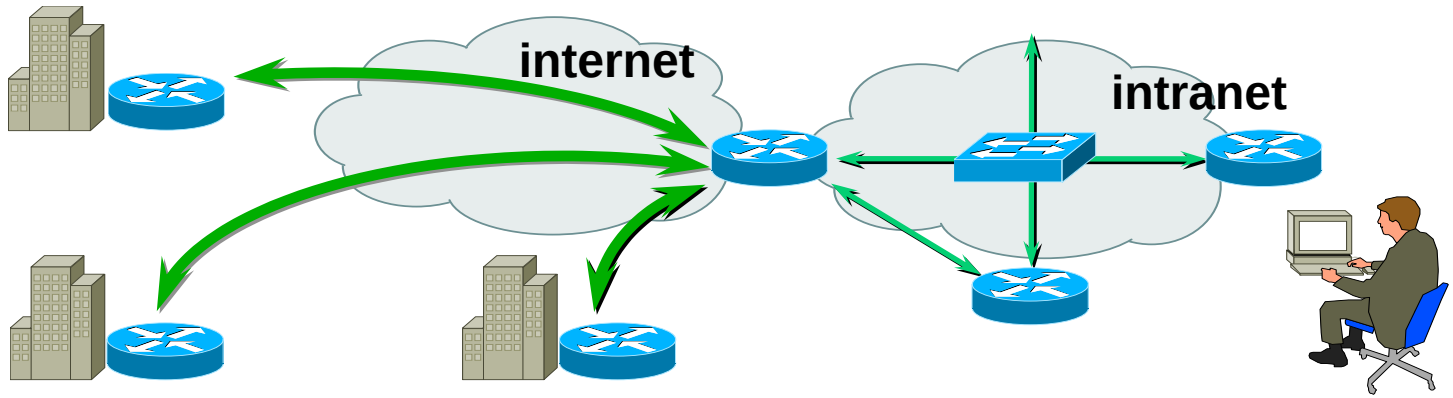
Solution: IP-in-IP secure tunnelling: IPSec

# What is IPSec?

- A security architecture
- Two IP security protocols
  - Authentication Header (AH)
  - Encapsulation Security Payload (ESP)
- Internet Key Exchange (IKE)
  - Exchange of IPSec security seeds
- An open standard (RFC 2401, 4301)

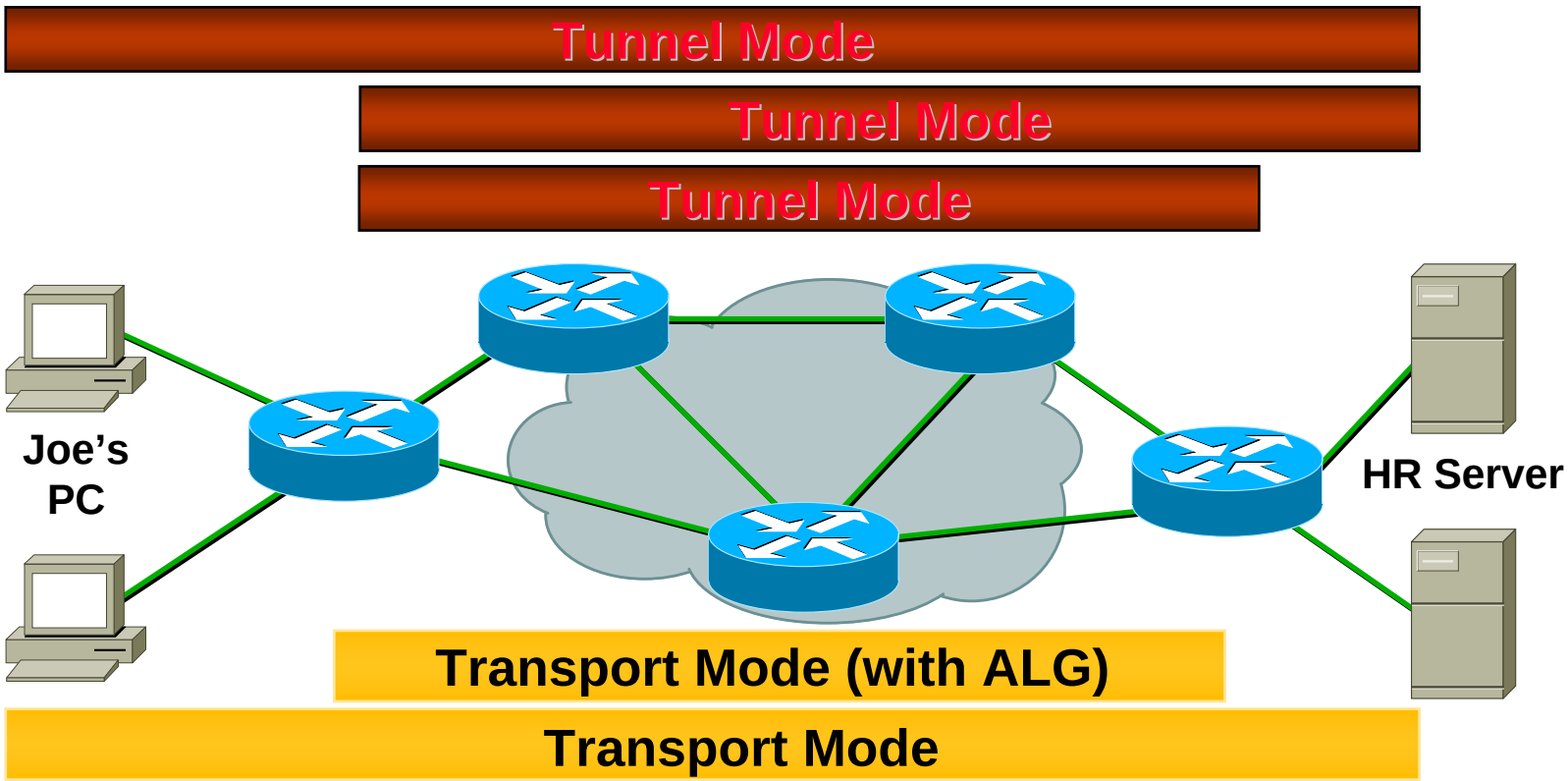
⇒ **An end-to-end security solution on the IP layer**

# Concepts of IPSec



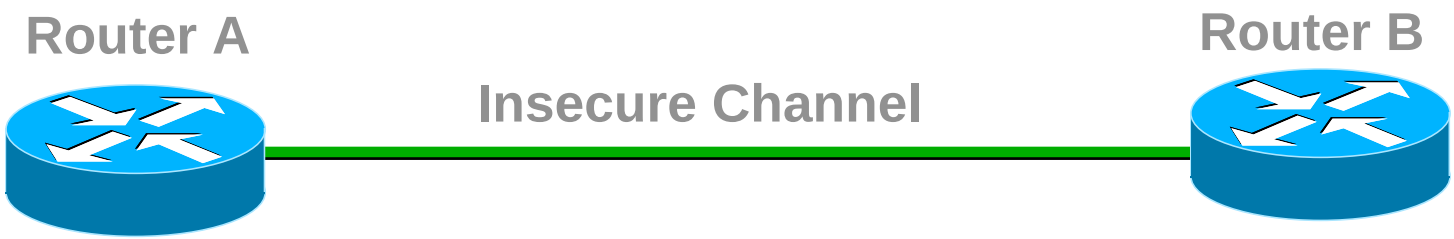

- Protects data transfers throughout the Internet, procuring
  Authentication, Integrity, Encryption
- Transparent to, but compliant with network infrastructure
- End-to-end concept

# Tunnel and Transport Mode




- Transport Mode End-to-End or via ALG
- Tunnel Mode for all connection types

# Security Association (SA)

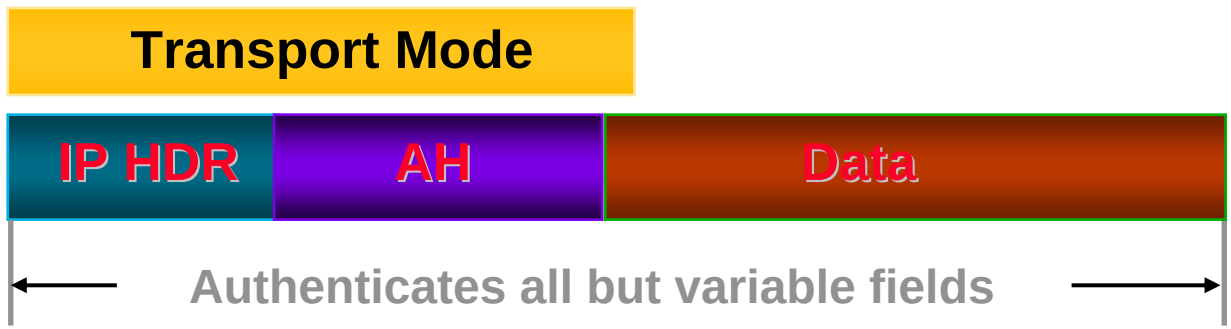



- Directional description of security services in use (unidirectional per connection)
- Valid for individual data flow
- Two-way communication uses two SAs
- Each SA identified by a Security Parameter Index (SPI)
  - as part of the IPSec Headers
  - number with strictly local scope

# Security Association (2)

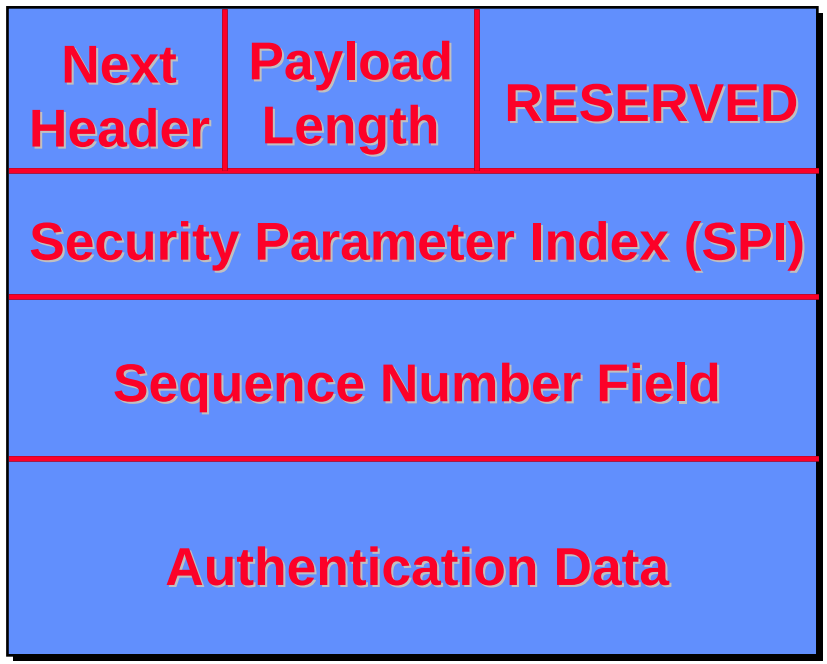
| | |
|---|---|
| **Destination Address** | **205.49.54.237** |
| **Security Parameter Index (SPI)** | **7A390BC1** |
| **IPSec Transform** | **AH, HMAC-MD5** |
| **Key** | **7572CA49F7632946** |
| ***Additional SA Attributes (e.g. lifetime)*** | **One Day or 100MB** |

# IPSec Authentication Header (AH)

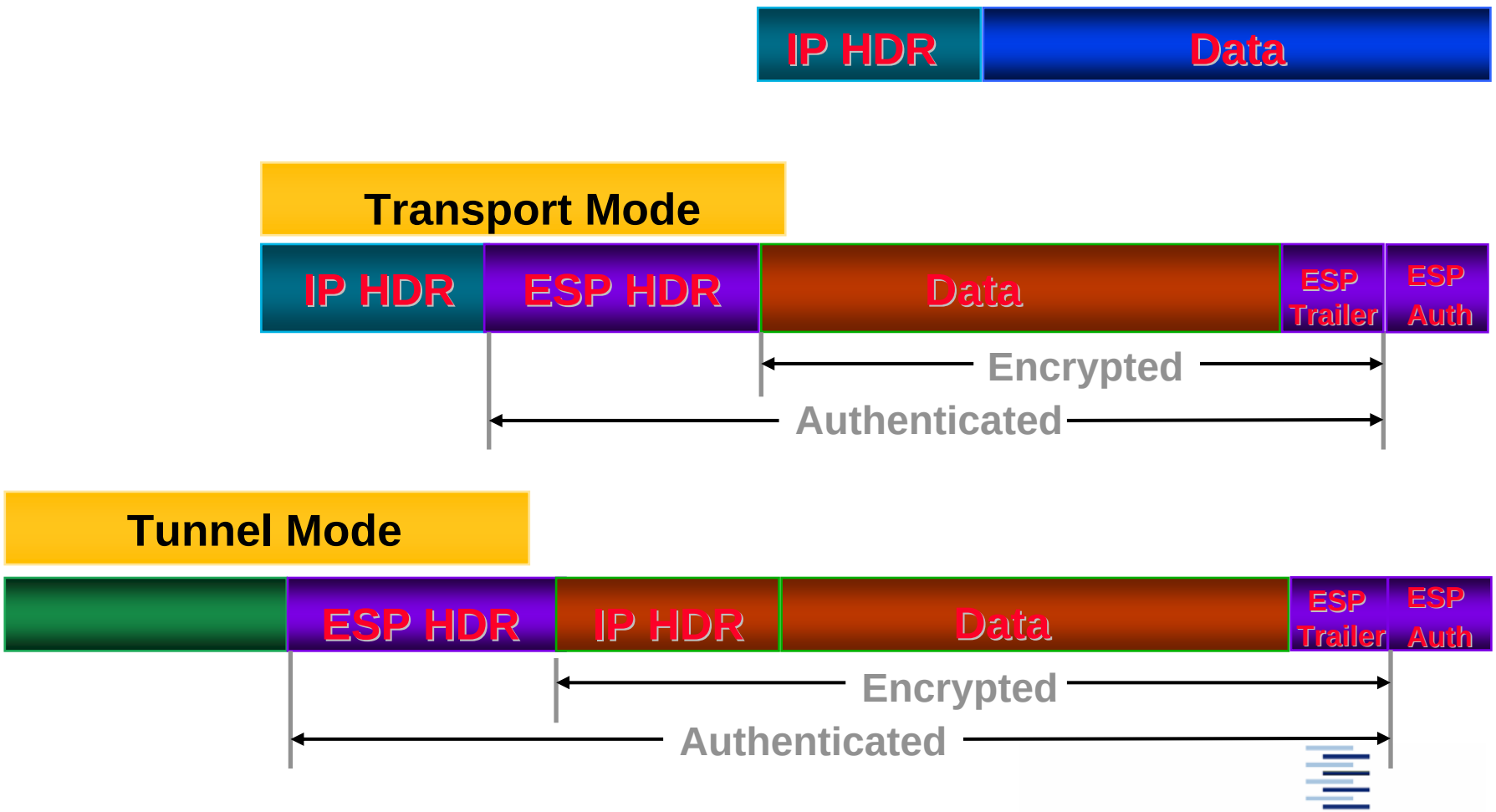




# Authentication Header (2)

- Authentication header is extension Header (IPv6).
- IPv4: placed prior to TCP/UDP header (change of IPv4-Stack) or as transport payload (lower efficiency)
- Authenticates data source and integrity by a Message Authentication Code (MAC).
- Remains unencrypted.

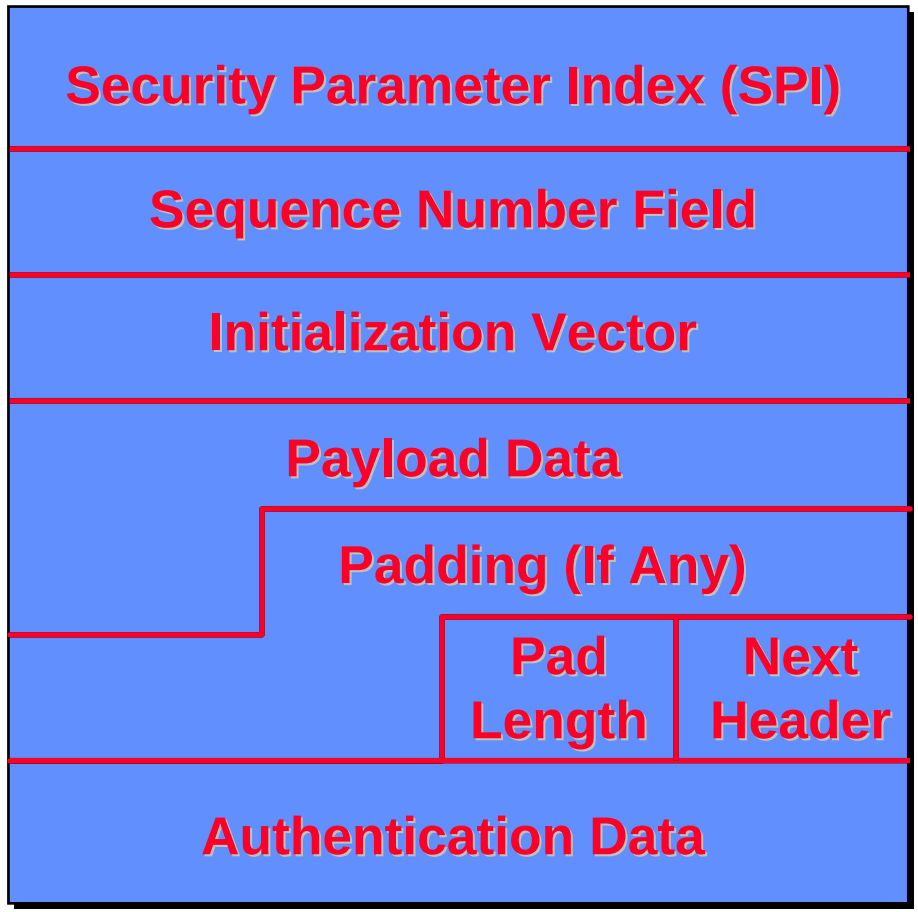


# Encapsulating Security Payload (ESP)



# Encapsulating Security Payload

- ESP Header handling as AH
- Ready to include encryption parameters (initialisation)
- ESP Header remains unencrypted, but authenticated with data
- ESP payload encrypted
- Trailer for terminating 0s and alignment.



# Encryption Methods

- IPSec can employ different encryption methods.
- To initiate a Security Association either a Public Key Infrastructure (PKI) or Preshared Secrets (offline) are needed.
- While an SA is running, data will be encrypted via symmetric encryption methods (performance).
- To regularly exchange keys an Internet Key Exchange Daemon is part of the IPSec concept.

# Internet Key Exchange (IKE)

- IKE-Protocol (RFC 2409, v2: RFC 4306) implements different key exchange schemes.
- Negotiates policies to use.
- Negotiates SAs to initiate IPSec.
- Modi: Main, Aggressive, Quick and New Group (depending on identification model).
- Authenticated Diffie-Hellman key exchange.

# Operation of IPSec

**Notable Traffic?**

↓

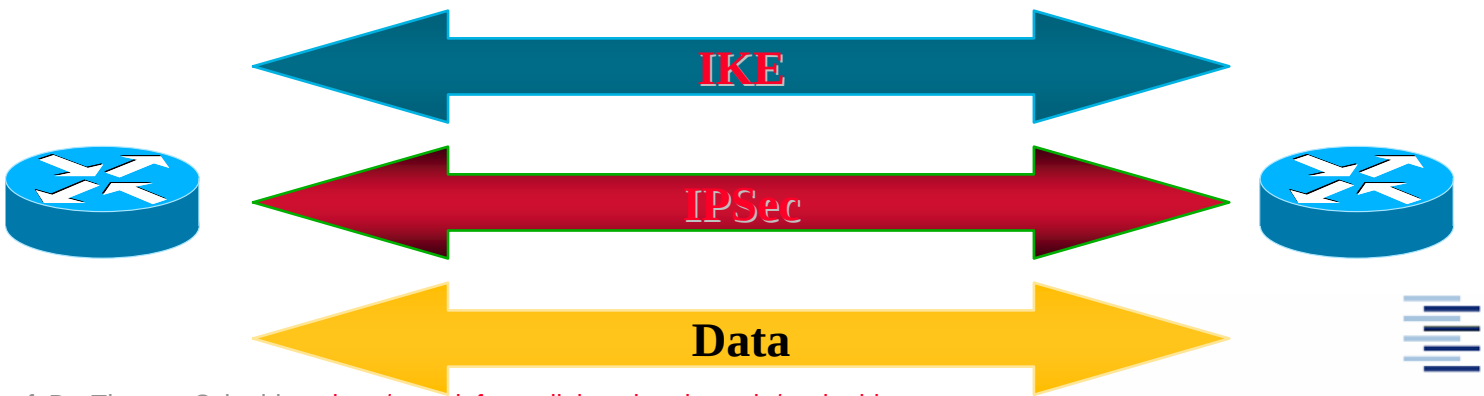
**IKE Negociation**

↓

**IPSec Negociation**

↓

**Tunnel Construction**



# IKE Initiation of a SA



**SA Request IPSec (triggered by ACL)**

**IKE SA Offer - des, sha, rsa sig, D-H group 1, lifetime**

**In the Clear**

**Policy Match accept offer**

**ISAKMP Phase 1 Oakley Main Mode**

**Fred D-H exchange : KE, nonce**

**Wilma D-H exchange : KE, nonce**

**Fred Authenticate D-H apply Hash**

**Wilma Authenticate D-H apply Hash**

**Protected**

**IKE Bi-directional SA Established**

# IPSec Constructing the SA



**IPSec SA Offer - transform, mode, pfs, authentication, lifetime**

**Policy Match accept offer**

**ISAKMP Phase 2 Oakley Quick Mode**

**Fred D-H exchange or refresh IKE key**

**Wilma D-H exchange or refresh IKE key**

**Protected by the IKE SA**

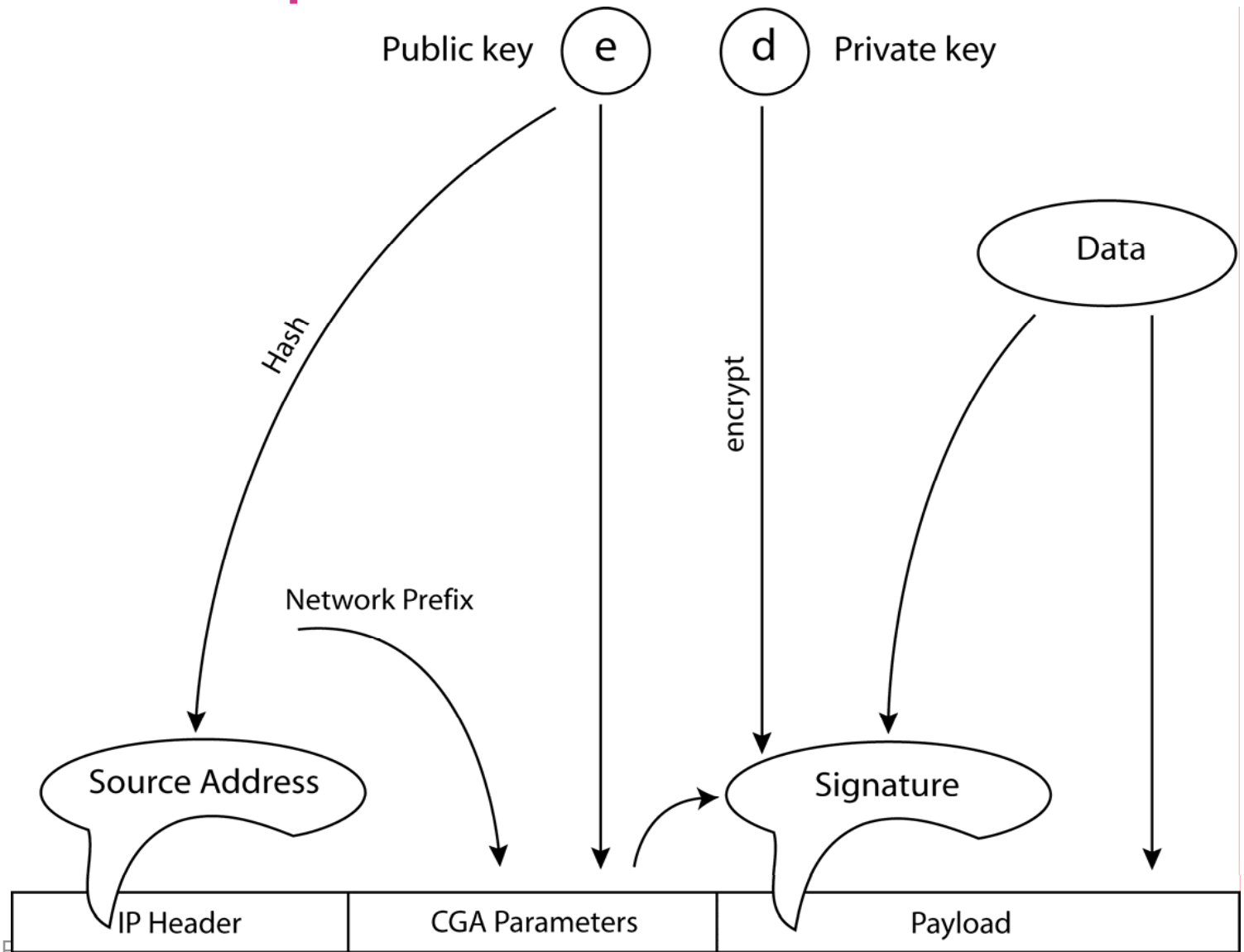
**IPSec Outbound SA Established**
**IPSec Inbound SA Established**

# Example 2: Cryptographically Generated Addresses (IPv6)

Problem: In IP infrastructure protocols the sender of a message frequently has to prove its 'ownership of address' to a receiver, it never met before. Authentication between unknown partners normally requires a public key infrastructure.

- Cryptographically Generated Addresses (CGAs) are source addresses formed from the public key (RFC 3972).
- This mechanism allows the authentication of sender's address and the signing of data without a PKI.
- Most important uses: SEND (RFC 3971), OMIPv6 (IEdraft)

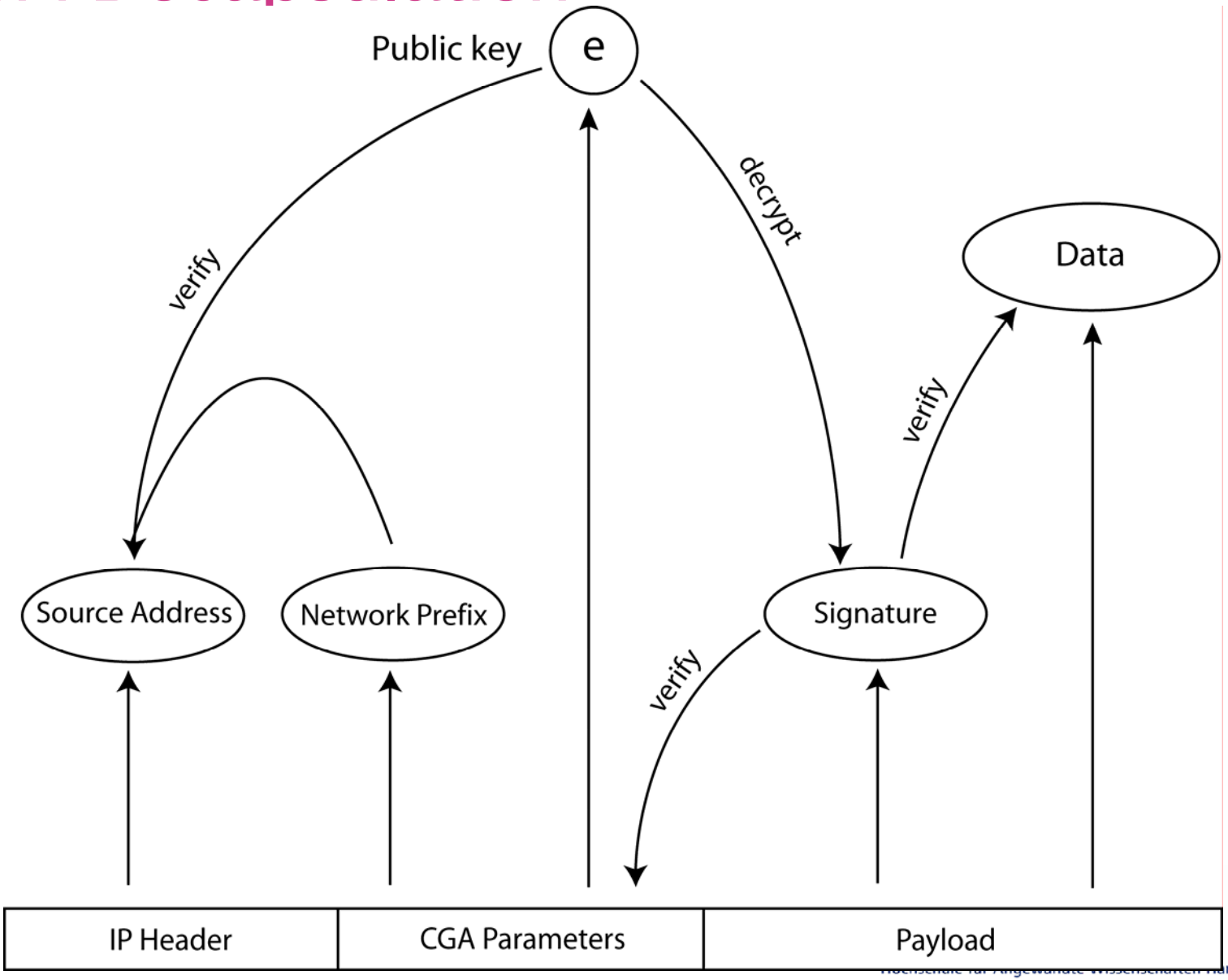
# CGA Encapsulation



# CGA: Encapsulation Steps

1. Sender forms public/private key pair *e* and *d,* calculates (node-)source address as a 64 bit hash from *e.*
2. Sender computes signature of network prefix, public key *e,* data *...* encrypted with its private key *d.*
3. Sender includes (unencrypted) network prefix, *e* and the signature in a CGA parameter header within the packet.
4. Sender adds data and sends packet.

# CGA Decapsulation



# CGA: Decapsulation Steps

1. Receiver extracts network prefix and the public key e from the CGA parameter header.
2. It decrypts the signature with the public key e and verifies the CGA parameters + Data.
3. Receiver re-calculates and verifies sender's source address as a 64 bit hash from e.
4. The receiver can now be sure, that the received packet has been originally sent by the owner of the claimed IP address.

# Summary

- Security in the net can be improved on many layers
- Final selection of a technology needs a careful need analysis
- The level of security achieved is determined by concept / algorithms, key strength and the management quality
- There is no such thing as ‘secure’ (only more secure)

# References


- William Stallings: *Cryptography and Network Security*, 3rd Ed., Prentice Hall, 2003.
- John Edney, William A. Arbaugh: *Real 802.11 Security,* Addison-Wesley, 2004.
- Hans Delfs, Hartmut Knebl: *Introduction to Cryptography*, Springer, 2002.
- Claudia Eckert: *IT Sicherheit*, 4th Ed., Oldenbourg Verlag, 2006.
- Internet Standards at: www.rfc-editor.org.