M2 CyberSecurity
Security Architecture: network, system, key management, cybersecurity of industrial system.

Systems and Network Security

Florent Autréau - florent.autreau@imag.fr
2016 /2017

Network Security - Part 3

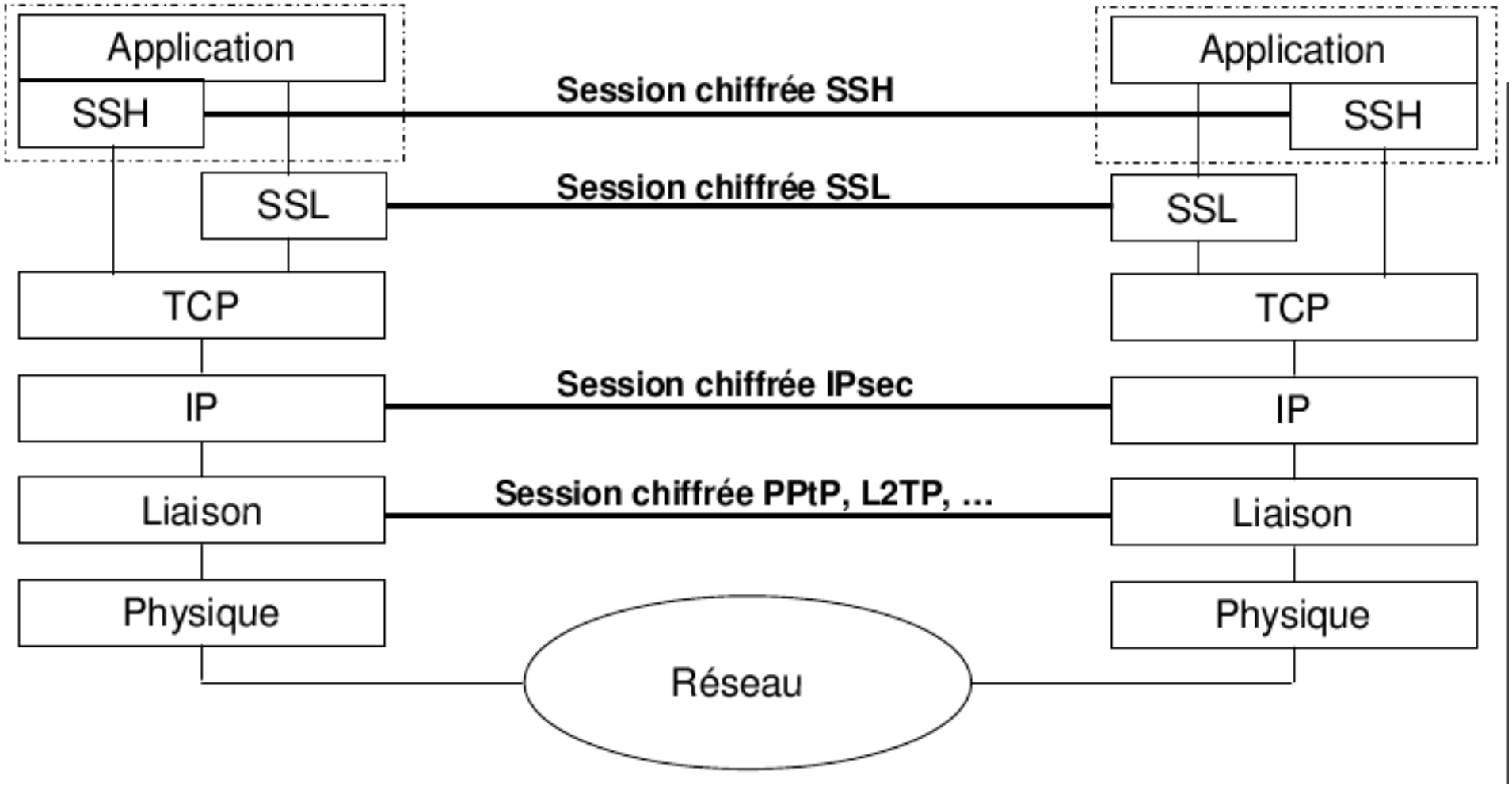
- Introduction
- Threat landscape
- Cryptography and Network Security
- Network Security – Vulnerabilities/Protection

Introduction

- Network architecture is layered
  - Lower layer vulnerabilities are inherited at higher levels
  - Describing exploitable features and vulnerabilities in the scope of each layer makes sense
- TCP/IP v.4 is dominant design in use
  - Many vulnerabilities can't be prevented without a major transition to a completely new design, or are hard problems
  - Most core vulnerabilities can't really be fixed

This is an important design consideration for any application that needs to use networks

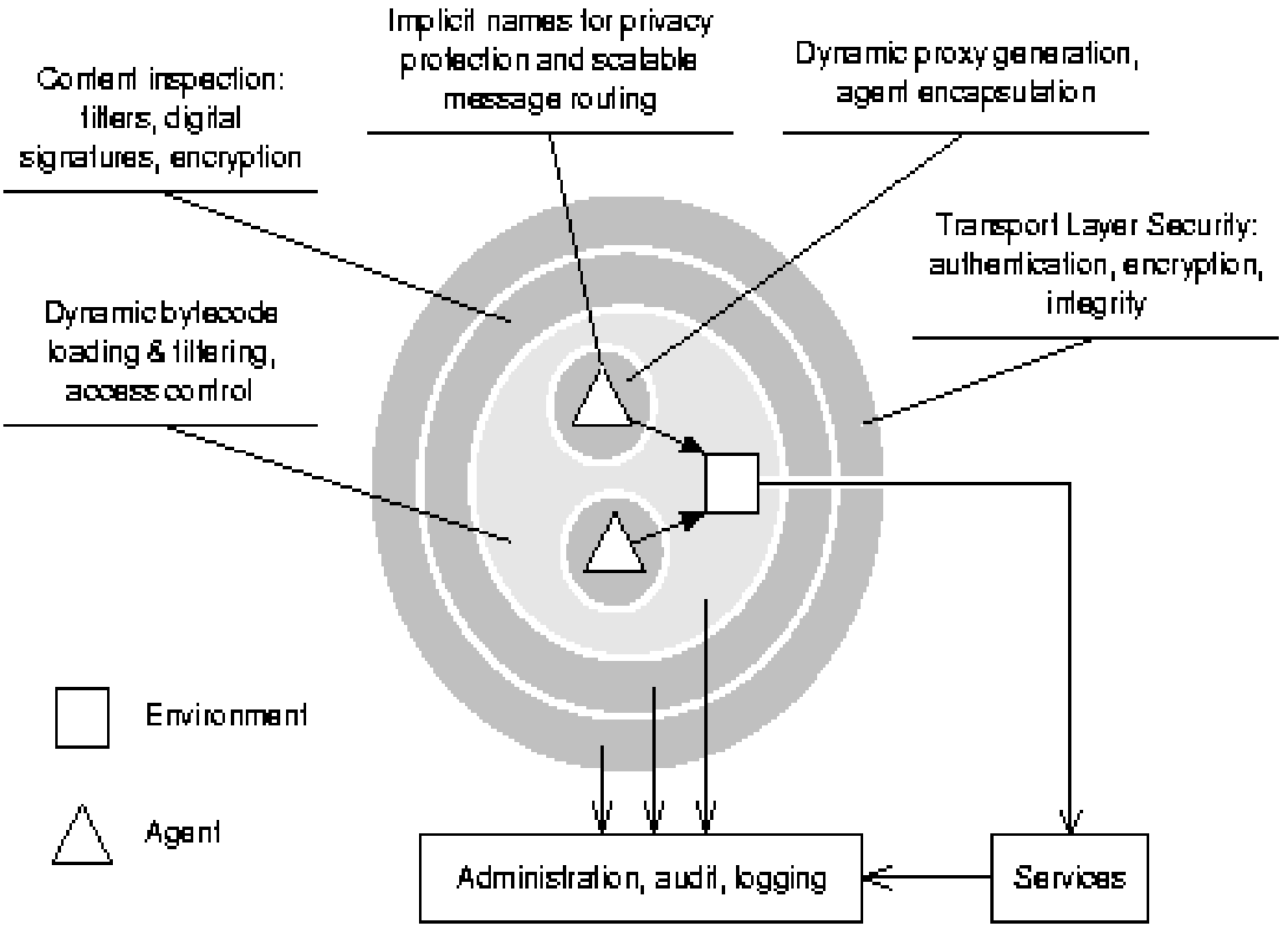
Network Security – Layered Model



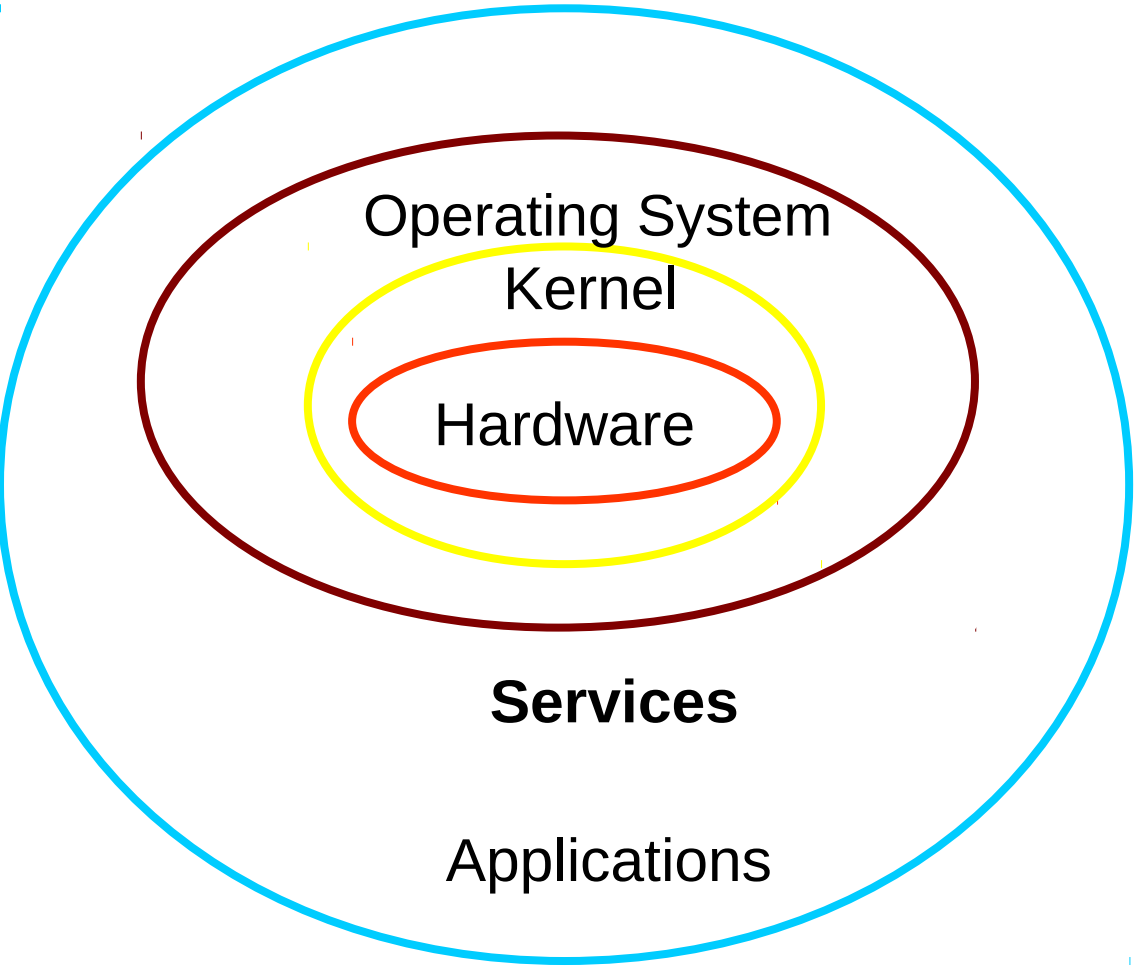
Strategies

- Prevention
  - Disabling features and functionality
  - Disabling exploitation paths
  - Some choice of network application-level protocols
- Mitigation (limiting consequences and impact)
  - Network configurations (de-militarized zones, etc...)
- Detection
- Response
- Providing guarantees at the application level
- Migrating to other, safer protocols

The Onion Model

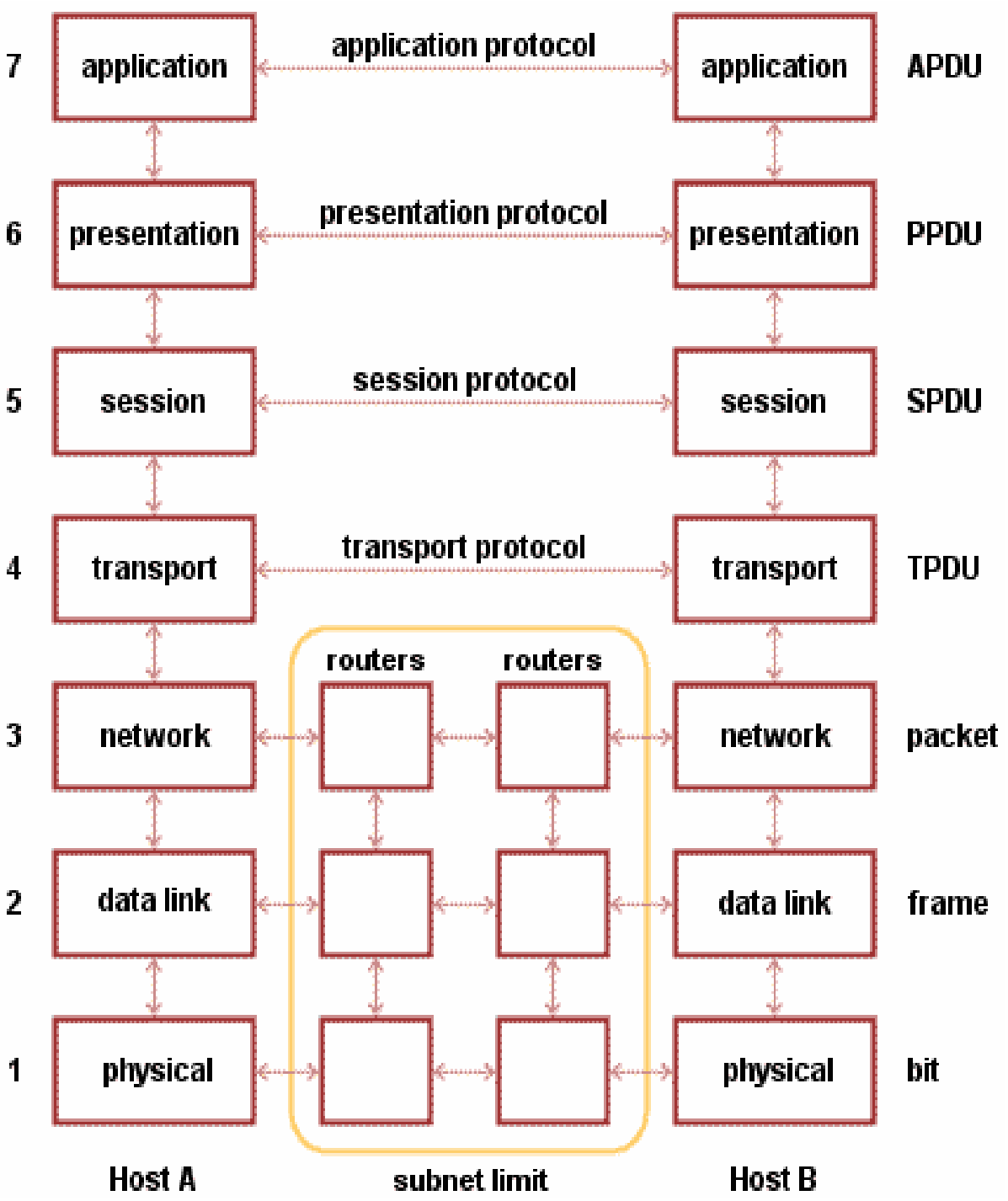


Layers of technology (and Onion Model)

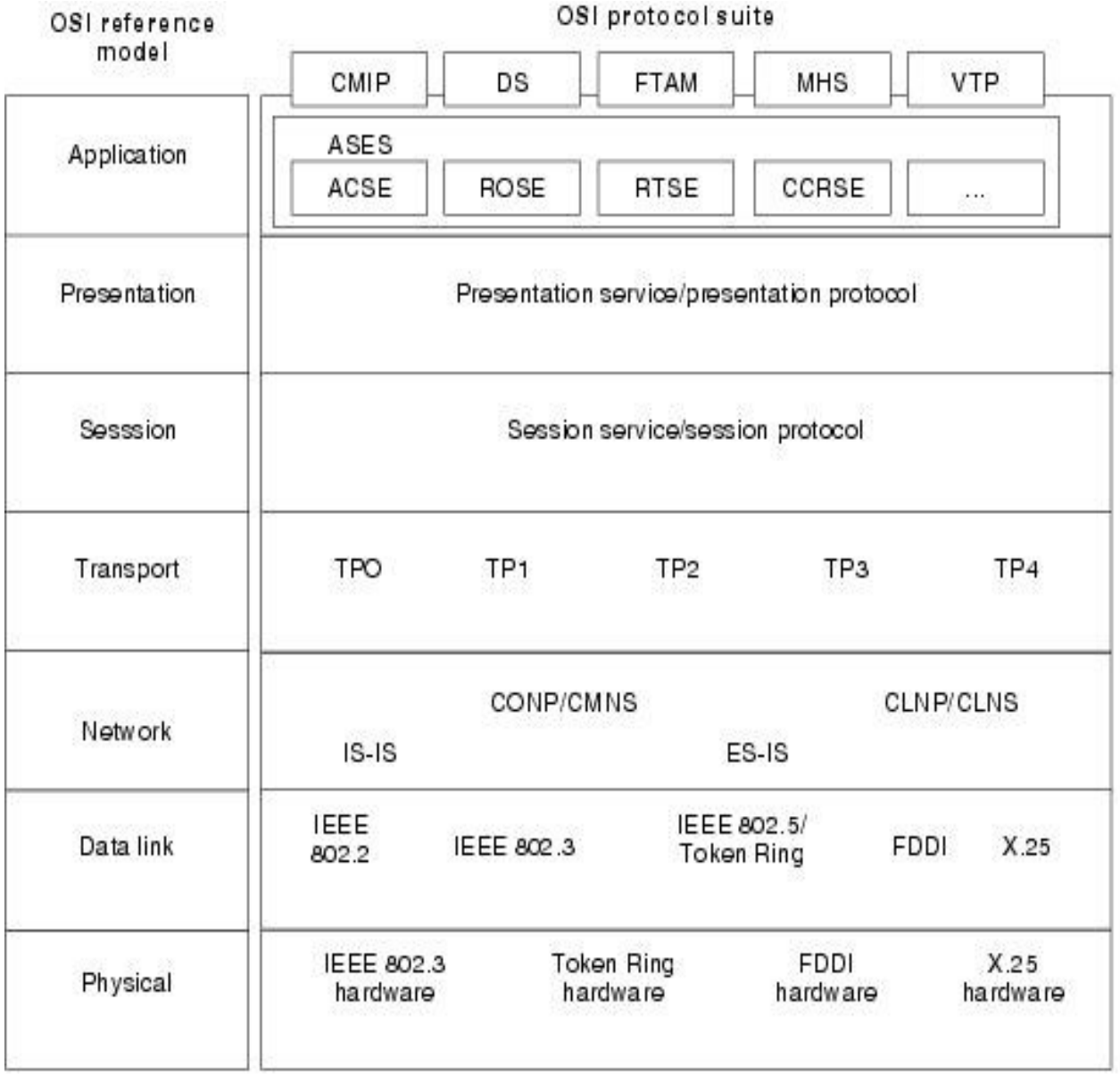


OSI Model

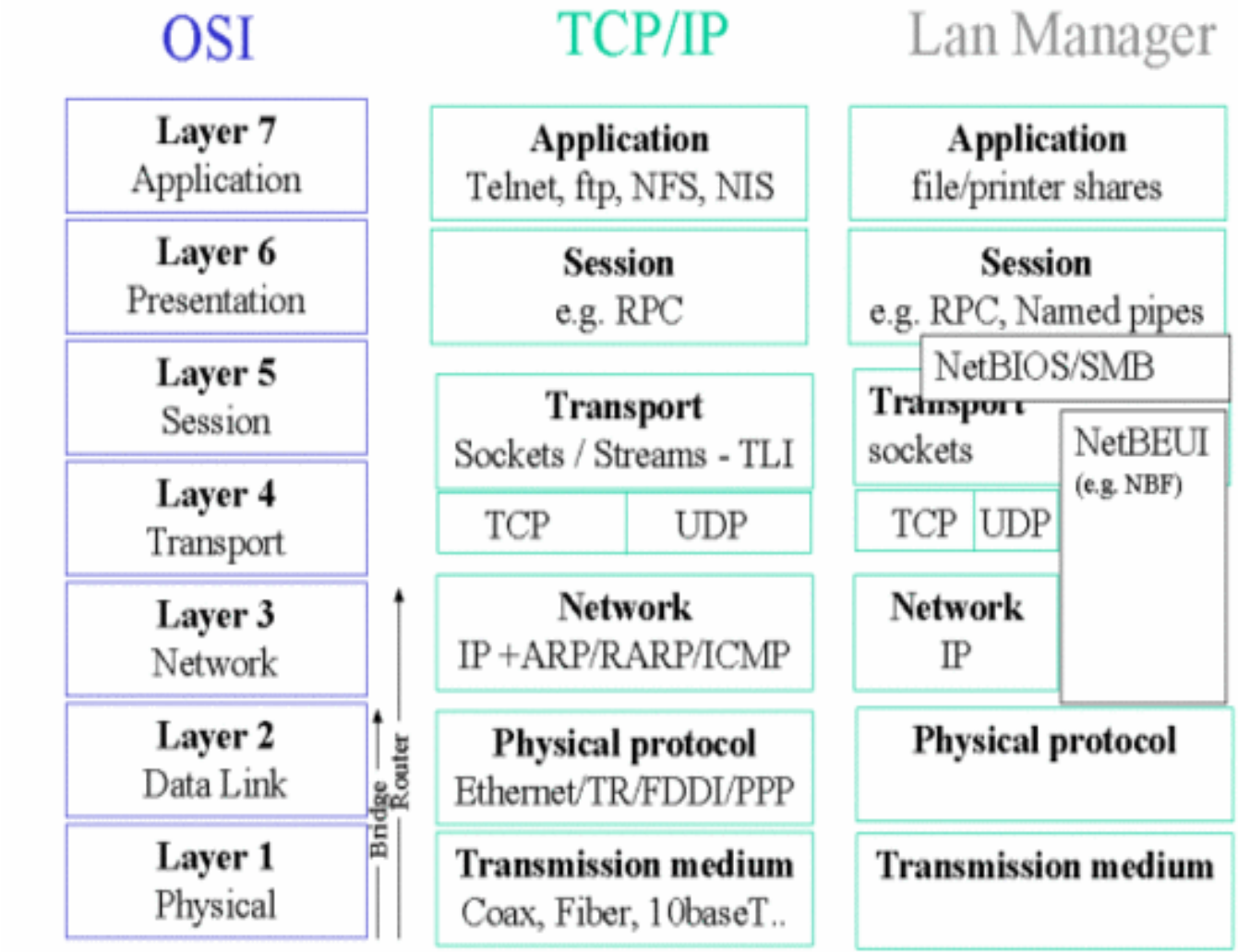
- Set of standards defining the architecture of computer networks
- OSI : Open Systems Interconnection
- 7 layers model
- Official references
  - ISO : IS 7498
  - CCITT / ITU-T : X200
  - AFNOR : NF.Z.70.001

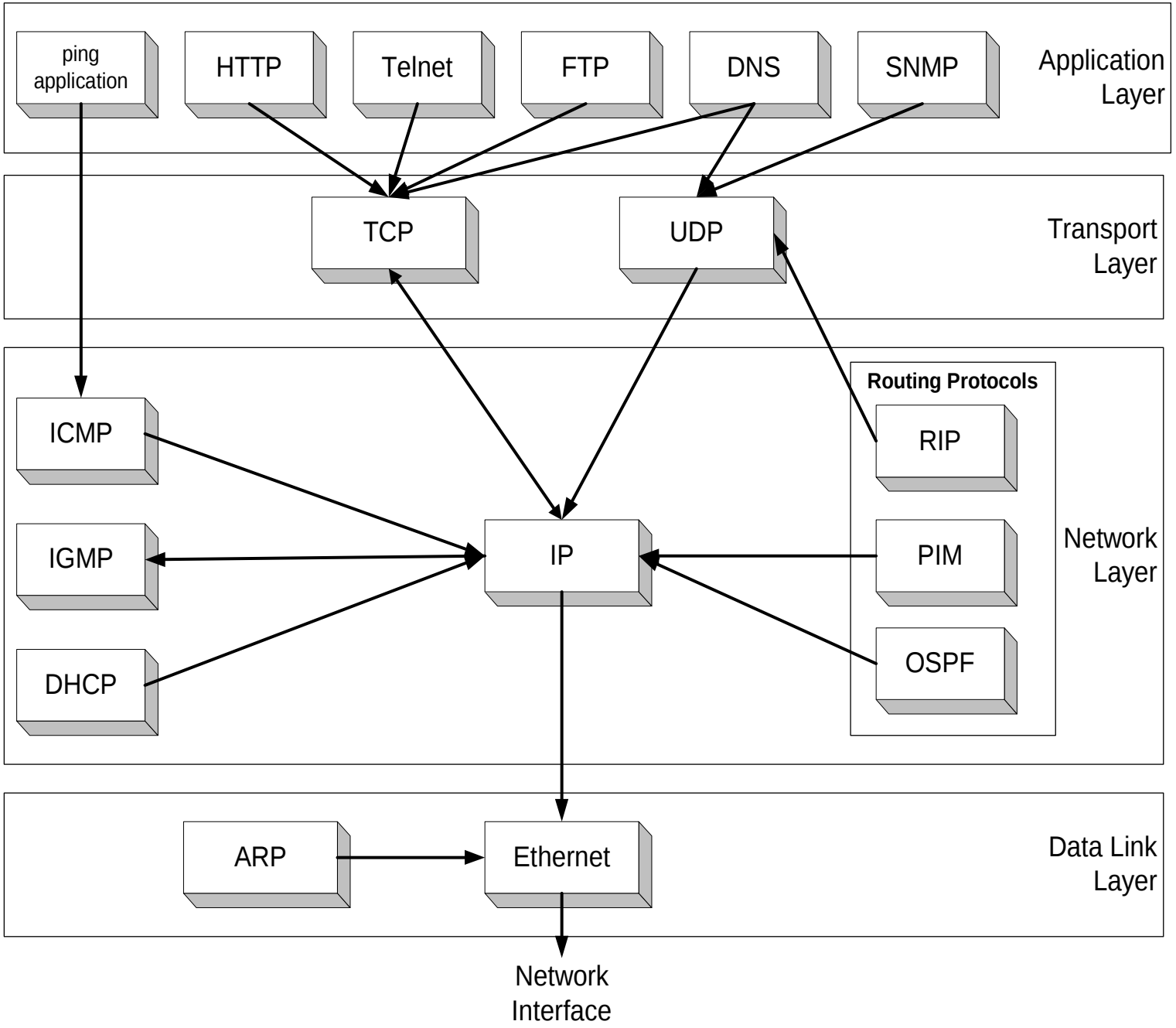

7
application
application protocol
application
APDU
6
presentation
presentation protocol
presentation
PPDU
5
session
session protocol
session
SPDU
4
transport
transport protocol
transport
TPDU
routers
routers
3
network
network
packet
2
data link
data link
frame
1
physical
physical
bit
Host A
subnet limit
Host B

OSI Reference Model



OSI, TCP/IP and Windows




ping application
HTTP
Telnet
FTP
DNS
SNMP
Application Layer
TCP
UDP
Transport Layer
Routing Protocols
ICMP
IGMP
DHCP
IP
RIP
PIM
OSPF
Network Layer
ARP
Ethernet
Data Link Layer
Network Interface

Layers

- The presence of layers is a feature of technology
- Separate layers often perform very different functions
- Similar functions are combined in one layer
- The boundary between two layers is usually easily defined
- Layers can often be *independently* implemented

The Onion Model works well with the standard OSI Model for Networking.

Physical layer – 1

- Purpose: Necessary infrastructure.
- Think "wires in the ground and connectors".
- This is the physical hardware of the network.
- Wires/optical cables/wireless links and other technologies provide a way for transmission of raw *bits* (0s and 1s).
- *Routers* and *switches* connect these cables and direct the traffic.

Physical Layer Vulnerabilities

- Loss of Power
- Loss of Environmental Control
- Physical Theft of Data and Hardware
- Physical Damage or Destruction of Data And Hardware
- Unauthorized changes to the functional environment (data connections, removable media, adding/removing resources)
- Disconnection of Physical Data Links
- Undetectable Interception of Data
- Keystroke & Other Input Logging
- Interference & Jamming

Physical Layer Controls

- Locked perimeters and enclosures
- Electronic lock mechanisms for logging & detailed authorization
- Video & Audio Surveillance
- PIN & password secured locks
- Biometric authentication systems
- Data Storage Cryptography
- Electromagnetic Shielding

Data link layer – 2

- Purpose: Provides basic connection between two logically connected machines.
- Think: “I stuff packets down a wire to my neighbour”
- Send raw packets between *hosts*.
- Basic error checking for lost data.
- In TCP/IP the "Physical layer" and the "Data Link" layer are grouped together and called the host-to-network layer.

Link Layer Vulnerabilities

- MAC Address Spoofing (station claims the identity of another)
- VLAN circumvention (station may force direct communication with other stations, bypassing logical controls such as subnets and firewalls.)
- Spanning Tree errors may be accidentally or purposefully introduced, causing the layer two environment to transmit packets in infinite loops.
- In wireless media situations, layer two protocols may allow free connection to the network by unauthorized entities, or weak authentication and encryption may allow a false sense of security.
- Switches may be forced to flood traffic to all VLAN ports rather than selectively forwarding to the appropriate ports, allowing interception of data by any device connected to a VLAN.

Link Layer Controls

- MAC Address Filtering- Identifying stations by address and cross-referencing physical port or logical access
- Do not use VLANs to enforce secure designs. Layers of trust should be physically isolated from one another, with policy engines such as firewalls between.
- Wireless applications must be carefully evaluated for unauthorized access exposure. Built-in encryption, authentication, and MAC filtering may be applied to secure networks.

Network Layer – 3

- Purpose: Provide end-to-end communication between any two machines.
- Think: “I try to get a packet to its destination”
- Tells data which link to travel down.
- Addresses the problem known as *routing.*
- Deals with the question "where do I go next to get to my destination?"
- Ensures packets get from source A to destination B.

Network Layer Vulnerabilities

- Route spoofing - propagation of false network topology
- IP Address Spoofing- false source addressing on malicious packets
- Identity & Resource ID Vulnerability - Reliance on addressing to identify resources and peers can be brittle and vulnerable

Network Layer Controls

- Route policy controls - Use strict anti-spoofing and route filters at network edges
- Firewalls with strong filter & anti-spoof policy
- ARP/Broadcast monitoring software
- Implementations that minimize the ability to abuse protocol features such as broadcast

Transport Layer - 4

- Purpose: Ensure that data gets between A and B.
- Think: “From the source and destination, I make sure that the data gets there”.
- Ensures a data gets between source and destination.
- If necessary ensure that connection is *lossless* (resend missing data).
- Provides *flow control* if necessary (send data faster or slower depending on the network conditions).

Transport Layer Vulnerabilities

- Mishandling of undefined, poorly defined, or “illegal” conditions
- Differences in transport protocol implementation allow “fingerprinting’ and other enumeration of host information
- Overloading of transport-layer mechanisms such as port numbers limit the ability to effectively filter and qualify traffic.
- Transmission mechanisms can be subject to spoofing and attack based on crafted packets and the educated guessing of flow and transmission values, allowing the disruption or seizure of control of communications.

Transport Layer Controls

- Strict firewall rules limiting access to specific transmission protocols and sub-protocol information such as TCP/UDP port number or ICMP type
- Stateful inspection at firewall layer, preventing out-of-state packets, “illegal” flags, and other phony packet profiles from entering the perimeter
- Stronger transmission and layer session identification mechanisms to prevent the attack and takeover of communications

Session Layer - 5

- Purpose: Provides a single connection for one application.
- Think: “I am in charge of the entire message.”
- This connection may be two way or may be synchronised.
- Not discussed much as it is never implemented.

Session Layer Vulnerabilities

- Weak or non-existent authentication mechanisms
- Passing of session credentials such as user ID and password in the clear, allowing intercept and unauthorized use
- Session identification may be subject to spoofing and hijack
- Leakage of information based on failed authentication attempts
- Unlimited failed sessions allow brute-force attacks on access credentials

Session Layer Controls

- Encrypted password exchange and storage
- Accounts have specific expirations for credentials and authorization
- Protect session identification information via random/cryptographic means
- Limit failed session attempts via timing mechanism, not lockout

Presentation Layer - 6

- Purpose: Provides commonly used functions for applications.
- Think: “I meet I18N standards”.
- The main job of the presentation layer is to ensure that character sets match – e.g. that Chinese characters are correctly received by the sends.
- Again not discussed much as it is never implemented.

Presentation Layer Vulnerabilities

- Poor handling of unexpected input can lead to application crashes or surrender of control to execute arbitrary instructions.
- Unintentional or ill-advised use of externally supplied input in control contexts may allow remote manipulation or information leakage.
- Cryptographic flaws may be exploited to circumvent privacy protections

Presentation Layer Controls

- Careful specification and checking of received input incoming into applications or library functions
- Separation of user input and program control functions- input should be sanitized and sanity checked before being passed into functions that use the input to control operation
- Careful and continuous review of cryptography solutions to ensure current security versus know and emerging threats

Application layer - 7

- Purpose: The computer programs which actually do things with the network.
- Think: “I deliver the mail, browse the web etc.”
- For example, your email *client* program which will talk to the email *server* at the other end.
- At this layer, we have many *protocols* (http, snmp, smtp, ftp, telnet) which different bits of software use.
- We often talk in terms of *client* and *server* architecture for the software.

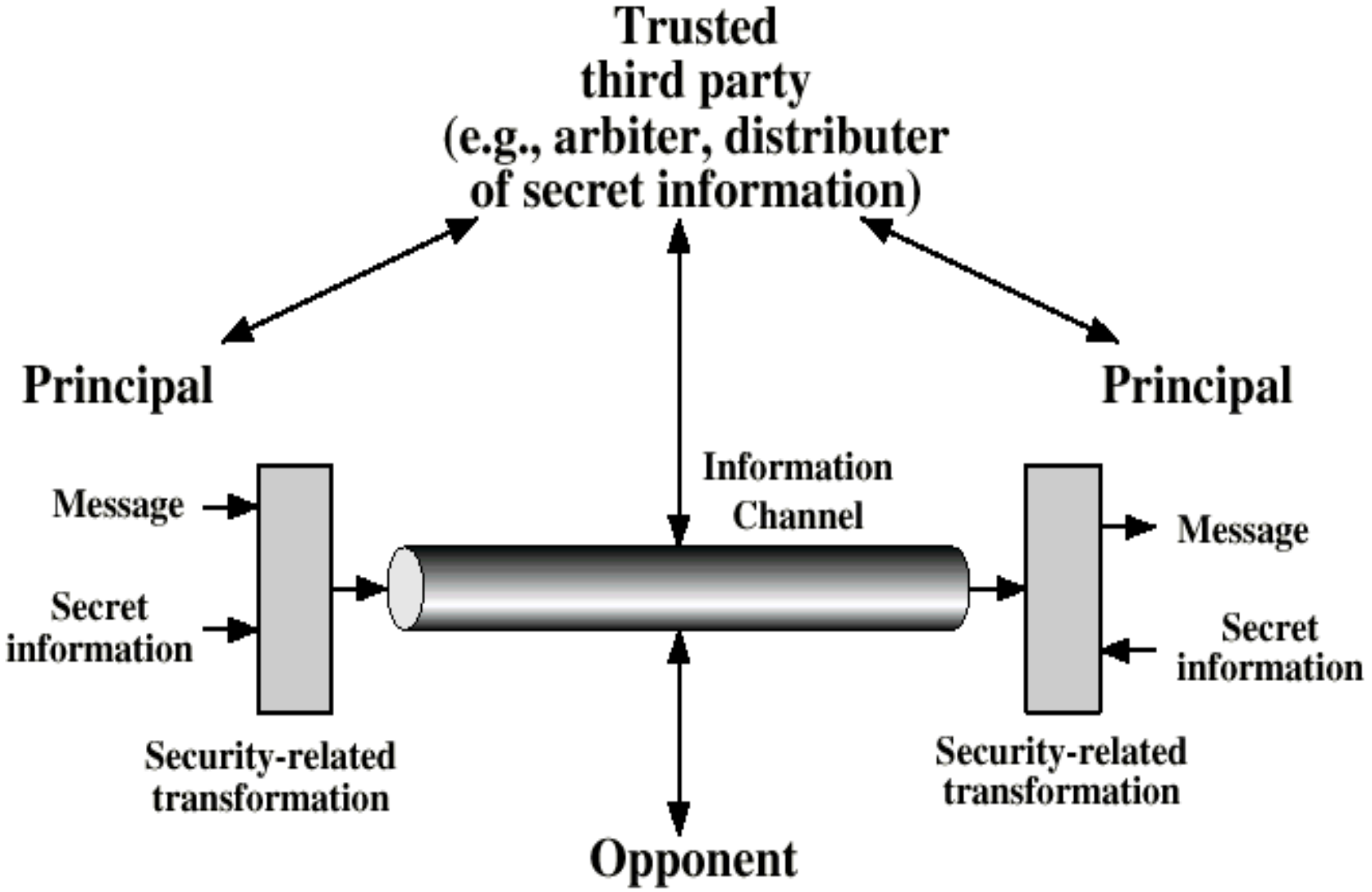
Application Layer Vulnerabilities

- Open design issues allow free use of application resources by unintended parties
- Backdoors and application design flaws bypass standard security controls
- Inadequate security controls force “all-or-nothing” approach, resulting in either excessive or insufficient access.
- Overly complex application security controls tend to be bypassed or poorly understood and implemented.
- Program logic flaws may be accidentally or purposely used to crash programs or cause undesired behavior

Application Layer Controls

- Application level access controls to define and enforce access to application resources. Controls must be detailed and flexible, but also straightforward to prevent complexity issues from masking policy and implementation weakness
- Standards, testing, and review of application code and functionality-A baseline is used to measure application implementation and recommend improvements
- IDS systems to monitor application inquiries and activity
- Some host-based firewall systems can regulate traffic by application, preventing unauthorized or covert use of the network.

Model for network security



Opponent – security threads and possible attacks

[Stallings'01]

Network Security - Part 3

- Introduction
- Threat landscape
- Cryptography and Network Security
- Network Security – Vulnerabilities/Protection
- Communication Security

Firewall

- System or combination of systems that supports an access control policy between two networks.

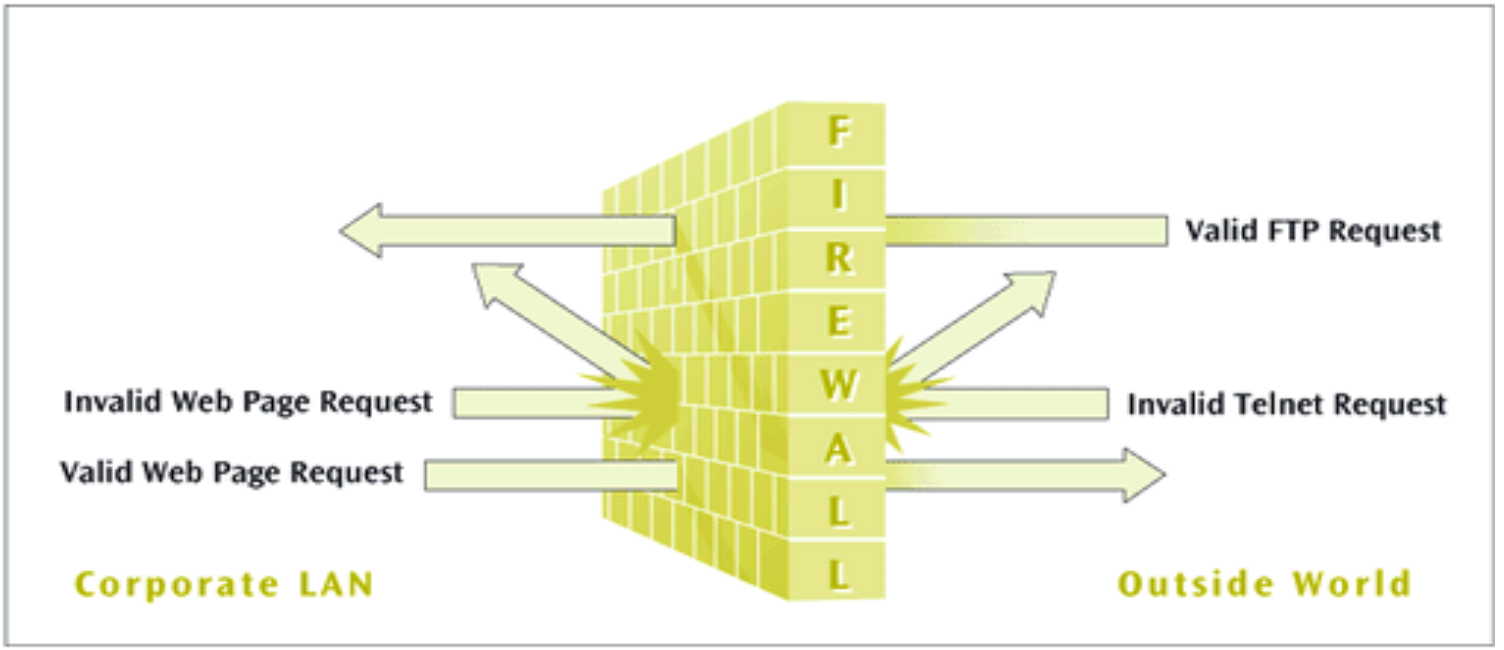
- All network packets entering firewall are filtered, or examined, to determine whether or not the network flow or the emitting/receiving hosts have authority to cross the boundaries.

- Firewall can limit types of transactions that enter system/network, as well as types of transactions that leave system/network.

- Firewall can be configured to block certain types or ranges of IP addresses, as well as certain types of TCP port numbers (applications).

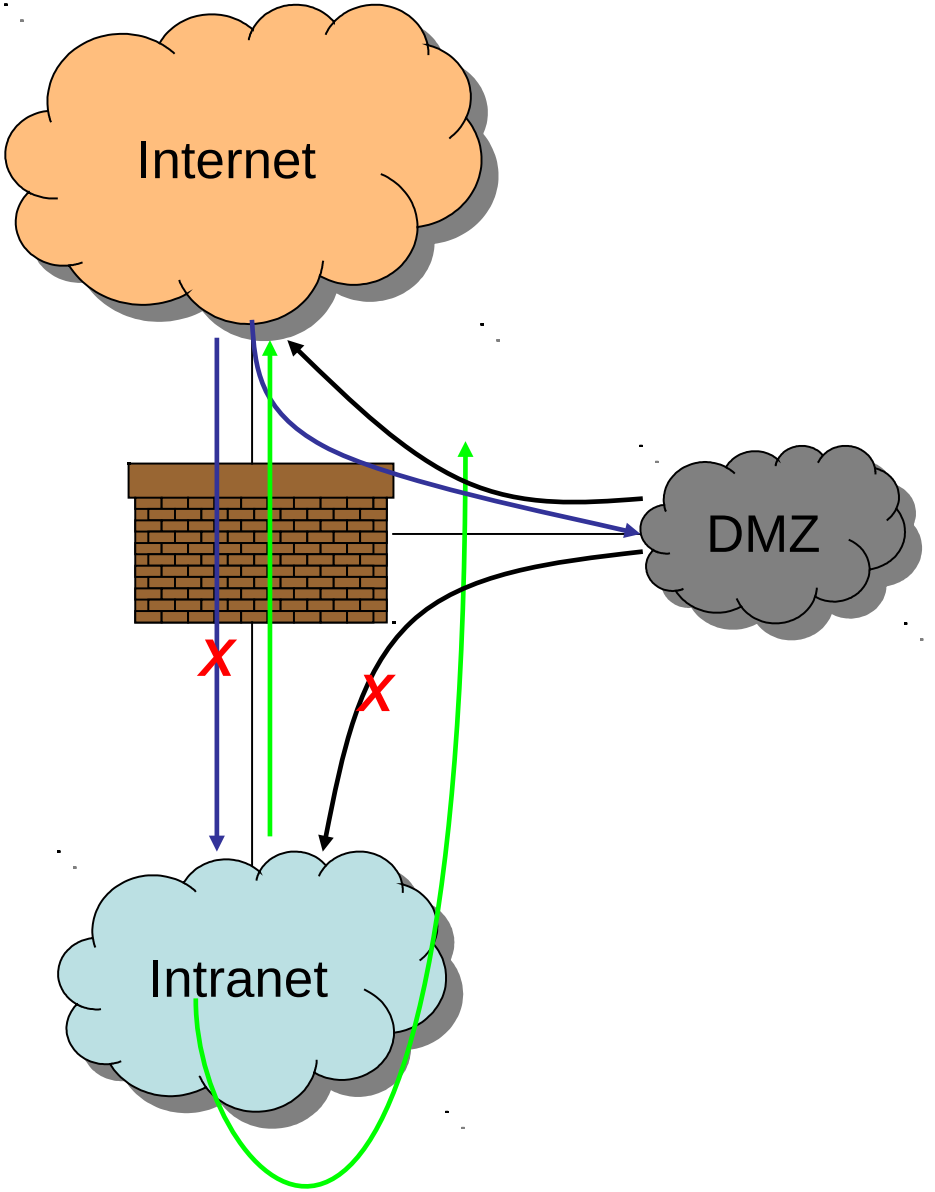
Firewall

Figure 13-8
A firewall as it stops certain internal and external transactions



Typical Firewall Configuration

- Internal hosts can access DMZ and Internet
- External hosts can access DMZ only, not Intranet
- DMZ hosts can access Internet only
- Advantages?
  - If a service gets compromised in DMZ it cannot affect internal hosts



Firewall

- *Packet filter* firewall - essentially router that has been programmed to filter out or allow in certain IP addresses or TCP port numbers.
- *Proxy server* - more advanced firewall that acts as doorman into corporate network. Any external transaction that requests something from corporate network must enter through proxy server.
- Proxy servers are more advanced but make external accesses slower.

Packet filtering

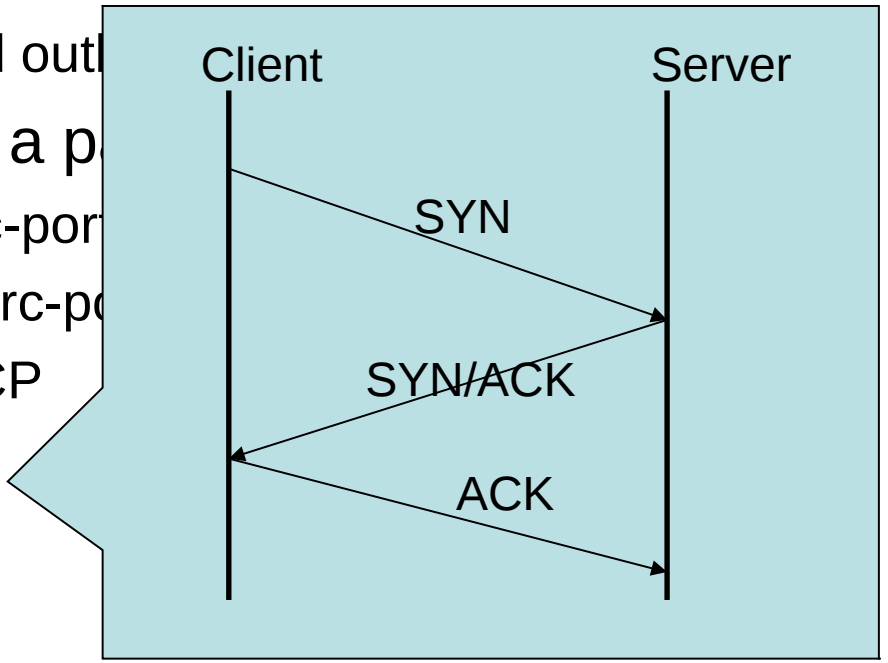
- Datagrams are identified by source address of host that issued message and destination address of remote.
- Filter - program that examines source address and destination address of incoming packet to firewall server.
- Filter tables - lists of addresses whose data packets and embedded messages are either allowed or prohibited from proceeding through the firewall.
- Filter tables can limit access of certain IP addresses to certain destination.

Packet filter

- A set of rules is applied to each incoming IP packet to decide whether it will be forwarded or discarded. The TCP/IP packet is parsed and filtered based on information that is usually found in packet headers:
  - Protocol number
  - Source and destination IP addresses
  - Source and destination port numbers
  - TCP connection flags (e.g. SYN and ACK flags)
- Most of packet filters are stateless:
  - each TCP/IP connection must be examined independently from what happened in the past
  - at the packet level, there is some statefulness:an outgoing connection with source port x opens the port x for incoming packets for the duration of the connection.

Sample Firewall Rule

- Allow SSH from external hosts to internal hosts
  - Two rules
    - Inbound and outb
  - How to know a p
    - Inbound: src-port
    - Outbound: src-p
    - Protocol=TCP
  - Ack Set?
  - Problems?



Rule	Dir	Src Addr	Src Port	Dst Addr	Dst Port	Proto	Ack Set?	Action
SSH-1	In	Ext	> 1023	Int	22	TCP	Any	Allow
SSH-2	Out	Int	22	Ext	> 1023	TCP	Yes	Alow

Default Firewall Rules

- Egress Filtering
  - Outbound traffic from external address → Drop
  - Benefits?
- Ingress Filtering
  - Inbound Traffic from internal address → Drop
  - Benefits?
- Default Deny
  - Why?

Rule	Dir	Src Addr	Src Port	Dst Addr	Dst Port	Proto	Ack Set?	Action
Egress	Out	Ext	Any	Ext	Any	Any	Any	Deny
Ingress	In	Int	Any	Int	Any	Any	Any	Deny
Default	Any	Any	Any	Any	Any	Any	Any	Deny

Packet Filters

- Advantages
  - Transparent to application/user
  - Simple packet filters can be efficient
- Disadvantages
  - Usually fail open
  - Very hard to configure the rules
  - Doesn’t have enough information to take actions
    - Does port 22 always mean SSH?
    - Who is the user accessing the SSH?

Alternatives

- Stateful packet filters
  - Keep the connection states
  - Easier to specify rules
  - More popular
  - Problems?
    - State explosion
    - State for UDP/ICMP?

Application Gateways

- Also called application level filters
- Port level filters determine legitimacy of party asking for information, application level filters assures validity of what they are asking for.
- Application level filters examine entire request for data rather than source and destination addresses.
- Application gateways are concerned with what services or applications message is requesting in addition to who is making request.
- Once legitimacy of request has been established, only proxy clients and servers actually communicate with each other.

Firewalls - Pro

Centralized management of security. Can even be stealth

Scaleability

Providing auditing, monitoring and recording capabilities

Records/logs that can ease forensic or incident analysis

Mutualise administration and configuration on PoE rather than the entire network

Firewalls - Con

Network bottleneck / SPOF (Single Point Of Failure)

Critical element of security architecture

Complexity :

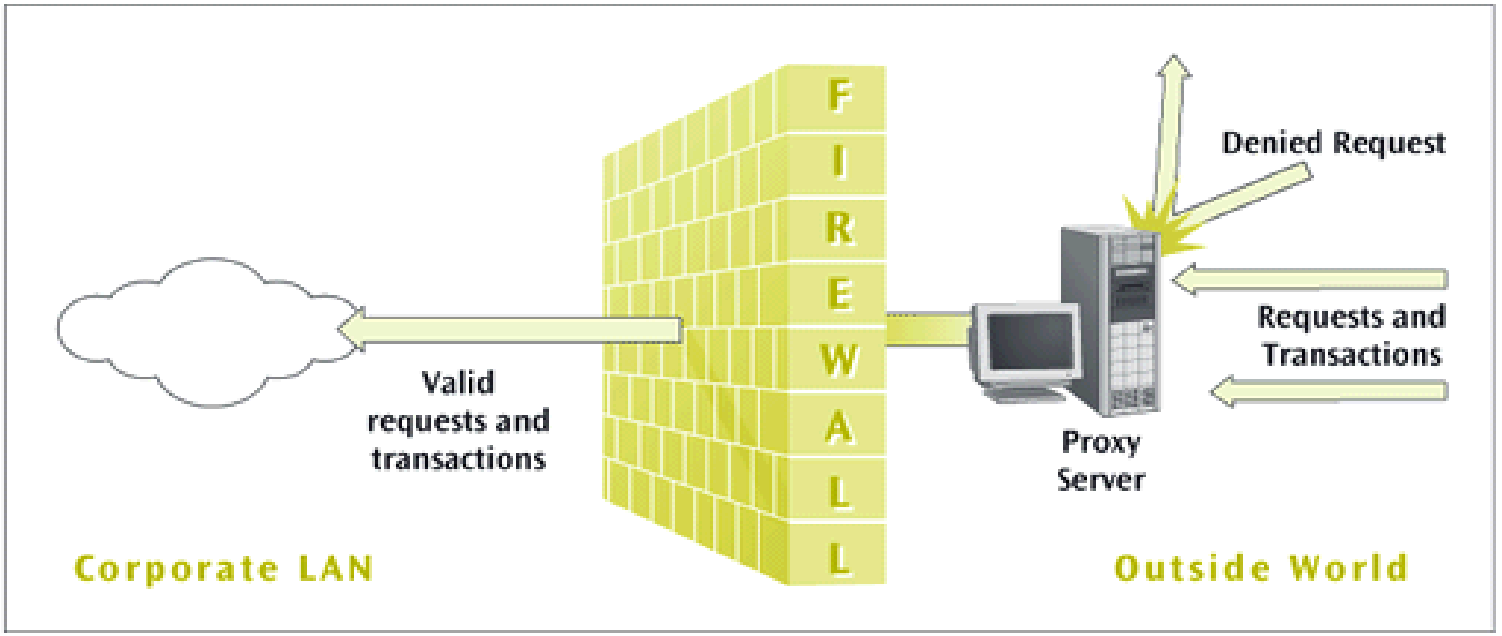
- Deep knowledge of filtered protocols (TCP/IP, HTTP, home-brewed services…)
- Require understanding of firewall features (interface with other filtering components , address translation…)

Alternatives

- Proxy Firewalls
  - Two connections instead of one
  - Either at transport level
    - SOCKS proxy
  - Or at application level
    - HTTP proxy
- Requires applications (or dynamically linked libraries) to be modified to use the proxy

Proxy Server

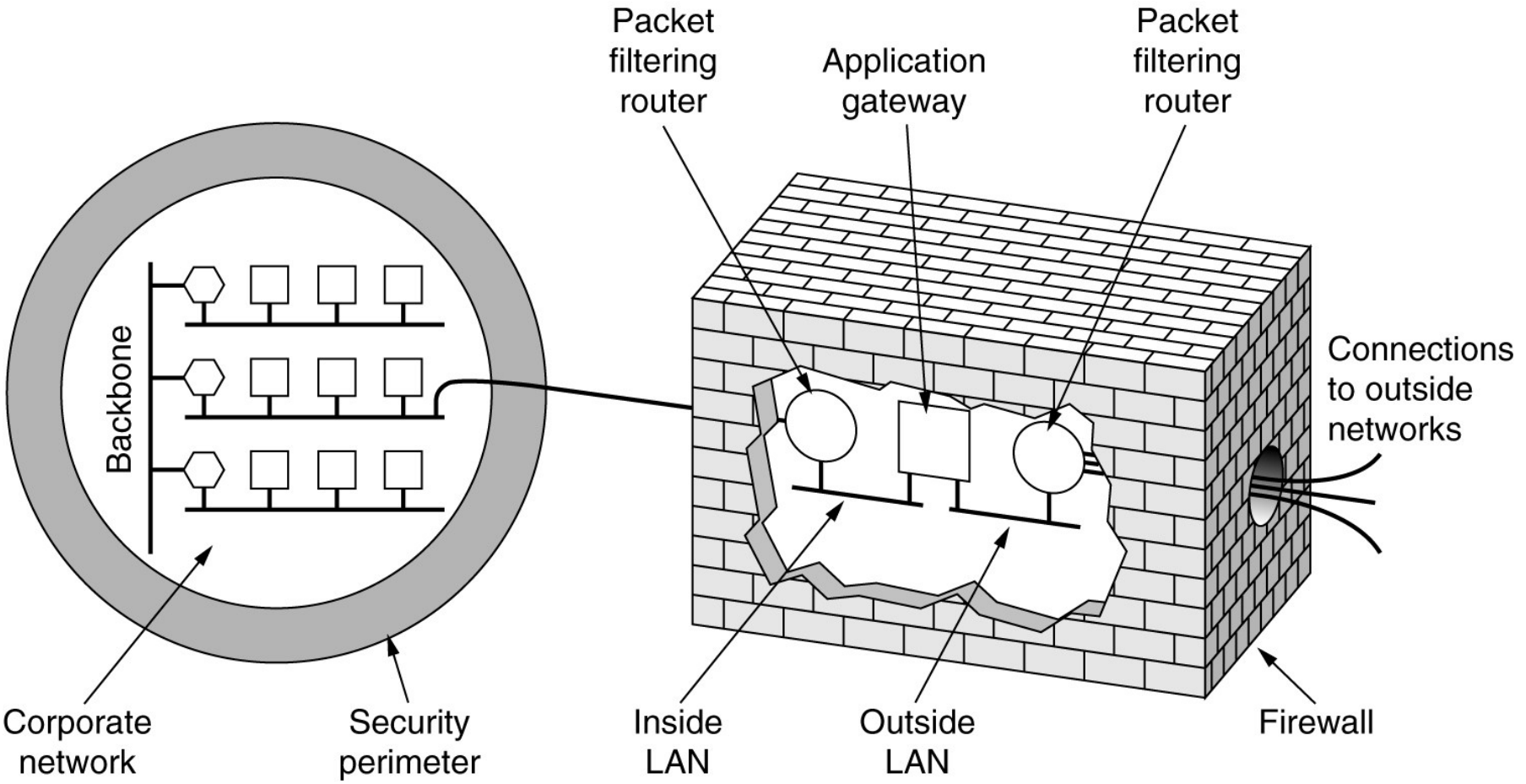
Figure 13-9
The proxy server sitting outside the protection of the corporate network



Proxy Firewall

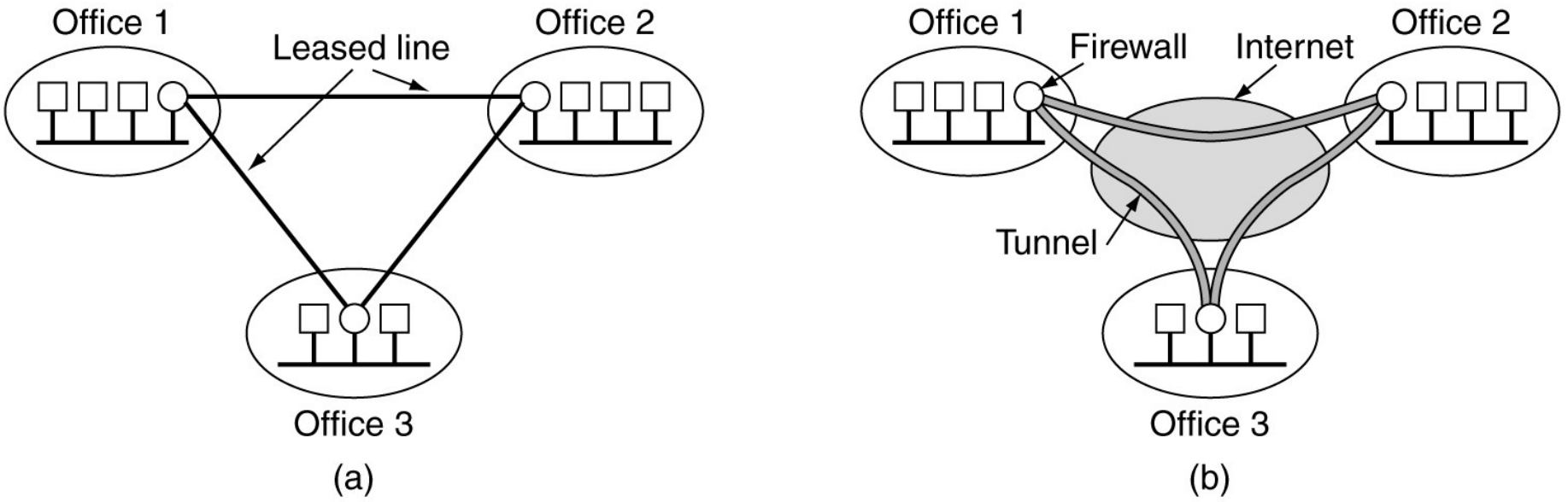
- Data Available
  - Application level information
  - User information
- Advantages?
  - Better policy enforcement
  - Better logging
  - Fail closed
- Disadvantages?
  - Doesn’t perform as well
  - One proxy for each application
  - Client modification

Internet Access Point



A firewall consisting of two packet filters and an application gateway.

Virtual Private Networks



(a) A leased-line private network. (b) A virtual private network.

VPN - categorization

Customer-managed VPN solution

- Layer 2: L2TP and PPTP
- Layer 3: IPSec

Provider-provisioned VPN solution

- Layer 3: MPLS-Based VPNs (RFC 2547bis)
- Layer 3: Non-MPLS-Based VPNs (Virtual Routers)
- Layer2: MPLS VPNs

MPLS Overview

- A forwarding scheme designed to speed up IP packet forwarding (RFC 3031)
- Idea: use a fixed length label in the packet header to decide packet forwarding
  - Label carried in an MPLS header between the link layer header and network layer header
- Support any network layer protocol and link layer protocol
- Simplify packet forwarding based on a fixed length label
- Enable explicit routing in IP networks
- Can be used for traffic management, QoS routing
- Enable fast restoration from failures.

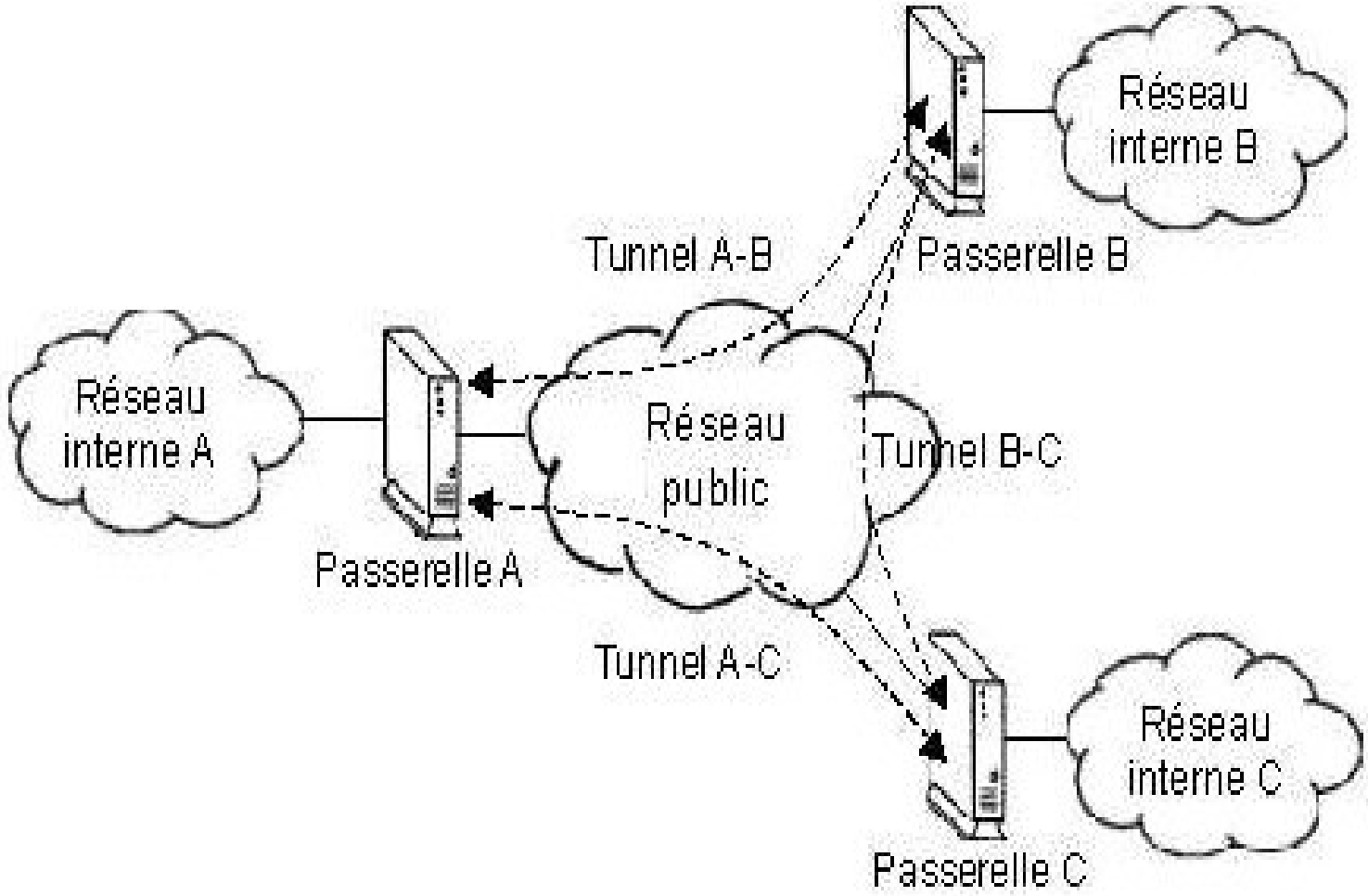
MPLS in a nutshell

- Simplify packet forwarding based on a fixed length label
- Enable explicit routing in IP networks
  - Can be used for traffic management, QoS routing
- Enable fast restoration from failures.

PPTP

- PPTP is essentially tunneling protocol that allows managers to choose whatever encryption or authentication technology they wish to hang off either end of the established tunnel.
- PPTP Microsoft tunneling protocol specific to Windows NT and remote access servers.
- PPTP concerned with secure remote access in that PPP-enabled clients would be able to dial into corporate network by Internet.

IPSEC & Virtual Private Networks



IPsec et Virtual Private Networks

- Protocole développé initialement dans le cadre d'IPv6. Il permet de chiffrer les informations au niveau réseau (IP).
- Standard IETF depuis 1995 (première version).
- Intègre un protocole de gestion des clés (IKE : Internet Key Exchange depuis 1998).
- L'avantage d'IPsec est qu'il permet le chiffrement des échanges de toutes les applications car il chiffre au niveau réseau.

IPsec

IP SECurity protocol issu d'une task force de l'IETF

- Authentification, confidentialité et intégrité (protection contre l'IP spoofing et le TCP session hijacking)
- Confidentialité

  Sécurisation au niveau de la couche transport (protection L3).

Algorithmes utilisés:

Authentification par signature DSS ou RSA

Intégrité par fonction de condensation (HMAC-MD5, HMAC-SHA-1, ...)

Confidentialité par chiffrement DES, 3-DES, AES

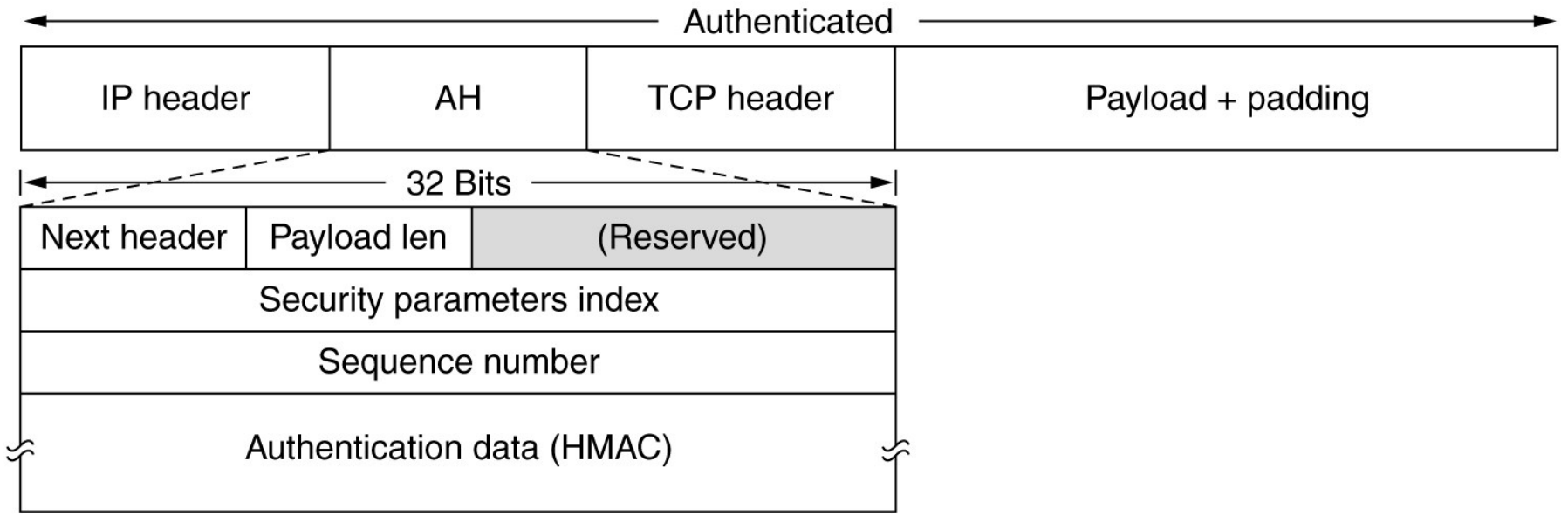
IPsec

- *Authentication Header* (AH) permettant d'authentifier les messages.
- *Encapsulating Security Payload* (ESP) permettant d'authentifier et de crypter les messages.

- mode **transport**: les machines source et destination sont les extrémités de la connexion sécurisée.

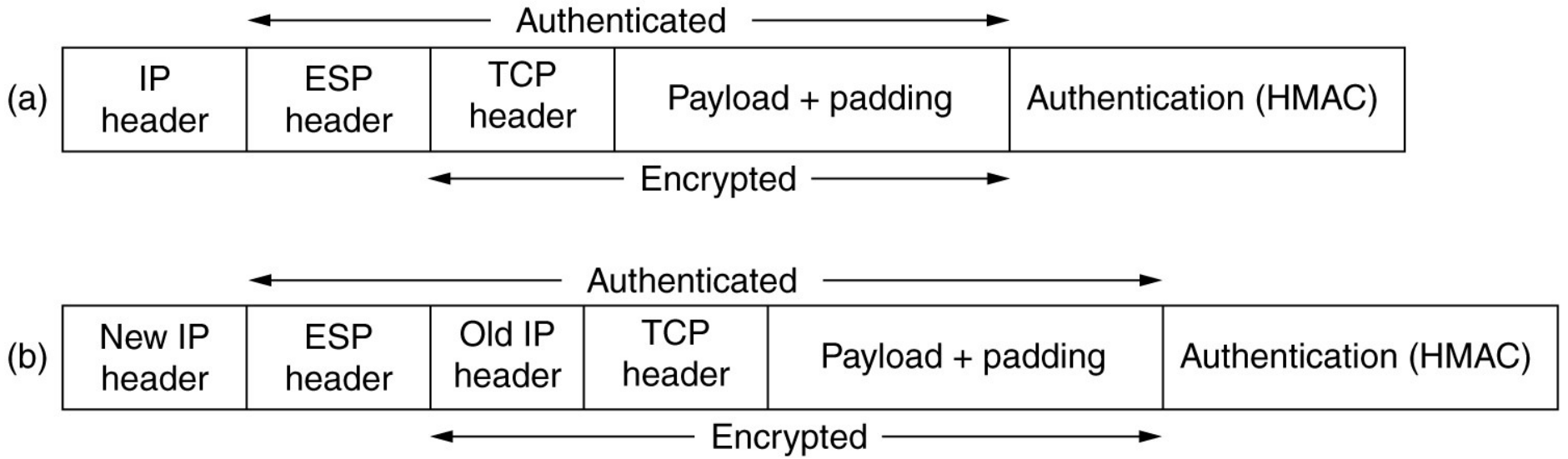
- mode **tunnel**: les extrémités de la connexion sécurisée sont des passerelles; les communications hôte à hôte sont encapsulées dans les entêtes de protocole de tunnel

IPsec



The IPsec authentication header in transport mode for IPv4.

IPsec

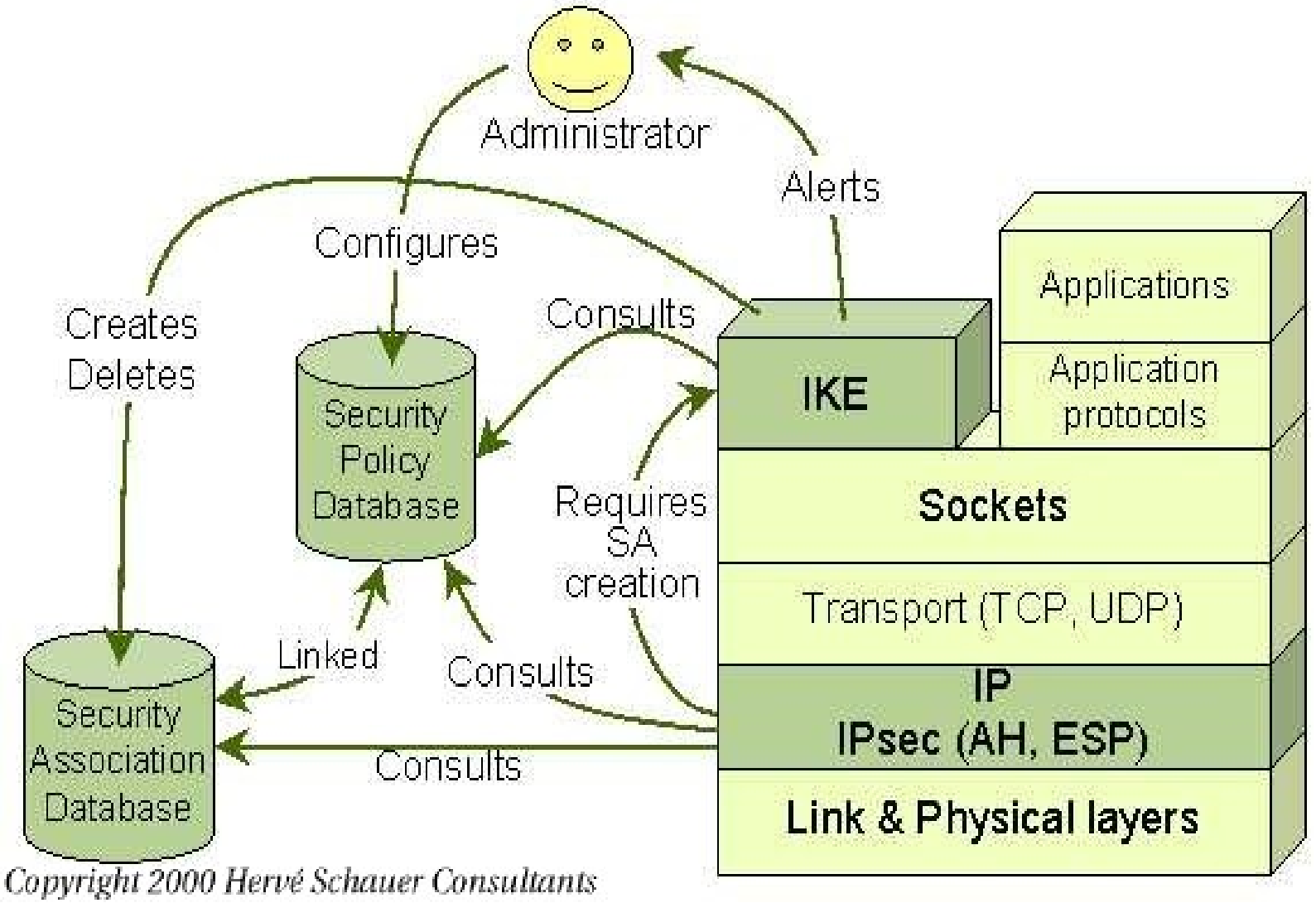


(a) ESP in transport mode. (b) ESP in tunnel mode.

IPsec

- IPsec est composé de plusieurs protocoles :
  - AH : Authentication Header qui permet d'authentifier les données
  - ESP : Encapsulating Security Payload qui permet de chiffrer et d'authentifier les données
  - IKE : Internet Key Exchange qui permet de gérer les clés de chiffrement et les SA
- À ces protocoles s'ajoutent :
  - Les associations de sécurité (SA : Security Associations) qui sont des structures de données permettant de stocker les paramètres relatifs à la sécurisation d'un type de communication
  - La politique de sécurité qui définit quel traitement appliquer à quel type de flux

IPSEC


Administrator
Alerts
Configures
Creates
Deletes
Consults
Security
Policy
Database
Requires
SA
creation
Linked
Consults
Security
Association
Database
Consults
Applications
IKE
Application
protocols
Sockets
Transport (TCP, UDP)
IP
IPsec (AH, ESP)
Link & Physical layers
Copyright 2000 Hervé Schauer Consultants

RFC 2367 : PF_KEY Interface
RFC 2401 (remplacée par la RFC 4301) : Security Architecture for the Internet Protocol
RFC 2402 (remplacée par les RFC 4302 et RFC 4305) : Authentication Header
RFC 2403 : The Use of HMAC-MD5-96 within ESP and AH
RFC 2404 : The Use of HMAC-SHA-1-96 within ESP and AH
RFC 2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC 2406 (remplacée par les RFC 4303 et RFC 4305) : Encapsulating Security Payload
RFC 2407 (remplacée par la RFC 4306) : IPsec Domain of Interpretation for ISAKMP (IPsec DoI)
RFC 2408 (remplacée par la RFC 4306) : Internet Security Association and Key Management Protocol (ISAKMP)
RFC 2409 (remplacée par la RFC 4306) : Internet Key Exchange (IKE)
RFC 2410 : The NULL Encryption Algorithm and Its Use With IPsec
RFC 2411 (remplacée par la RFC 6071) : IP Security Document Roadmap
RFC 2412 : The OAKLEY Key Determination Protocol
RFC 2451 : The ESP CBC-Mode Cipher Algorithms
RFC 2857 : The Use of HMAC-RIPEMD-160-96 within ESP and AH
RFC 3526 : More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)
RFC 3706 : A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers
RFC 3715 : IPsec-Network Address Translation (NAT) Compatibility Requirements
RFC 3947 : Negotiation of NAT-Traversal in the IKE
RFC 3948 : UDP Encapsulation of IPsec ESP Packets
RFC 4106 : The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
RFC 4301 (remplace la RFC 2401) : Security Architecture for the Internet Protocol
RFC 4302 (remplace la RFC 2402) : IP Authentication Header
RFC 4303 (remplace la RFC 2406) : IP Encapsulating Security Payload (ESP)
RFC 4304 : Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)
RFC 4305 : Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)
RFC 4306 (remplace les RFC 2407, RFC 2408, et RFC 2409) : Internet Key Exchange (IKEv2) Protocol
RFC 4307 : Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)
RFC 4308 : Cryptographic Suites for IPsec
RFC 4309 : Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)
RFC 4478 : Repeated Authentication in Internet Key Exchange (IKEv2) Protocol
RFC 4543 : The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH
RFC 4555 : IKEv2 Mobility and Multihoming Protocol (MOBIKE)
RFC 4621 : Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol
RFC 4718 : IKEv2 Clarifications and Implementation Guidelines
RFC 4806 : Online Certificate Status Protocol (OCSP) Extensions to IKEv2
RFC 4809 : Requirements for an IPsec Certificate Management Profile
RFC 4835 (remplace la RFC 4305) : Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)
RFC 4945 : The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX
RFC 6071 (remplace la RFC 2411) : IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap

SSL / TLS

The terms, Secure Socket Layer (SSL) and Transport Layer Security (TLS) are often used interchangeably.

SSL – originally developed by Netscape
TLS – standardized by IETF

SSL v3.1 ~ TLS v1.0.

SSL – Secure Sockets Layer

- SSL mis en oeuvre initialement par la société Netscape Communications mais qui a depuis été repris par l'IETF sous le nom TLS (Transport Layer Security)
- Repose sur le protocole TCP avec des numéros de port spécifiques : HTTPS (443), NNTPS (563), LDAPS (636), FTPS(989 pour les données, 990 pour le contrôle), telnets (992), IMAPS (993), POP3S (995)
- Principale utilisation : HTTPS

SSL – Secure Sockets Layer

- C'est un système qui permet d'échanger des informations entre 2 ordinateurs de façon sûre. SSL assure 3 choses:
- Confidentialité: Il est difficile d'espionner les informations échangées.
- Intégrité: Il est difficile de truquer les informations échangées.
- Authentification: Il permet de s'assurer de l'identité du programme, de la personne ou de l'entreprise avec laquelle on communique.

SSL Protocol Stack

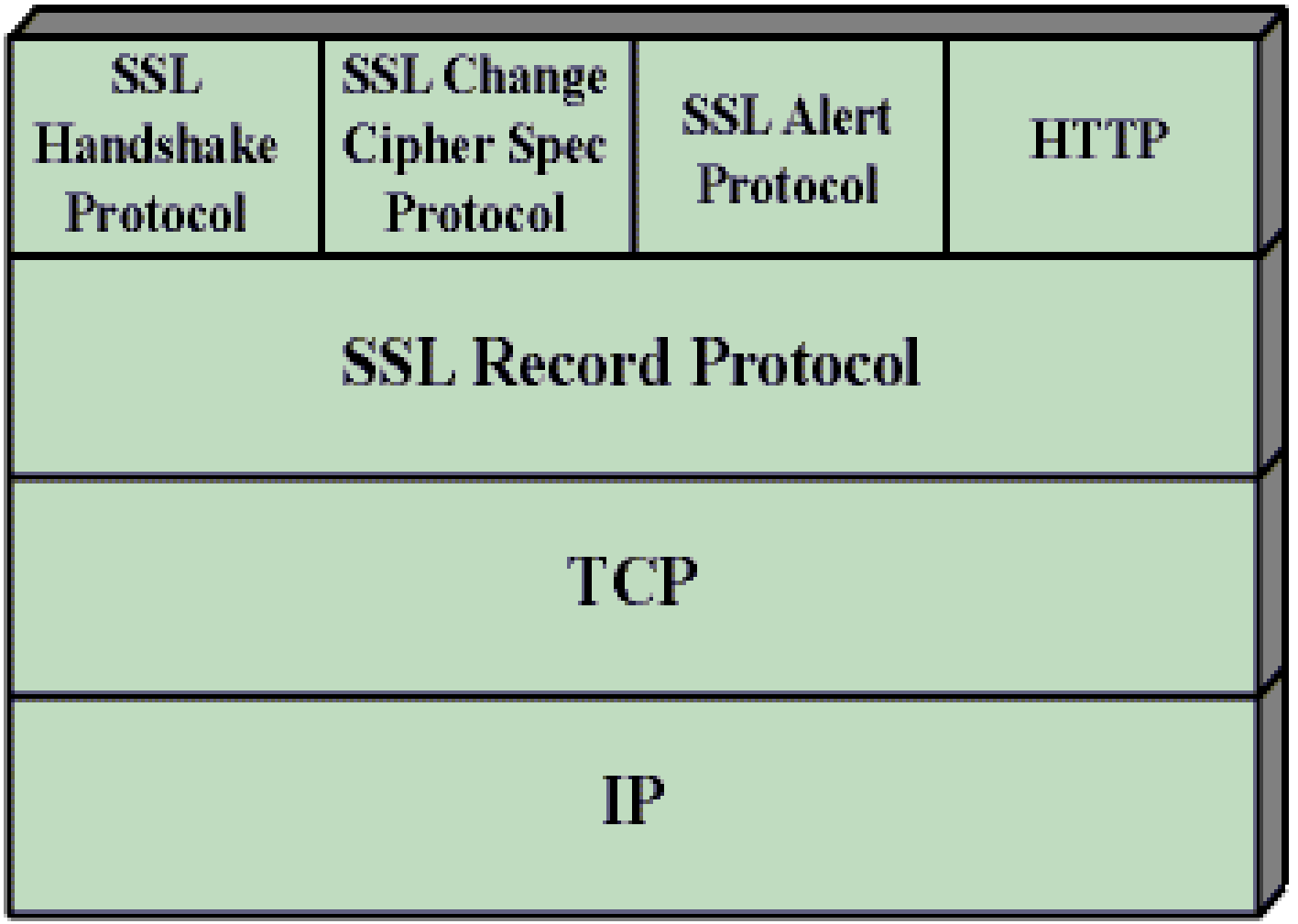


Figure 14.2 SSL Protocol Stack

SSL – Secure Socket Layer (2)

- SSL Handshake protocol: avant de communiquer, les 2 programmes SSL négocient des clés et des protocoles de chiffrement communs.
    - Version SSL, méthodes de chiffrement et de compression, certificats.
- SSL Record protocol: Une fois négociés, ils chiffrent toutes les informations échangées et effectuent divers contrôles.
- HTTPS: c'est HTTP+SSL
- Mais aussi FTTPS, SSH, Stunnel

SSL – Secure Sockets Layer (3)

•Les services de sécurité fournis sont :
- Confidentialité des données transmises (RC4, DES, 3DES...)
- Authentification et intégrité des données (MD5, SHA-1)
- Authentification optionnelle des interlocuteurs par cryptographie à clefs publiques

•Le serveur s'authentifie à l'aide d'un certificat. Bien qu'optionnelle, cette authentification est pratiquement toujours réalisée

•L'authentification du client n'a pas été prévue à l'origine

•Cependant :
- Depuis la version 3.0 du protocole, il est possible d'authentifier le client à l'aide d'un certificat
- On peut toujours authentifier le client à l'aide d'un autre mécanisme dans la connexion SSL (mot de passe)

•Les certificats sont au format X509v3
•A lire : The SSL Protocol Version 3.0, Internet draft expiré, novembre 1996 : http://www.netscape.com/eng/ssl3/draft302.txt
•
•The TLS Protocol Version 1.0, RFC 2246 : http://www.ietf.org/rfc/rfc2246.txt

SSL Handshake

Negotiate the cipher suite

Establish a shared session key

Authenticate the server (optional)

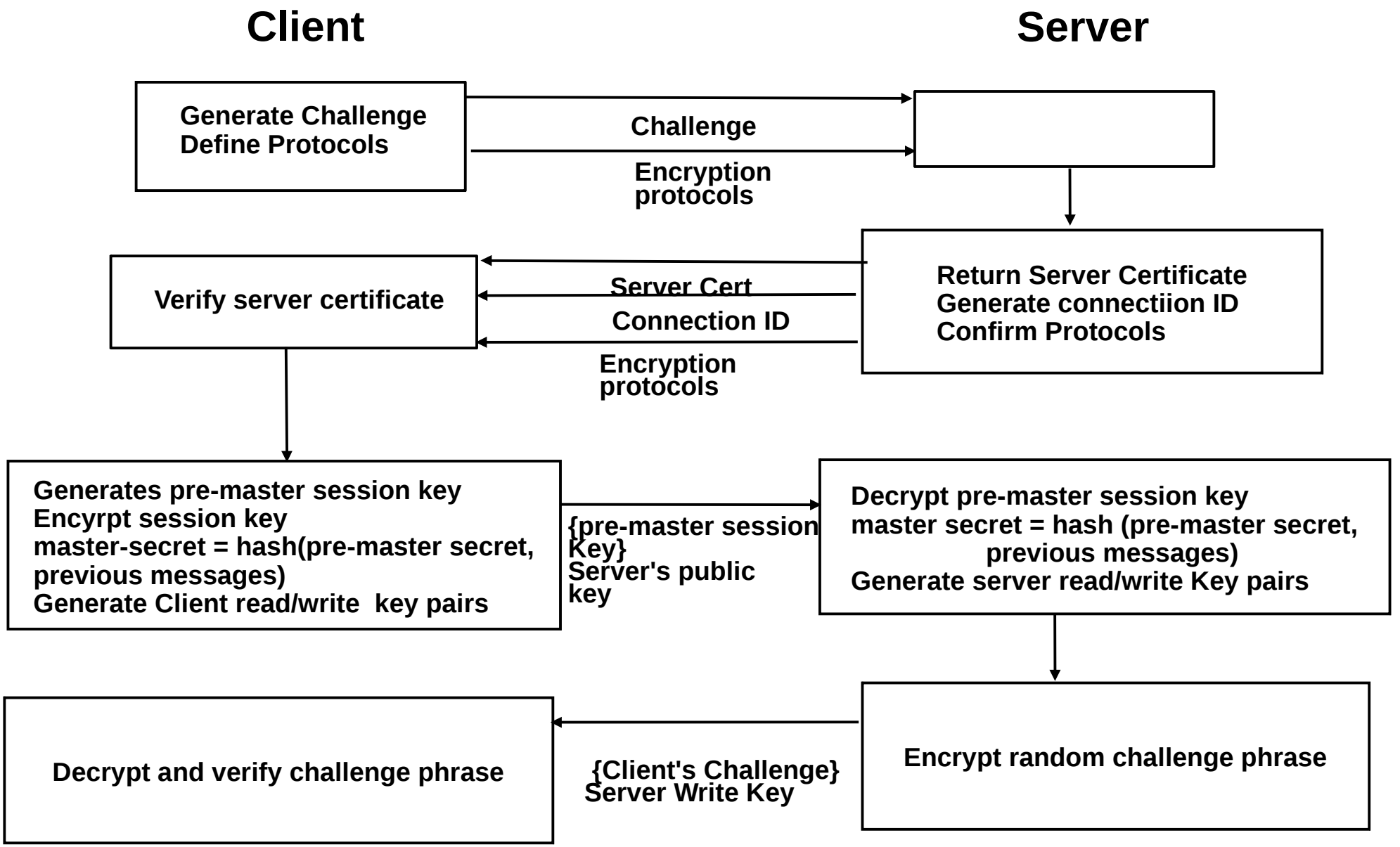
Authenticate the client (optional)

Authenticate previously exhanged data

SSL handshake details

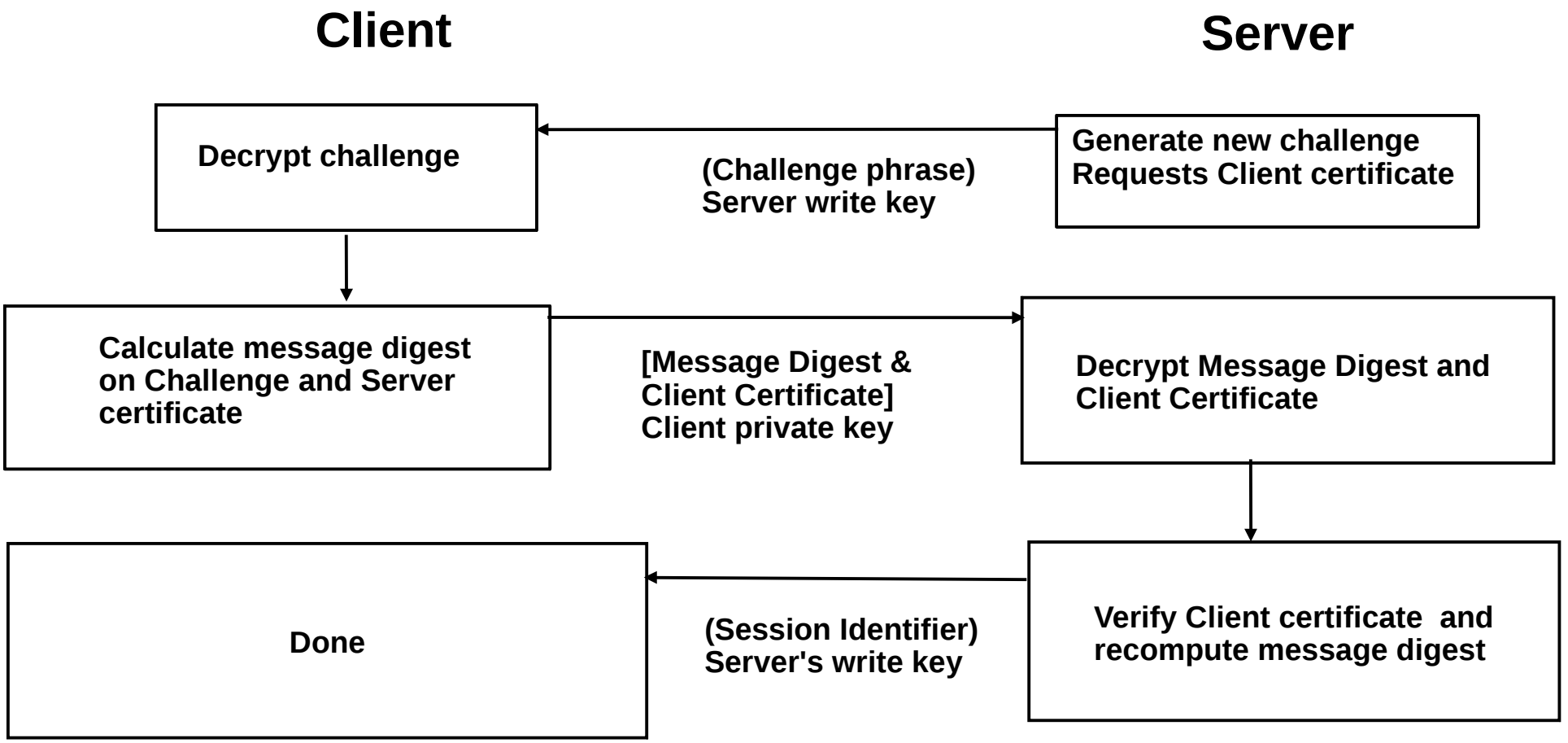
- Client hello:
  - Client challenge, client nonce
  - Available cipher suites (eg RSA + RC4/40 + MD5)
- Server hello:
  - Server certificate, server nonce
  - Connection ID
  - Selected cipher suite
- Server adapts to client capabilities
- Optional certificate exchange to authenticate server/client
  - Commercial sites only use server authentication

SSL Handshake - details



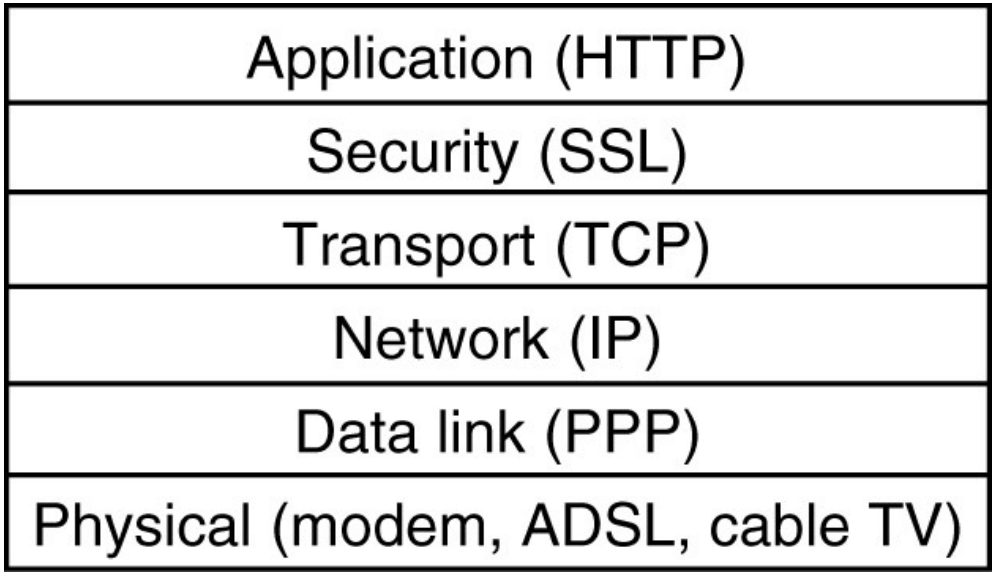
SSL Handshake

Client Authentication

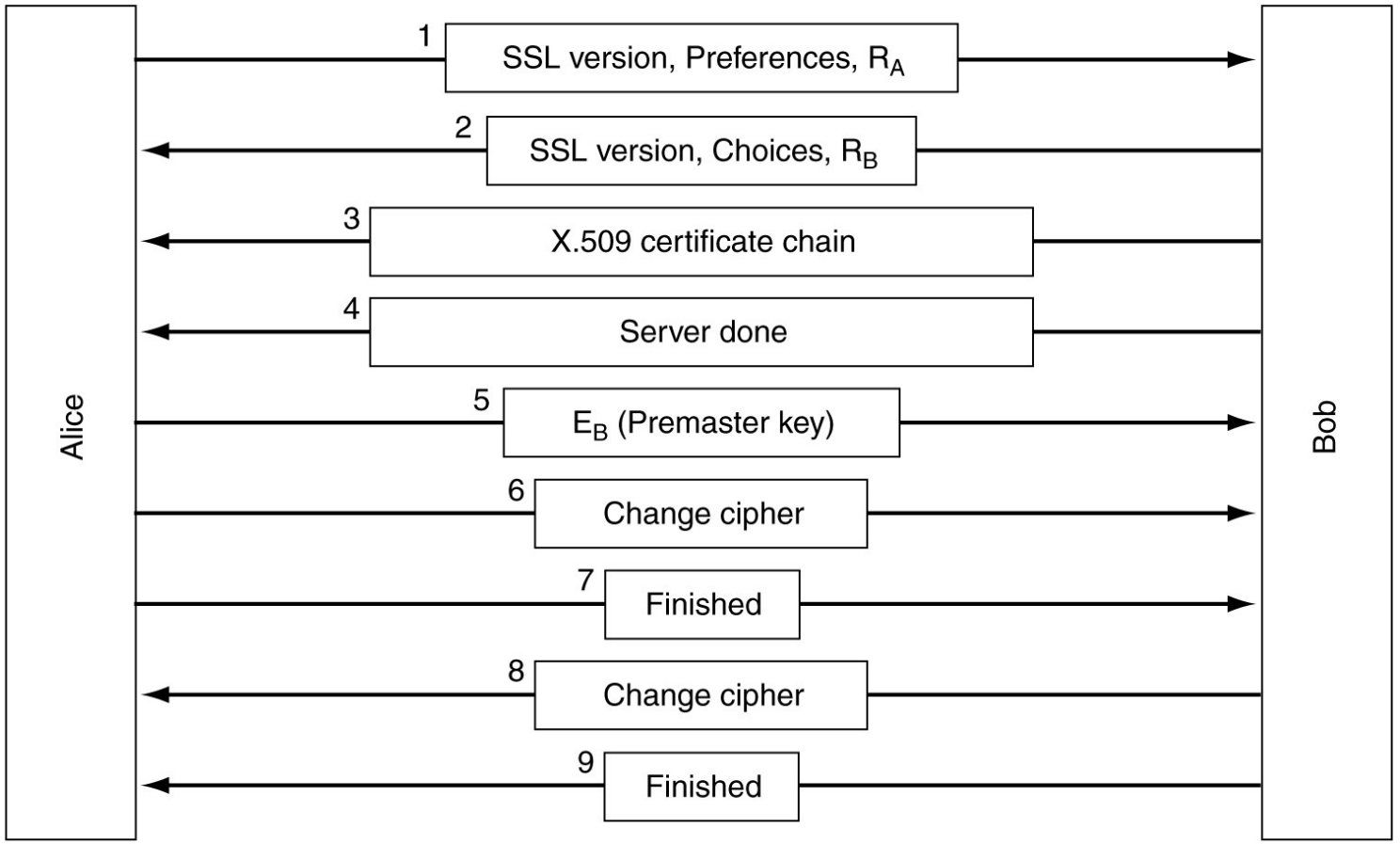


SSL - Secure Sockets Layer

Layers (and protocols) for a home user browsing with SSL.

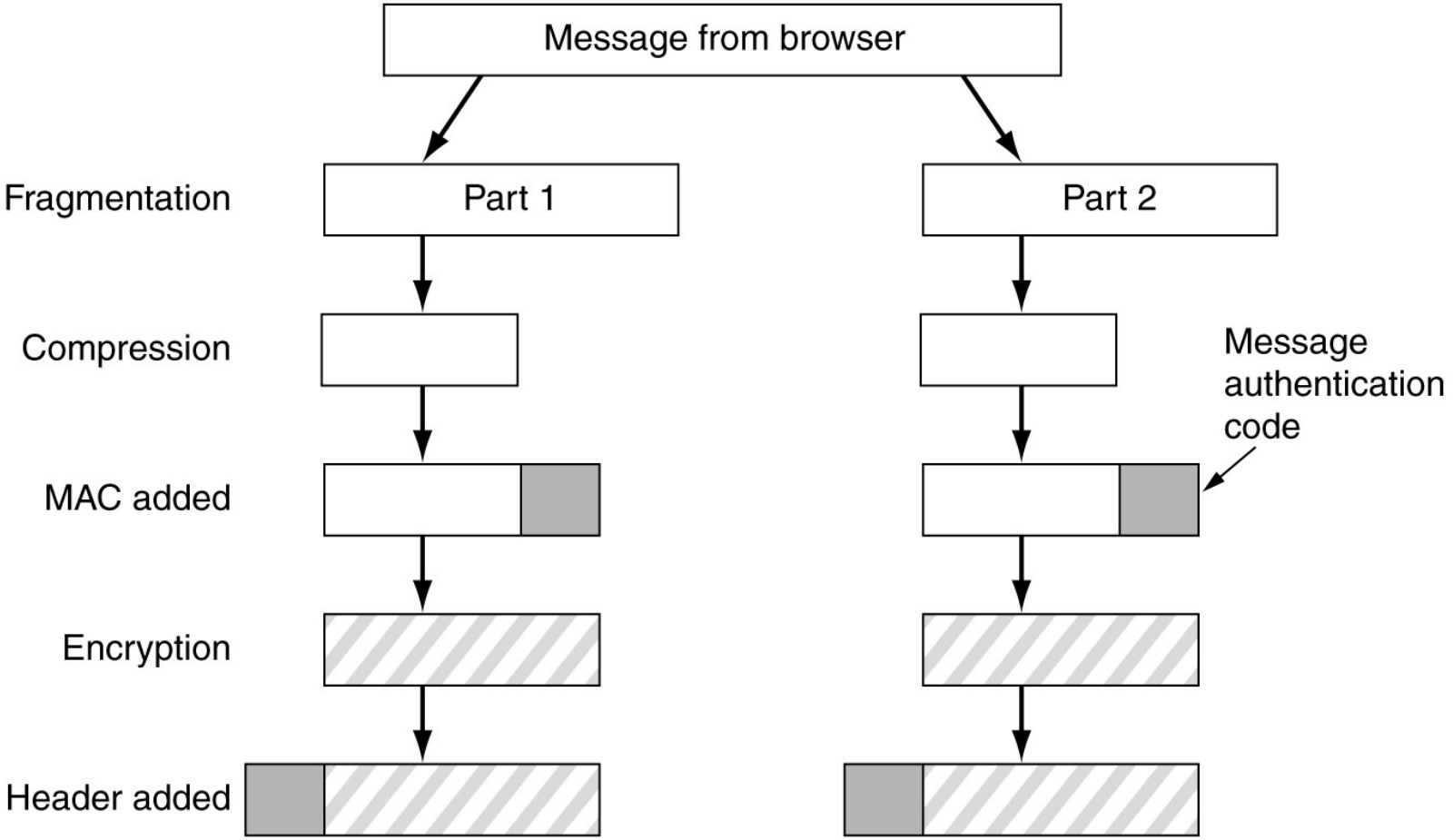


SSL



A simplified version of the SSL connection establishment subprotocol.

SSL



Data transmission using SSL.

Secure Server Design

- Use TLS or Other Strong Transport Everywhere
- Do Not Provide Non-TLS Pages for Secure Content
- Do Not Mix TLS and Non-TLS Content
- Use "Secure" Cookie Flag
- Keep Sensitive Data Out of the URL
- Prevent Caching of Sensitive Data
- Use HTTP Strict Transport Security
- Use Public Key Pinning

Secure Certificate

- Use Strong Keys & Protect Them
- Use Fully Qualified Names in Certificates
- Do Not Use Wildcard Certificates
- Do Not Use RFC 1918 Addresses in Certificates
- Use an Appropriate Certification Authority for the Application's User Base
- Always Provide All Needed Certificates
- Be aware of and have a plan for the SHA-1 deprecation plan

Server Protocol and Cipher Configuration

- Only Support Strong Protocols
- Prefer Ephemeral Key Exchanges
- Only Support Strong Cryptographic Ciphers
- Support TLS-PSK and TLS-SRP for Mutual Authentication
- Only Support Secure Renegotiations
- Disable Compression

Assessing your SSL config

- O-Saft - OWASP SSL advanced forensic tool
  - https://www.owasp.org/index.php/O-Saft
- SSLScan - Fast SSL Scanner
  - http://sourceforge.net/projects/sslscan/
- SSLyze
  - https://github.com/iSECPartners/sslyze
- SSL Audit
  - http://www.g-sec.lu/sslaudit/sslaudit.zip
- Openssl
  - Openssl cyphers

Exercice – Client security

Investigate and describe your settings (configuration, certificate mgt, ...)

- for Windows :
  - Certificate manager – certmgr.msc
  - Registry
  - IE Certificate Store
- for Firefox
- for Chromium

SSH- Secure SHell

- SSH is a protocol for secure remote login and other secure network services over an insecure network
- Specified in a set of Internet drafts

SSH

- Fonctionnalités
  - Connexion à distance (remplaçant sécurisé de rlogin, rsh et telnet)
  - Transfert de fichiers (scp = remplaçant sécurisé de rcp)
  - Redirection de sessions X11
  - Redirections de ports TCP
- Services de sécurité
  - Authentification du serveur par clef publique (pas de certificat)
  - Authentification du client par clef publique, mot de passe (transmis chiffré) ou liste de machines de confiance (.rhost, hosts.equiv)
  - Chiffrement des données échangées (IDEA, DES, 3DES, aRC4, Blowfish)
  - Compression optionnelle des données échangées
  - Pas de mécanisme spécifique d'authentification des données

SSH

- Initialement, programme écrit par Tatu Ylönen, chercheur au Laboratory of Information Processing Science à Helsinki.
- Version 1.5 du protocole publiée sous forme d'Internet draft en novembre 1995
- Création de la société SSH Communications Security (www.ssh.fi) en 1995 qui distribue les versions gratuites
- Data Fellows (www.datafellows.com) possède la licence pour les versions commerciales.
- Création d'OpenSSH en 1999 par les développeurs du projet OpenBSD : version 100% libre (license BSD)
- Unix : Versions fournies par SSH Communications Security :
- OpenSSH du projet OpenBSD :http://www.openssh.com : libre pour tout usage
- Windows : F-Secure SSH : http://www.datafellows.com/f-secure/
- TTSSH: extension pour Teraterm : http://www.zip.com.au/~roca/ttssh.html

Protocole SSH (connexion sécurisée)

- Interfaçage de terminaux et applications à travers Internet. Fournit les règles de base pour permettre de relier un client (système composé d'un affichage et d'un clavier) à un interpréteur de commande (côté serveur).
- S'appuie sur une connexion TCP pour envoyer des données au format ASCII codées sur 8 bits entre lesquelles s'intercalent des séquences de contrôle Telnet. Il fournit ainsi un système orienté communication, bi-directionnel (half-duplex), codé sur 8 bits facile à mettre en oeuvre.

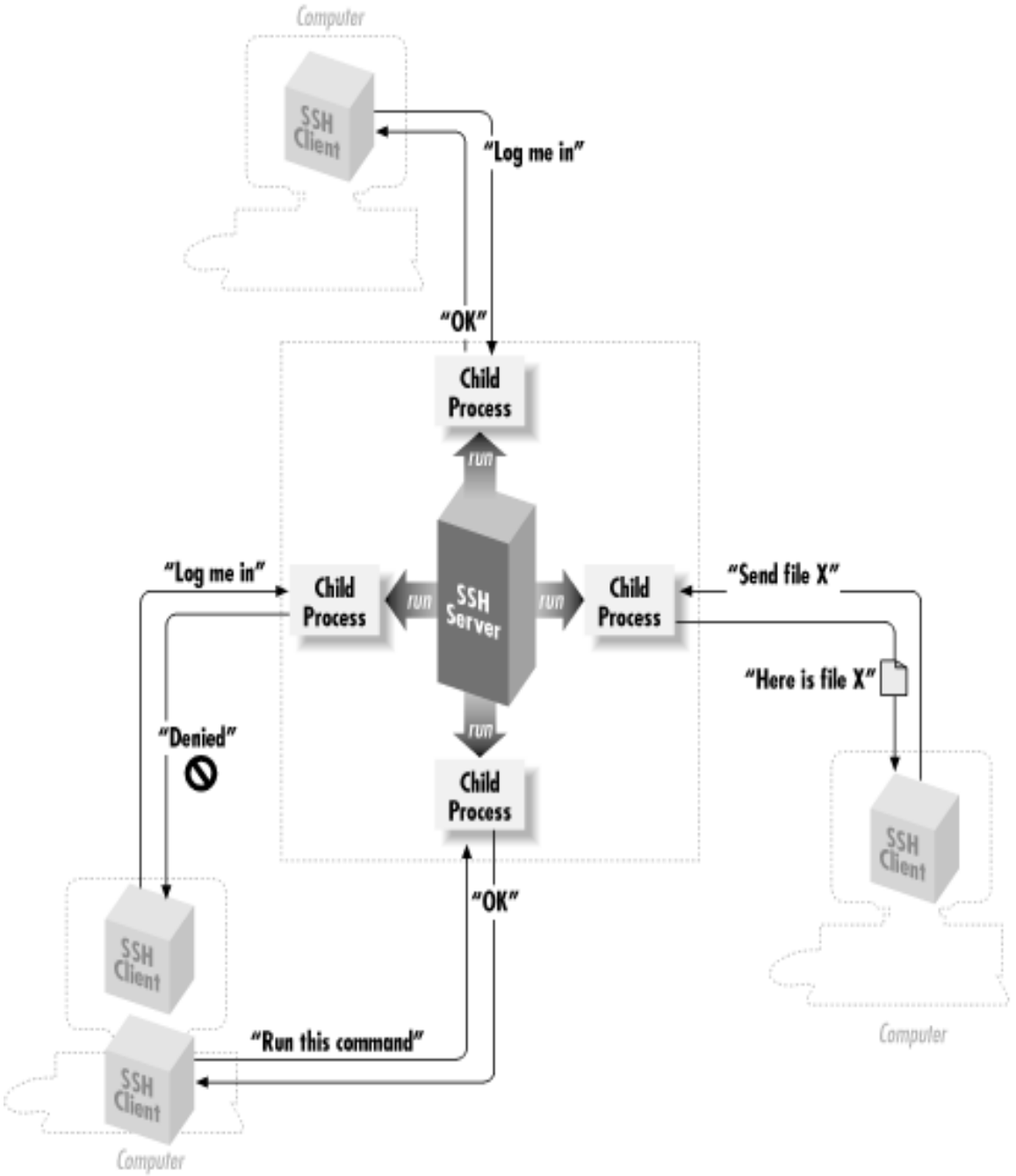
- Trois concepts fondamentaux : paradigme du terminal réseau virtuel (NVT, Network Virtual Terminal) ; principe d'options négociées ; règles de négociation.
- Base et appui pour certains autres protocoles de la suite TCP/IP (FTP, SMTP, POP3, ...).
- Sécurité : Pas d'authentification Transfert de données non sûr, en clair sur le réseau (de manière non chiffrée)
- A lire : RFC 854, tandis que les nombreuses options sont décrites par les RFC 855 à 861. http://www.commentcamarche.net/contents/internet/telnet.php3

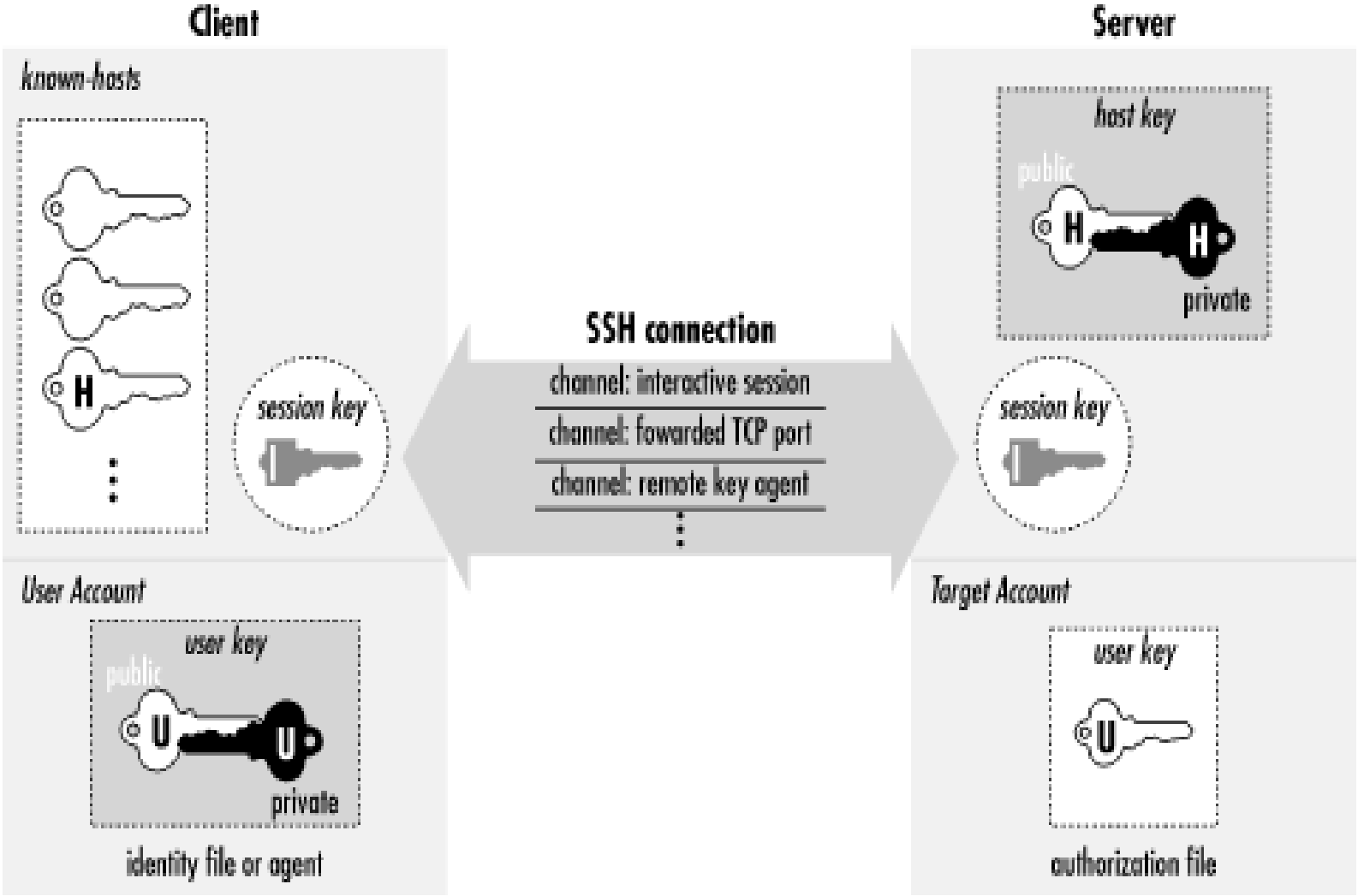
Major SSH components

- SSH Transport Layer Protocol
  - provides server authentication, confidentiality, and integrity services
  - it may provide compression too
  - runs on top of any reliable transport layer (e.g., TCP)
- SSH User Authentication Protocol
  - provides client-side user authentication
  - runs on top of the SSH Transport Layer Protocol
- SSH Connection Protocol
  - multiplexes the secure tunnel provided by the SSH Transport Layer and User Authentication Protocols into several logical channels
  - these logical channels can be used for a wide range of purposes
    - secure interactive shell sessions
    - TCP port forwarding
    - carrying X11 connections

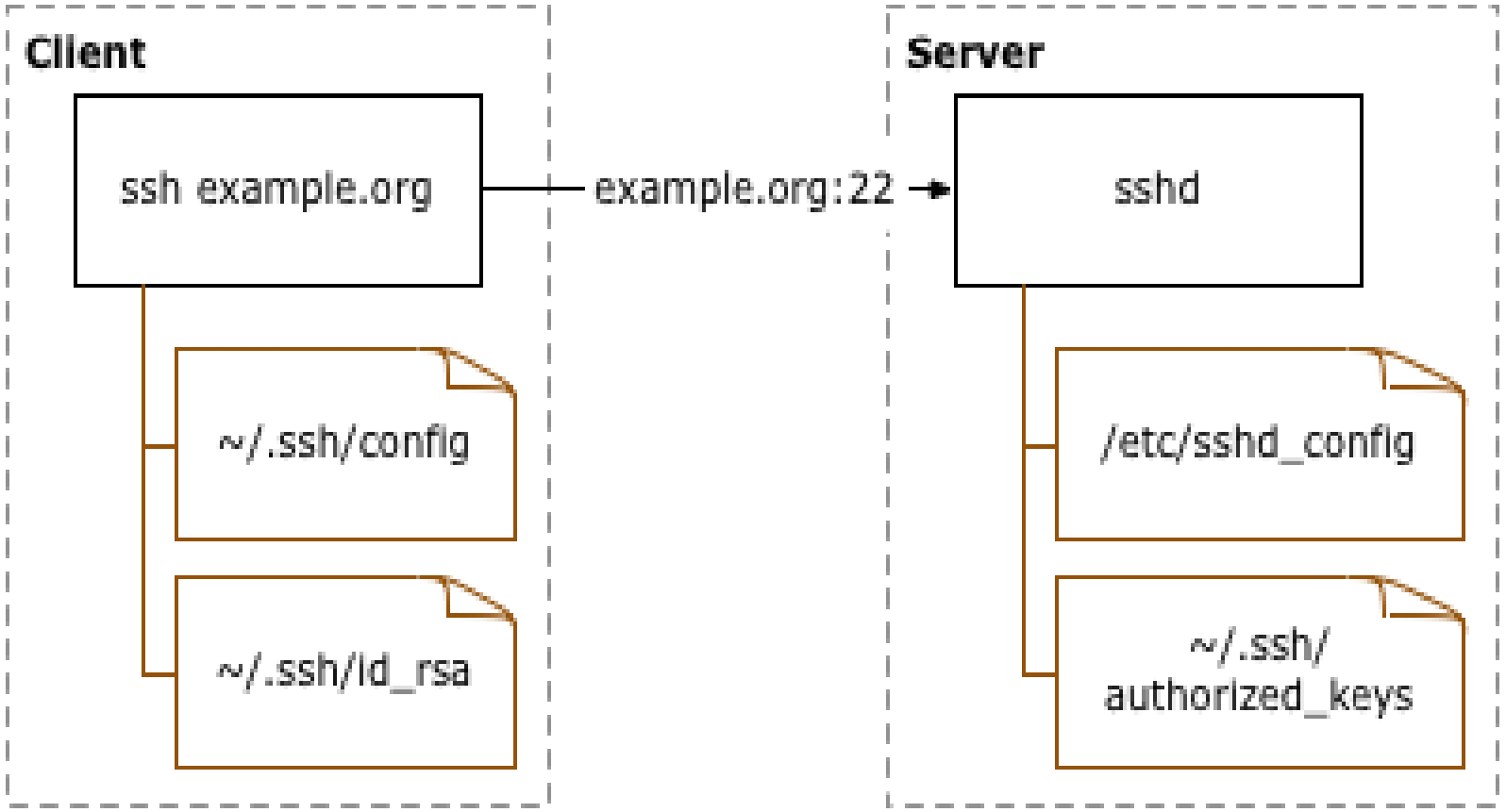
SSH security features

- strong algorithms
  - uses well established strong algorithms for encryption, integrity, key exchange, and public key management
- large key size
  - requires encryption to be used with at least 128 bit keys
  - supports larger keys too
- algorithm negotiation
  - encryption, integrity, key exchange, and public key algorithms are negotiated
  - it is easy to switch to some other algorithm without modifying the base protocol


Computer
SSH Client
"Log me in"
"OK"
Child Process
run
"Log me in"
Child Process
run
SSH Server
run
Child Process
"Send file X"
"Here is file X"
"Denied"
Child Process
SSH Client
"OK"
SSH Client
"Run this command"
SSH Client
Computer
Computer

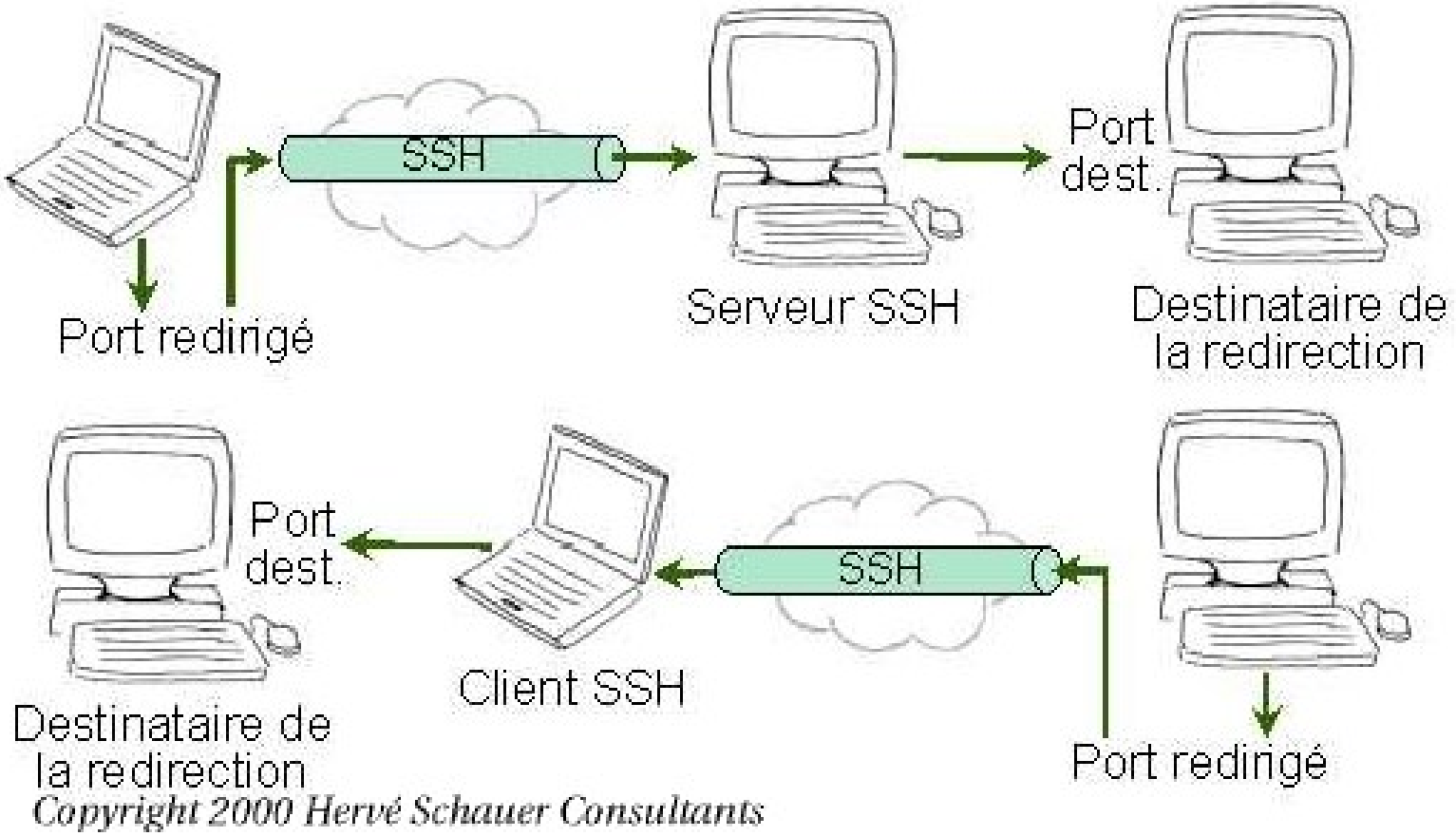

Client
known-hosts
H
session key
User Account
user key
public
U
U
private
identity file or agent
SSH connection
channel: interactive session
channel: fowarded TCP port
channel: remote key agent
Server
host key
public
H
H
private
session key
Target Account
user key
U
authorization file

Authentification avec clé ssh



SSH(port 22)

La redirection de port TCP permet de chiffrer certaines connexions et d'accéder à des machines non accessibles en direct C'est utilisable de façon symétrique



Other mechanism

- WS Secure message
  - Security information along with message header used to establish identity
  - Each message individually secured
  - End to end security is feasible
- WS Secure Conversation
  - Handshake mechanism like SSL, but each message is individually secured
- Username/password
  - Used along with other methods to encrypt password.

Books – recommended readings

- 'Computer Networks', Andrew S. Tanenbaum
- 'TCP/IP Illustrated Volume 1,2 and 3', W.Richard Stevens and al.
- 'Routing in the Internet', Christian Huitema
- 'Interconnections', Radia Perlman
- 'Patterns in Network Architecture', John Day
- 'Practical Unix and Internet Security', Simon Garfinkel and Gene Spafford