Practical Network Defense

Master's degree in Cybersecurity 2020-21

VPN, SSL/TLS and IPSec

Angelo Spognardi
spognardi@di.uniroma1.it
Dipartimento di Informatica
Sapienza Università di Roma

Today's agenda

- VPN principles
- SSL Tunneling
- VPN device placement
- IPsec

VPN principles

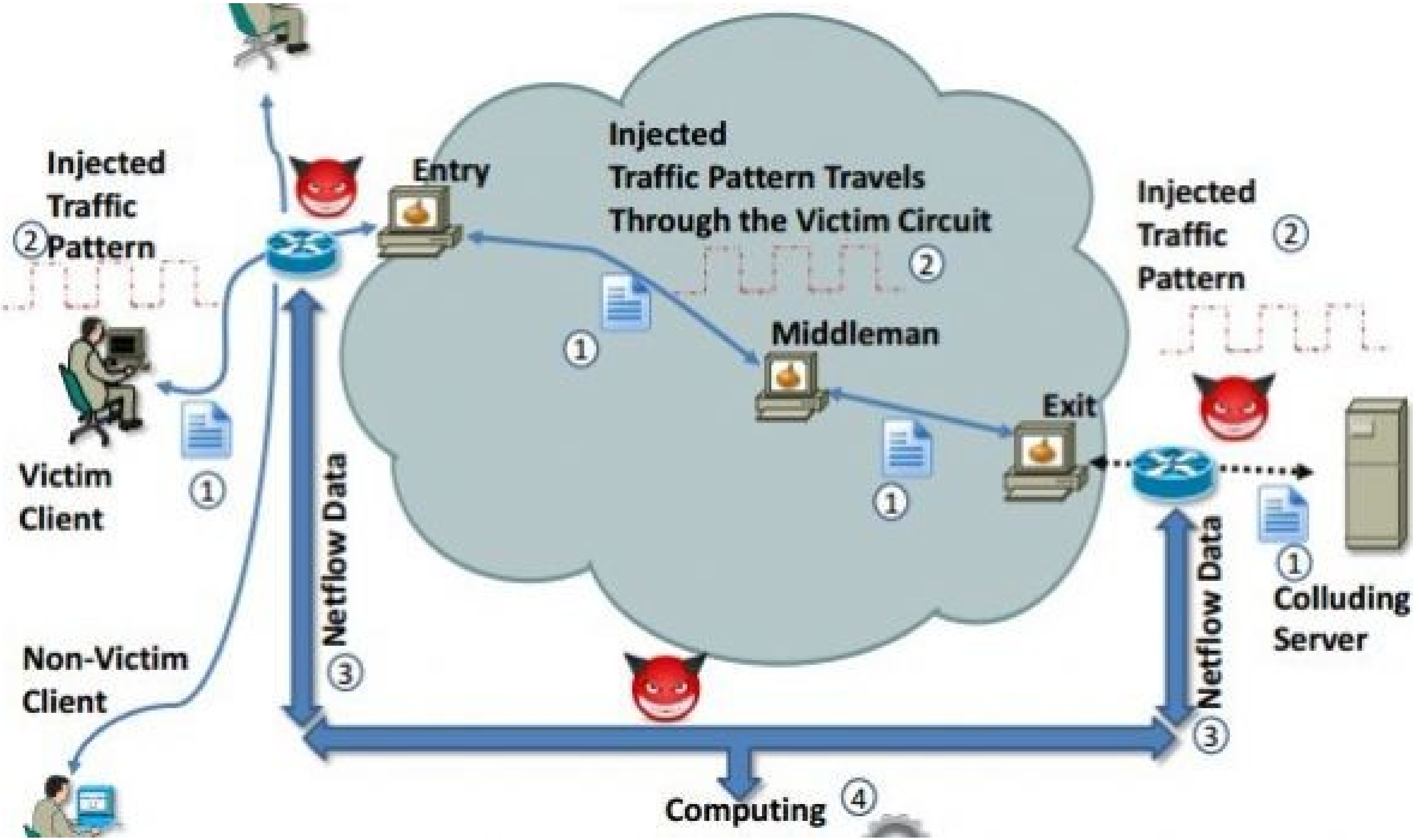
Virtual Private Networks

- Definition (NIST SP800-113): A virtual network, built on top of an existing network infrastructure, which can provide a secure communications mechanism for data and other information transferred between two endpoints
- Typically based on the use of encryption, but several possible choices for:
  - How and where to perform the encryption
  - Which parts of communication should be encrypted
- Important subsidiary goal: usability
  - If a solution is too difficult to use, it will not be used → poor usability leads to no security

Security Goals for a VPN

- Traditional
  - Confidentiality of data
  - Integrity of data
  - Peer Authentication
- Extended
  - Replay Protection
  - Access Control
  - Traffic Analysis Protection

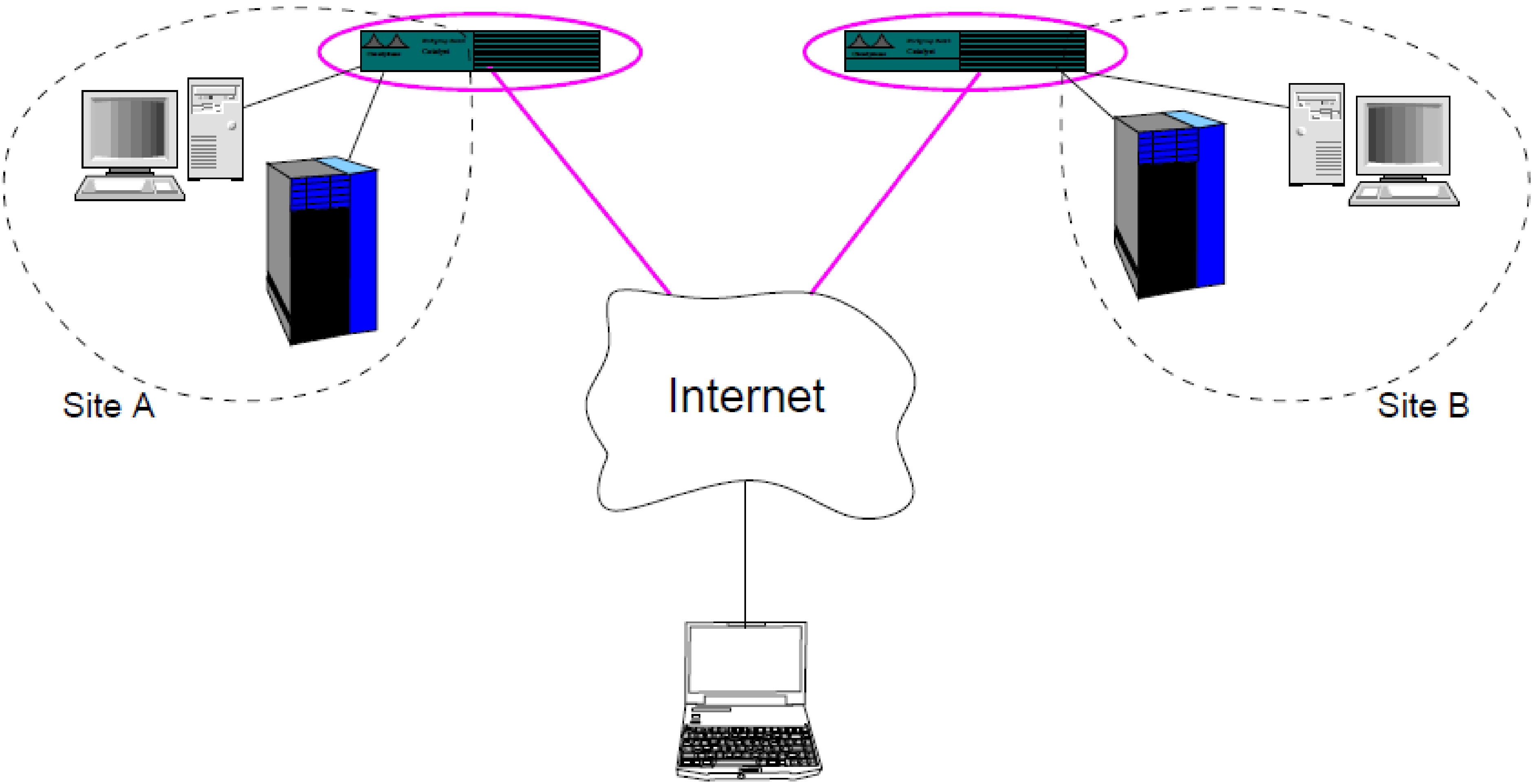
Traffic analysis



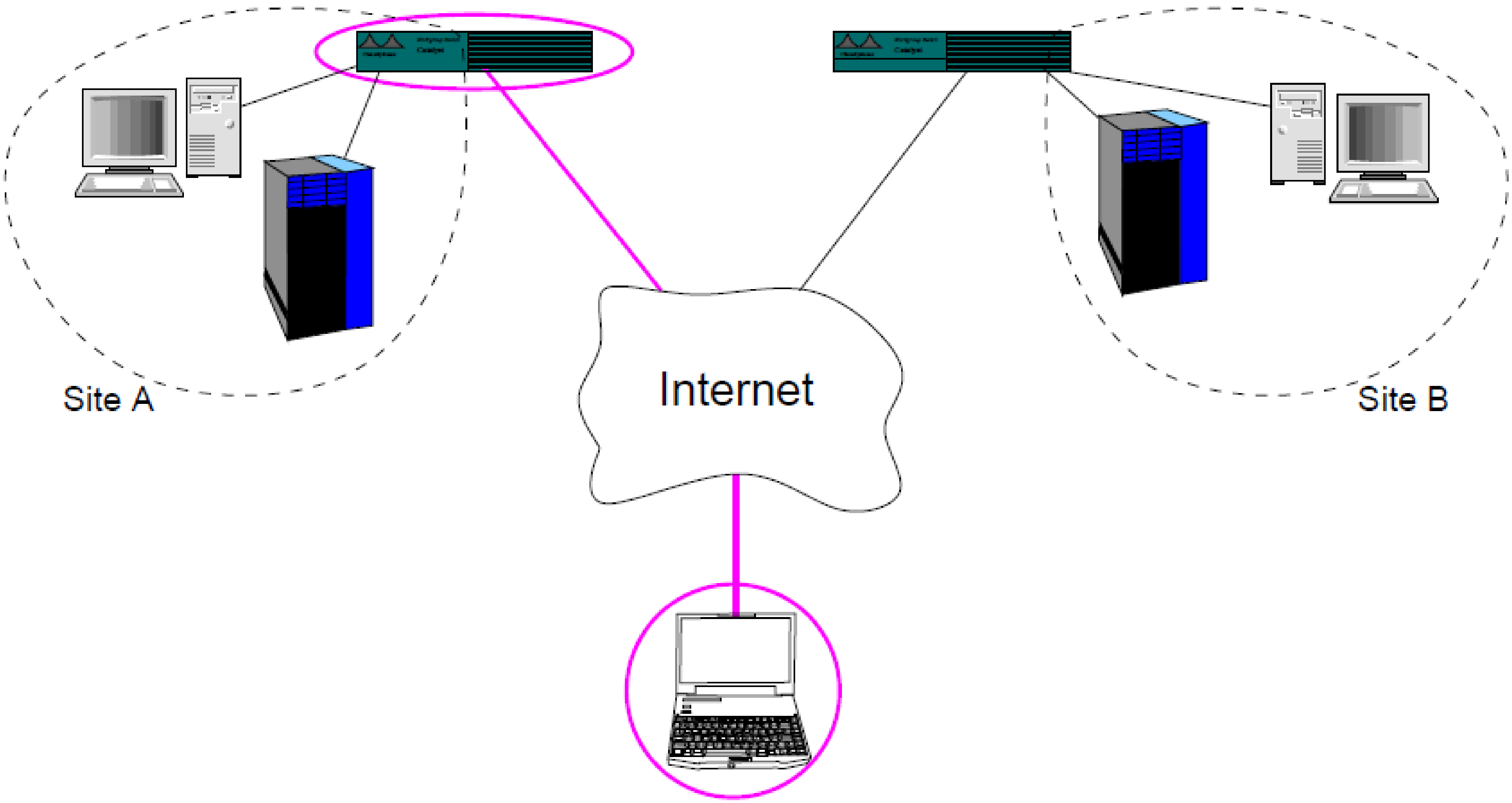
Usability goals

- Transparency
  - VPN should be invisible to users, software, hardware.
- Flexibility
  - VPN can be used between users, applications, hosts, sites.
- Simplicity
  - VPN can be actually used

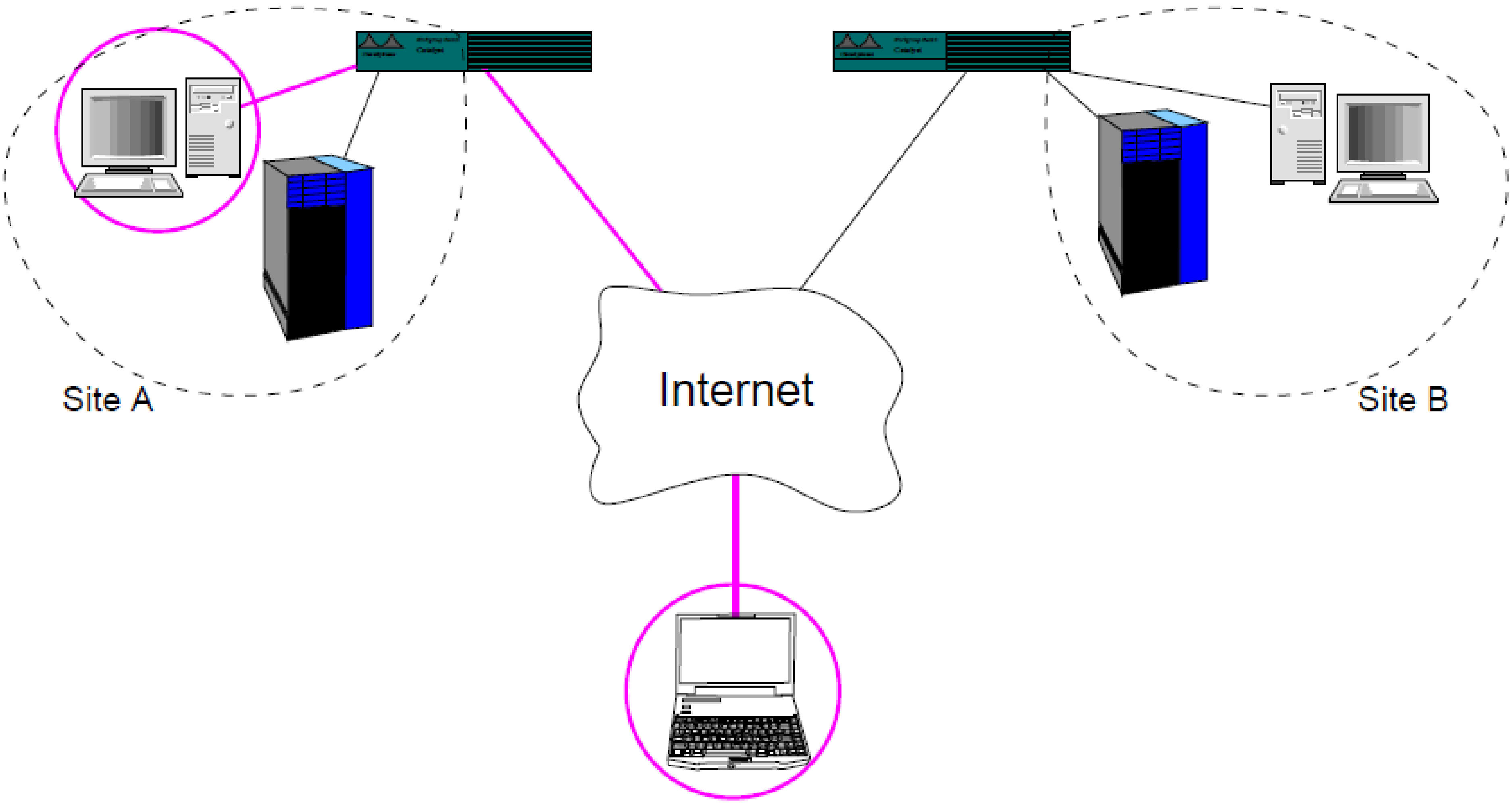
Site-to-site security



Host-to-site security

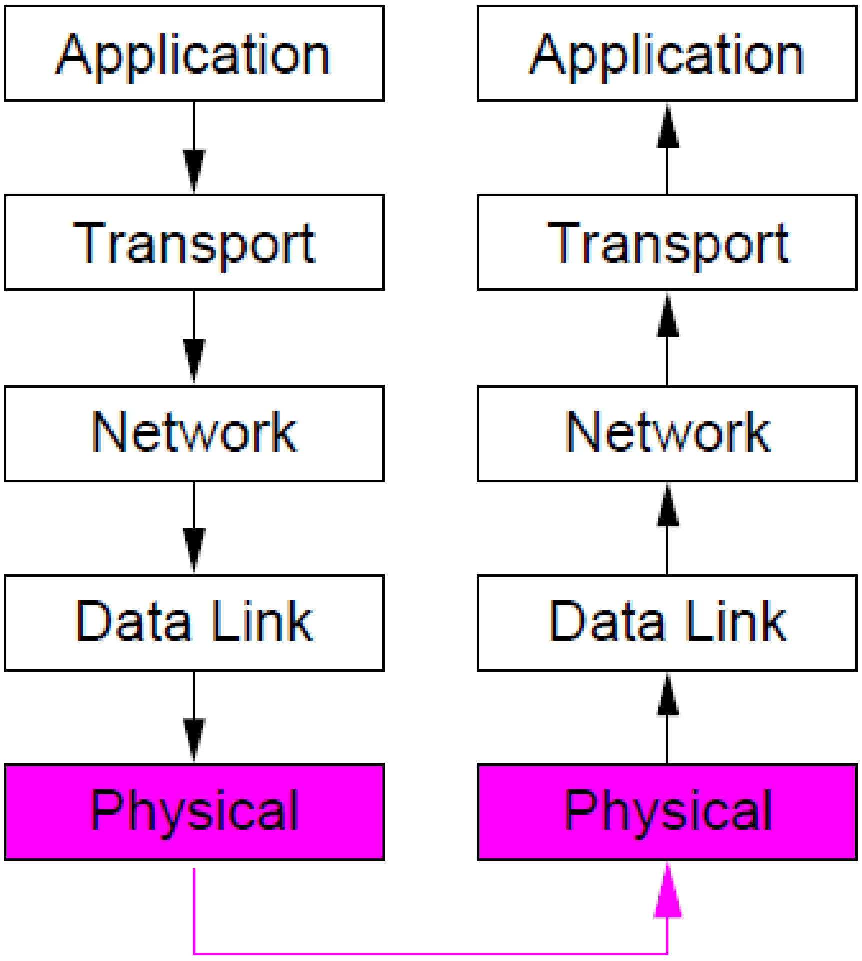


Host-to-host security

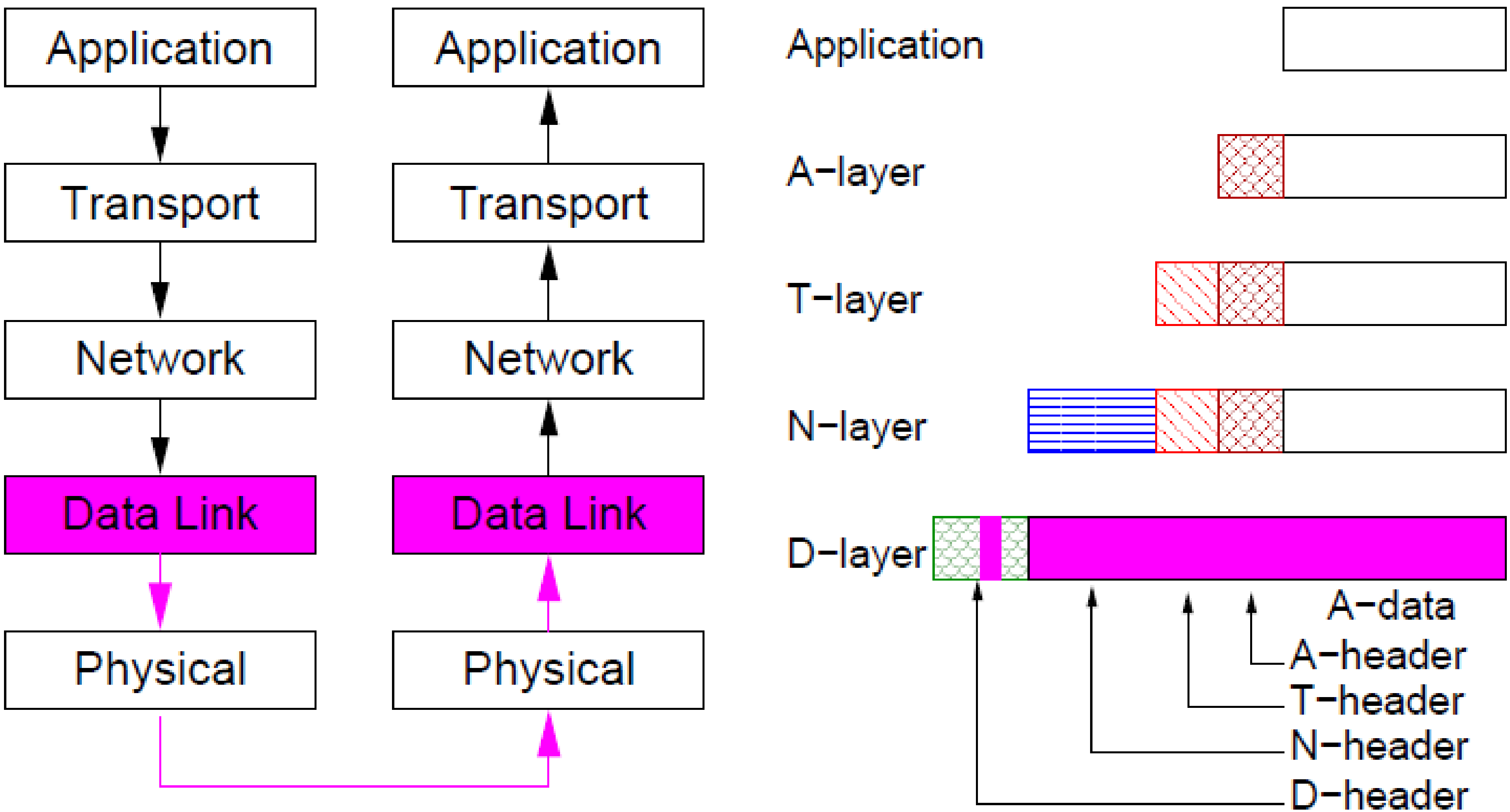


Physical layer

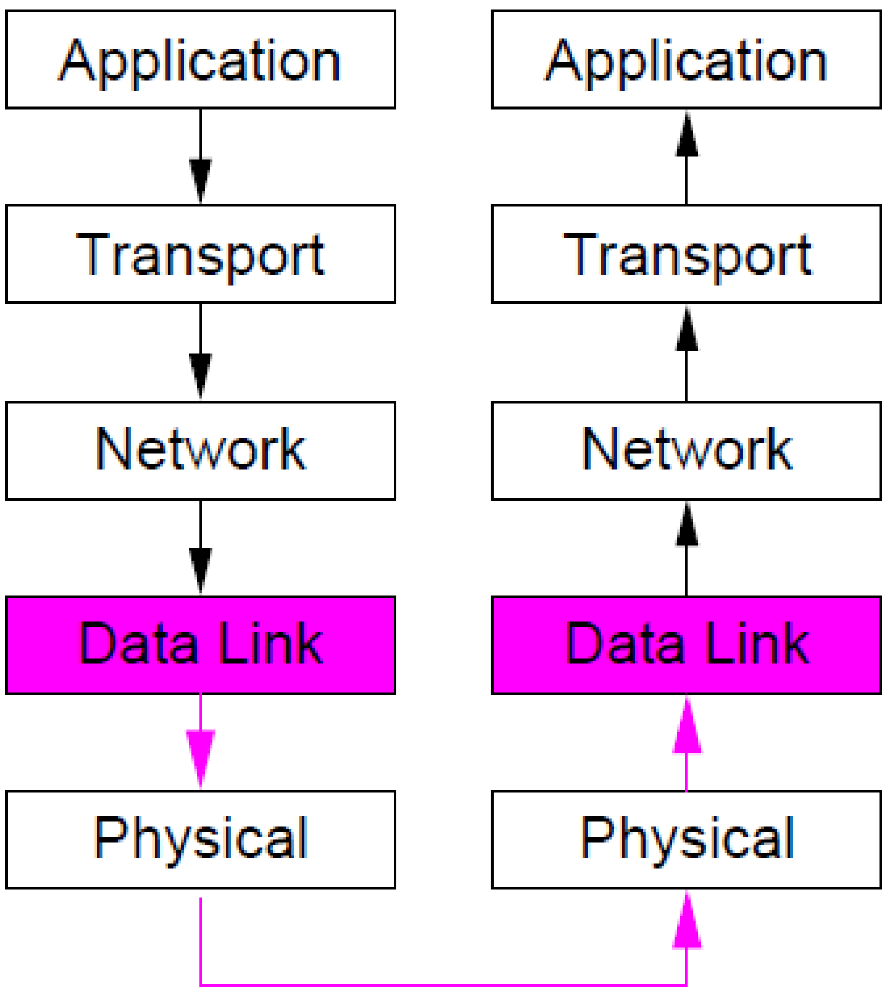
- Confidentiality: on cable
- Integrity: on cable
- Authentication: none
- Replay protection: none
- Traffic analysis protection: on cable
- Access control: physical access
- Transparency: full transparency
- Flexibility: can be hard to add new sites
- Simplicity: excellent!



Datalink layer: protect a single link

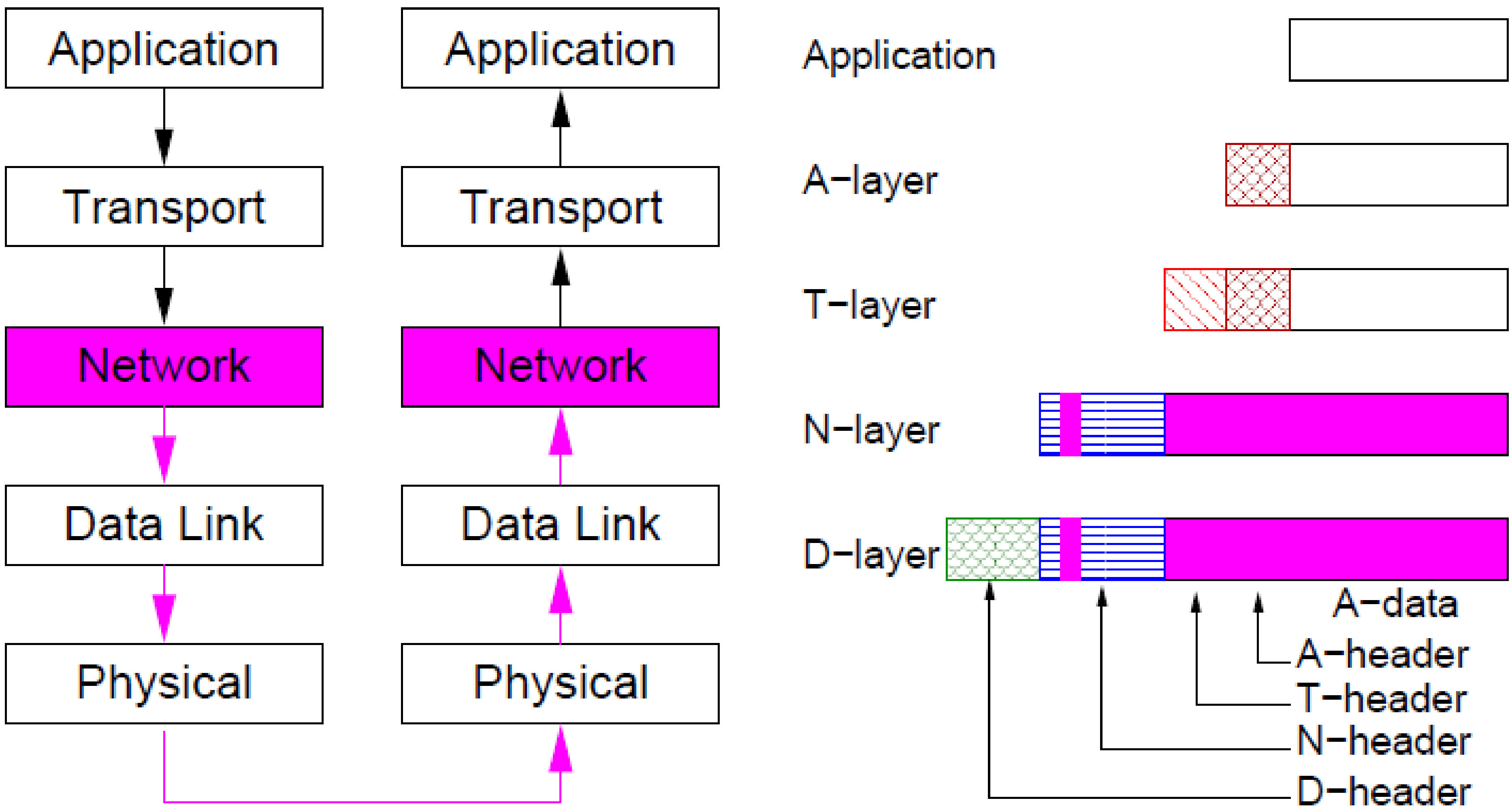


Datalink layer: protect a single link

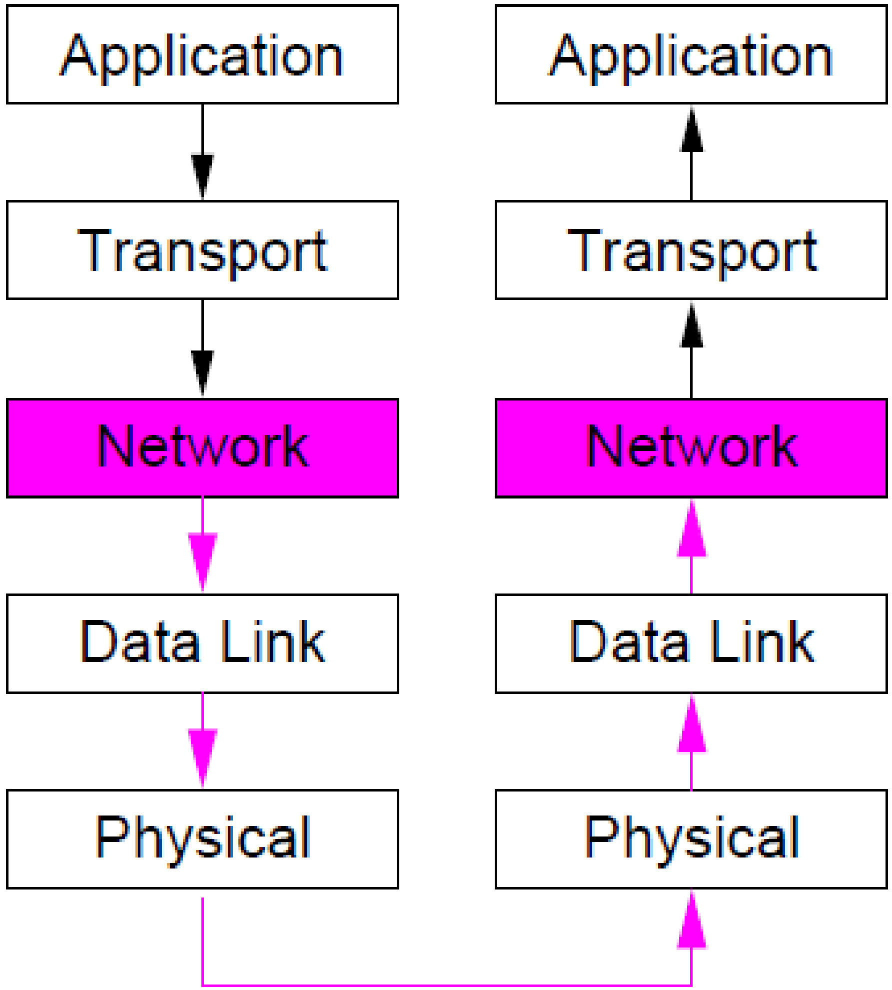


- Confidentiality: on link ("virtual cable")
- Integrity: on link
- Authentication: none
- Replay protection: none
- Traffic analysis protection: on link
- Access control: physical access
- Transparency: full transparency
- Flexibility: can be hard to add new sites
- Simplicity: excellent!

Network layer: protect end-to-end between systems

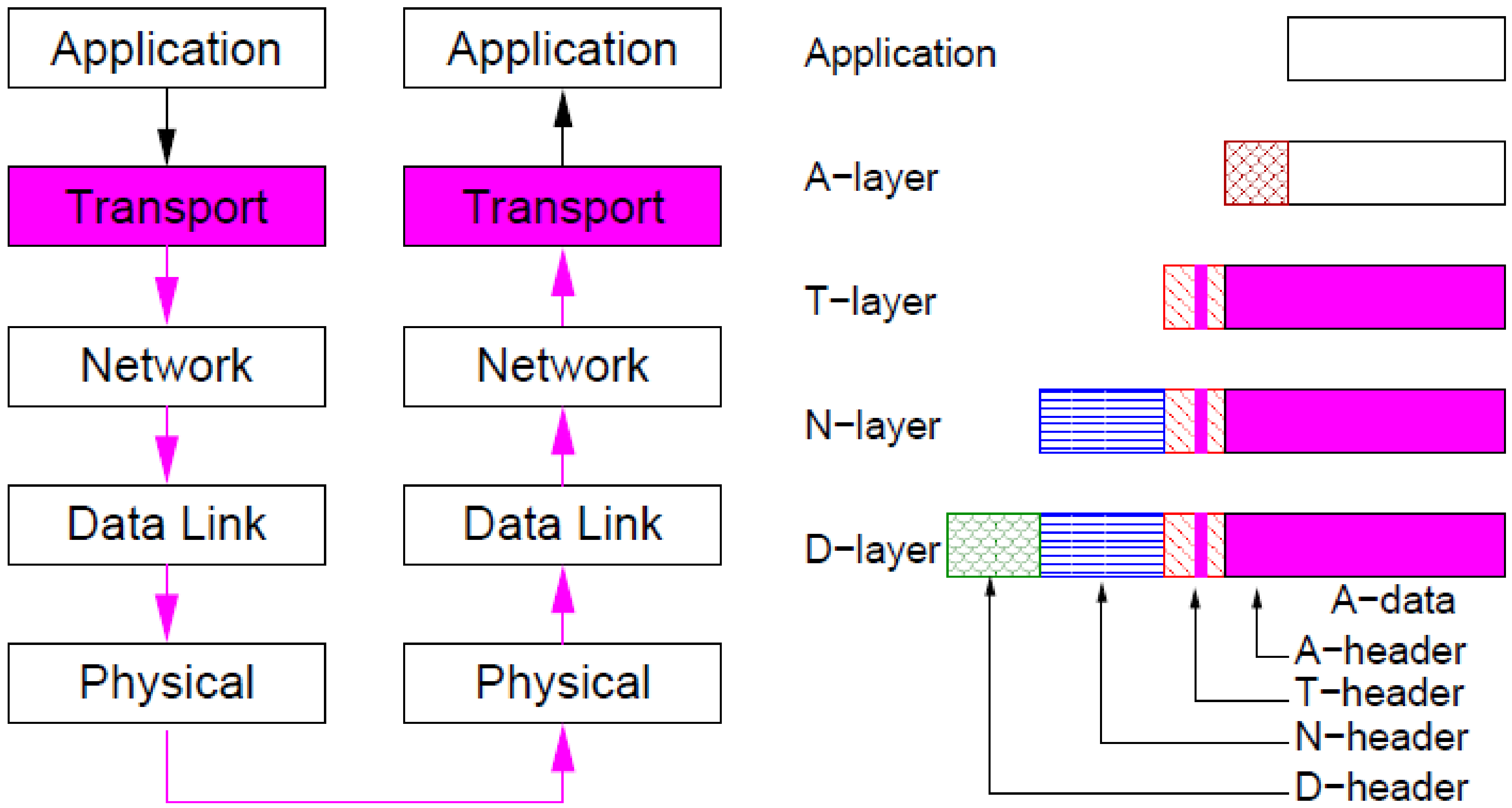


Network layer: protect end-to-end between systems

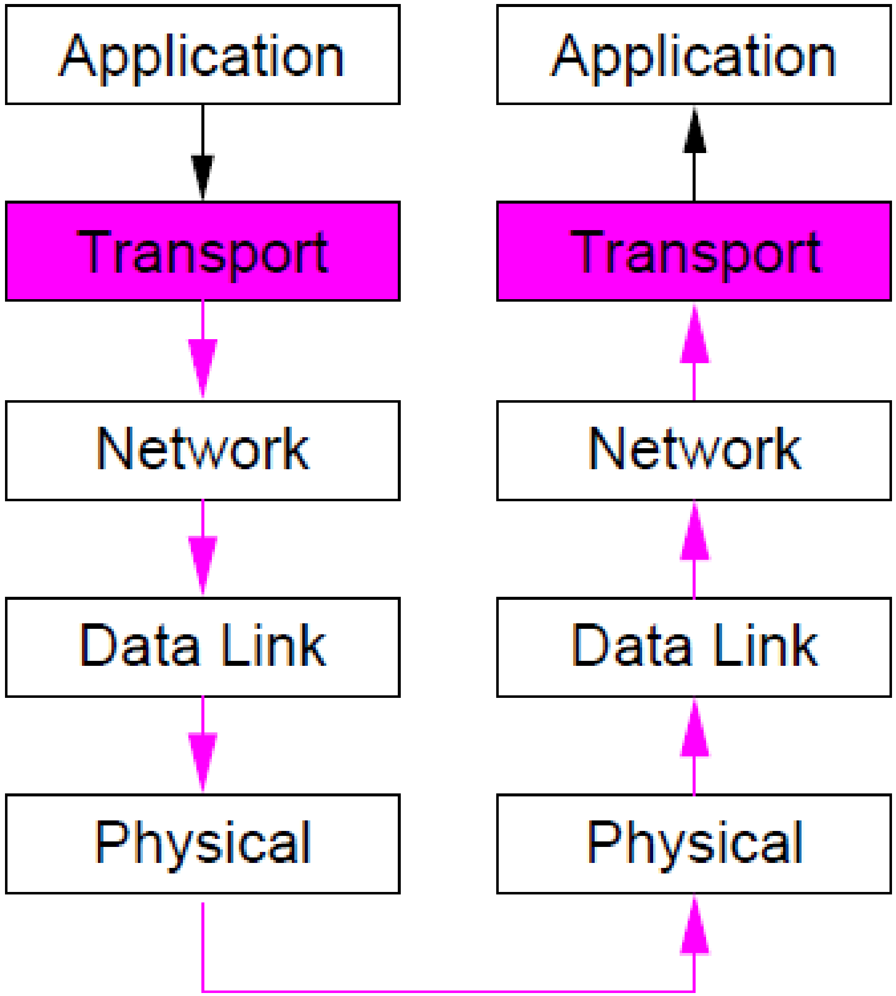


- Confidentiality: between hosts/sites
- Integrity: between hosts/sites
- Authentication: for host or site
- Replay protection: between hosts/sites
- Traffic analysis protection: host/site information exposed
- Access control: to host/site
- Transparency user and SW transparency possible
- Flexibility: may need HW or SW modifications
- Simplicity: good for site-to-site, not good for host-to-site

Transport layer: Protection end-to-end between processes

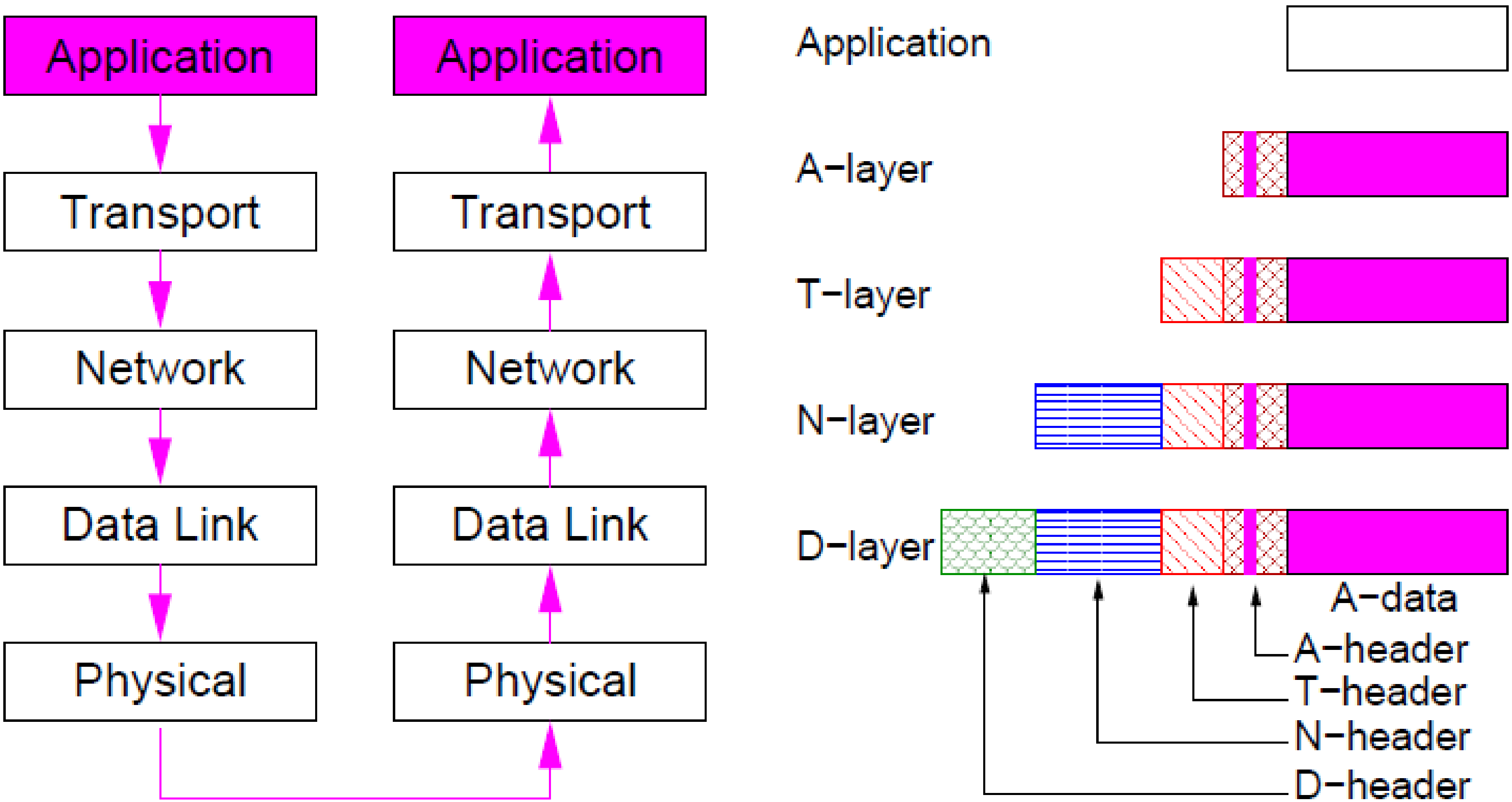


Transport layer: Protection end-to-end between processes

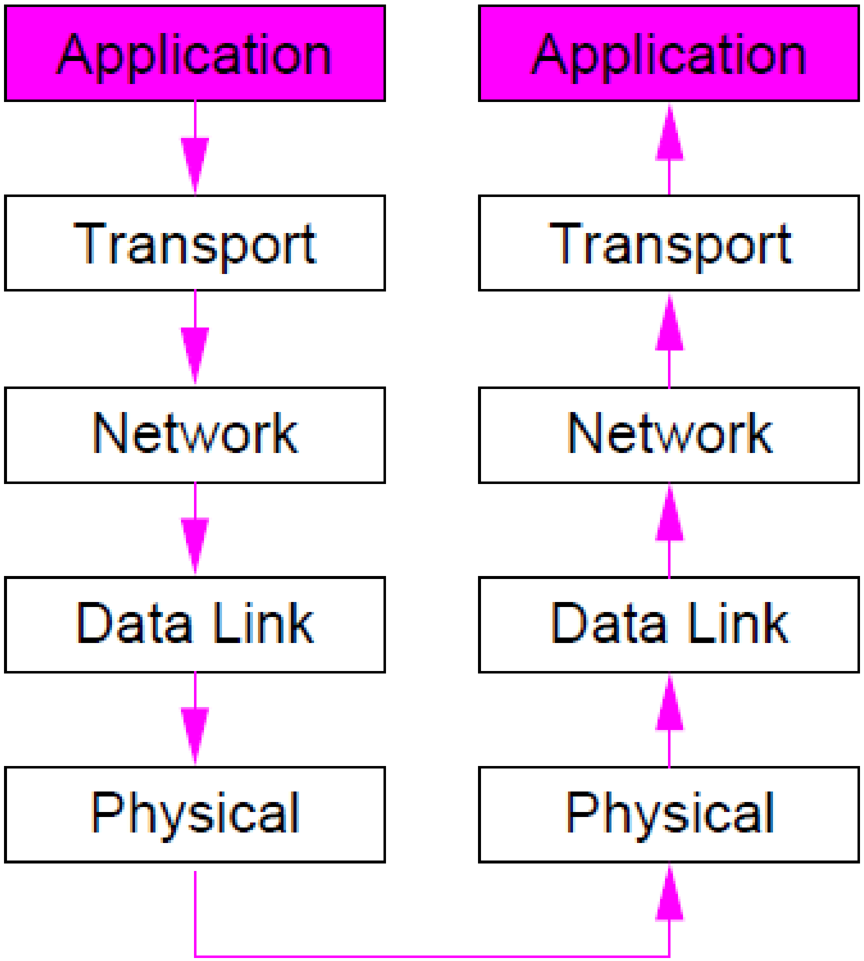


- Confidentiality: between apps/hosts/sites
- Integrity: between apps/hosts/sites
- Authentication: for user, host, site
- Replay protection: between apps/hosts/sites
- Traffic analysis protection: protocol/ host/site info. exposed
- Access control: user/host/site
- Transparency user and SW transparency possible
- Flexibility: HW or SW modifications
- Simplicity: good for site-to-site, not good for host-to-site

Application layer: Security for a single application



Application layer: Security for a single application



- Confidentiality: between users/apps
- Integrity: between users/apps
- Authentication: user
- Replay protection: between apps
- Traffic analysis protection: all but data exposed
- Access control: only data access secured
- Transparency: only user transparency
- Flexibility: SW modifications
- Simplicity depends on application

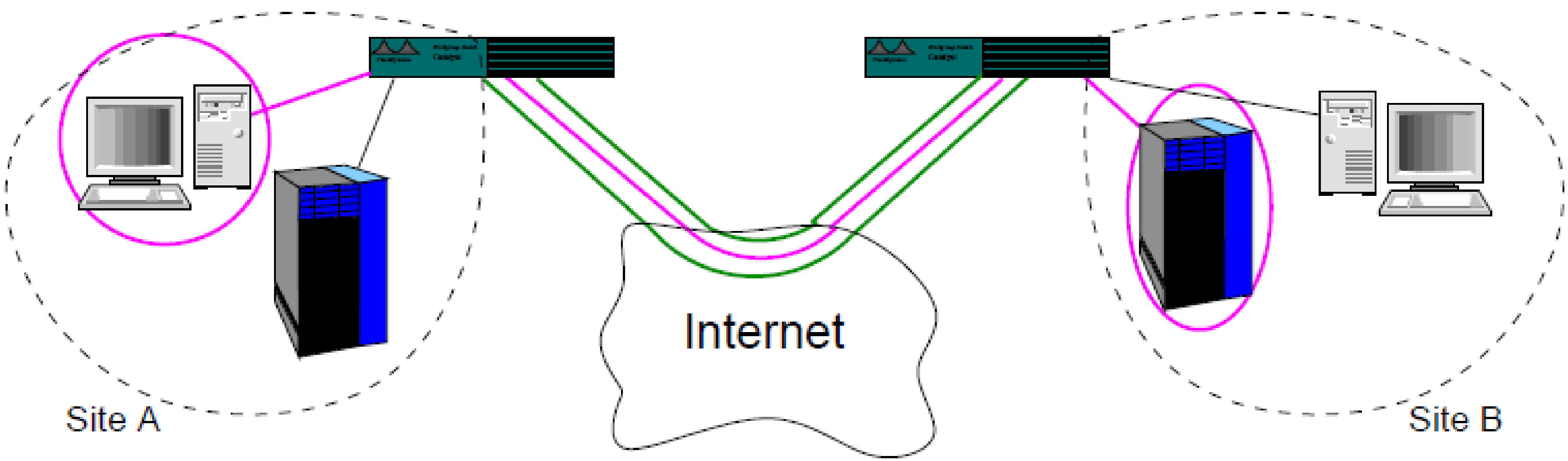
VPN: then?

- It looks best to introduce security in the
  - Transport layer
  - Network layer
- These are the most popular choices for VPNs
- Other options:
  - Secure Application layer protocols: only protect a single application, but are often used for specialized purposes, e.g. S/MIME or PGP for secure e-mail
  - Secure Data Link layer protocols: are mostly used with PPP or other modem-based communication. e.g. PPTP, L2TP, LTF

SSL Tunneling

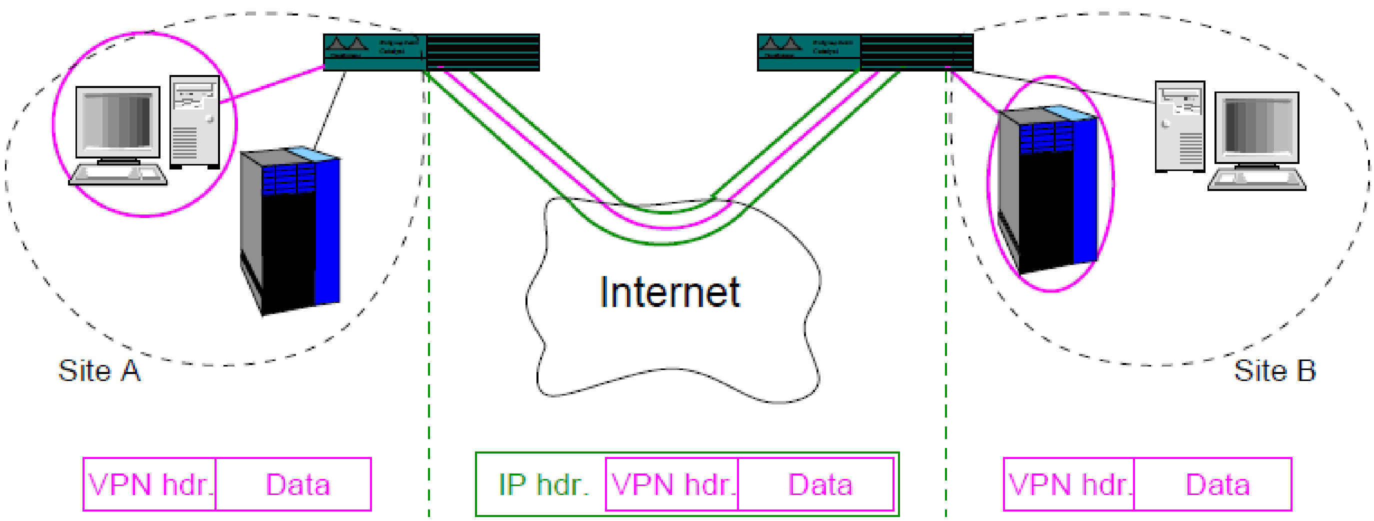
Tunneling

- Operation of a network connection on top of another network connection
- It allows two hosts or sites to communicate through another network that they do not want to use directly

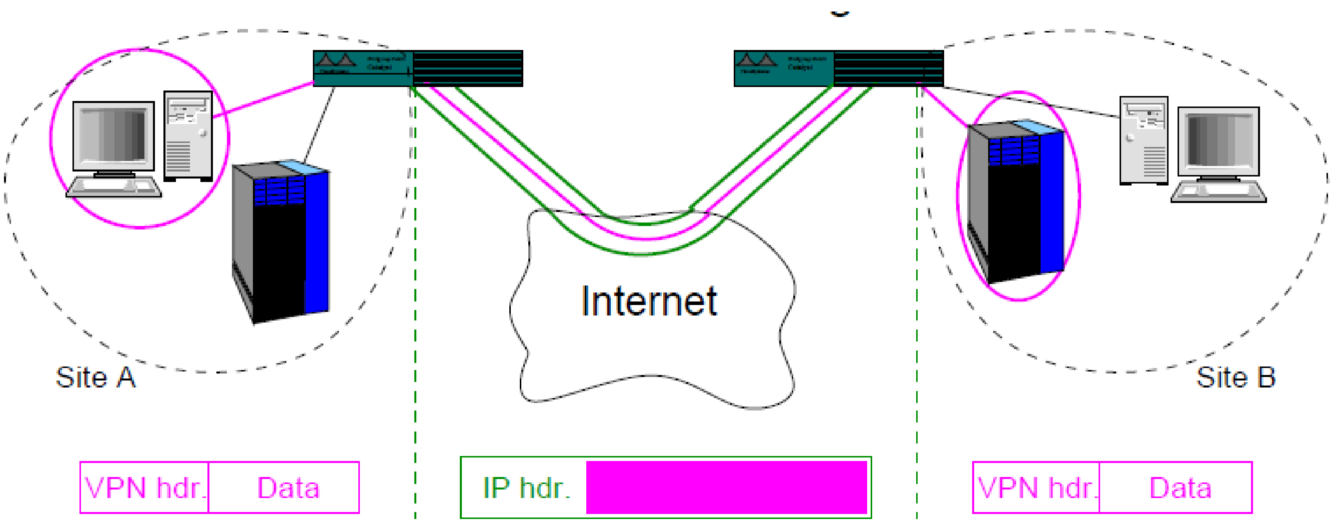


Site-to-site tunneling

- Enables a PDU to be transported from one site to another without its contents being processed by hosts on the route.
- Idea: Encapsulate the whole PDU in another PDU sent out on the network connecting the two sites.
  - Encapsulation takes place in edge router on src. site.
  - Decapsulation takes place in edge router on dst. site.
- Note that the host-to-host communication does not need to use IP



Secure tunneling



- Enables a PDU to be transported from one site to another without its contents being seen or changed by hosts on the route.
- Idea: Encrypt the PDU, and then encapsulate it in another PDU sent out on the network connecting the two sites.
  - Encryption can take place in edge router on src. site.
  - Decryption can take place in edge router on dst. site.
- Note: dst. address in IP header is for dst. edge router.

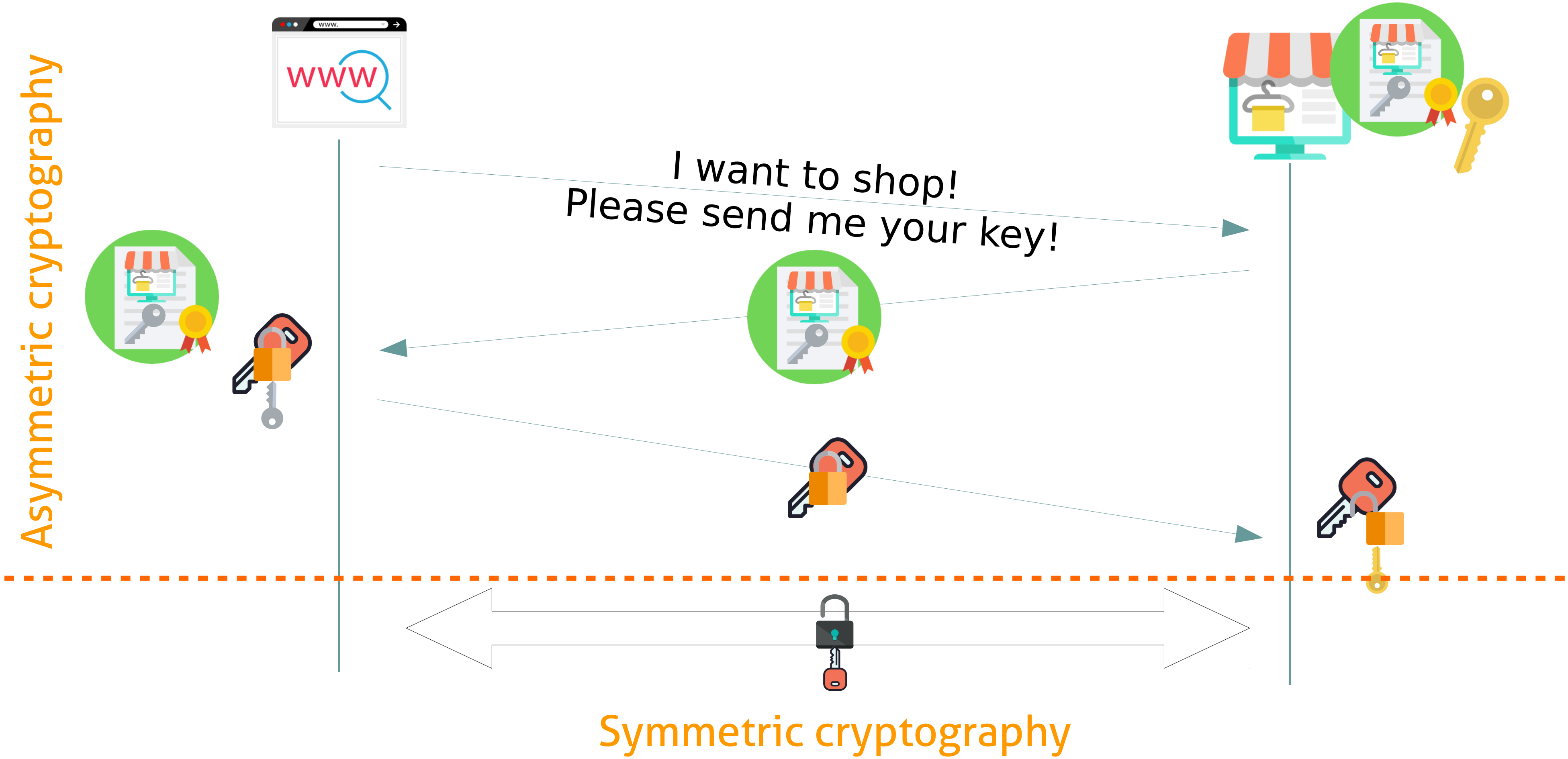
Tunneling for VPNs

- Tunneling offers the basic method for providing a VPN.
- Where in the network architecture to initiate and terminate the tunnel:
  - Router/firewall?
  - Special box?
  - Host?
  - Application?
- Which layer to do the tunneling in:
  - Transport layer?
  - Network layer?
- Other possibilities (see previous discussion)
- And of course: Is tunneling the only possible technique?

Secure Socket Layer

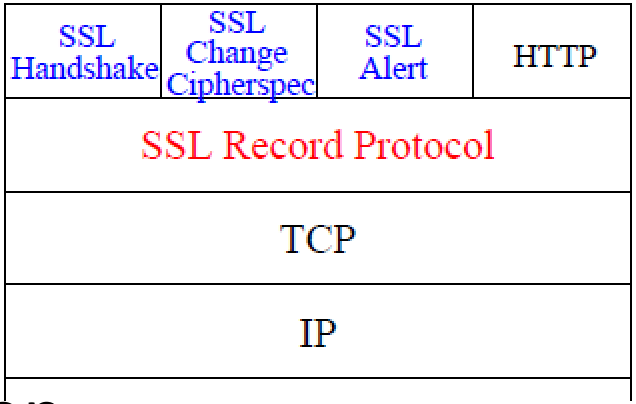
- SSL 3.0 has become TLS standard (RFC 2246) with small changes
- Applies security in the Transport layer.
- Originally designed (by Netscape) to offer security for client-server sessions.
- If implemented on boundary routers (or proxies), can provide a tunnel between two sites – typically LANs.
- Placed on top of TCP, so no need to change TCP/IP stack or OS.
- Provides secure channel (byte stream)
  - Any TCP-based protocol
  - https:// URIs, port 443
  - NNTP, SIP, SMTP...
- Optional server authentication with public key certificates
  - Common on commercial sites

How HTTPS (HTTP on top of TLS) works



SSL protocol Architecture

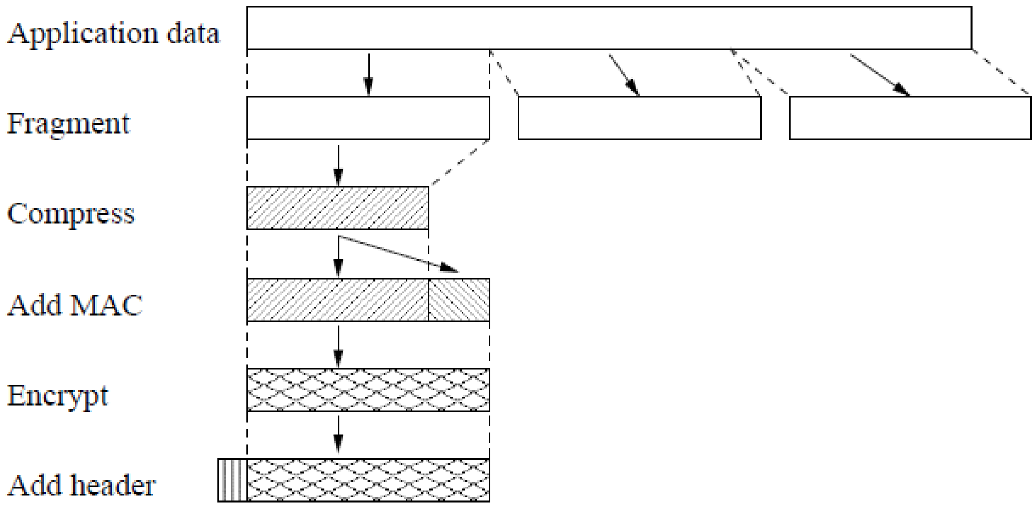
- Adds extra layer between T- and A-layers, and extra elements to A-layer
- Record Protocol: Protocol offering basic encryption and integrity services to applications



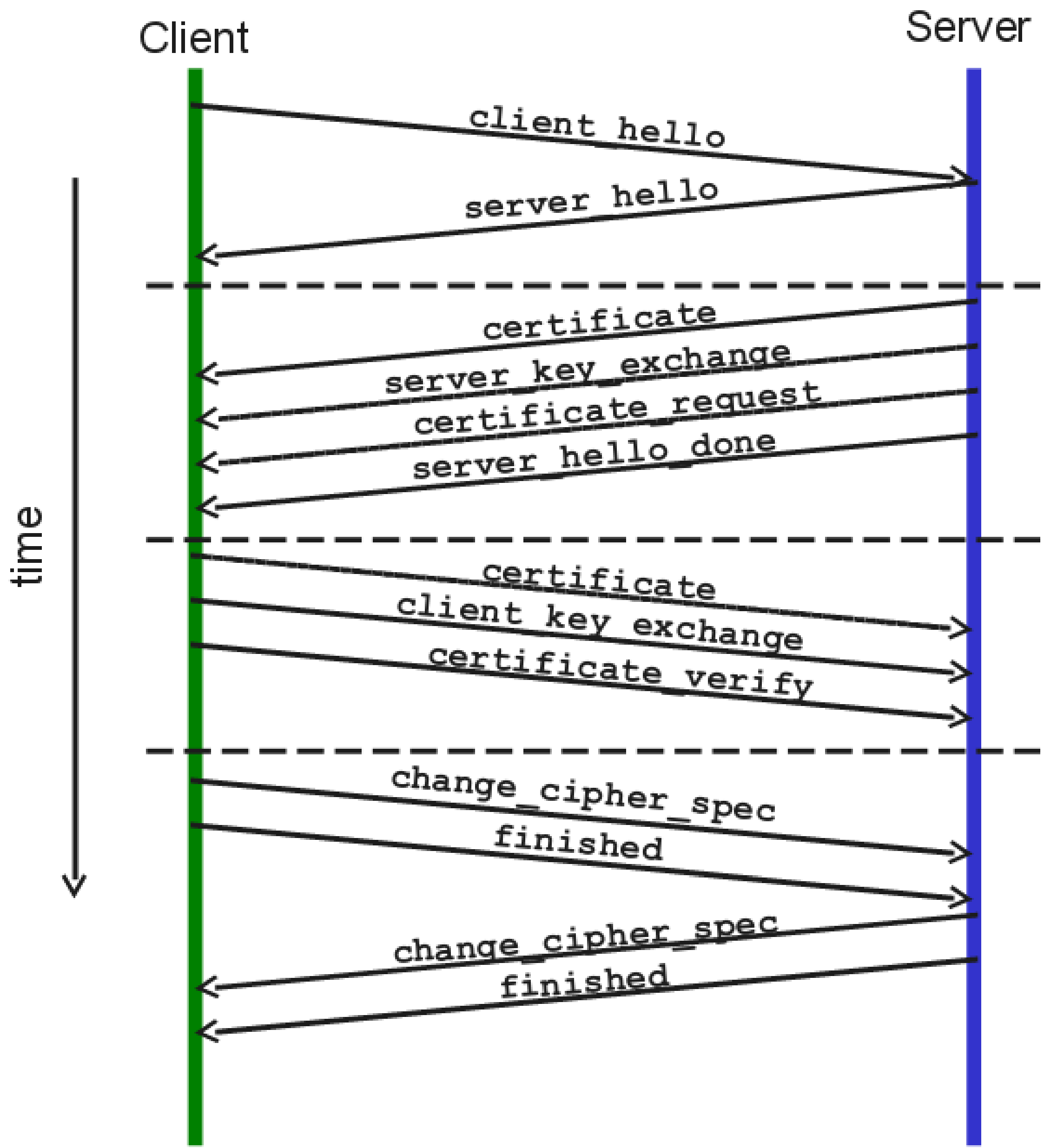
- Application Protocols: control operation of the record protocol
  - Handshake: Used to authenticate server (and optionally client) and to agree on encryption keys and algorithms.
  - Change cipher spec: Selects agreed keys and encryption algorithm until further notice.
  - Alert: Transfers information about failures.

SSL/TLS Record Protocol

- Offers to apply the following steps to PDUs:
  - 1 Fragmentation into blocks of ≤ 2^{14} bytes.
  - 2 (optional) Lossless compression.
  - 3 Addition of a keyed MAC, using a shared secret MAC key.
  - 4 Encryption, using a shared secret encryption key.
  - 5 Addition of header indicating Application protocol in use.



SSL Handshake



1) Hello phase

2) Server authentication

3) Client authentication

4) Finish

SSL/TLS Handshake Protocol

4-phase "Client/Server" protocol to establish parameters of the secure connection ("Client" is the initiator):

1) **Hello**: Establishment of security capabilities: Client sends list of possibilities, in order of preference. Server selects one, and informs Client of its choice. Parties also exchange random noise for use in key generation.

2) **Server authentication and key exchange**: Server executes selected key exchange protocol (if needed). Server sends authentication info. (e.g. X.509 cert.) to Client.

3) **Client authentication and key exchange**: Client executes selected key exchange protocol (mandatory). Client sends authentication info. to Server (optional).

4) **Finish**: Shared secret key is derived from pre-secrets exch. in 2, 3. Change Cipher Spec. protocol is activated. Summaries of progress of Handshake Protocol are exchanged and checked by both parties.

SSL/TLS Security Capabilities

- Conventionally expressed by a descriptive string, specifying:
  - Version of SSL/TLS
  - Key exchange algorithm
  - Grade of encryption (previous to TLSv1.1)
  - Encryption algorithm
  - Mode of block encryption (if block cipher used)
  - Cryptographic checksum algorithm
- Example: TLS_RSA_WITH_AES_128_CBC_SHA
  - TLS → (Latest version of) TLS
  - RSA → RSA key exchange
  - WITH → (merely filler…)
  - AES_128 → 128-bit AES encryption
  - CBC → Cipher Block Chaining
  - SHA → Use HMAC-SHA digest

Key exchange and authentication

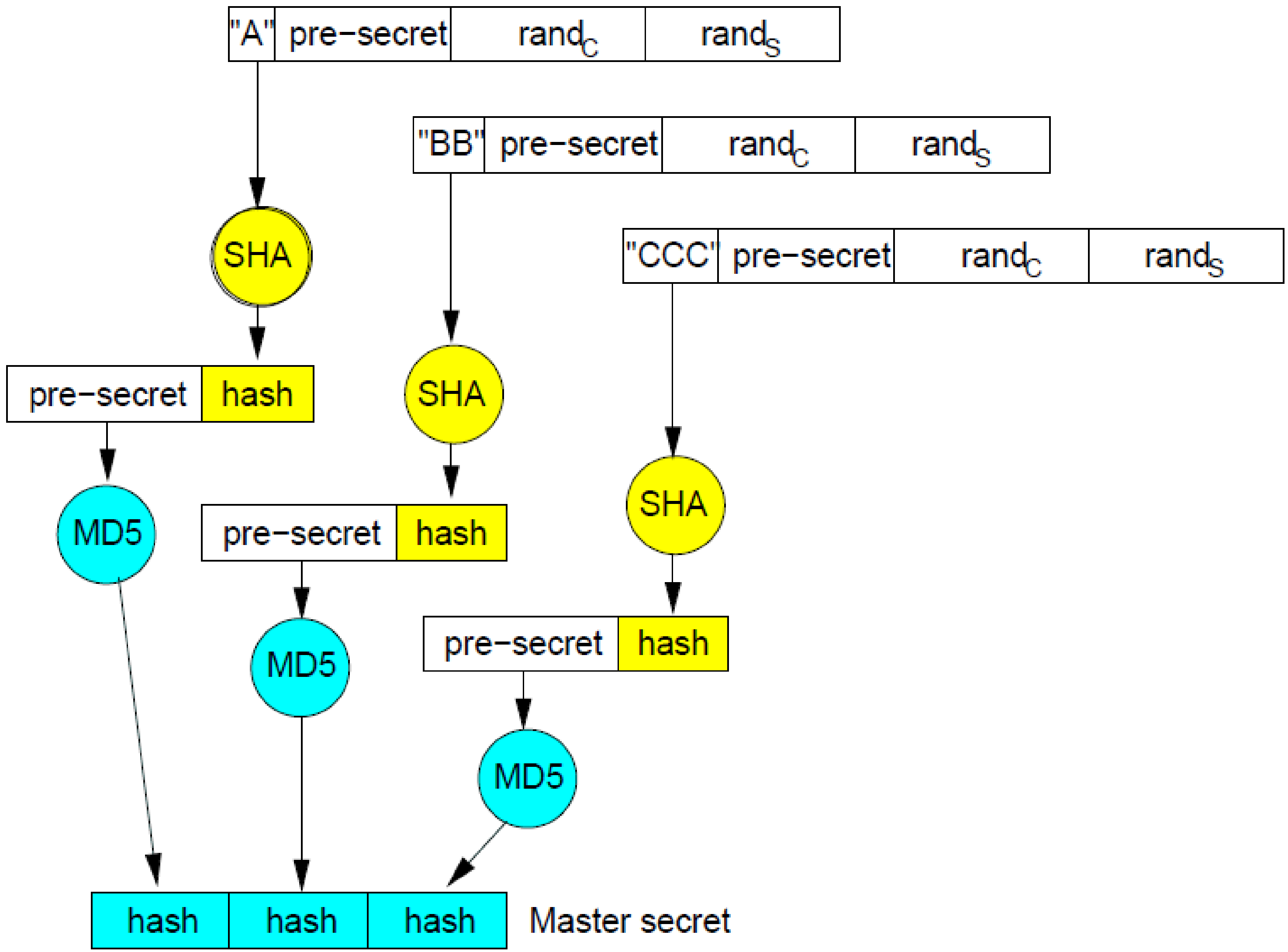
Possible ways of agreeing on secrets in TLS are:

- RSA: RSA key exch. (secret encrypted with recipient's publ. key)
- DHE RSA: Ephemeral Diffie-Hellman with RSA signatures
- DHE DSS: Ephemeral Diffie-Hellman with DSS signatures
- DH DSS: Diffie-Hellman with DSS certificates
- DH RSA: Diffie-Hellman with RSA certificates
- DH anon: Anonymous Diffie-Hellman (no authentication)
- NULL No key exch.

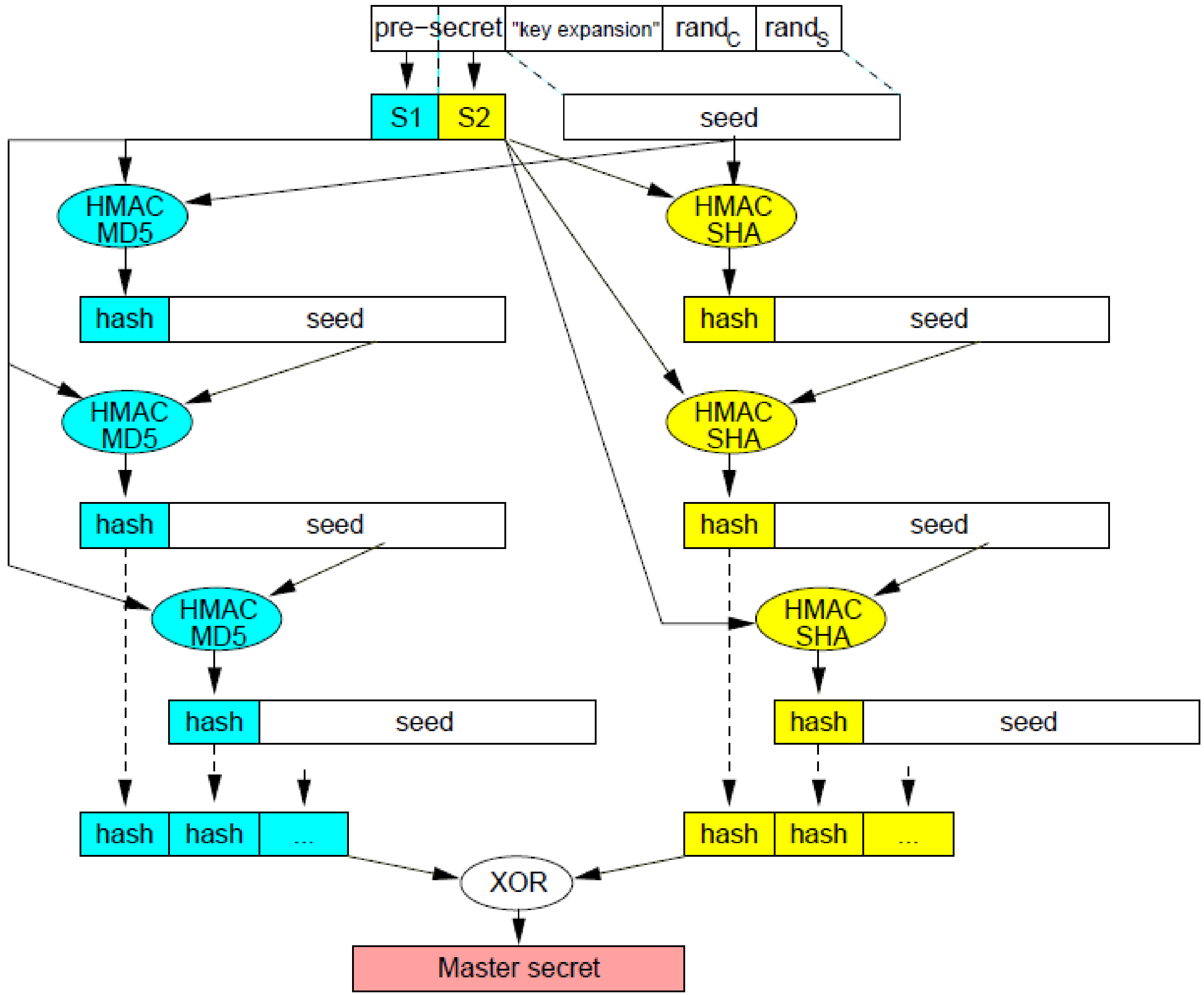
Variant: If followed by "EXPORT_", weak encryption is used. (This option only available prior to TLSv1.1)

- **Note**: "Key exchange" only establishes a pre-secret! From this, a master secret is derived by a pseudo-random function (PRF). Shared secret encryption key is derived by expansion of master secret with another PRF. (In TLS several keys are derived for different purposes.)

SSL Master Secret



TLS Master Secret



SSL/TLS Heartbeat

- It is an extension (RFC 6520) that allows to keep an established session alive
  - That is, as soon as the data exchange between two endpoints terminates, the session will also terminate
- To avoid the re-negotiation of the security parameters for establishing a secure session, we can keep using the same parameters even if there is no exchange of data
- It introduces two messages: **HeartbeatRequest** and **HeartbeatResponse**

Heartbeat exchange

- When one endpoint sends a HeartbeatRequest message to the other endpoints, the former also starts what is known as the **retransmit timer**
  - During the time interval of the retransmit timer, the sending endpoint will not send another HeartbeatRequest message.
- An SSL/TLS session is considered to have terminated in the absence of a HeartbeatResponse packet within a time interval

Heartbeat payload

- As a protection against a replay attack, HeartbeatRequest packets include a payload that must be returned without change by the receiver in its HeartbeatResponse packet
- The Heartbeat message is defined as

```
struct {
      HeartbeatMessageType type;
      uint16 payload_length;
      opaque payload[HeartbeatMessage.payload_length];
      opaque padding[padding_length];
} HeartbeatMessage;
```

Heartbleed bug

- Bug in OpenSSL library (4/4/2014)
- The receiver of request did not check that the **size of the payload in the packet** actually equaled the **value given** by the sender to the payload length field in the request packet
  - The attacker sends little data but sets the size to **max**
  - The receiver allcates that amount of memory for the response and copied **max bytes** from the mem location where the request packet was received
  - Then, the actual payload returned could potentially include objects in the memory that **had nothing to do** with the received payload
    - Objects could be private keys, passwords, and such…

SSL VPN Architecture

- Two primary models:
- 1 SSL Portal VPN
  - Allow remote users to:
    - Connect to VPN gateway from a Web browser
    - Access services from Web site provided on gateway
- 2 SSL Tunnel VPN
  - Allow remote users to:
    - Access network protected by VPN gateway from
    - Web browser allowing active content.
  - More capabilities than portal VPNs, as easier to provide more services.

SSL VPN functionalities

Most SSL VPNs offer one or more core functionalities:

- Proxying: Intermediate device appears as true server to client. E.g. Web proxy.
- Application Translation: Conversion of information from one protocol to another.
  - e.g. Portal VPN offers translation for applications which are not Web-enabled, so users can use Web browser to access applications with no Web interface.
- Network Extension: Provision of partial or complete network access to remote users, typically via Tunnel VPN.
  - Two variants:
    - Full tunneling: All network traffic goes through tunnel.
    - Split tunneling: Organisation's traffic goes through tunnel, other traffic uses remote user's default gateway.

SSL VPN Securty Services

Typical services include:

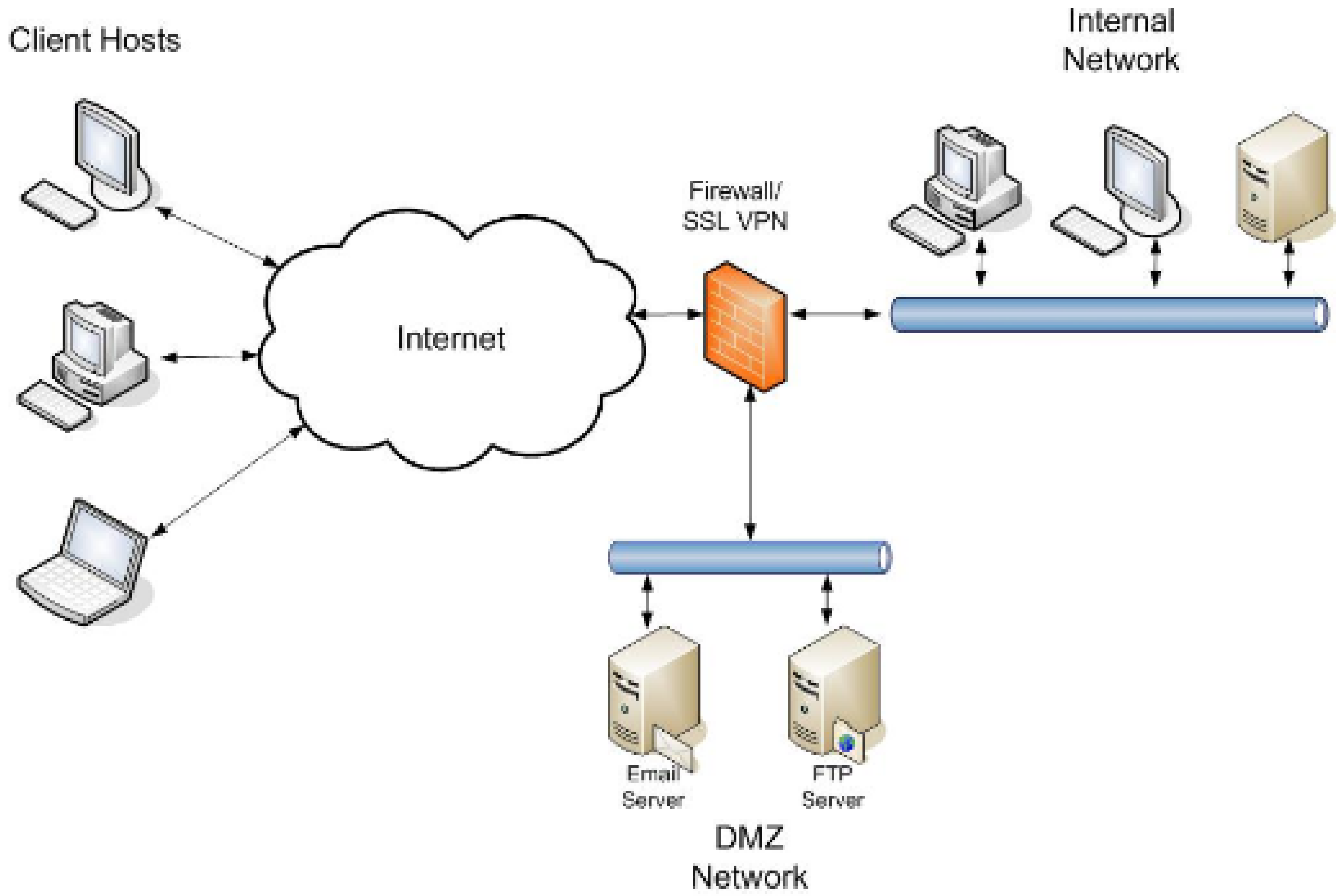
- **Authentication** Via strong authentication methods, such as two-factor authent., X.509 certificates, smartcards, security tokens etc. May be integrated in VPN device or external authent. server.
- **Encryption** and integrity protection: Via the use of the SSL/TLS protocol.
- **Access control**: May be per-user, per-group or per-resource.
- **Endpoint security controls**: Validate the security compliance of clients attempting to use the VPN.
  - e.g. presence of antivirus system, updated patches etc.
- **Intrusion prevention**: Evaluates decrypted data for malicious attacks, malware etc.

VPN device placement

SSL VPN Device placement

- Device placement is a challenge because it affects:
  - Security
  - Functionality
  - Performance
- Main options for placement:
  - VPN functionality in firewall
  - VPN device in internal network
  - Single-interface VPN device in DMZ
  - Dual-interface VPN device in DMZ
- Remember: Cryptographic protection only extends from VPN client systems to the SSL VPN device.

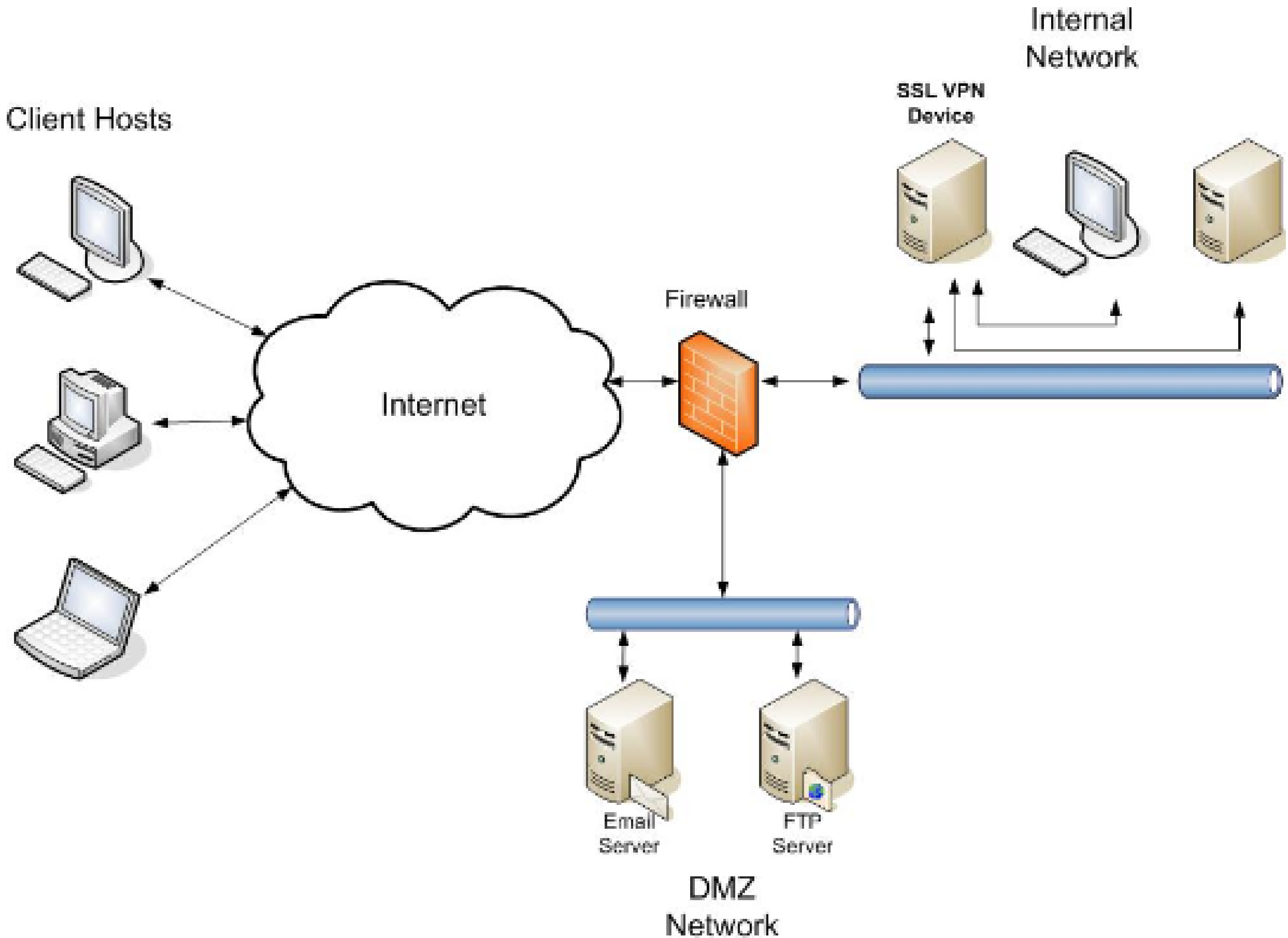
Firewall with an SSL VPN



VPN-enabled firewall

- The VPN device communicates directly with internal hosts
- Advantages
  - No holes in FW between external VPN device and internal network.
  - Traffic between device and internal network must go through FW.
  - Simple network administration since only one “box” to administer.
- Disadvantages
  - Limited to VPN functionality offered by FW vendor.
  - FW directly accessible to external users via port 443.
  - Adding VPN functionality to FW can introduce vulnerabilities.
- Note: TCP port 443 (standard) must be open on external FW interface, so clients can initiate connections.

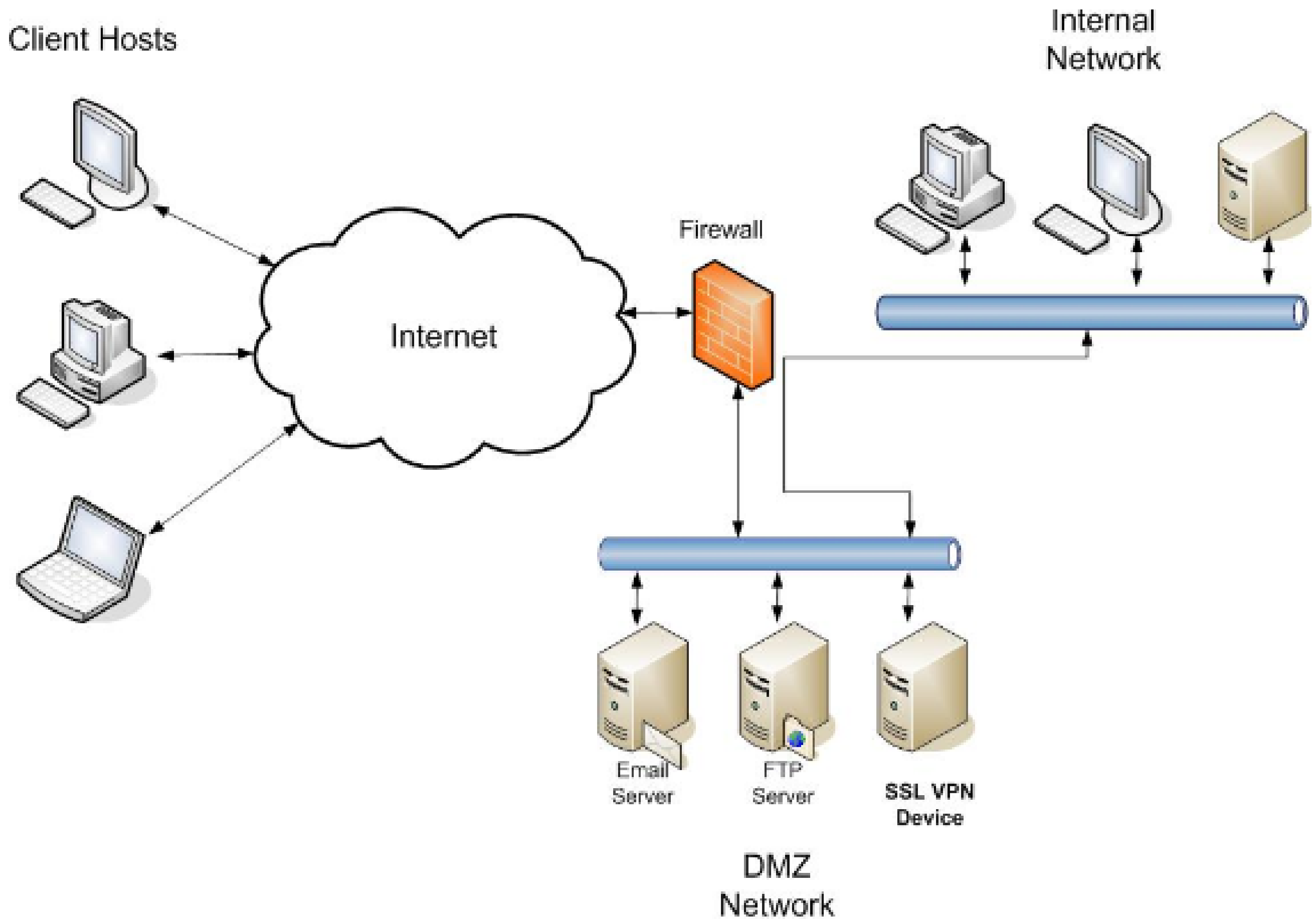
SSL VPN in internal network



VPN internal

- Advantages
  - Only single rule for single address to be added to FW.
  - No “holes” needed in FW between VPN device and internal network.
  - VPN traffic is behind FW, so protected from attacks by machines in DMZ.
- Disadvantages
  - VPN traffic passes through FW on tunnel, so it is not analyzed.
  - Unsolicited traffic can be sent into internal network from outside to internal VPN device.
  - Internal network is compromised if VPN device is compromised.
- Note: TCP port 443 (standard) opened on FW for the address of the device.

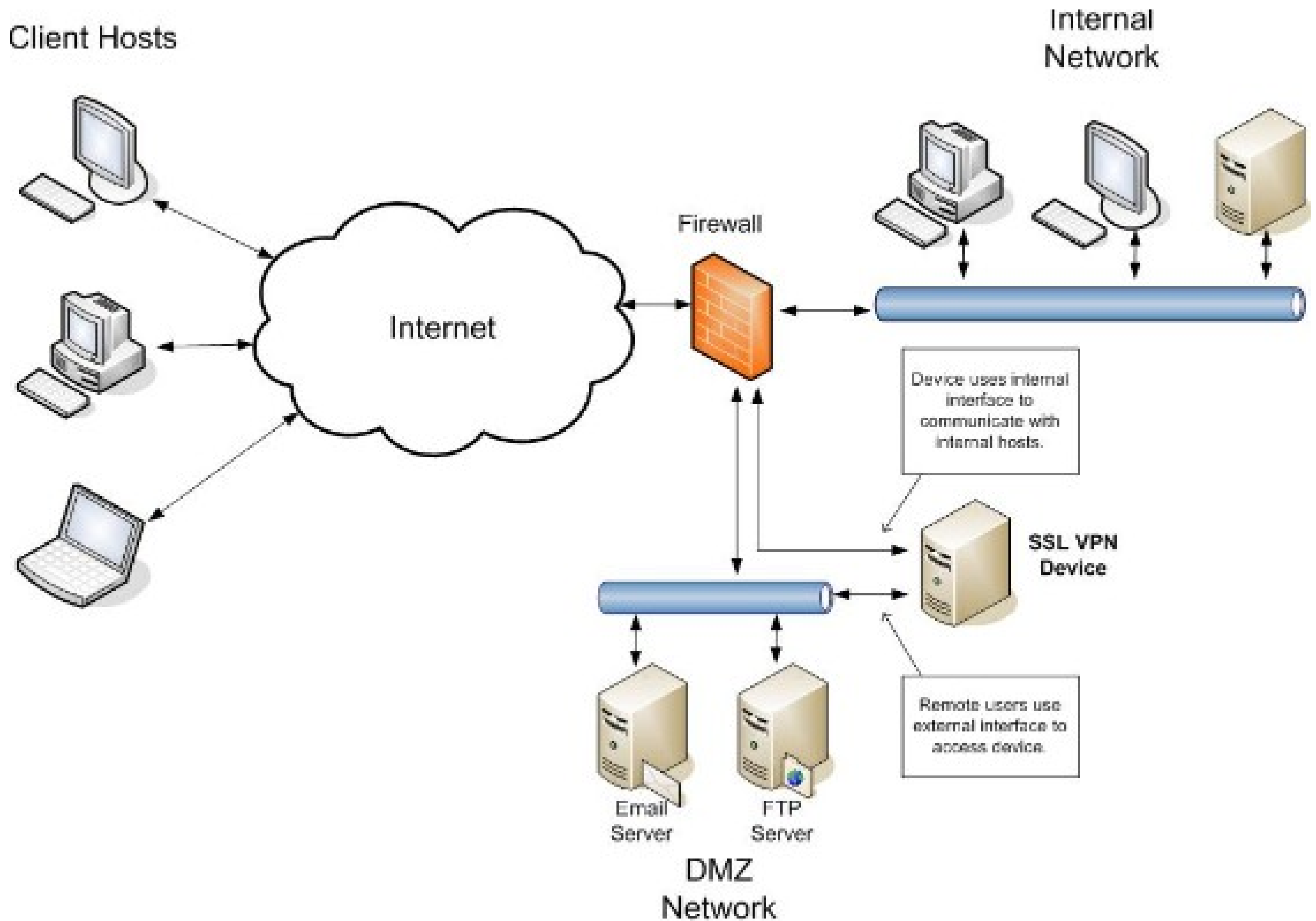
SSL VPN In DMZ



DMZ with VPN

- Advantages
  - Internal network protected against compromised VPN device.
  - Traffic between device and internal network must go through FW.
  - IDS in DMZ can analyze traffic destined for internal network.
- Disadvantages
  - Numerous ports open in FW between device and internal hosts.
  - Decrypted traffic from device to internal network must be sent through DMZ.
  - FW bypassed when user traffic is destined for hosts in DMZ.
- Note: TCP port 443 (standard) opened on FW for the address of the device

Dual interfaces VPN device in DMZ



VPN with two interfaces in DMZ

- Clients connect to external device interface, internal traffic uses internal interface.
- Advantages
  - All advantages of placing VPN device DMZ.
  - Unencrypted traffic to internal hosts is protected from other hosts in DMZ.
  - Only FW interface connected to device's internal interface needs to permit traffic from VPN device.
- Disadvantages
  - Numerous ports open in FW between device and internal hosts.
  - May introduce additional routing complexity.
  - FW bypassed if split tunneling is not used and user traffic is destined for hosts in DMZ

IPSec

IPsec

- A Network Layer protocol suite for providing security over IP.
- Part of IPv6; an add-on for IPv4.
- Can handle all three possible security architectures:

Feature	Gateway-to-Gateway	Host-to-Gateway	Host-to-Host
Protection between client and local gateway	No	N/A (client is VPN endpoint)	N/A (client is VPN endpoint)
Protection between VPN endpoints	Yes	Yes	Yes
Protection between remote gateway and remote server (behind gateway)	No	No	N/A (client is VPN endpoint)
Transparency to users	Yes	No	No
Transparency to users' systems	Yes	No	No
Transparency to servers	Yes	Yes	No

IPsec services

- Basic functions, provided by separate (sub-)protocols:
  - Authentication Header (AH): Support for data integrity and authentication of IP packets.
  - Encapsulated Security Payload (ESP): Support for encryption and (optionally) authentication.
  - Internet Key Exchange (IKE): Support for key management etc.

Service	AH	ESP (encrypt only)	ESP(encrypt+authent.)
Access Control	+	+	+
Connectionless integrity	+		+
Protection between VPN endpoints	+		+
Data origin authentication	+		+
Reject replayed packets		+	+
Payload confidentiality		+	+
Metadata confidentiality		partial	partial
Traffic flow confidentiality		(*)	(*)

IPsec Security Associations

- Think of it as an IPsec connection: all of the parameters needed, like crypto algorithms (AES, SHA1, etc.), modes of operation (CBC, HMAC, etc.), key lengths, traffic to be protected, etc.
- Both sides must agree on the SA for secure communications to work
- For a two-way communication, two SAs must be defined.
- SA parameters must be negotiated (using IKE) between sender and receiver before secure communication can start.
- Each SA is identified by:
  - Security Parameters Index (SPI): 32-bit integer chosen by sender. Enables receiving system to select the required SA.
  - Destination Address: Only unicast IP addresses allowed!
  - Security Protocol Identifier: AH or ESP.

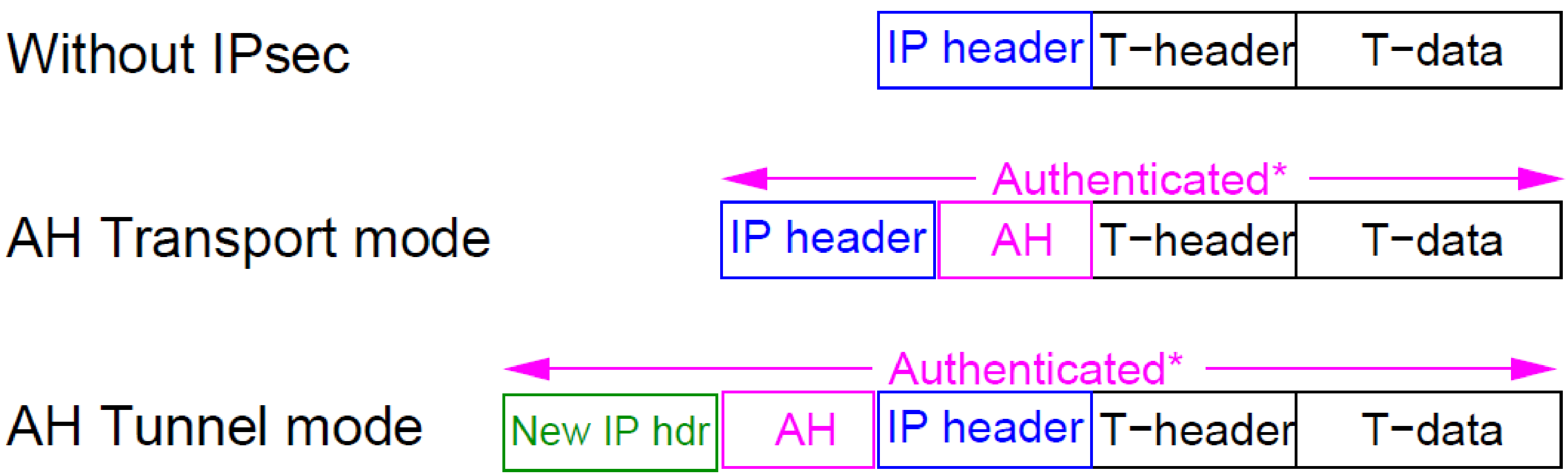
IPsec modes

- Transport Mode
  - Provides protection for a T-layer packet embedded as payload in an IP packet.
- Tunnel Mode
  - Provides protection for an IP packet embedded as payload in an IP packet.

	Transport Mode SA	Tunnel Mode SA
AH	Authenticate IP payload and selected parts of IP header and IPv6 extension headers.	Authenticate entire inner IP packet and selected parts of outer IP header and outer IPv6 extension headers.
ESP	Encrypt IP payload + any IPv6 extension headers after ESP header.	Encrypt inner IP packet.
ESP + authent.	Encrypt IP payload + any IPv6 extension headers after ESP header. Authenticate IP payload.	Encrypt and authenticate inner IP packet.

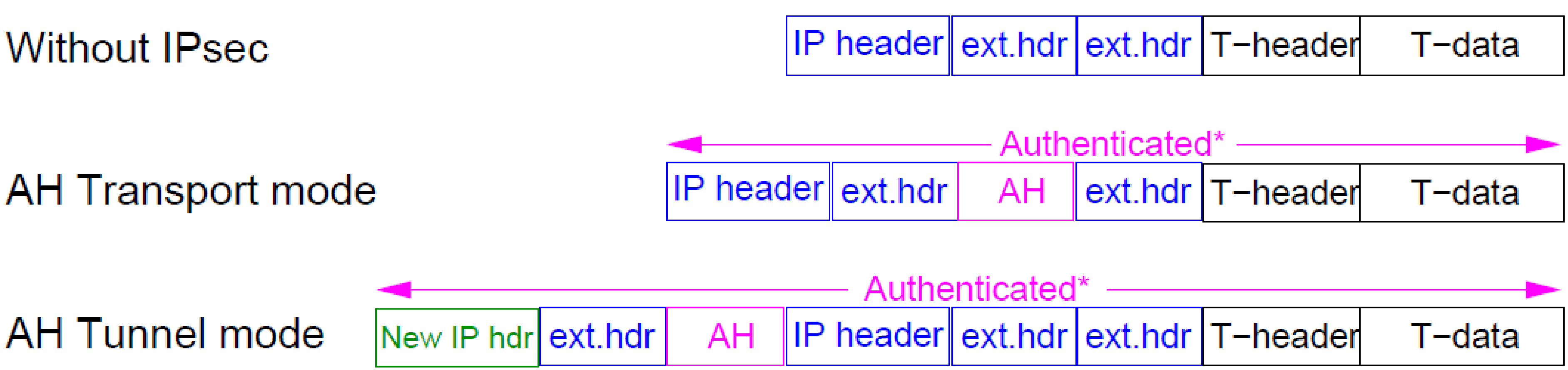
Authentication with IPv4

- AH header inserted after the outermost IP header – depending on whether Transport or Tunnel mode is used.
  - Do not forget that integrity check (and thus authentication) does not cover any mutable, unpredictable header fields.



Authentication with IPv6

- AH header inserted after the outermost IP header – depending on whether Transport or Tunnel mode is used.
  - Do not forget that integrity check (and thus authentication) does not cover any mutable, unpredictable header fields.

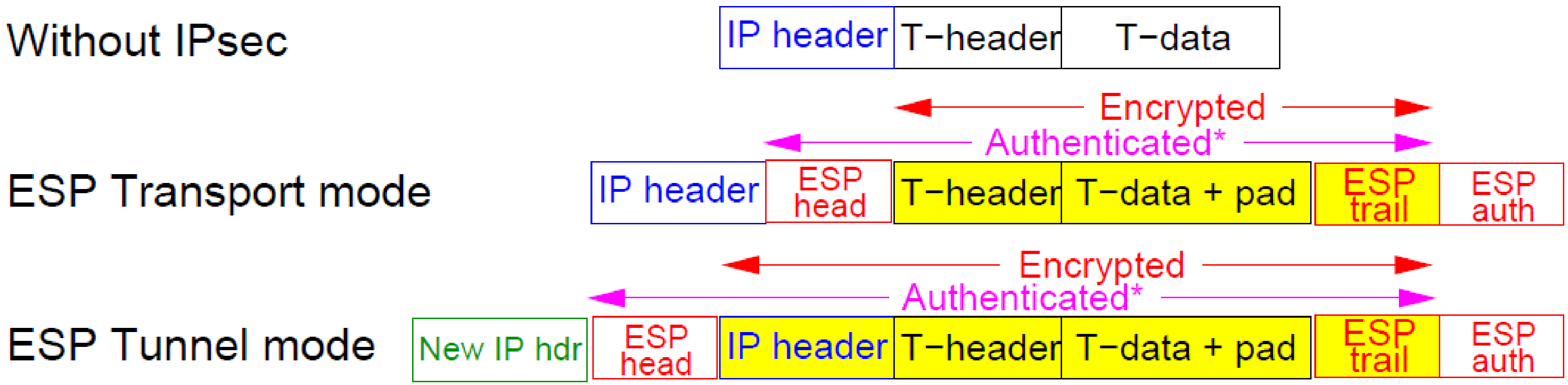


Authentication Header

- One of the (many possible) IP header fields. Contains:
  - Next Header: Type of following header field.
  - Payload Length: (Length - 2), in 32-bit words, of AH.
  - SPI: Identifies SA in use.
  - Sequence Number: Monotonically increasing packet counter value.
  - Authentication Data (AD): (variable length) HMAC based on MD5 or SHA-1 criptorgraphic hashing algorithm, or AES-CBC, evaluated over:
    - Immutable or predictable IP header fields. (Other fields assumed zero when MAC is calculated.)
    - Rest of AH header apart from AD field.
    - All embedded payload (from T-layer or embedded IP packet), assumed immutable.
- Immutable fields do not change as the packet traverses the network.
  - Example: Source address.
- Mutable but predictable fields may change, but can be predicted.
  - Example: Destination address.
- Mutable, unpredictable fields include Time-to-live, Header checksum.

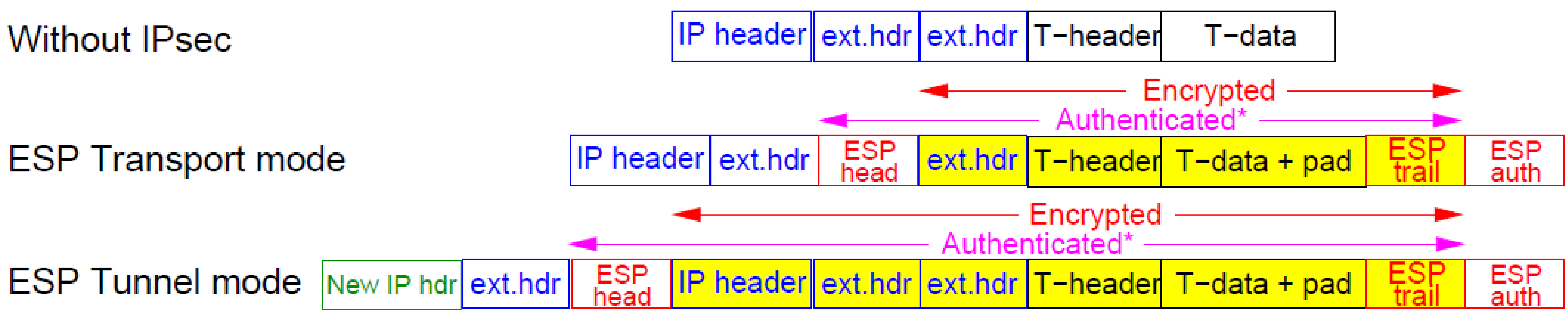
ESP with IPv4

- ESP header inserted after the outermost IP header – depending on whether Transport or Tunnel mode is used:
- Padding is added to end of T-layer payload to give (a certain amount) of traffic analysis protection.
- ESP trailer and (optional) ESP authentication field added after the end of the padded T-layer payload.
- As usual, integrity check (and thus authentication) does not cover any mutable, unpredictable header fields.



ESP with IPv6

- ESP header inserted after the outermost IP header – depending on whether Transport or Tunnel mode is used:
- Padding is added to end of T-layer payload to give (a certain amount) of traffic analysis protection.
- ESP trailer and (optional) ESP authentication field added after the end of the padded T-layer payload.
- As usual, integrity check (and thus authentication) does not cover any mutable, unpredictable header fields.



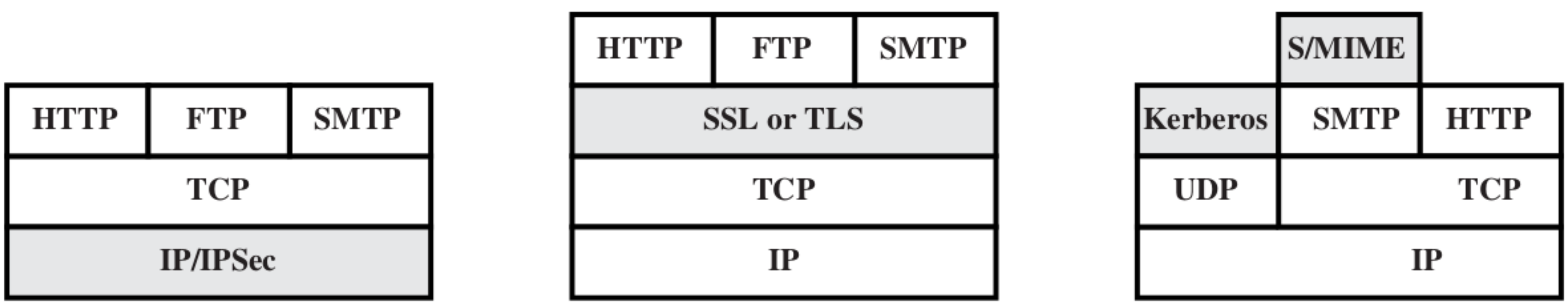
Encryption + Authentication

A common combination, can be achieved by:

1) ESP with Authentication. First apply ESP to data, then add AH field. Two subcases:
    1) Transport mode: E+A apply to IP payload, but IP header not protected.
    2) Tunnel mode: E+A apply to entire inner packet.
2) Transport Adjacency. Use bundled SAs, first ESP, then AH.
3) Encryption covers original IP payload. Authentication covers ESP + original IP header, including source and destination IP addresses
4) Transport-Tunnel bundle. Used to achieve authentication before encryption, for example via inner AH transport SA and outer ESP tunnel SA.
5) Authentication covers IP payload + IP immutable header. Encryption is applied to entire authenticated inner packet.

IPsec vs TLS

- TLS much more flexible because is in the upper levels
- TLS also provides application end-to-end security, best for web applications → HTTPS
- IPsec hast to run in kernel space
- IPsec much more complex and complicated to manage with



(a) Network level (b) Transport level (c) Application level

SSL again

- Crypto is insufficient for Web security
- One issue: linkage between crypto layer and applications
- Trust: what does the server really know about the client?
  - Unless client-side certificates are used, absolutely nothing
  - SSL provides a secure pipe. “Someone” is at the other end; you don’t know who
  - Usually there is no user authentication in SSL, but in the application layer!

SSL: the Client's Knowledge of the Server

- The client receives the server's certificate
- Does it help?
- A certificate means that someone has attested to the binding of some name to a public key
- Who has done the certification? Is it the right name?
  - Every browser has a list of built-in certificate authorities
  - Hundreds of certificate authorities! Do you trust them all to be honest and competent? Do you even know them all?
- It's all a matter of trust…

Conclusions on SSL

- The cryptography itself seems correct
  - The human factors are dubious
    - Most users don't know what a certificate is, or how to verify one
- Even when they do know, it's hard to know what it should say in any given situation
- There is no rational basis for deciding whether or not to trust a given CA

That’s all for today

- **Questions?**
- See you on Thursday (12:40)
- Resources:
  - Chapter 24 textbook
  - “Virtual private networking“, Gilbert Held, Wiley ed.
  - http://www.tcpipguide.com/free/t_IPSecurityIPSecProtocols.htm
  - “Guide to IPsec VPNs”, NIST800-77
  - “Guide to SSL VPNs”, NIST-SP800-113