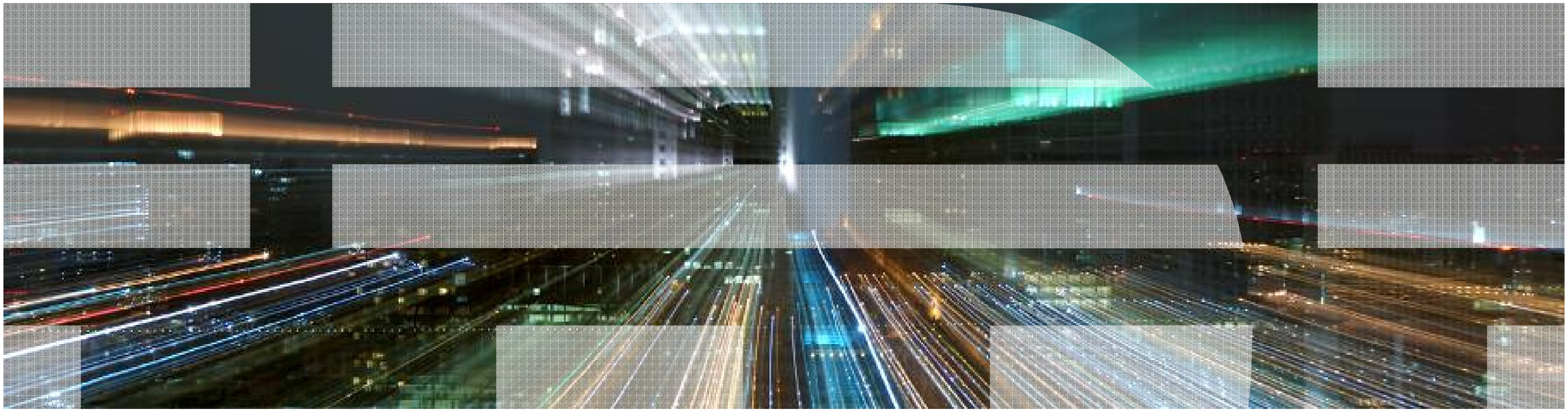
Thomas Cosenza - tcosenza@us.ibm.com
STG LAB Services

AST 11 & 12
z/OS Communications Server Network A Security Practitioner's Overview



Trademarks and Notices

Introduction

- Thomas Cosenza
  - Lab Services Leader for XI50z enablement services
  - Network and IT Security Consultant for the last 8 years
  - CISSP in good standing

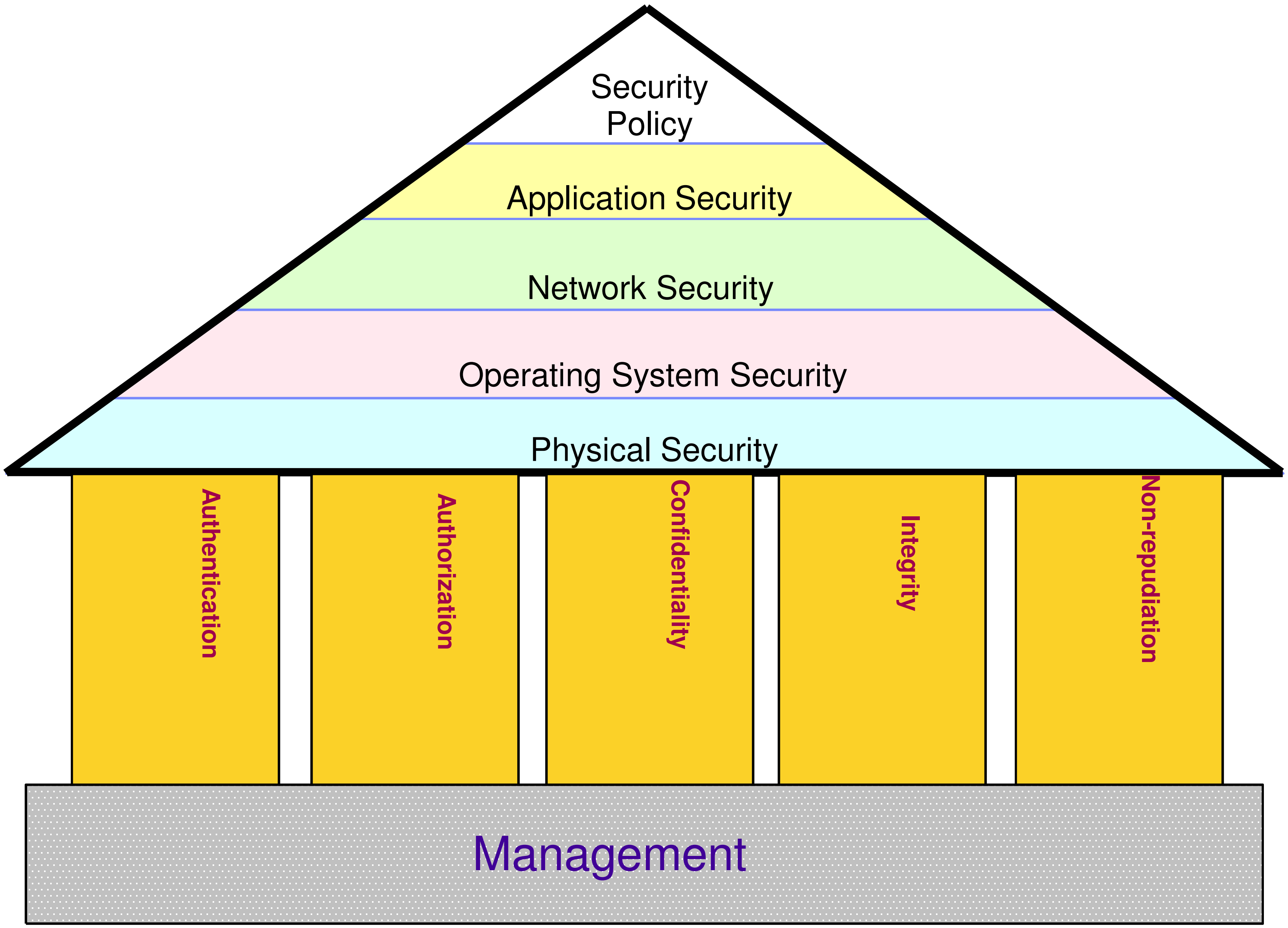
- This presentation is designed to give you a complete overview of
  - IPSecurity Technologies
  - How these are implemented at a high level

- Due to time constraints some of the slides are hidden … download the full slide deck to get all the goodies

Security Architecture

- **Authentication:**
  - Is this person or process what you think he/she/it is? How can you be sure?
  - Without strong authentication everything else is useless!
- **Authorization:**
  - Does this person or process have the right to access this resource?
- **Data Integrity:**
  - Has this data been modified since it was created by its rightful owner?
- **Confidentiality:**
  - Has this data been read by those who should not see it?
- **Non-repudiation:**
  - Is there positive proof that the identity of the creator of this message is authentic?

The Security Edifice



Implementing Security on Your System

- **What kind of security do you need?**
  - Users
    - Internet, Intranet, Business partners
    - Legitimate Users or Hackers and Thieves?
    - Who can be Trusted, and how far?
  - Data
    - What applications need protecting?
    - How is the data transferred?
    - Do we need just integrity, or confidentiality?
  - Denial of Service Attacks
    - External
    - Internal (Trojans, zombies)

- **Where do you do Security?**
  - Routers?
  - Servers?
  - Applications?
  - Firewalls?
  - All of these?

- **What will it cost?**
  - Risk Analysis

Protecting TCP/IP system resources and data in the network

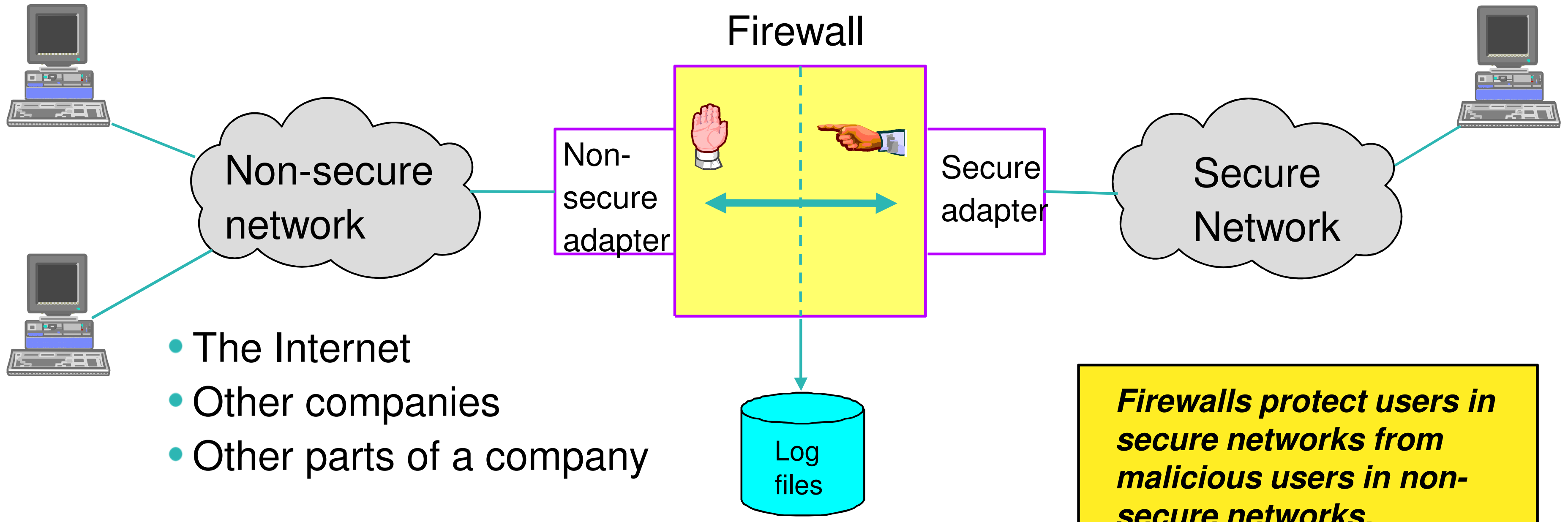
- **Protect against malicious or accidental access attempts**
  - Solution: IP packet filtering
- **Protect against malicious or accidental DoS attacks**
  - Solution: Intrusion detection
- **Protect confidentiality and integrity of data in the network**
  - Solution: Encryption
  - Solution: Virtual Private Networks
- **Build multiple barriers between you and the bad guys**
  - Solution: Gateways (Proxy and SOCKS)
  - Solution: Demilitarized Zone

(not really security but worth a mention)

- Network Address Translation

Overview of Network Security Resource Protection

What is a Firewall?



- The Internet
- Other companies
- Other parts of a company

Firewalls protect users in secure networks from malicious users in non-secure networks.

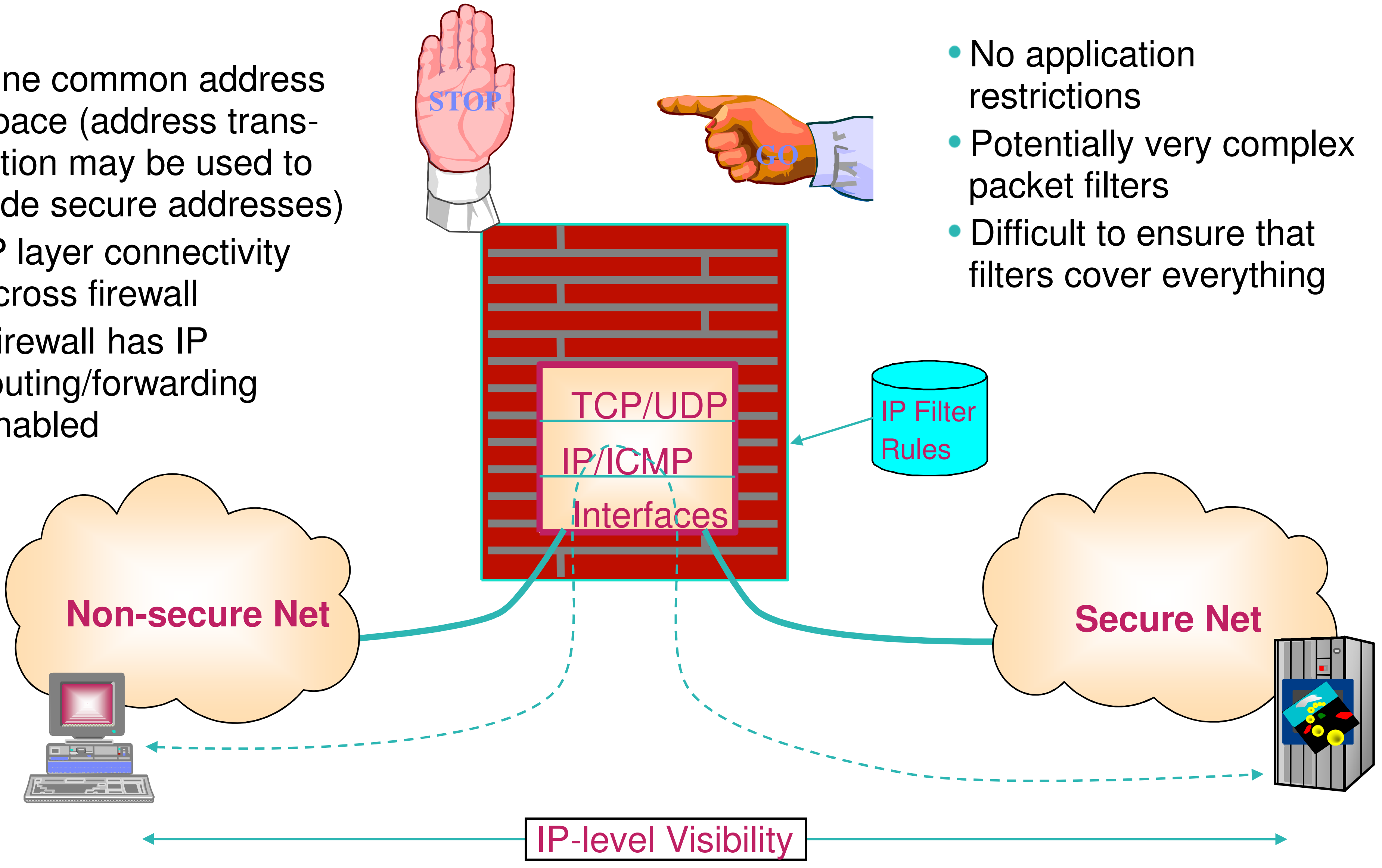
N.B. Don't forget that you might have malicious users inside your intranet too!!!!

1. A Firewall is most often a dedicated machine, but need not be
2. It controls TCP/IP traffic at the IP level and/or at the application level in and out of the secure network
3. A Firewall maintains log files of suspicious access patterns and rejected access attempts.

Packet Filtering Firewall

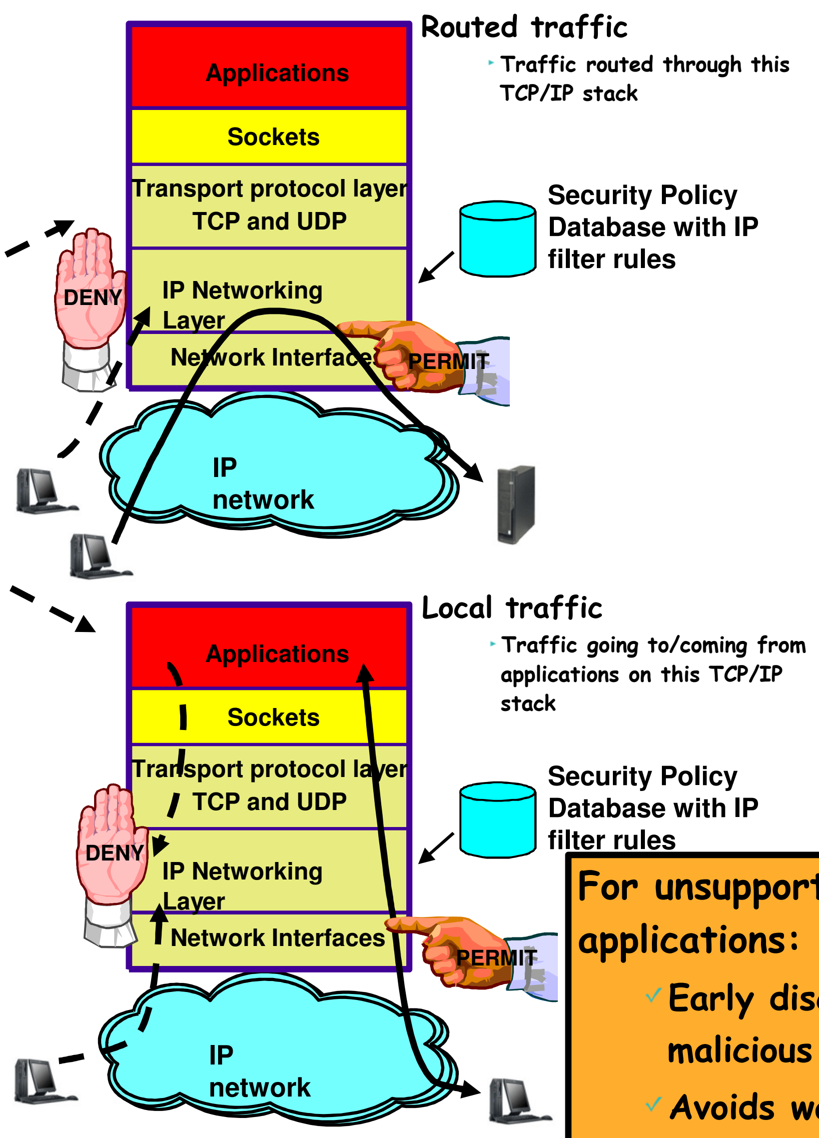
- One common address space (address translation may be used to hide secure addresses)
- IP layer connectivity across firewall
- Firewall has IP routing/forwarding enabled

- No application restrictions
- Potentially very complex packet filters
- Difficult to ensure that filters cover everything



Blocking unnecessary IP traffic through IP filtering

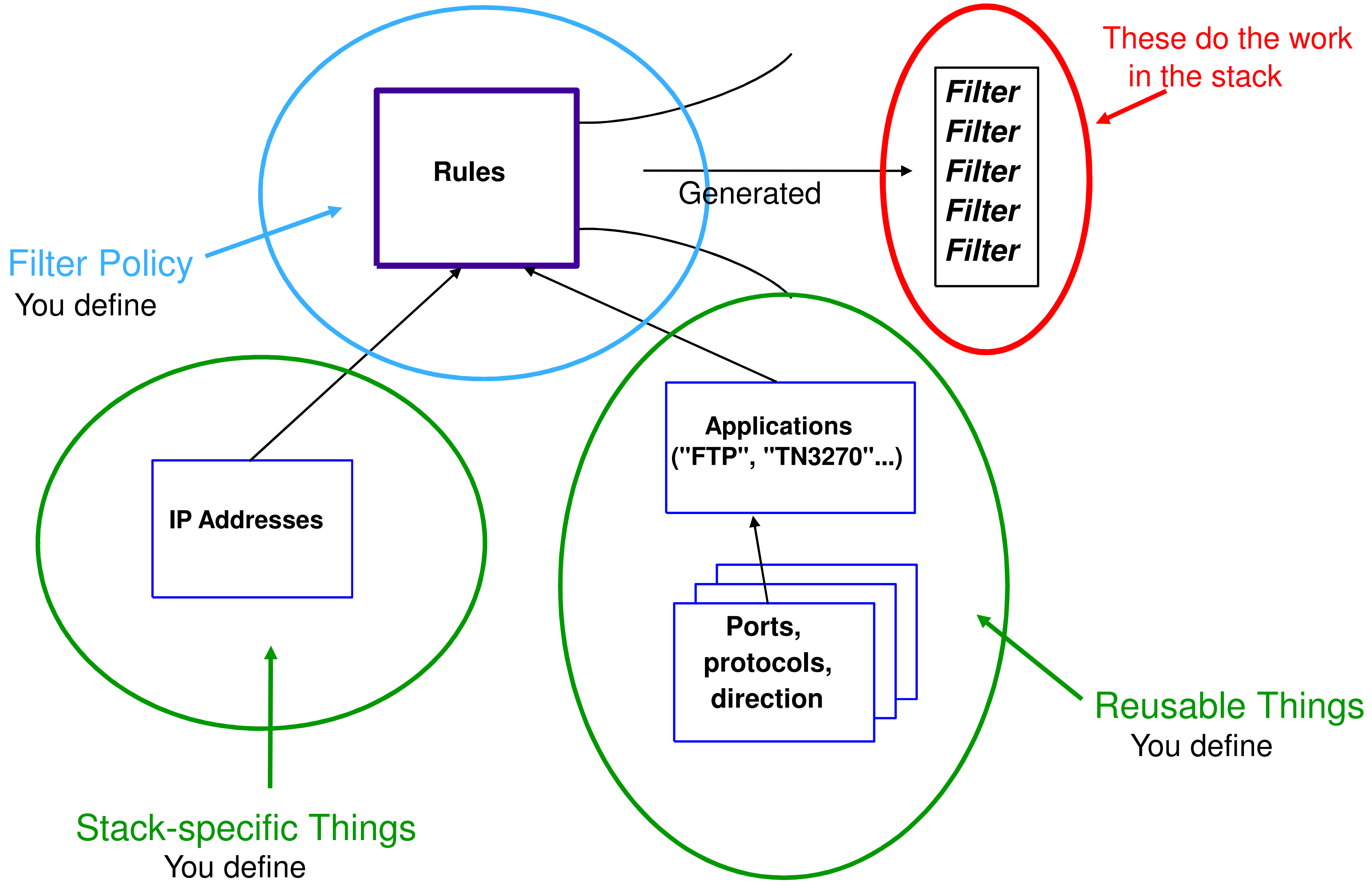
- **IP filtering at the IP Layer**
  - Filter rules defined to deny or permit packets based on:
    - IP source/destination address
    - Protocol (TCP, TCP with ACK, UDP, ICMP, ?)
    - Source/destination Port
    - Direction of flow
    - Local or routed traffic
    - Time
    - Network interface
  - Used to control
    - traffic being routed
    - traiic to this host (local)
  - When IP filtering is active, a default rule (implicit deny all) will lose all packets that are not specifically permitted



Routed traffic
Traffic routed through this TCP/IP stack
Applications
Sockets
Transport protocol layer TCP and UDP
IP Networking Layer
Network Interfaces
DENY
PERMIT
Security Policy Database with IP filter rules
IP network
Local traffic
Traffic going to/coming from applications on this TCP/IP stack
Applications
Sockets
Transport protocol layer TCP and UDP
IP Networking Layer
Network Interfaces
DENY
PERMIT
Security Policy Database with IP filter rules
IP network

For unsupported local applications:
- ✓ Early discard of potential malicious packets
- ✓ Avoids wasting CPU cycles checking validity of packets

Typical Packet Filter Implementation



Packet Filter Definitions

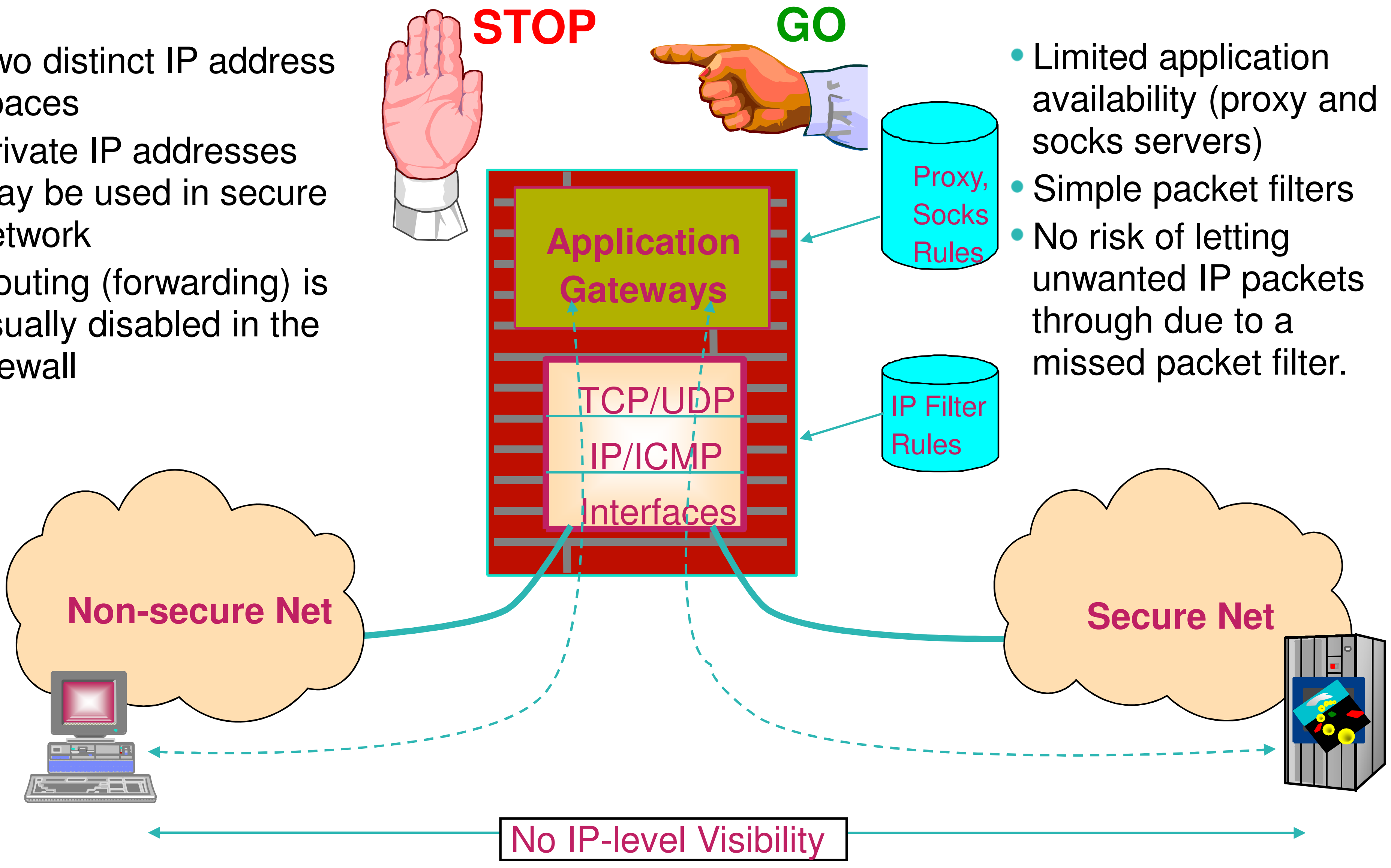
```
permit 9.0.0.0 255.0.0.0 0.0.0.0 0.0.0.0 tcp any 0 eq 20 both both inbound l=no f=yes t=0
permit 9.0.0.0 255.0.0.0 0.0.0.0 0.0.0.0 tcp any 0 eq 21 both both inbound l=no f=yes t=0
permit 9.0.0.0 255.0.0.0 0.0.0.0 0.0.0.0 tcp any 0 eq 23 both both inbound l=no f=yes t=0
permit 9.0.0.0 255.0.0.0 0.0.0.0 0.0.0.0 udp eq 53 any 0 both both inbound l=no f=yes t=0
permit 9.0.0.0 255.0.0.0 0.0.0.0 0.0.0.0 udp any 0 eq 68 both both inbound l=no f=yes t=0
permit 9.0.0.0 255.0.0.0 0.0.0.0 0.0.0.0 tcp eq 515 any 0 both both inbound l=no f=yes t=0
permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 udp eq 53 any 0 both both inbound l=no f=yes t=0
permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tcp/ack eq 80 any 0 both both inbound l=no f=yes t=0
permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tcp/ack eq 443 any 0 both both inbound l=no f=yes t=0
permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tcp eq 20 any 0 both both inbound l=no f=yes t=0
permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tcp/ack eq 21 any 0 both both inbound l=no f=yes t=0
permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tcp/ack eq 23 any 0 both both inbound l=no f=yes t=0
permit 192.168.0.0 255.255.0.0 0.0.0.0 0.0.0.0 tcp any 0 eq 20 both both inbound l=no f=yes t=0
permit 192.168.0.0 255.255.0.0 0.0.0.0 0.0.0.0 tcp any 0 eq 21 both both inbound l=no f=yes t=0
permit 192.168.0.0 255.255.0.0 0.0.0.0 0.0.0.0 tcp any 0 eq 23 both both inbound l=no f=yes t=0
permit 192.168.0.0 255.255.0.0 0.0.0.0 0.0.0.0 udp any 0 eq 68 both both inbound l=no f=yes t=0
deny 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tcp any 0 lt 1024 both both inbound l=yes f=yes t=0
deny 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 udp any 0 lt 1024 both both inbound l=yes f=yes t=0
permit 9.0.0.0 255.0.0.0 0.0.0.0 0.0.0.0 icmp any 0 any 0 both both inbound l=no f=yes t=0
permit 192.168.0.0 255.255.0.0 0.0.0.0 0.0.0.0 icmp any 0 any 0 both both inbound l=no f=yes t=0
deny 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 icmp any 0 any 0 both both inbound l=yes f=yes t=0
permit 9.0.0.0 255.0.0.0 0.0.0.0 0.0.0.0 all any 0 any 0 both both inbound l=no f=yes t=0
permit 192.168.0.0 0.0.0.0 0.0.0.0 0.0.0.0 all any 0 any 0 both both inbound l=no f=yes t=0
permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 all any 0 any 0 both both outbound l=no f=yes t=0
```

- Editing this is NOT recommended.
- It's even less intelligible on z/OS.

Application Layer Gateway

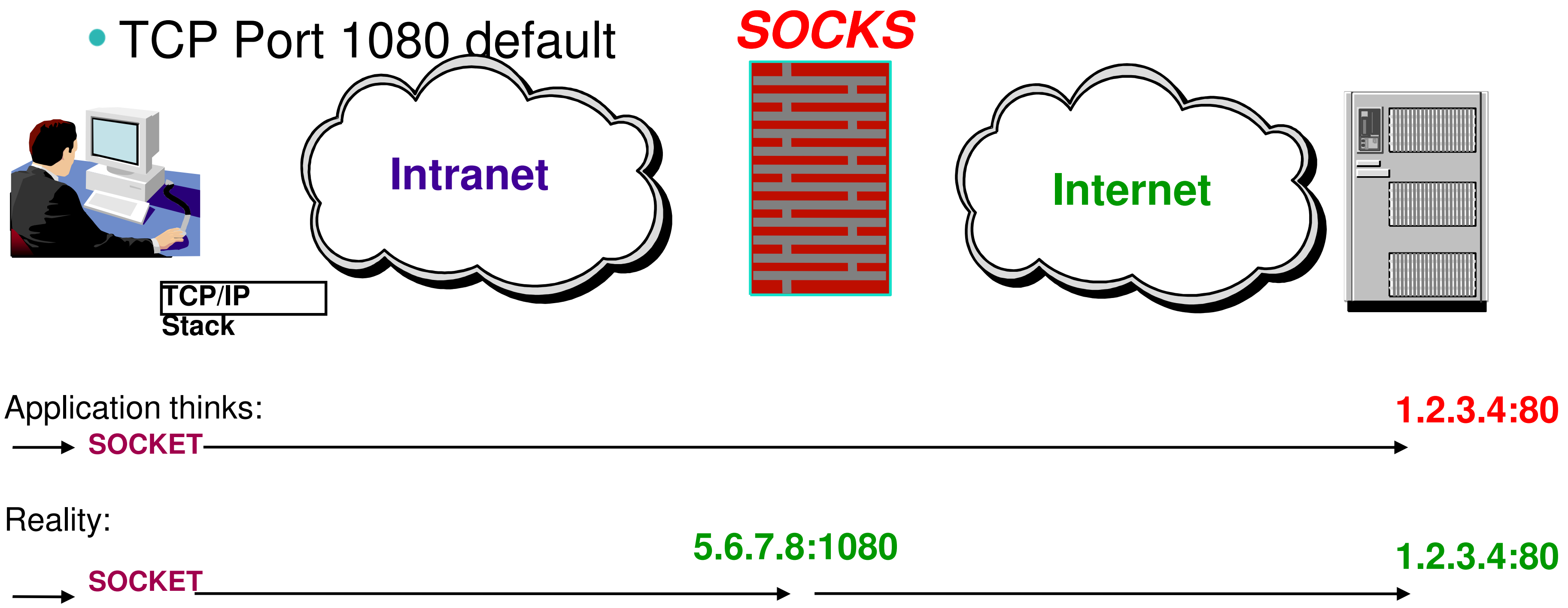
- Two distinct IP address spaces
- Private IP addresses may be used in secure network
- Routing (forwarding) is usually disabled in the firewall

- Limited application availability (proxy and socks servers)
- Simple packet filters
- No risk of letting unwanted IP packets through due to a missed packet filter.



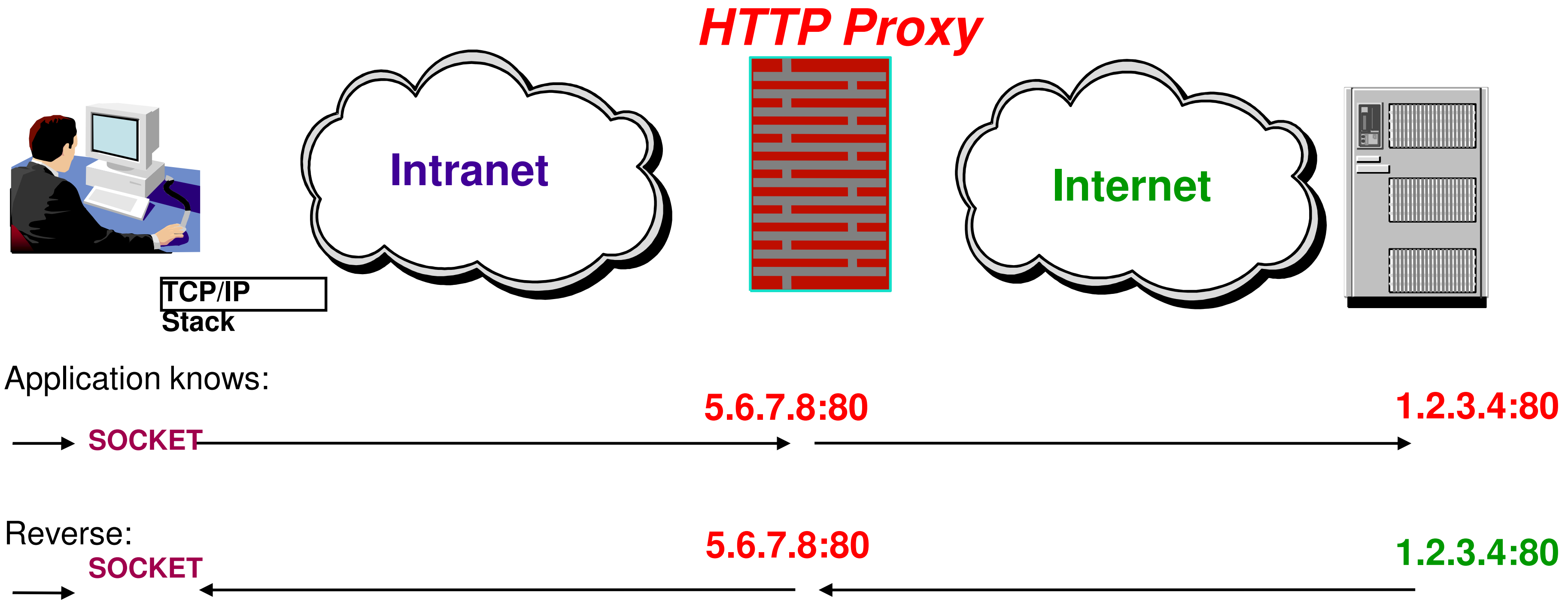
SOCKS and Proxy

- SOCKS
  - Stack-specific gateway
  - Transparent to client / server
  - Outbound Connections
  - TCP Port 1080 default

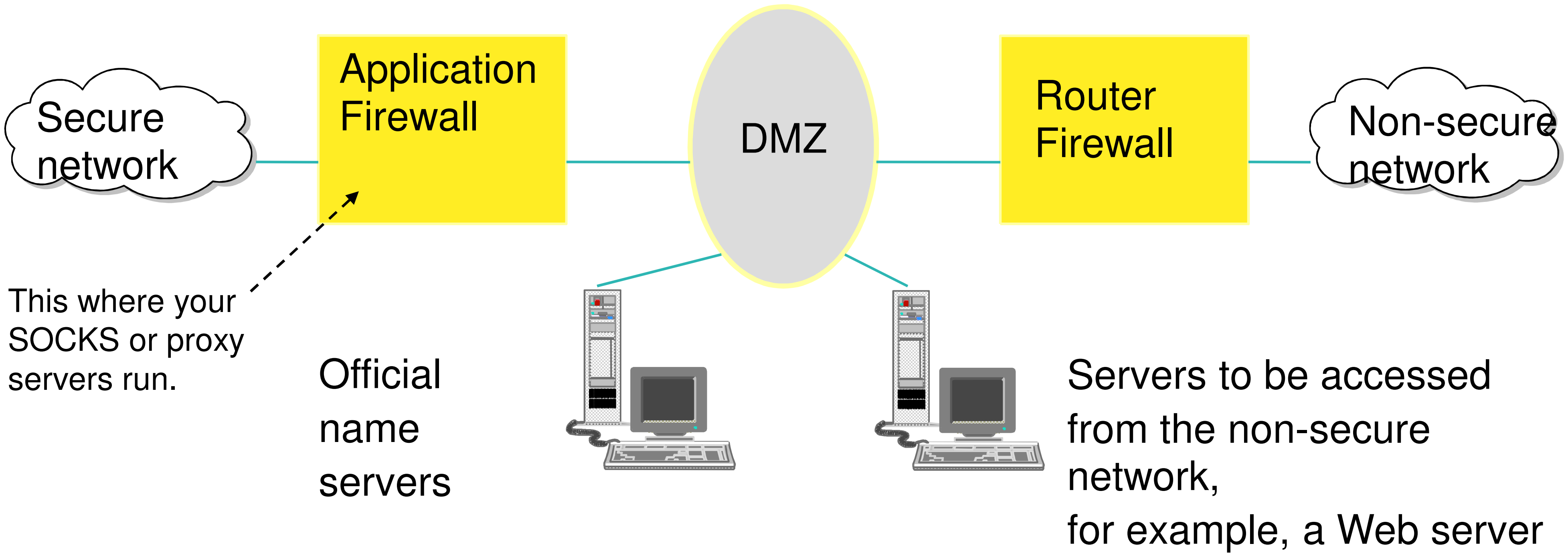


SOCKS and Proxy

- Proxy Server
  - Application-specific gateway
  - Outbound (needs client modification)
  - Inbound (Reverse) needs no modification



Demilitarized Zone

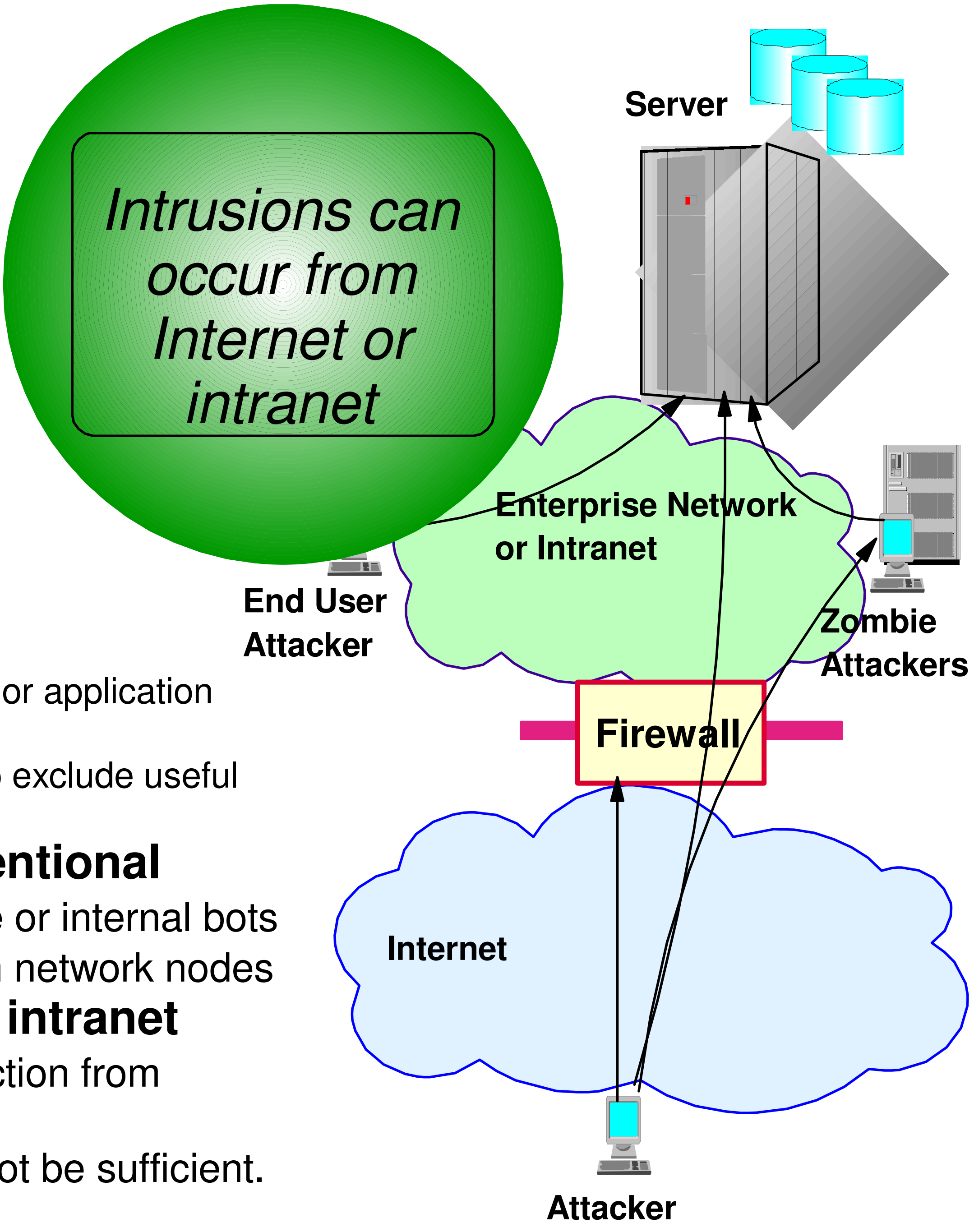

Secure network
Application Firewall
DMZ
Router Firewall
Non-secure network
This where your SOCKS or proxy servers run.
Official name servers
Servers to be accessed from the non-secure network, for example, a Web server

May have multiple DMZs

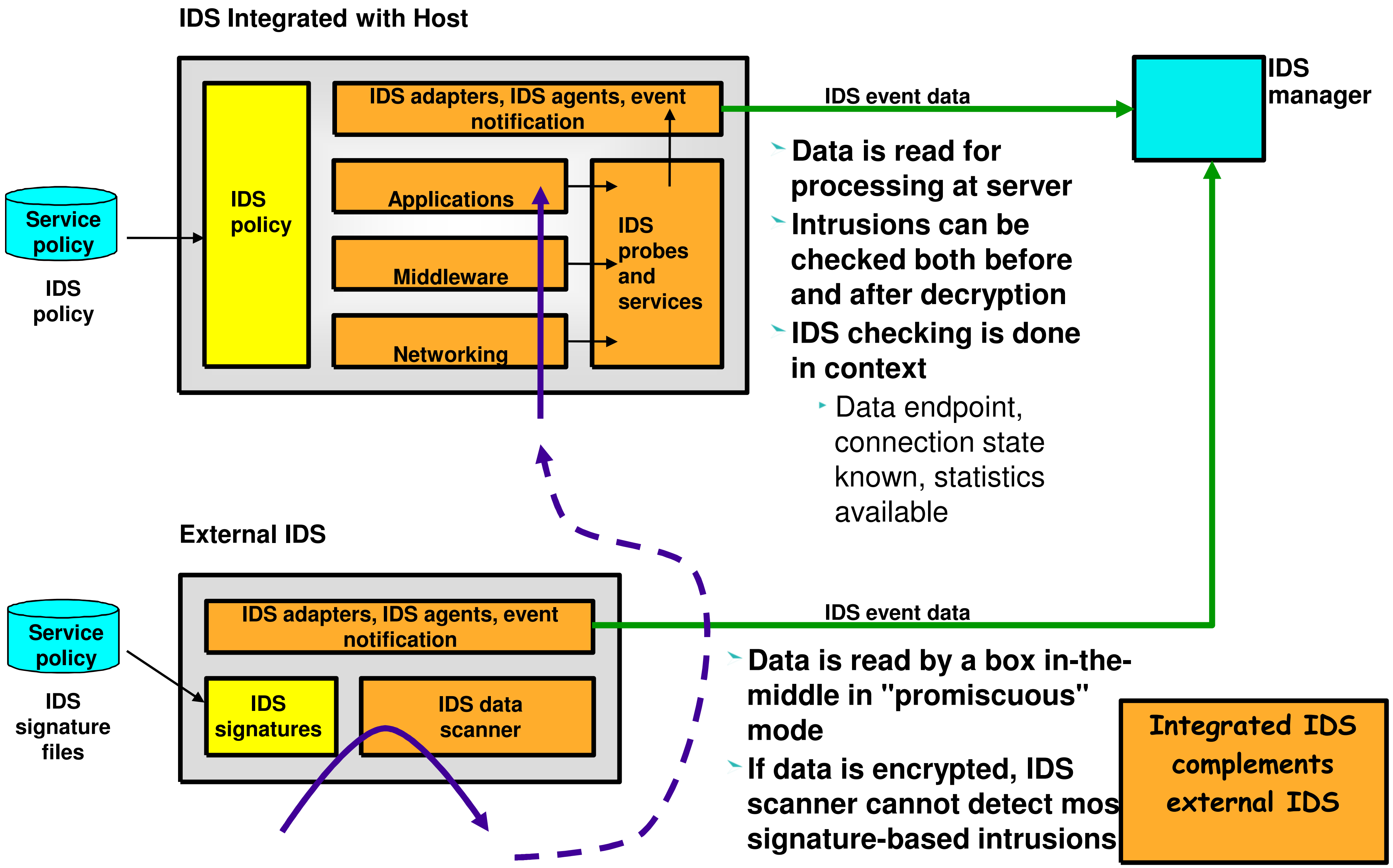
Parallel or serial

The Intrusion Threat

- **What is an intrusion?**
  - ➤ Information Gathering
    - –Network and system topology
    - –Data location and contents
  - ➤ Eavesdropping / Impersonation / Theft
    - –On the network / on the server
    - –Based for further attacks on others
      - ✓ Amplifiers
      - ✓ Robot or zombie
  - ➤ Denial of Service
    - –Attack on availability
      - ✓ Single Packet attacks - exploits system or application vulnerability
      - ✓ Multi-Packet attacks - floods systems to exclude useful work
- **Attacks can be deliberate or unintentional**
  - ➤ Deliberate: malicious intent from outside or internal bots
  - ➤ Unintentional: various forms of errors on network nodes
- **Attacks can occur from Internet or intranet**
  - ➤ Firewall can provide some level of protection from Internet
  - ➤ Perimeter Security Strategy *alone* may not be sufficient.
    - –Considerations:
      - ✓ Access permitted from Internet
      - ✓ Trust of intranet


Intrusions can occur from Internet or intranet
Server
Enterprise Network or Intranet
End User Attacker
Zombie Attackers
Firewall
Internet
Attacker

Integrated versus external Intrusion Detection Services (IDS)

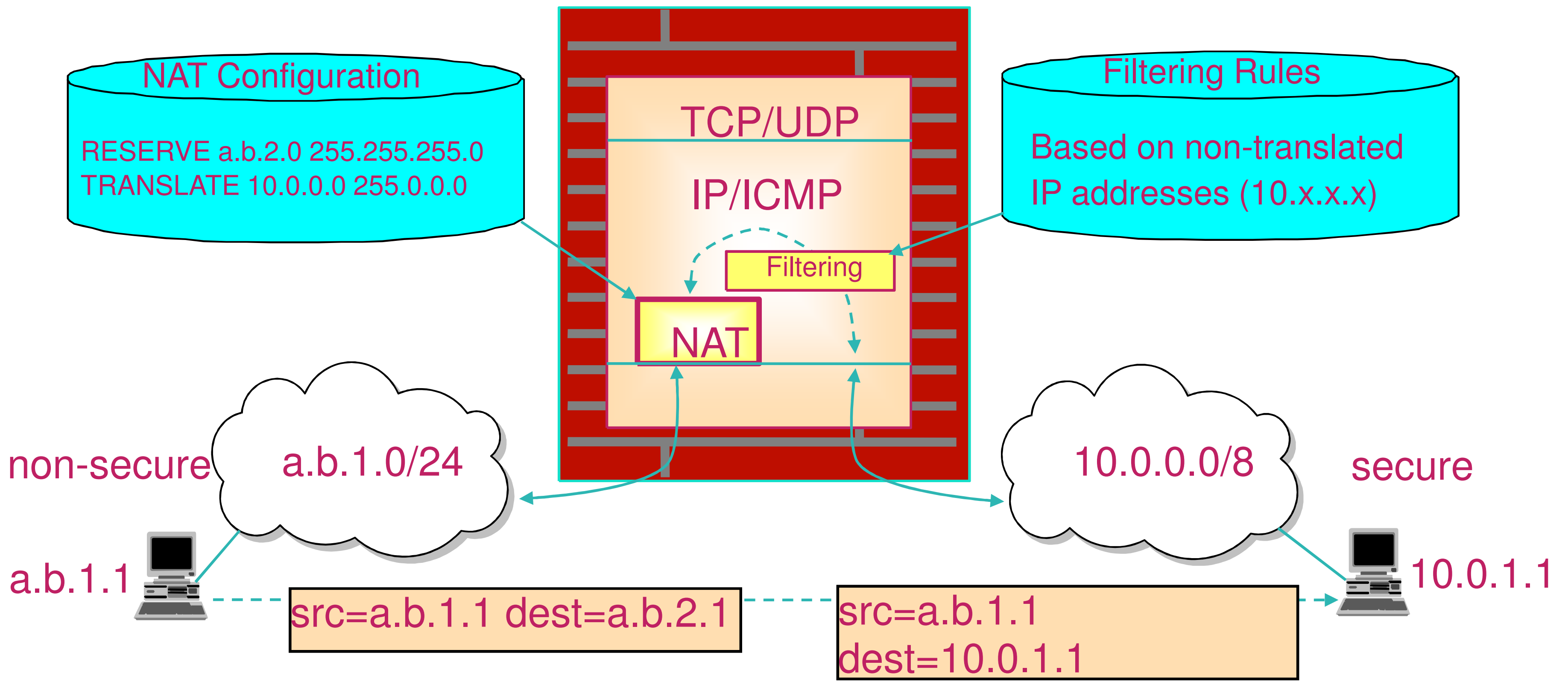


- Data is read for processing at server
- Intrusions can be checked both before and after decryption
- IDS checking is done in context
  - Data endpoint, connection state known, statistics available

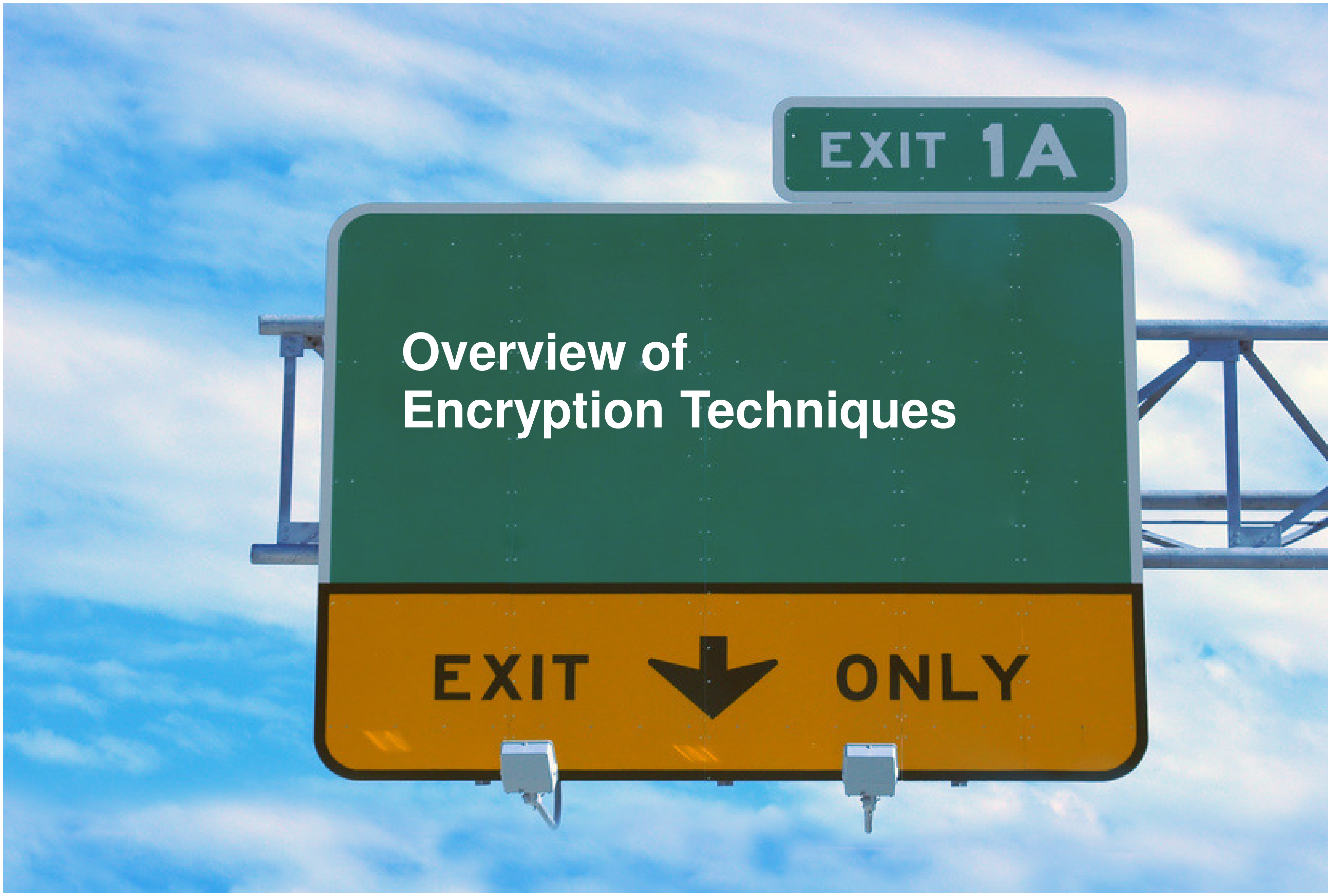
- Data is read by a box in-the-middle in "promiscuous" mode
- If data is encrypted, IDS scanner cannot detect mos signature-based intrusions

Integrated IDS complements external IDS

Network Address Translation



- NAT was done to conserve IP addresses
- NAT is "security by obscurity"
- NAPT is also used (port translation)
- NAT and NAPT make crypto-based security more difficult!



Overview of Encryption Techniques

Protecting the Data in the Network

- **Ensure confidentiality of data**
  - Solution: Symmetric encryption
    - RC2, RC4, DES, 3DES, AES, CAST, IDEA, Blowfish
- **Protect the encryption keys**
  - Solution: Asymmetric encryption
    - RSA, DSA, Diffie-Hellman, Elliptic Curve
- **Ensure data integrity and non-repudiation**
  - Solution: Digital signatures
    - MD5, SHA
- **Manage identities and encryption keys**
  - Solution: Digital certificates
  - Solution: Public Key Infrastructure

KEY TO NETWORK ENCRYPTION

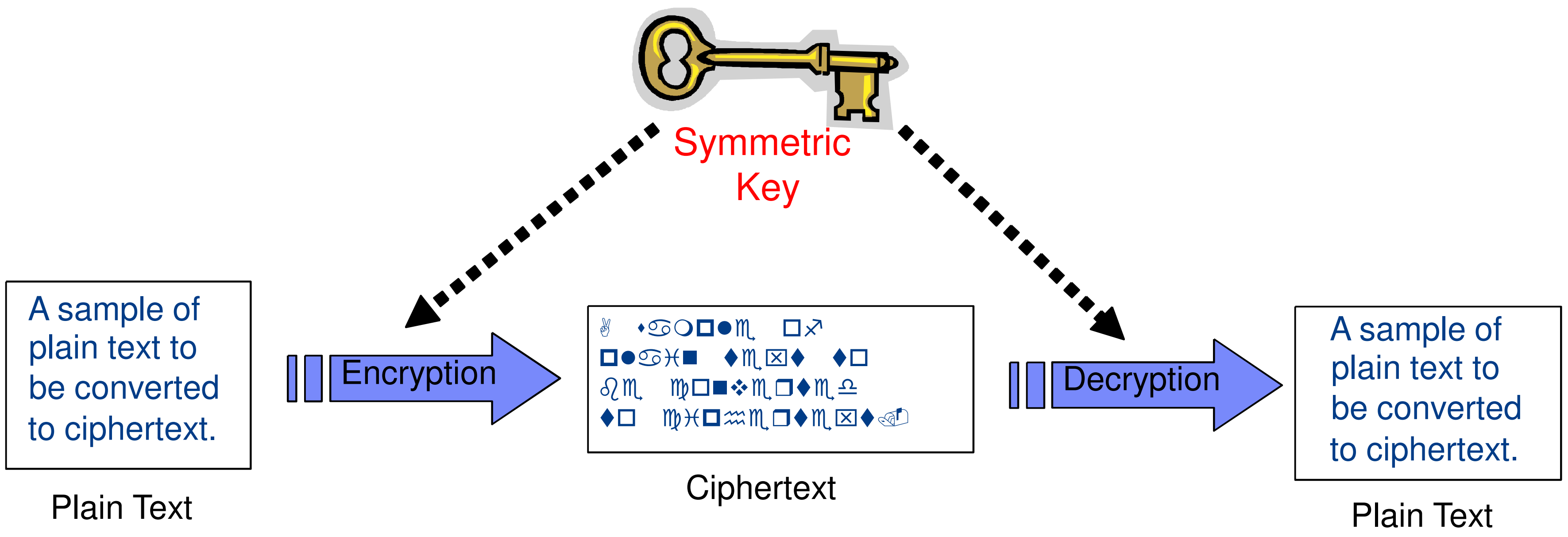
Encryption Techniques Through the Ages

- Simple Substitution (Monoalphabetic)
- Multiple Substitution
- Multiple Ciphers in a Message
- Mechanical Ciphers
- Enigma
- Computers - DES etc.
- Quantum Cryptography

- One Time Pad is best, but impractical

Symmetric Encryption

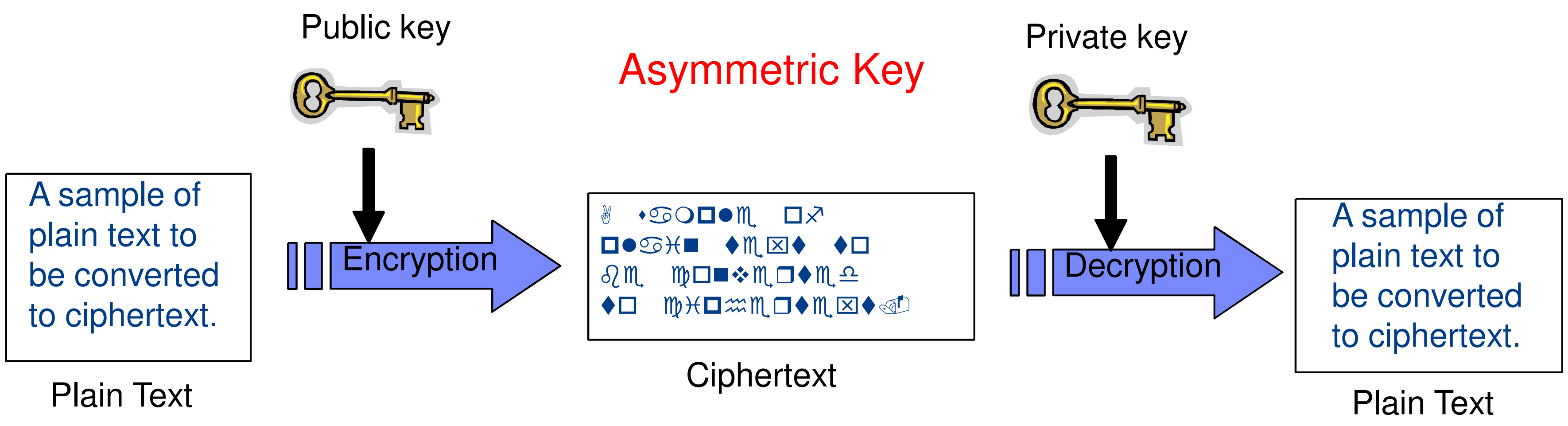
- Symmetric Key - a secret key that is used both to encrypt and to decrypt messages
- Known only by sender and receiver
- Relatively fast and light on CPU power
- Until recently, the ONLY form of encryption
- Challenge : exchanging keys with many people


Symmetric Key
A sample of plain text to be converted to ciphertext.
Plain Text
Encryption
Ciphertext
Decryption
A sample of plain text to be converted to ciphertext.
Plain Text

- Commonly used algorithms are Triple DES and AES
- DES is discouraged - not secure enough

Asymmetric Encryption

- Asymmetric key - public/private key pairs
- Message encrypted with partner's public key and decrypted with your private key
- Slower and more CPU-intensive than symmetric key cryptography
- The private key cannot be derived from the public key
- Either key can undo what the other does
- Used to exchange symmetric keys, and optionally for authentication


Public key
Asymmetric Key
Private key
A sample of plain text to be converted to ciphertext.
Encryption
Decryption
Plain Text
Ciphertext
A sample of plain text to be converted to ciphertext.
Plain Text

- Commonly used algorithms are RSA and Diffie-Hellman

Basic Principles of Cryptography

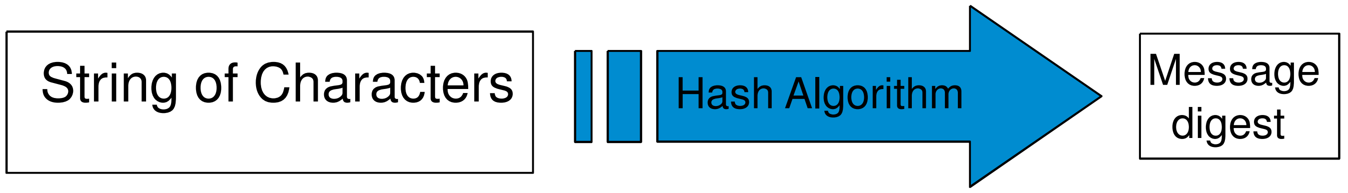
1. Publish the algorithm but protect the keys
   - Cracking the formula should not expose the data
2. Ensure that the algorithm exhibits no patterns
   - Only a brute force attack can break the key
3. Make the key as long as practicable
   - Difficulty of cracking goes up exponentially as key length increases

Security Issues... so far

- Confidentiality - how do I keep anyone from reading my message? ✓ YES!
- Integrity - how do I know if anyone has tampered with my message? ✗ NOT YET
- Authentication - how do I know that my communication partner is who it claims to be? ✓ GETTING THERE
- Non-repudiation - how do I know that the identity of the message sender is authentic? ✗ NOT YET

Data Integrity

- How do I know if anyone has looked at my message and changed it?
- Answer : Use a hash algorithm
  - Like a weapons-grade check digit
- A hash algorithm transforms a message into a short string of a fixed length. This string is called a *message digest*.
- Any tampering with the message will result in a different message digest.
- Creating two messages with the same message digest is *extremely* difficult.



But.... the hacker knows the algorithm too!

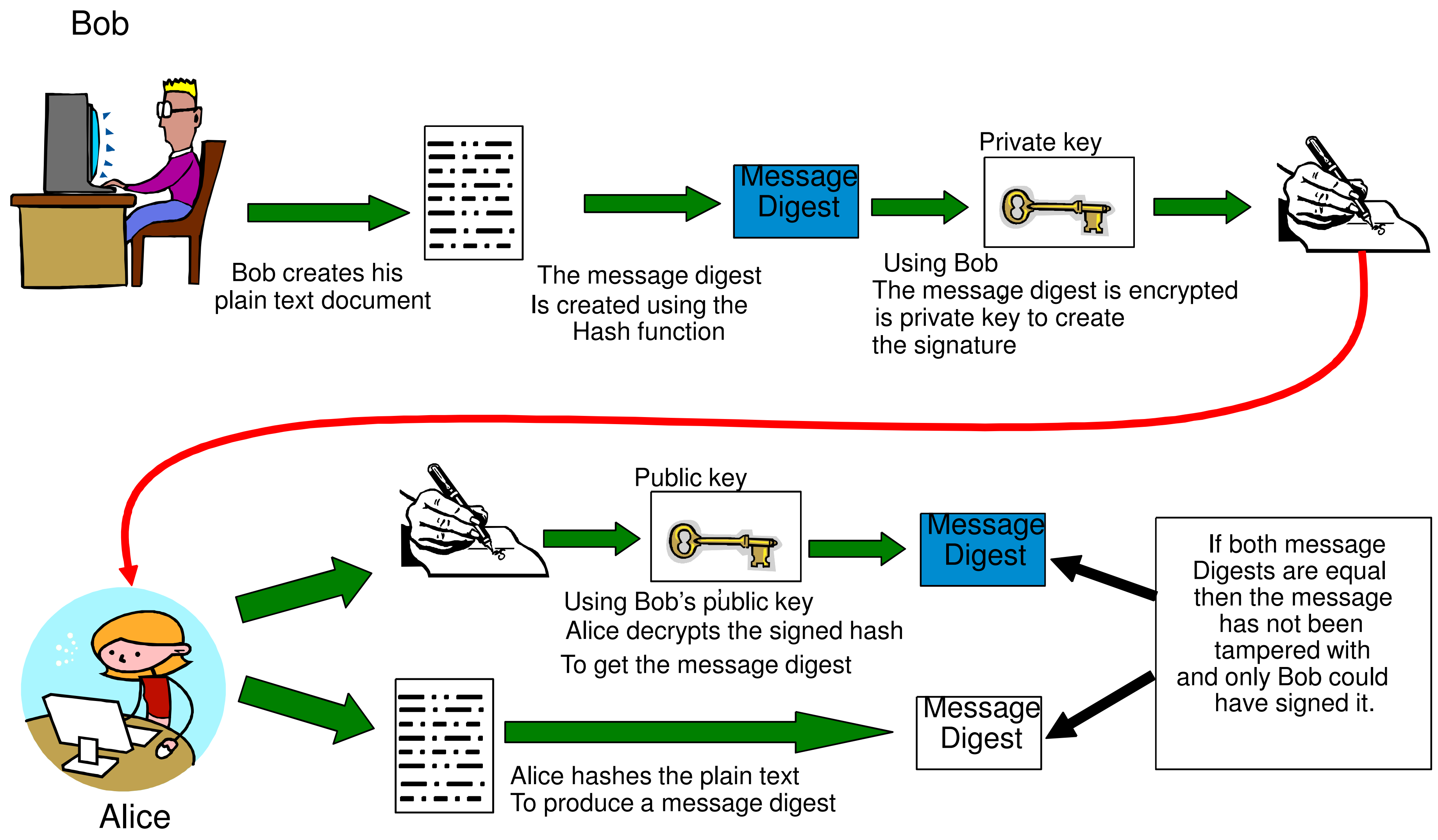
Cryptographic Hash Algorithms

Data integrity!

- Answer: Encrypt the message digest!
  - With a shared symmetric key (during communication)
    - In practice, do the digest over the message plus the key
  - With your own private key (long term)
    - This is known as a ***digital signature.***
- Cryptographic Hash algorithms include:
  - MD4, MD5 (no longer considered secure)
  - SHA-1, SHA-128, SHA-256, SHA-512
    - Use the later ones, SHA-1 is suspect

Non-repudiation!

Digital Signature


Bob
Bob creates his plain text document
Message Digest
The message digest Is created using the Hash function
Private key
Using Bob The message digest is encrypted is private key to create the signature
Public key
Using Bob's public key Alice decrypts the signed hash To get the message digest
Message Digest
If both message Digests are equal then the message has not been tampered with and only Bob could have signed it.
Alice hashes the plain text To produce a message digest
Message Digest
Alice

Impersonation / Authentication, revisited

- How do I REALLY know who the party on the other end is?
- By the use of digital signatures
  - Provide the ability to authenticate who sent the message
  - Incorporate the use of Asymmetric keys and cryptographic hash functions
  - The signature is encrypted with the sender's private key
  - If the sender's public key can decrypt the signature then the sender must be authentic.
- Final problem: How to ensure that this public key belongs to this sender?

Digital Certificates

- Digital Certificates address the authentication problem.
- A Digital Certificate can be thought of as an electronic identity card that establishes your credentials (authenticates you) when communicating securely.
- A Digital Certificate contains:
  - Your name
  - A serial number
  - Start and expiry dates
  - Your public key
  - And a digital signature, to verify that this public key belongs to this entity.

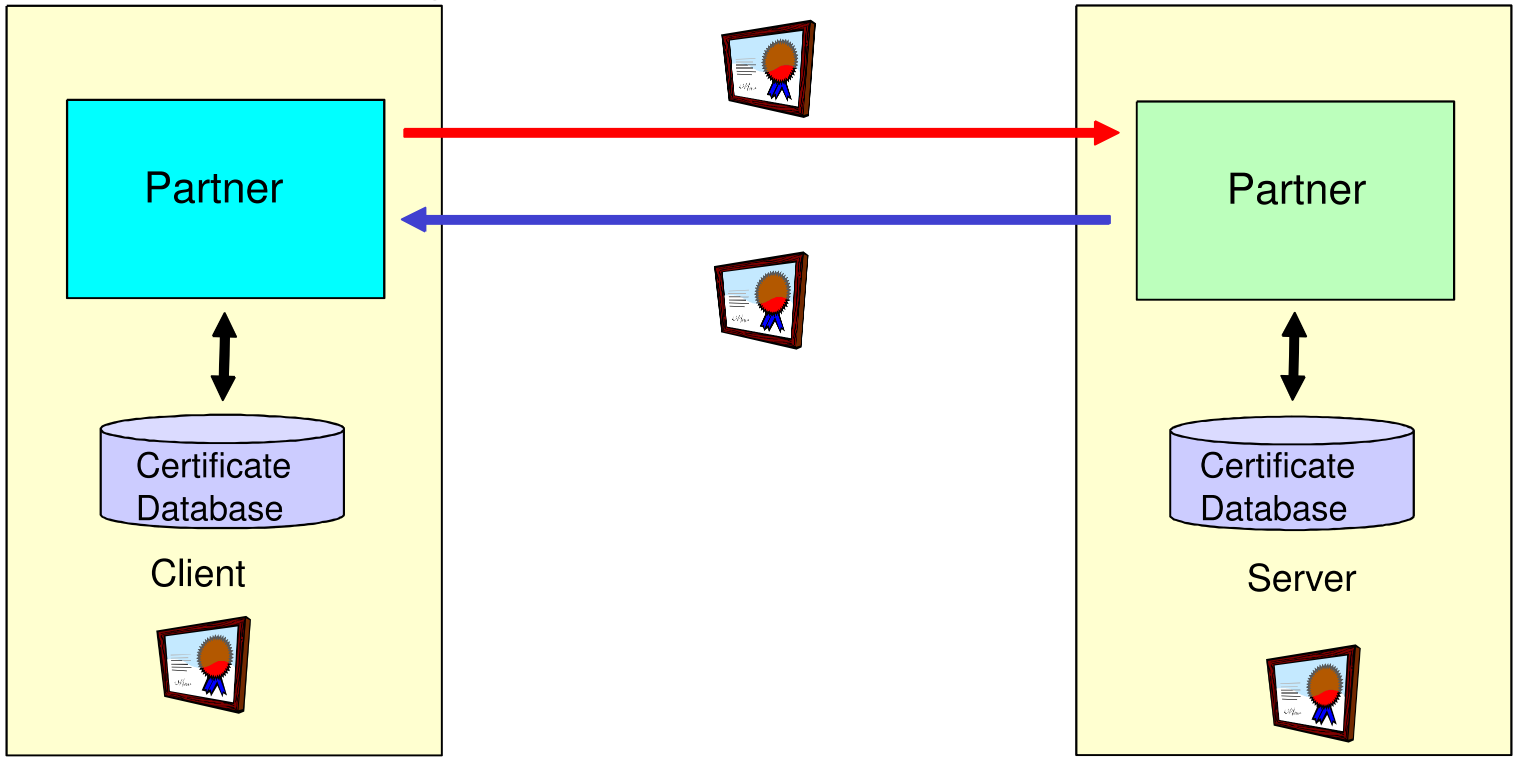
But whose signature?

Where do certificates come from?

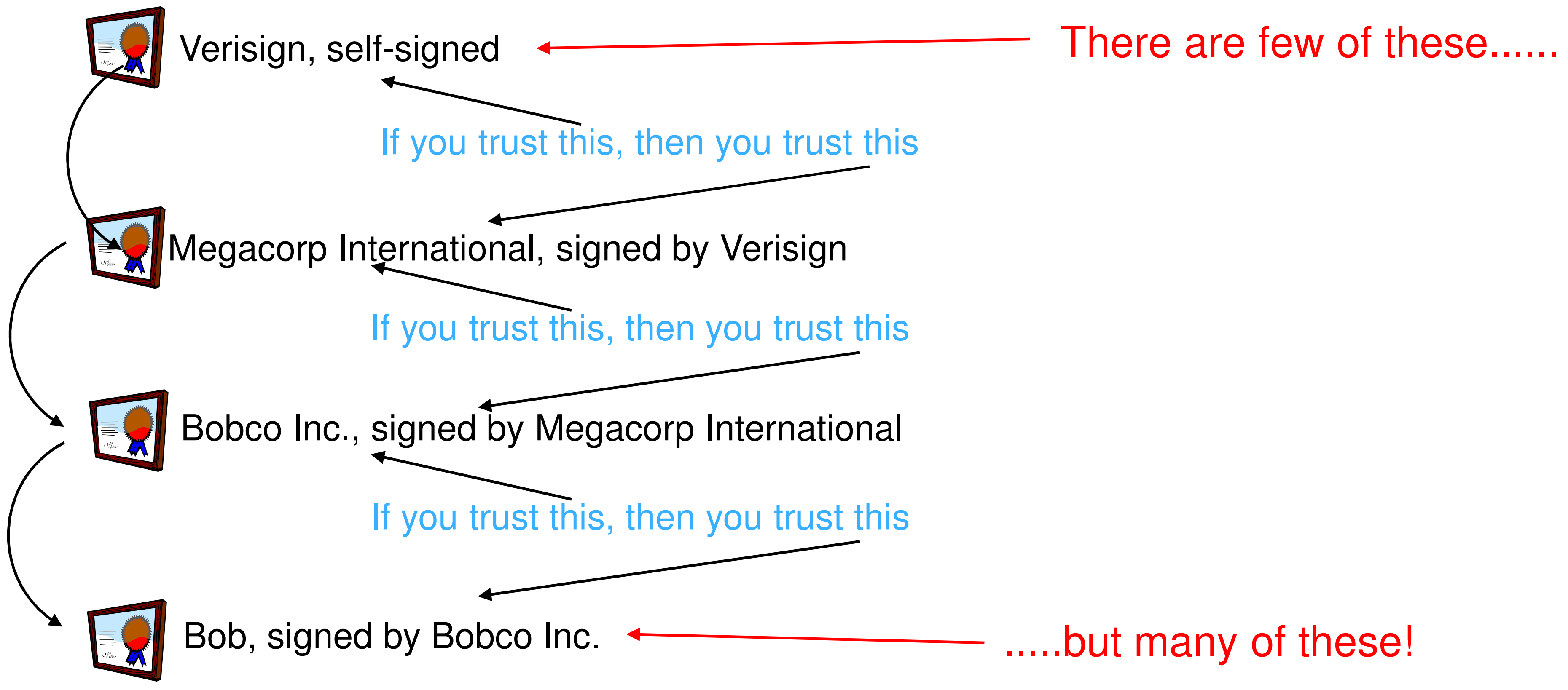
- Ultimately, the only way to be sure of authenticity is a ***physical*** exchange of digital certificates.
  - Any attempt at exchanging them over a network is doomed unless it is done securely.
  - But to do it securely you need a digital certificate to start with....

Self Signed Certificate (the poor man's method)

- Minimum requirement :
  - At least one end must have a copy of the other's certificate.
  - Often, both ends must have each other's certificates.
  - They must be exchanged securely (physically, or over a controlled Intranet connection)
  - Suitable for testing, but **NOT** for Internet communication



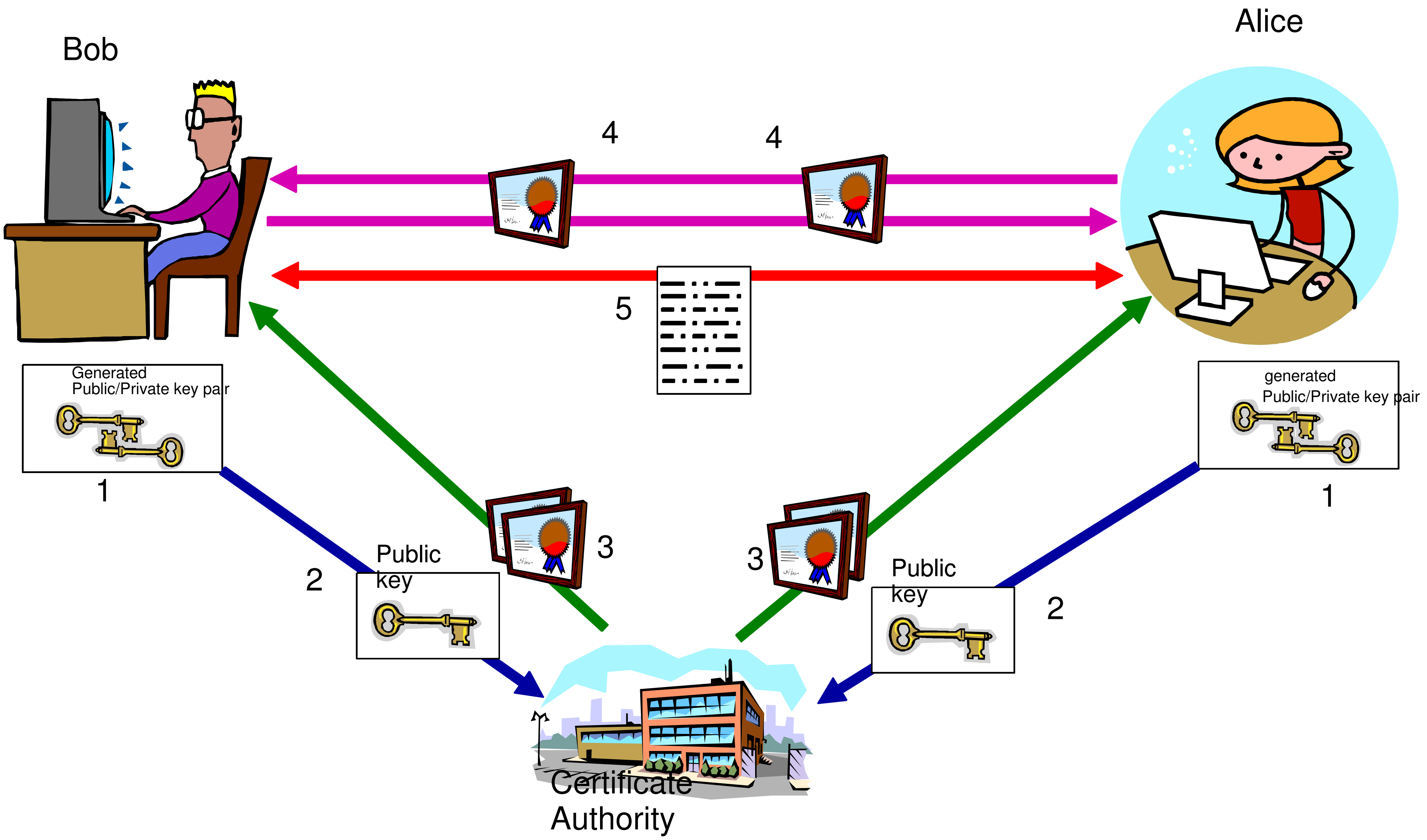
Certificate Authorities


Verisign, self-signed
There are few of these......
If you trust this, then you trust this
Megacorp International, signed by Verisign
If you trust this, then you trust this
Bobco Inc., signed by Megacorp International
If you trust this, then you trust this
Bob, signed by Bobco Inc.
.....but many of these!

CA Certificates

- A CA can be thought of as a certificate distributor.
- To obtain your personal certificate, you must send your information to a CA:
  - Create an Asymmetric public/private key pair.
  - Send your identity with your public key to the CA.
  - Wait for them to check you out and (maybe) pay them a large sum of money.
- The CA sends you back your certificate, signed by them.
- Now, if your communication partner trusts the CA then you are authenticated.

Using CA Certificates

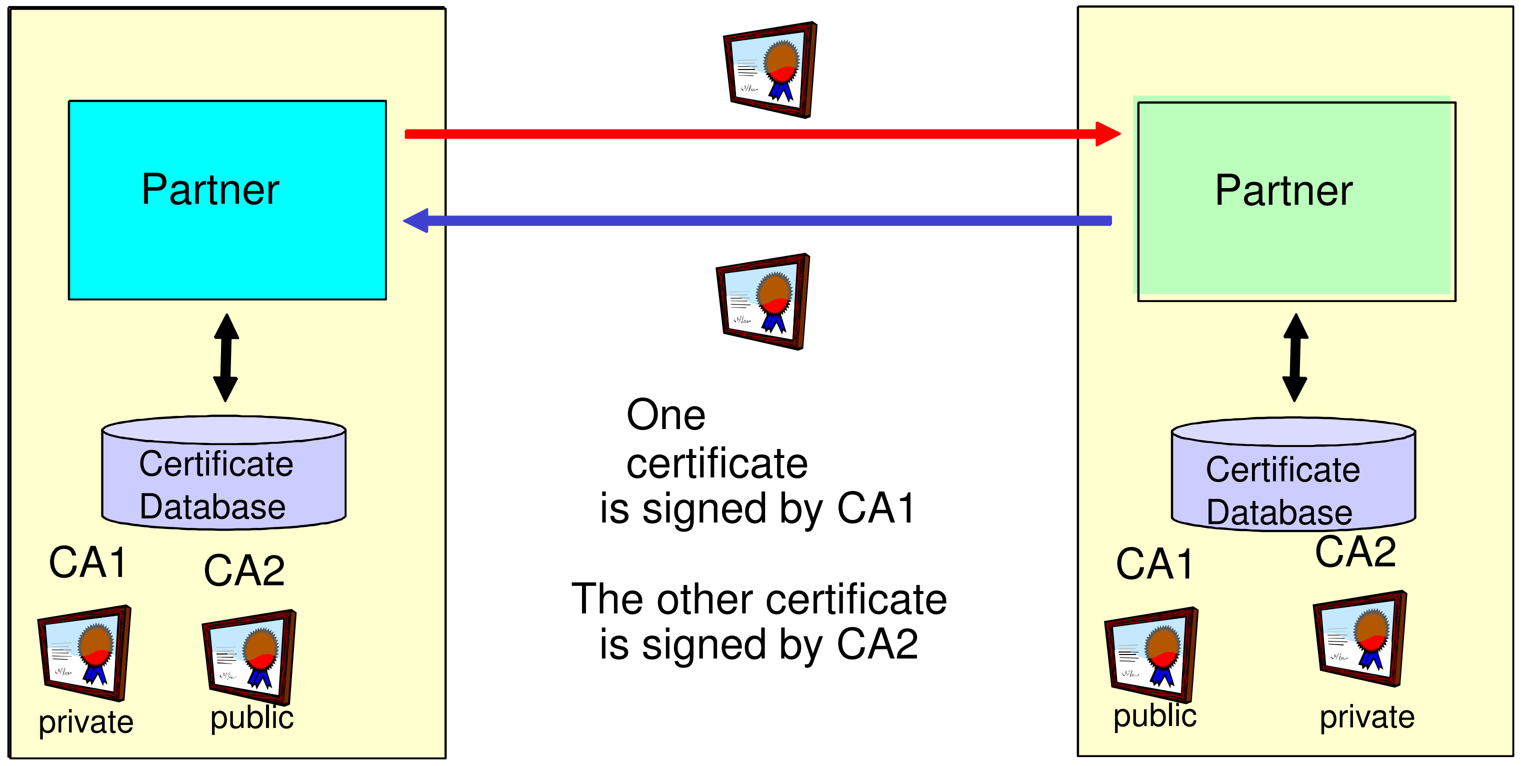


The Final Step

- If you trust the small group of CAs then you can authenticate any partner down the chain of trust.
- You need the CA's self-signed certificate, which contains their all-important public key.
- How can you be sure that the CA certificate is authentic?
- Ultimately, physical receipt of the certificate is necessary.
  - They are delivered with the machine!
  - With the operating system (z/OS - RACF)
  - With the application
    - Web browser (Internet Explorer, Firefox)
    - TN3270 client (PComm)
    - Web server (IHS, WAS...)
- You can verify the message digest of the CA (root) certificate on the CA's web site.

Certificate Management - CA Certificates

Before any communication can take place, at least one end must already have a copy of the CA certificate that signed the other end's personal certificate.



Certificate Management

- Certificates are used to authenticate the partner and (with the embedded public key) to encrypt and exchange symmetric keys.
- Three types of certificate may be found in a certificate database:
  1. CA certificates (for verifying partners' personal certificates)
  2. Personal certificates (to identify you, signed by your preferred CA)
  3. Self-signed certificates (to identify you, but not so secure)
- Certificates have a life span
  - If a certificate has expired, it should no longer be trusted.
  - If on the Internet, it MUST not be trusted.
- Certificates can be revoked
  - If the private key has been compromised, for example
  - The issuer places the revoked certificate on an LDAP server (the CRL - certificate revocation list)
  - The partner attempting authentication SHOULD check the CRL
  - Most often, this is not done.

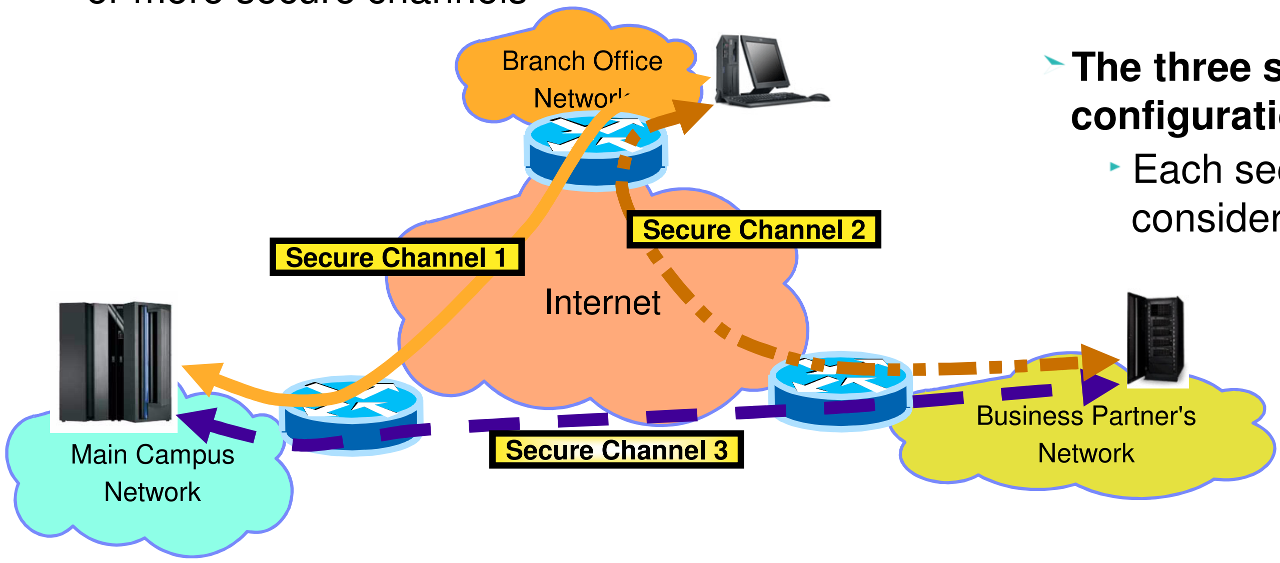
Overview of IPSEC and SSL/TLS

IPSec Architecture

- Security Association
- Authentication Header (AH) Protocol
- Encapsulated Security Payload (ESP) protocol
- Transport mode and Tunnel mode
- ISAKMP/Oakley
- Internet Key Exchange (IKE)
- IPSec Data Flows

IPSec Overview

- **Virtual Private Network**
  - Logical network of connected nodes that communicate over unsecure networks using one or more secure channels



- **The three secure channels in this sample configuration make up a VPN**
  - Each secure channel in itself can be considered a VPN

- **IPSec is a set of standards (RFCs) defining how to do VPNs.**
- **A secure channel is commonly called an IPSec security association (SA).**
  - The term "tunnel" is also sometimes used in this context, but it is ambiguous and can be confusing
- **A secure channel provides point-to-point security: Authentication, Data Integrity and optionally Confidentiality.**
- **There is no "client" or "server" - only "initiator" and "responder".**
- **IPSec utilizes IP security protocols defined by the IPSec working group**
  - Original Ones : RFC 2401 to RFC 2412
  - New Ones : RFC 4301 to RFC 4308

IPSec - How to encapsulate?

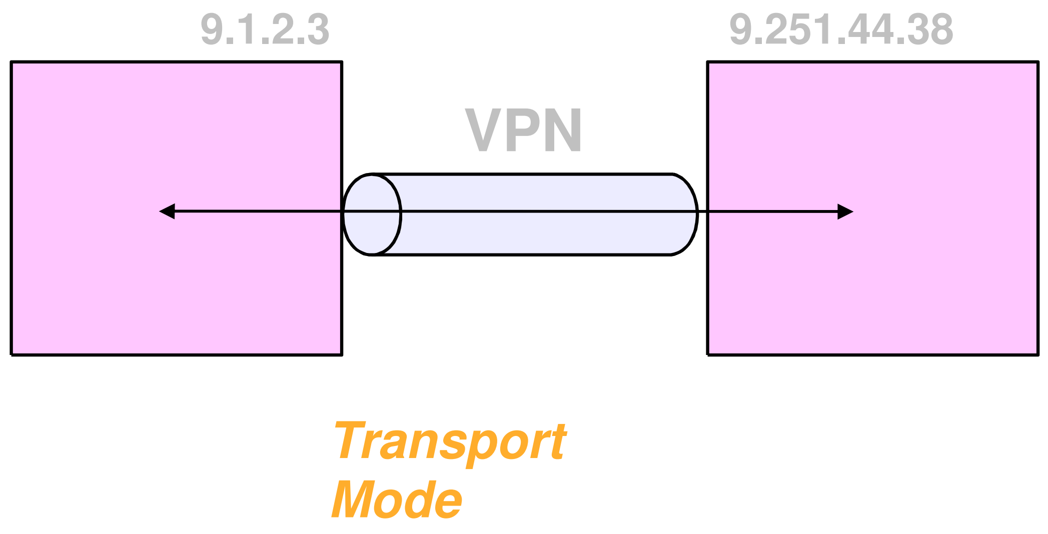
- **Two modes**
  - Transport mode
    - Inserts IPSec headers between original IP header and protected data
  - Tunnel mode
    - Creates a new IP header with an IPSec header
    - IPSec header followed by original IP header and protected data

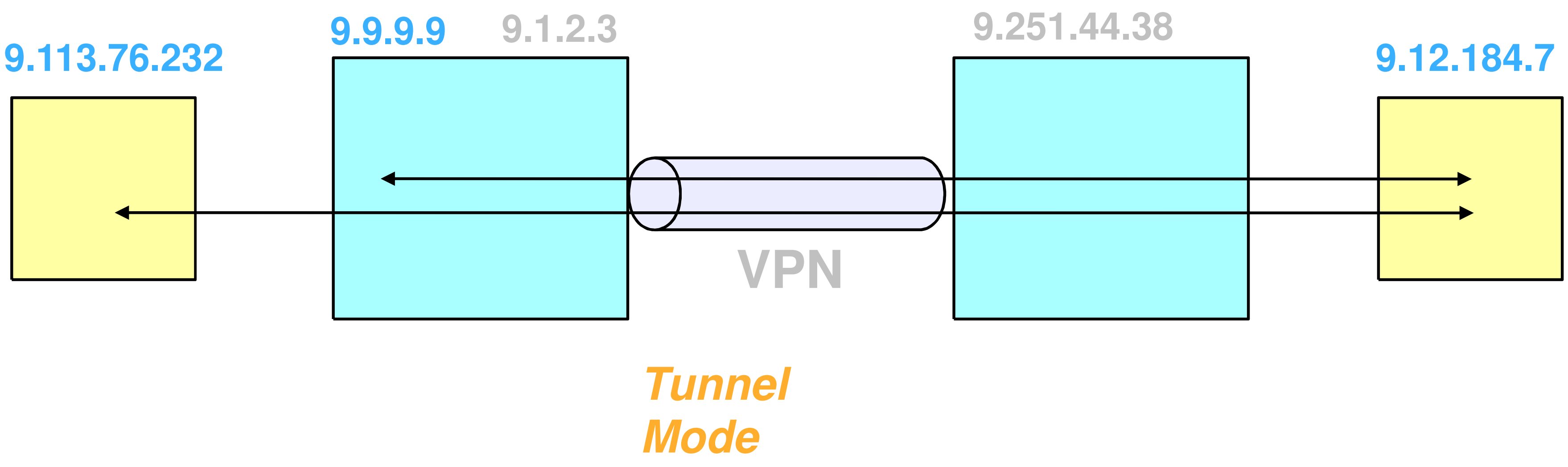
- **If one or both security endpoints are acting as a gateway**
  - Tunnel mode must be selected
  - The endpoints are ADDRESSES not HOSTS (z/OS usually has many addresses)

- **If neither security endpoint is acting as a gateway**
  - Tunnel or transport may be selected
  - Usually transport mode is used in this case
    - No need for extra cost of adding a new IP header in this case

- **The counterpart to encapsulation is decapsulation**

Transport & Tunnel mode with z/OS





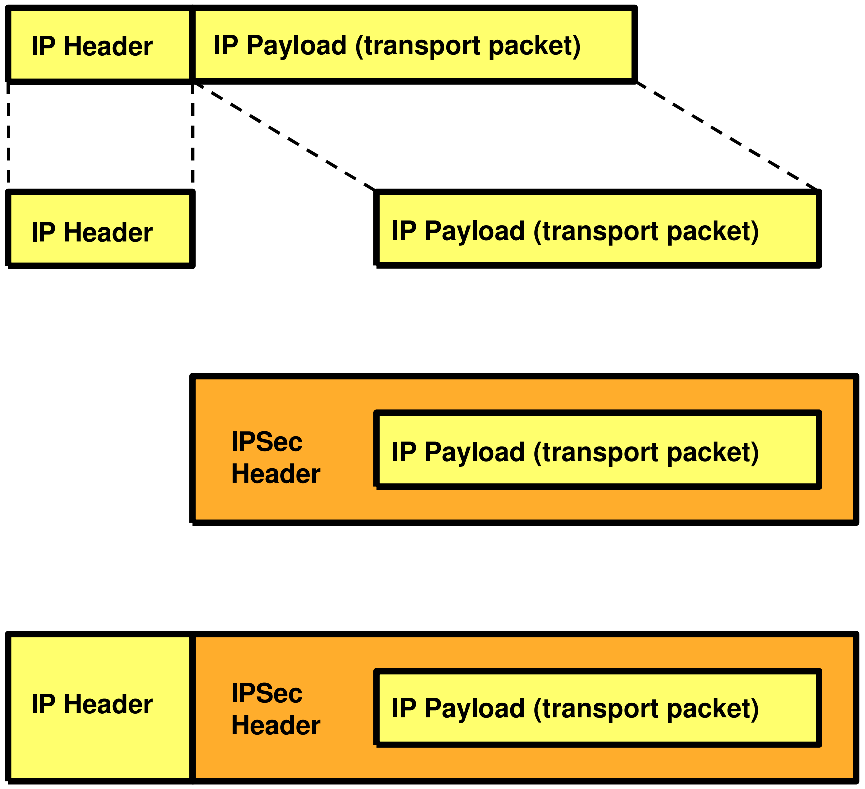
IPSec Packet Using Transport Mode

Original IP packet

Separate IP header and transport packet

Create IPSec packet

Attach and modify original IP header to IPSec packet


IP Header
IP Payload (transport packet)
IP Header
IP Payload (transport packet)
IPSec Header
IP Payload (transport packet)
IP Header
IPSec Header
IP Payload (transport packet)

Transport mode is typically used between two hosts that establish an IPSec VPN end-to-end between them.

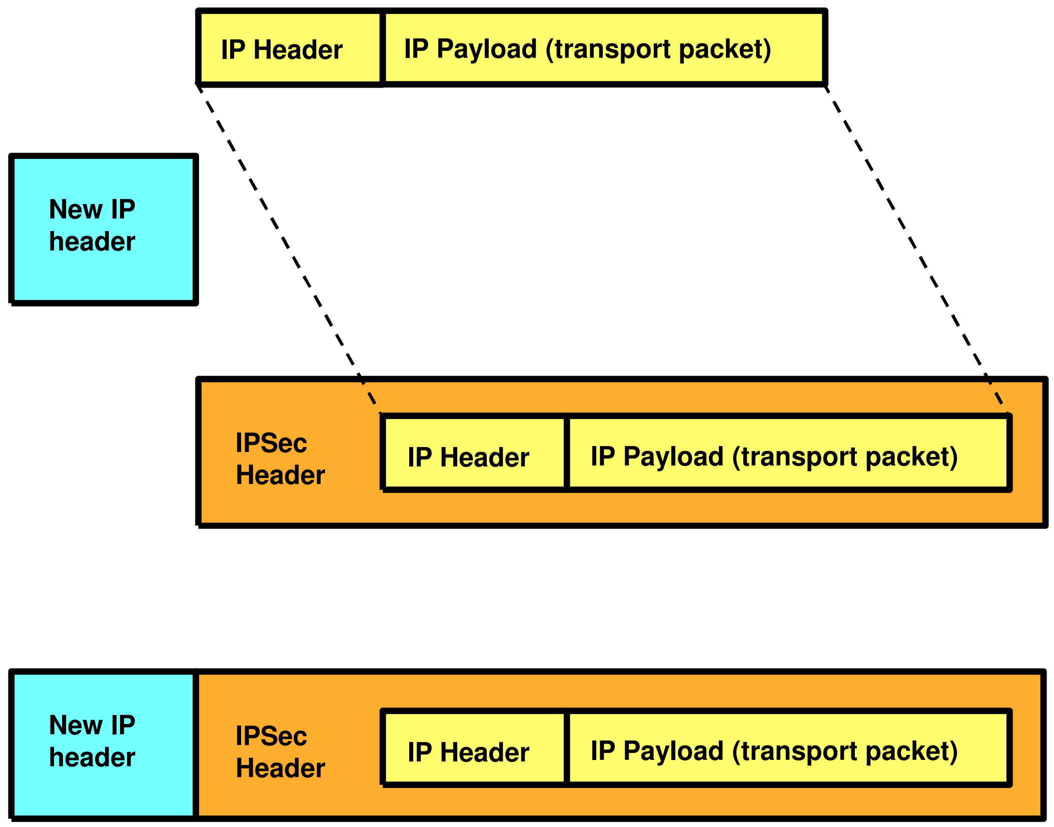
IPSec Packet Using Tunnel Mode

Original IP packet

Create new IP header

Create IPSec packet

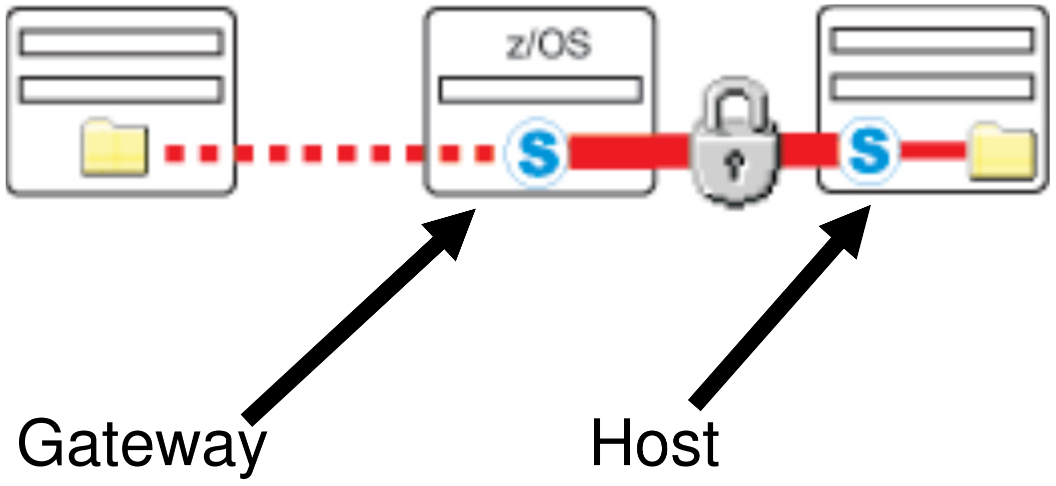
Update and attach new IP header to IPSec packet


IP Header
IP Payload (transport packet)
New IP header
IPSec Header
IP Header
IP Payload (transport packet)
New IP header
IPSec Header
IP Header
IP Payload (transport packet)

Tunnel mode is used if at least one of the two IPSec VPN endpoints is a gateway.

Security Endpoints

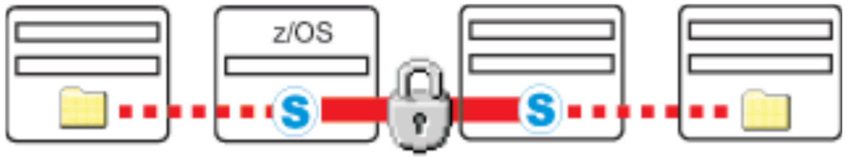
- **The endpoints of an IPSec secure channel**
  - Where IPSec protection is applied
- **Endpoint roles**
  - Host
    - Local data endpoint and secure channel endpoint are the same IP address
  - Gateway (or Security Gateway)
    - Local data endpoint and secure channel endpoint are different IP addresses



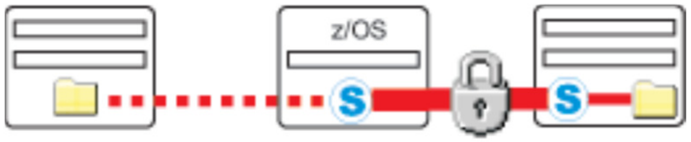
Encapsulation Mode Rules

- **Must use tunnel mode:**

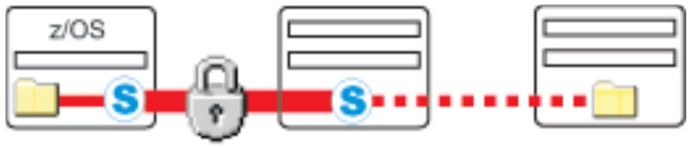
Gateway to Gateway



Gateway to Host

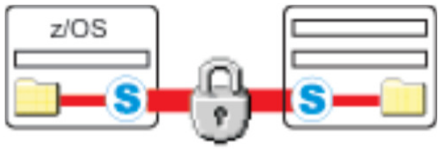


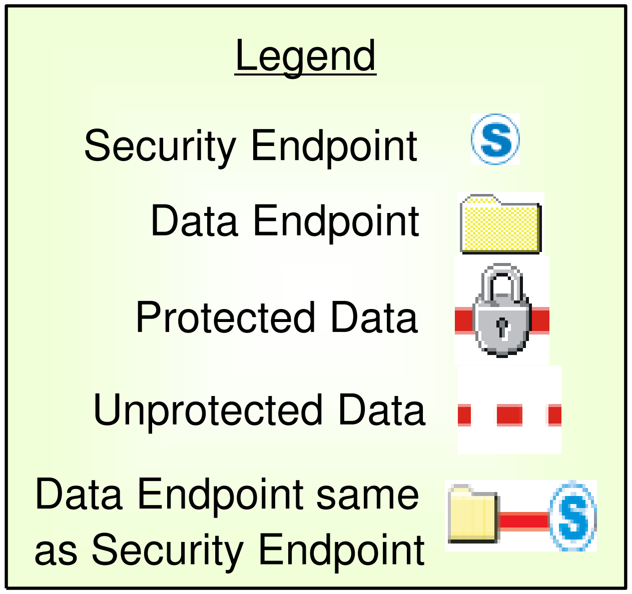
Host to Gateway



- **May use tunnel or transport mode:**

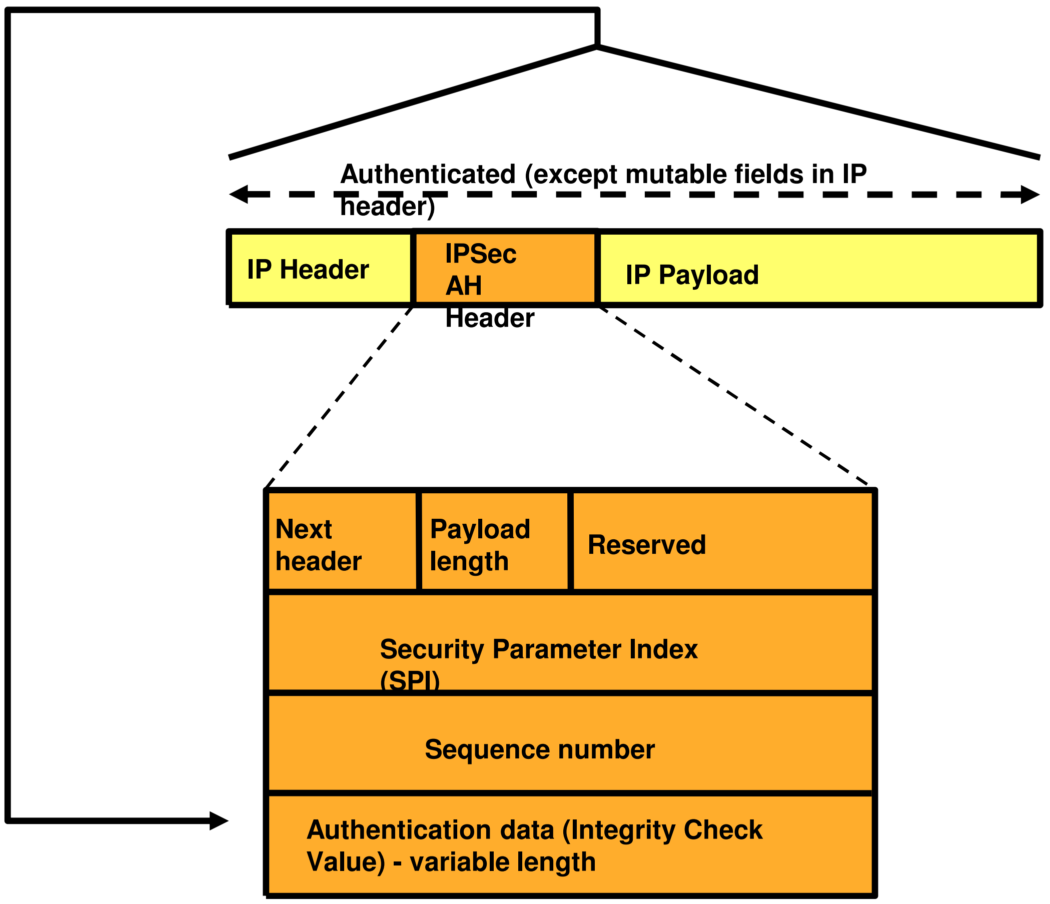
Host to Host





Authentication Header Protocol

IP Protocol number 51


Authenticated (except mutable fields in IP header)
IP Header
IPSec AH Header
IP Payload
Next header
Payload length
Reserved
Security Parameter Index (SPI)
Sequence number
Authentication data (Integrity Check Value) - variable length

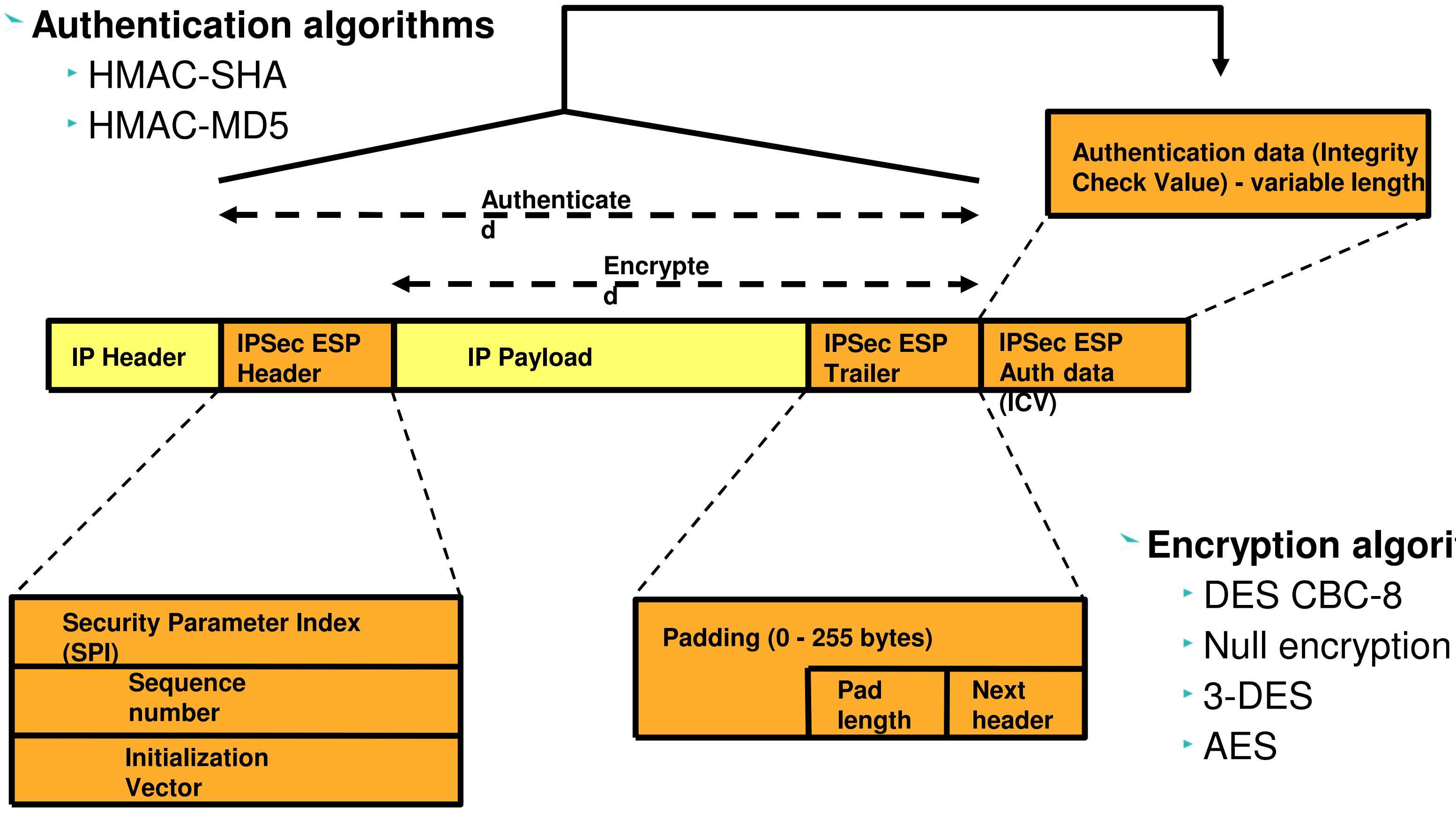
- **Authentication algorithms**
  - HMAC-SHA
  - HMAC-MD5

- **If transport mode, then "Payload" contains the original transport header and original data**
- **If tunnel mode, then "Payload" contains the original IP header, original transport header, and original data**

Encapsulating Security Payload

IP Protocol number 50

- **Authentication algorithms**
  - HMAC-SHA
  - HMAC-MD5

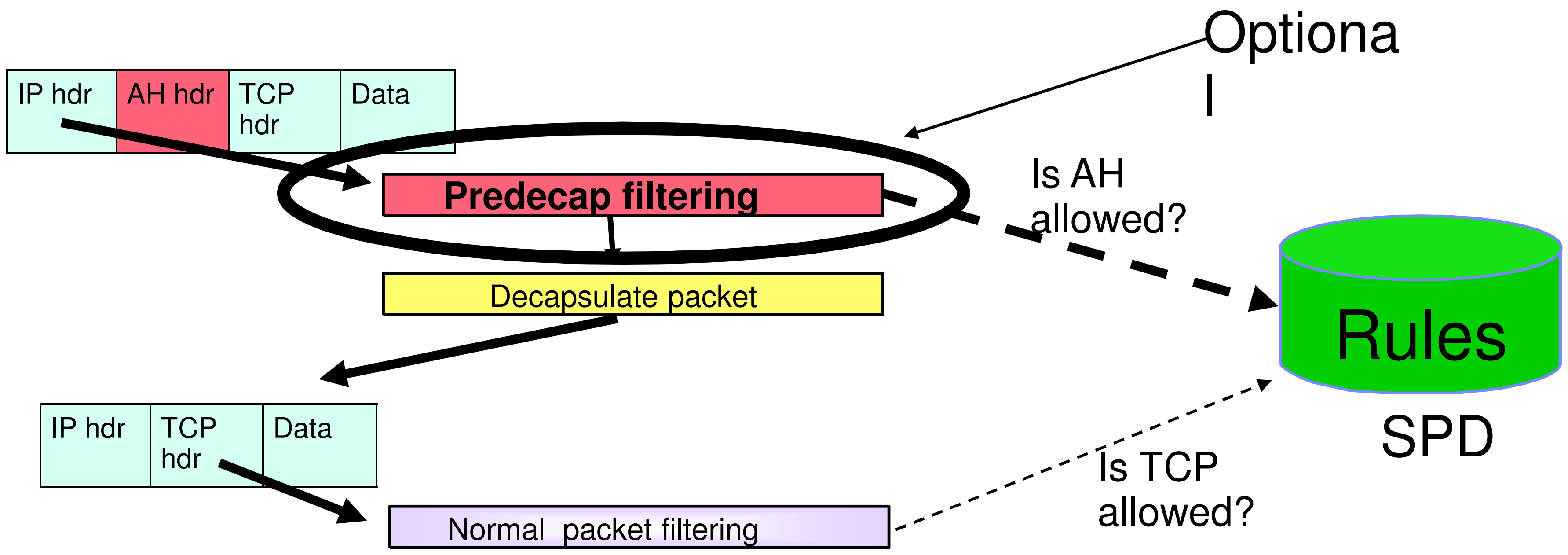


- **Encryption algorithms**
  - DES CBC-8
  - Null encryption
  - 3-DES
  - AES

- **If transport mode, then "Payload" contains the original transport header and original data (possibly encrypted)**
- **If tunnel mode, then "Payload" contains original IP header, original transport header, and original data**
  - "Payload" can be encrypted

Predecap Filtering

- IPSec protected traffic arrives as an AH or ESP packet (UDP-encapsulated ESP packets are interpreted as ESP packets; see charts on UDP-encapsulation)
- The stack can optionally perform filtering on AH/ESP packets before decapsulation
  - Known as predecap filtering
  - Prevents decapsulation of AH/ESP traffic from unacceptable sources
- The AH/ESP packet is then decapsulated revealing the original packet
  - Filtering is always performed on the decapsulated packet



Security Associations

- **IPSec secure channel endpoints must agree on how to protect traffic**
  - Security protocol
    - AH
    - ESP
  - Algorithms to be used by the security protocols
    - Encryption Algorithm
      - DES or Triple DES or AES
    - Authentication Algorithm
      - HMAC_MD5 or HMAC_SHA
  - Cryptographic keys
  - Encapsulation mode
    - Tunnel
    - Transport
  - Lifetime/lifesize (for dynamic SAs)
- **This agreement is known as a "security association" - or for short, an SA**

Security Associations

- **Used to protect IP traffic**

- **Unidirectional**
  - Need one for inbound and another for outbound - each IPSec secure channel endpoint consists of two SAs
    - Generally symmetrical with regards to algorithms used
    - Cryptographic keys will be different
  - A pair of matching SAs are, on z/OS, referred to as a "Tunnel ID" - in a sense identifying the secure channel

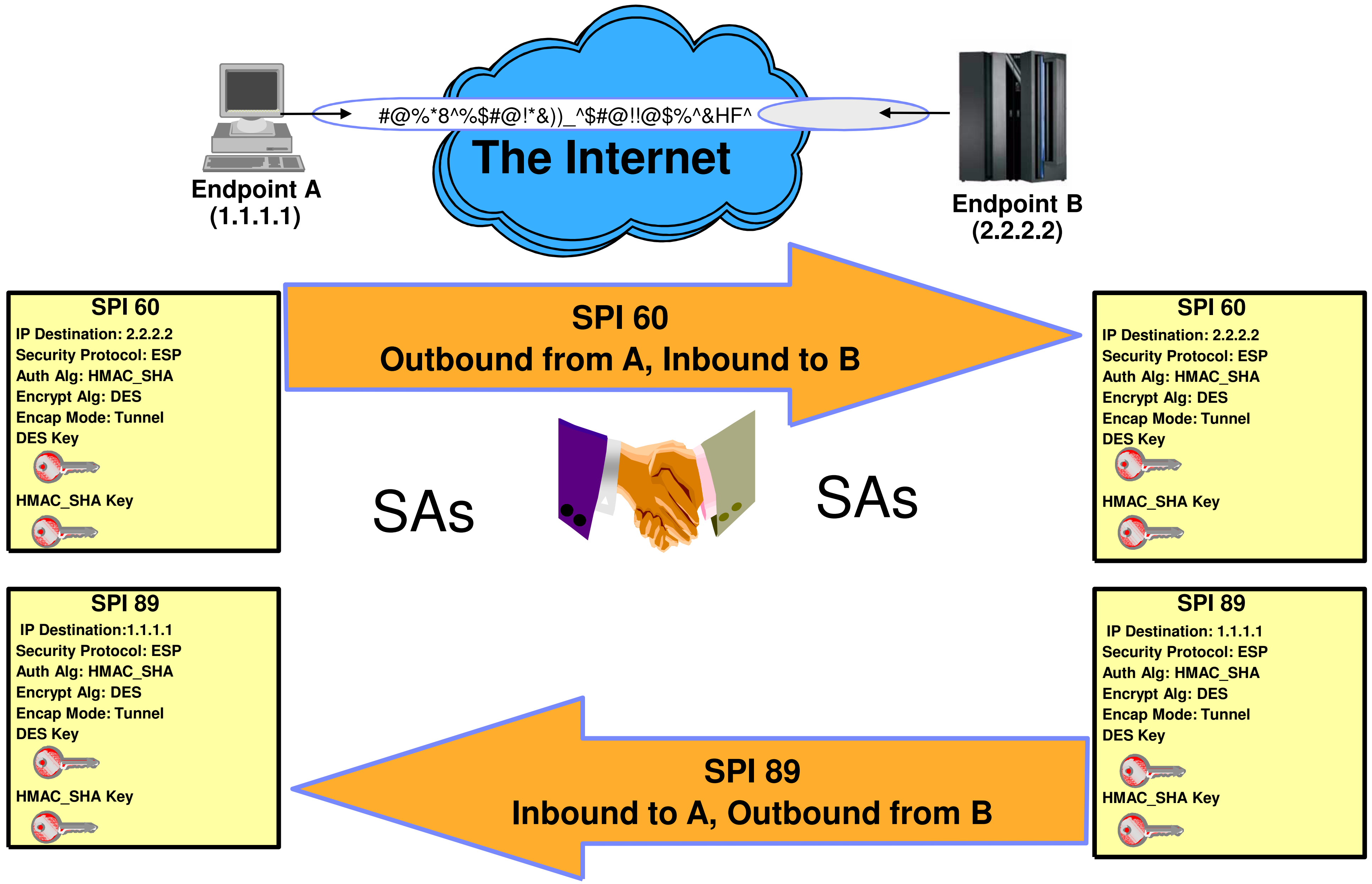
- **An SA is identified by:**
  - A Security Parameter Index (SPI)
    - The SPI is a 32-bit value
    - SPI numbers in themselves may not be unique on a given IPSec node
    - The SPI is carried in the IPSec headers
  - IPSec protocol
  - Destination IP address information

- **Manually defined SAs**
  - Statically defined in the Security Policy Database (SPD - Pagent IPSec config file)

- **Dynamically defined SAs**
  - Negotiated using the Internet Key Exchange protocol
  - Acceptable values (policy) defined in the SPD (Pagent IPSec config file)

- **Security Association Database (SAD)**
  - The collection of all SAs known to the stack

SA Example



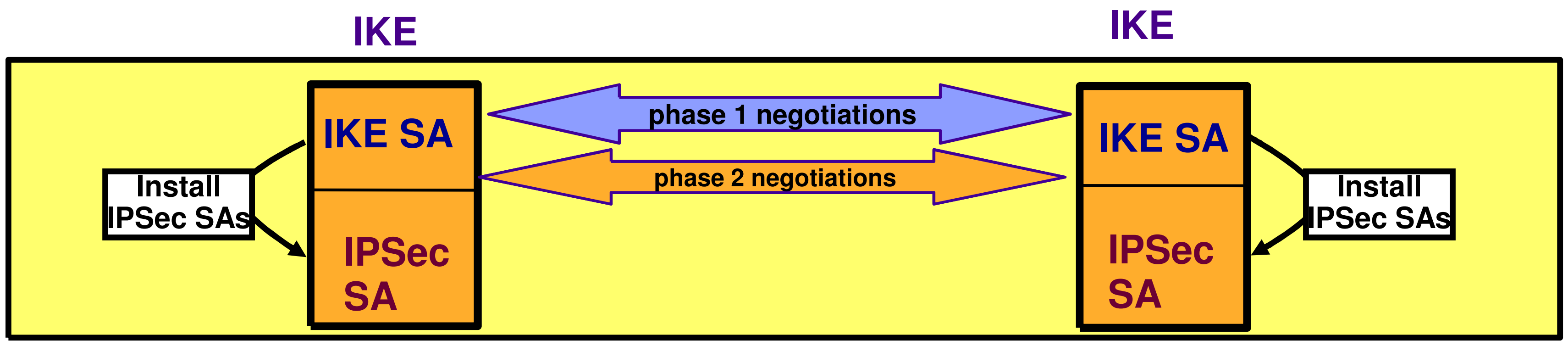
#@%*8^%$#@!*&))_^$#@!!@$%^&HF^
The Internet
Endpoint A (1.1.1.1)
Endpoint B (2.2.2.2)
SPI 60
IP Destination: 2.2.2.2
Security Protocol: ESP
Auth Alg: HMAC_SHA
Encrypt Alg: DES
Encap Mode: Tunnel
DES Key
HMAC_SHA Key
SPI 60
Outbound from A, Inbound to B
SAs
SAs
SPI 60
IP Destination: 2.2.2.2
Security Protocol: ESP
Auth Alg: HMAC_SHA
Encrypt Alg: DES
Encap Mode: Tunnel
DES Key
HMAC_SHA Key
SPI 89
IP Destination:1.1.1.1
Security Protocol: ESP
Auth Alg: HMAC_SHA
Encrypt Alg: DES
Encap Mode: Tunnel
DES Key
HMAC_SHA Key
SPI 89
Inbound to A, Outbound from B
SPI 89
IP Destination: 1.1.1.1
Security Protocol: ESP
Auth Alg: HMAC_SHA
Encrypt Alg: DES
Encap Mode: Tunnel
DES Key
HMAC_SHA Key

Manual or Dynamic SAs

- Manual SA
  - Not often used - less secure
  - Symmetric keys defined at each end
  - Authentication is done via possession of key
  - Keys need to be updated regularly
- Dynamic SA
  - Symmetric keys generated securely
  - Authentication may be done by certificates, or shared keys
  - Keys are automatically refreshed at intervals
  - Standards used are
    - IKE (Internet Key Exchange) and
    - ISAKMP (Internet Security Association Key Management Protocol, a.k.a. Oakley)
    - Done using UDP port 500

Two Phases of Key Negotiation for a Dynamic Tunnel

- **Phase 1 negotiation**
  - Creates a secure channel with a remote security endpoint
    - Negotiates an IKE SA
      - Generates cryptographic keys that will be used to protect Phase 2 negotiations and Informational exchanges
      - Authenticates the identity of the parties involved
      - Bidirectional, and not identified via SPIs
  - Requires processor-intensive cryptographic operations
  - Done infrequently
- **Phase 2 negotiation**
  - Negotiates a pair of IPSec SAs with a remote security endpoint
    - Generates cryptographic keys that are used to protect data
      - Authentication keys for use with AH
      - Authentication and/or encryption keys for use with ESP
  - Performed under the protection of an IKE SA
  - Done more frequently than phase 1



ISAKMP Security Associations

- **Used to protect Phase 2 negotiations**
- **Bidirectional**
- **Endpoints must agree on**
  - Encryption algorithm
    - DES/Triple DES/AES...
  - Hash Algorithm
    - MD5/SHA1...
  - Authentication Method
    - Preshared Key
    - RSA Signature
  - Diffie-Hellman Group
  - Lifetime/Lifesize
- **Policy definition is based on identities exchanged during phase 1**
  - Key Exchange Policy
    - A set of filter rules for IKE

IKE/ISAKMP Details

- There are two different phase 1 exchange modes. Both exchange the same information, but one utilizes fewer messages.
  - Main mode
    - All IPSec implementations must support main mode. Main mode utilizes 6 messages. The last two messages contain identity information and are encrypted. This provides identity protection.
  - Aggressive mode
    - Some IPSec implementations do not support aggressive mode. Aggressive mode utilizes 3 messages. No messages are encrypted.
- Identity information is used to locate policy. Phase 1 identity types supported by Integrated IPSec include:
  - An IPv4 address (this identity type should not be used when behind a NAT)
  - RFC 822 name (for example, email address)
  - Fully qualified domain name (FQDN)
  - x500 distinguished name (DN)
- Authentication modes
  - Preshared key
    - Security endpoint administrators agree to this value. The key is not directly used to encrypt data.
    - Often used during the initial stages of dynamic SA deployment
  - RSA signature
    - Require X509 certificates.
      - Certificates need to contain an endpoint's identity in the certificate's SubjectName (for DNs) or the SubjectAlternate name (for RFC 822 names, FQDNs, or IPv4 addresses).
    - Often used when dynamic SA are widely deployed.
- Diffie-Hellman is an algorithm that allows IKE to produce cryptographic keying material. Diffie-Hellman groups are defined in RFC 2409 (IKE). Original options are groups 1 and 2. Group 2 provides better security characteristics, but it also requires more computational power. Groups 5 and 14 are new for use with AES.

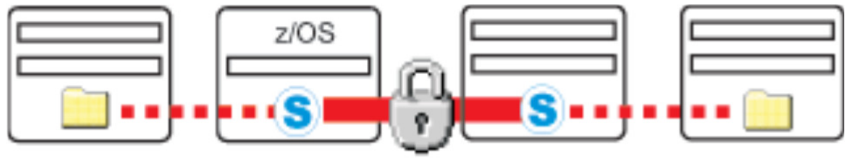
Perfect Forward Secrecy

- Perfect forward secrecy
  - Refers to the notion that the compromise of a single key will only permit access to data protected by that key
    - Compromise of the keys negotiated in phase 1 will not compromise keys generated in phase 2
    - Compromise of the keys negotiated in phase 2 will not compromise future phase 2 keys or previously generated phase 2 keys
- PFS is optional
  - Accomplished by performing an optional Diffie-Hellman exchange during phase 2
    - The Diffie-Hellman exchange during Phase 1 SA is not optional
- Factors to consider
  - Frequency that IKE SAs are refreshed (Phase 1)
  - Frequency that IPSec SAs are refreshed (Phase 2)
  - Key size

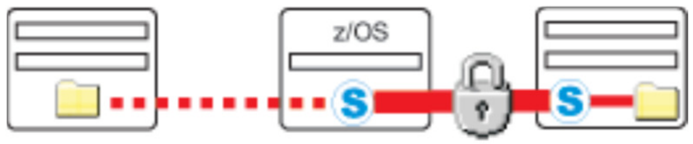
IPSec without NAT Traversal

- Tunnel mode with AH and/or ESP

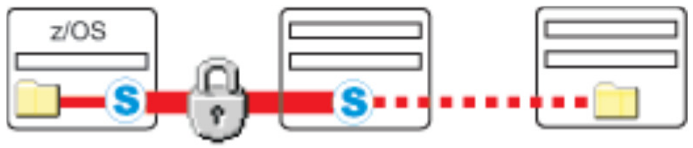
Gateway to Gateway



Gateway to Host

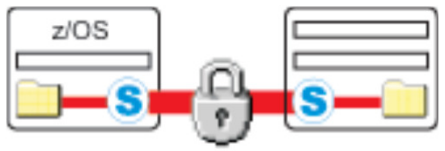


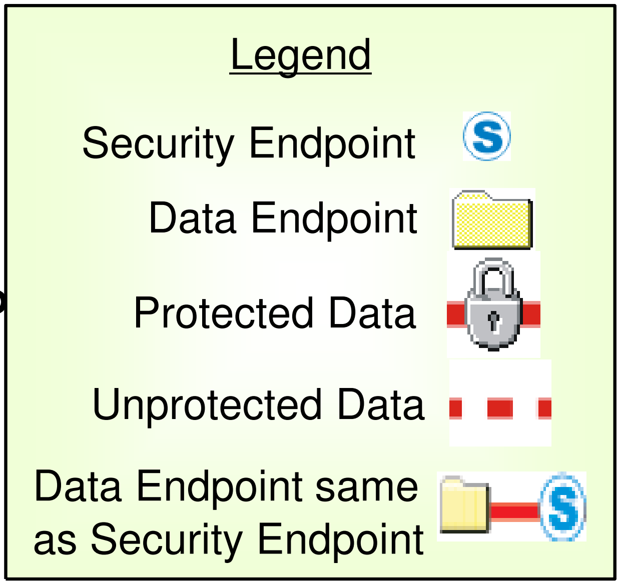
Host to Gateway



- Tunnel or transport mode with AH and/or ESP

Host to Host

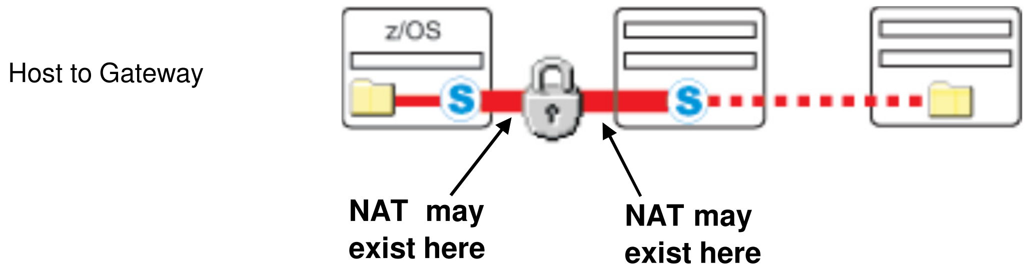




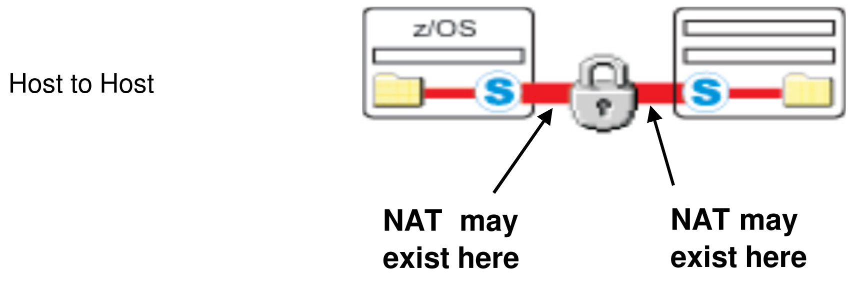
IPSec with NAT (or NAPT) in the way

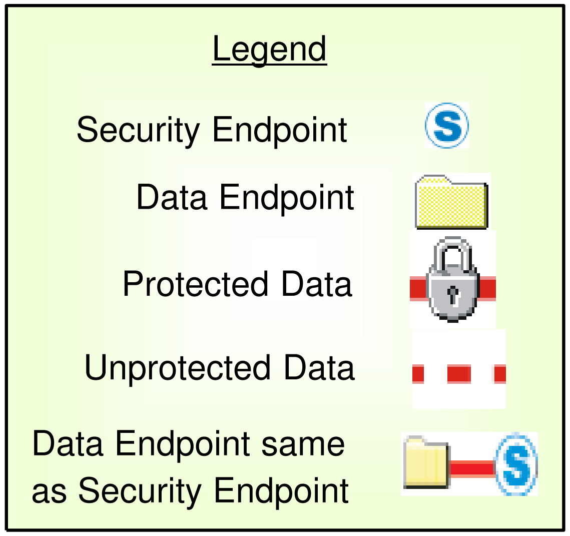
- **IPSec carries endpoint IP addresses in data (and they are secured)**
- **Solution is to encapsulate ESP packet in UDP packet**
- **NAT is discovered dynamically by IKE daemon**
  - Therefore, dynamic tunnels only
  - Additional flows in Phase 1 Exchange
- **If the responder of an SA negotiation is behind a NAT or NAPT firewall, a static NAT mapping should be used**

- Tunnel mode with ESP



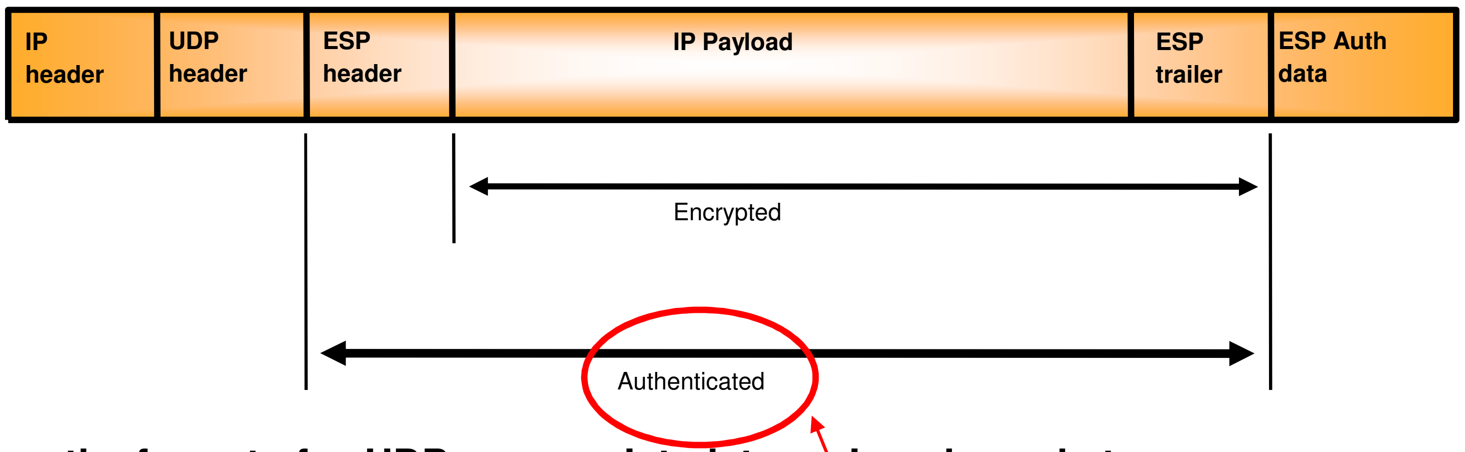
- Tunnel or transport mode with ESP



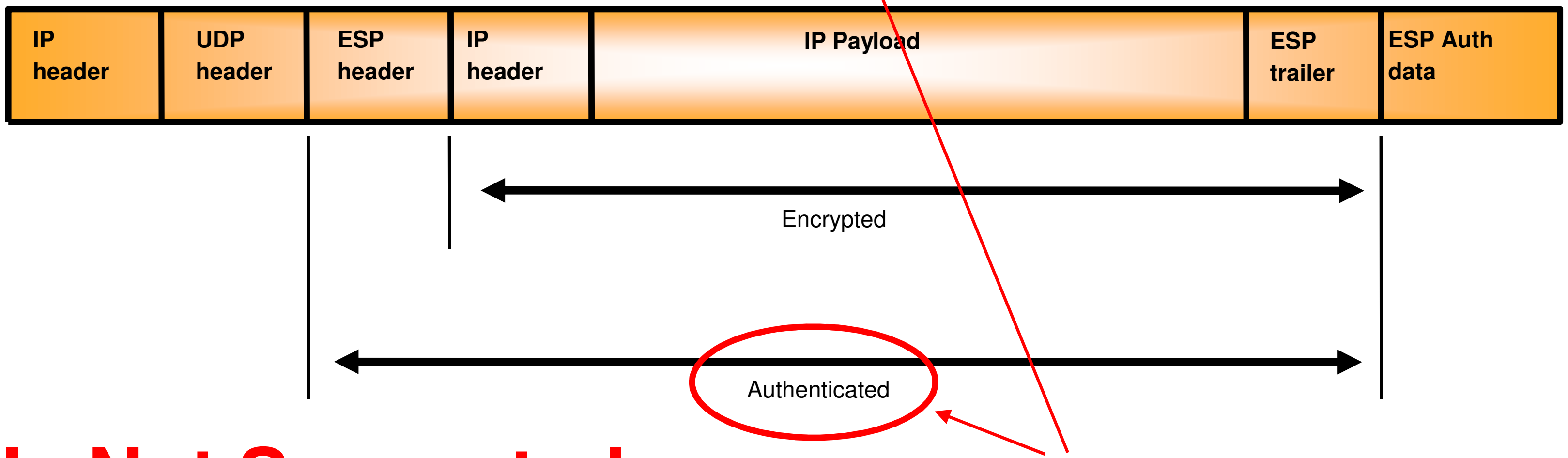


NAT Traversal - Another Encapsulation Mode

- **This shows the format of a UDP-encapsulated transport mode packet**



- **This shows the format of a UDP-encapsulated tunnel mode packet**

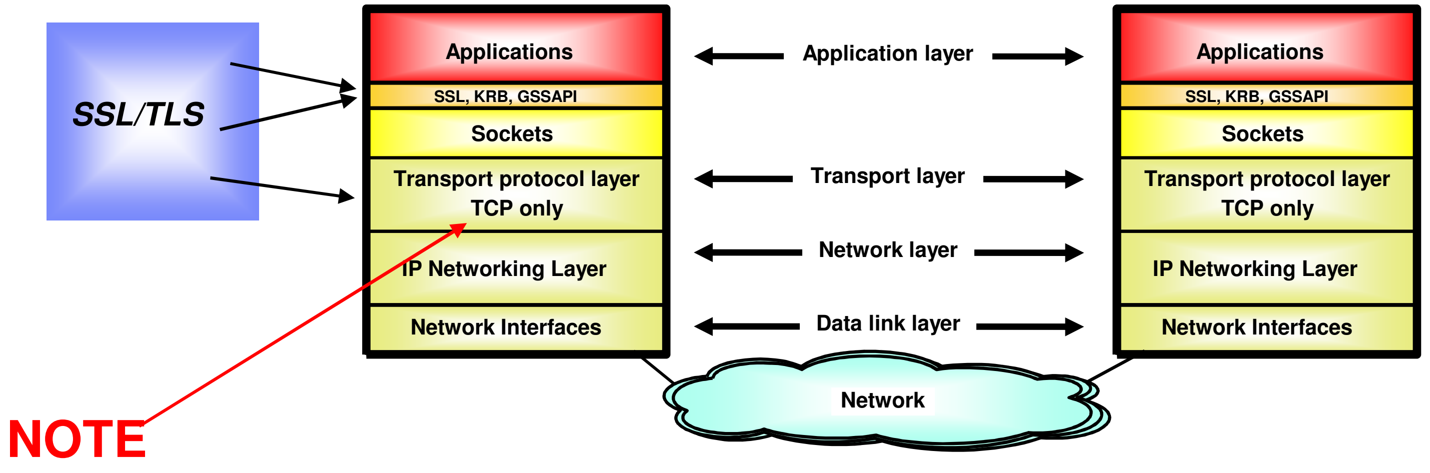


AH Is Not Supported - and this is why

Secure Sockets Layer and Transport Layer Security

- **SSL History**
- **Client and Server authentication**
- **SSL/TLS Protocols and Flows**

SSL/TLS Overview



- **Transport Layer Security (TLS) is defined by the IETF (RFC 2246)**
  - Based on Secure Sockets Layer (SSL)
    - Above sockets layer, below application layer
    - SSL originally defined by Netscape to protect HTTP traffic
    - V1 no longer supported
    - V2 still exists, concerns about efficacy
    - V3 is common
    - TLS is V3.1; TLS implementers should drop to V3 if partner is back level
- **End to End Application "pipe"**
  - Requires reliable transport layer, therefore TCP only
  - Server always authenticated, client is optional
    - Always uses certificate
    - Client authentication is often ID/password, but encrypted
  - Application source code changes are generally needed to enable a Sockets program for TLS

SSL/TLS Functions

- TLS provides:
  - Message privacy
    - Using symmetric key encryption.
    - Uses a handshake when initiating contact. The handshake establishes a session key and encryption algorithm, between both parties, prior to any messages being sent.
  - Message integrity
    - By using the combination of shared secret key and cryptographic hash functions.
    - This ensures that the content of any messages does not change.
  - Mutual authentication
    - Server and (optionally) client authenticate to each other.
    - This happens during the handshake.

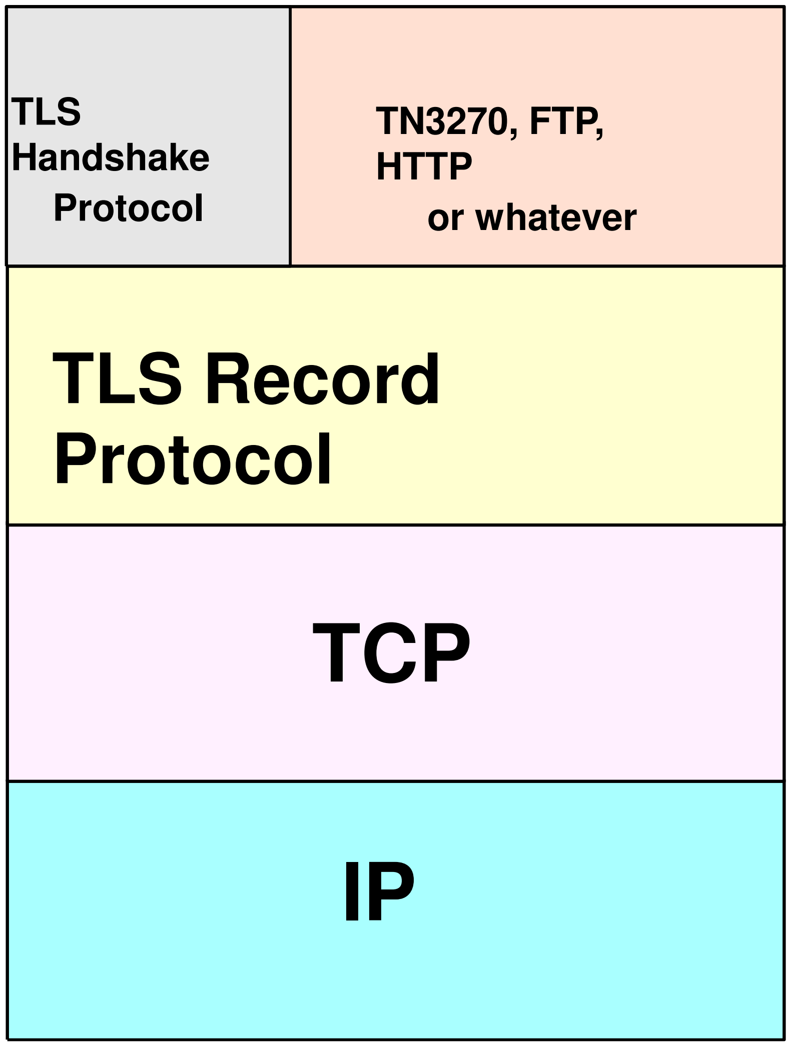
SSL/TLS Contrasted with IPSec

1. IPSec is stack-level
2. IPSec protects all traffic between specified address ranges
3. IPSec can protect any IP protocol
4. IPSec can do manual or dynamic tunnels
5. IPSec can do shared keys or certificate authentication
6. IPSec must authenticate both partners
7. IPSec can do data integrity, and optionally privacy (encryption)

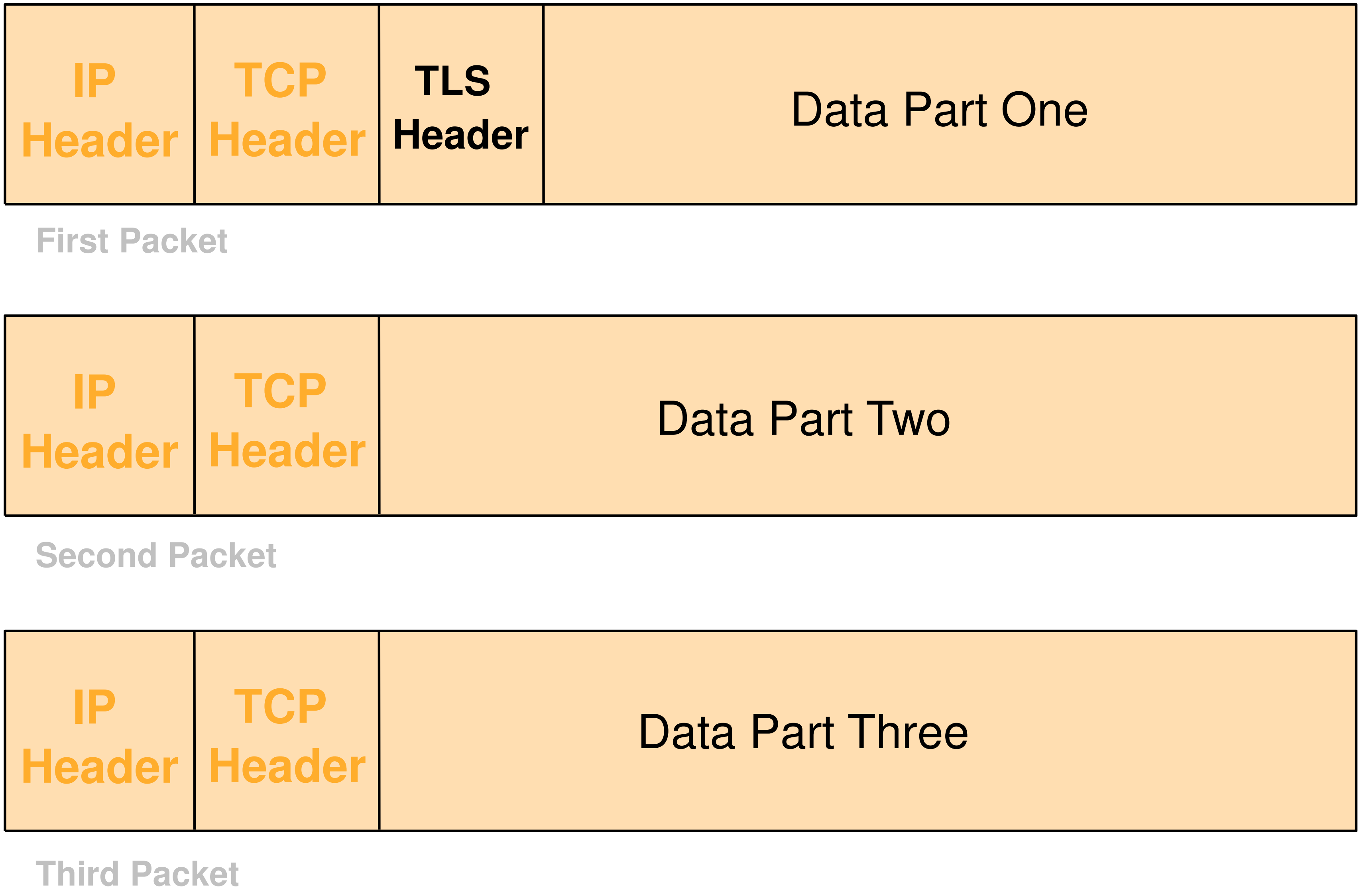
1. SSL/TLS is application-specific
2. SSL/TLS protects all traffic on a specified TCP connection
3. SSL/TLS can only protect TCP
4. SSL/TLS can only do dynamic sessions
5. SSL/TLS must always use certificates
6. SSL/TLS must authenticate the server, and optionally the client
7. SSL/TLS always does privacy (encryption)

The same algorithms (DES, 3DES, AES, MD5, SHA, RSA, DSA.... are used by both SSL/TLS and IPSec.

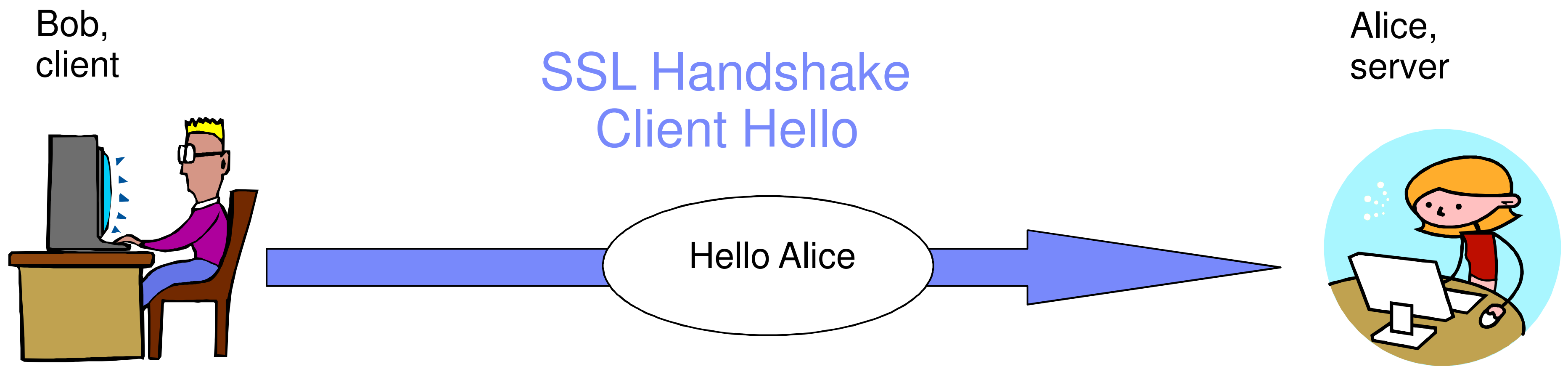
SSL/TLS Protocol Stack



SSL/TLS Packet Format

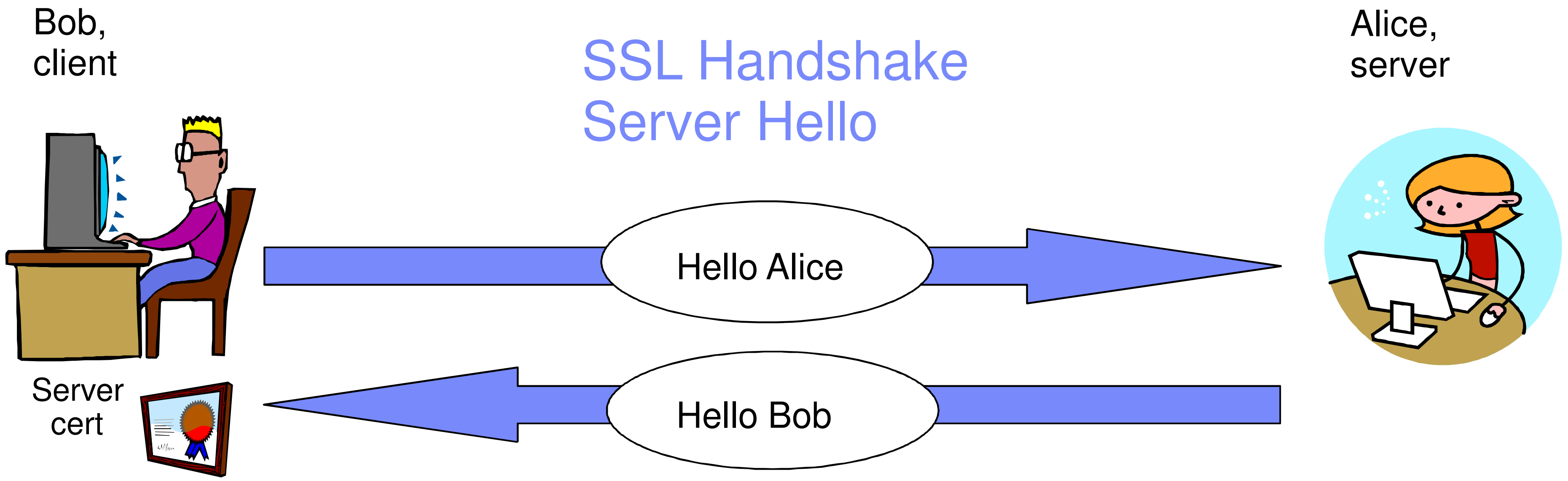


SSL/TLS Handshake (1)



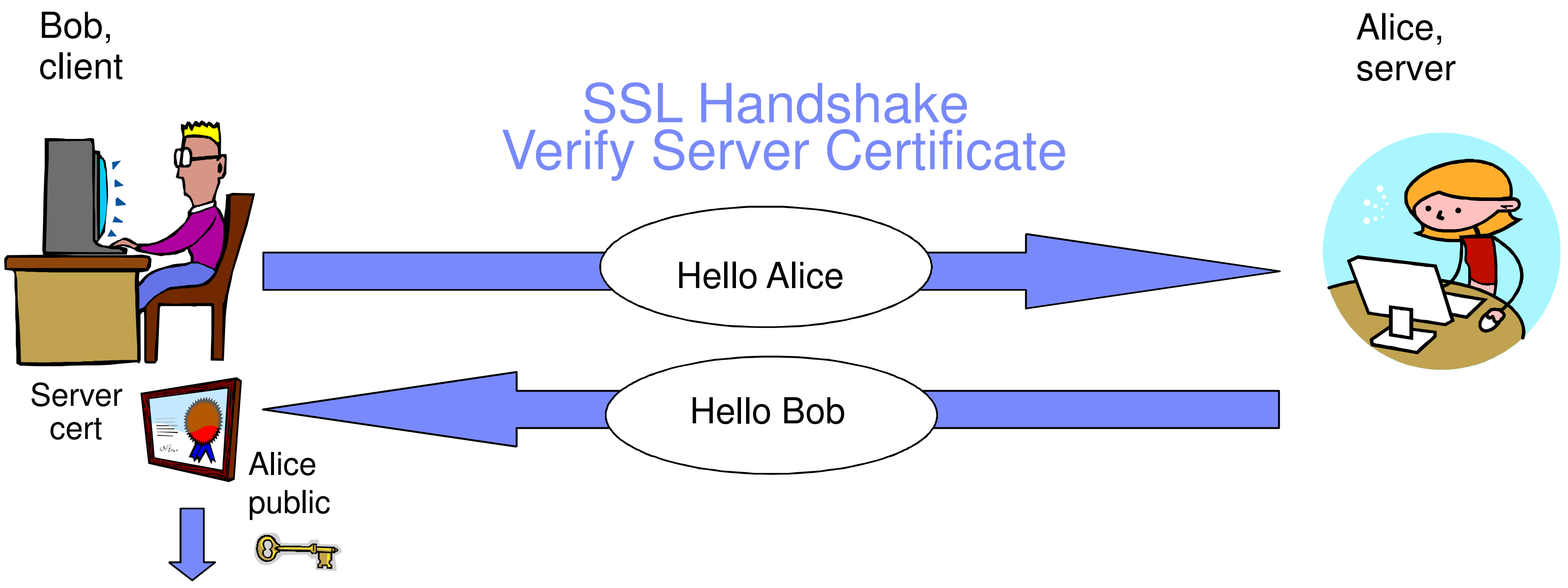
1. Bob sends Alice a hello message.
2. Within this initial contact message is a list of Cipher Suites that Bob (the client) can use.
3. The list is in client preference.
4. Bob is considered the client since he initiated communication.

SSL/TLS Handshake (2)



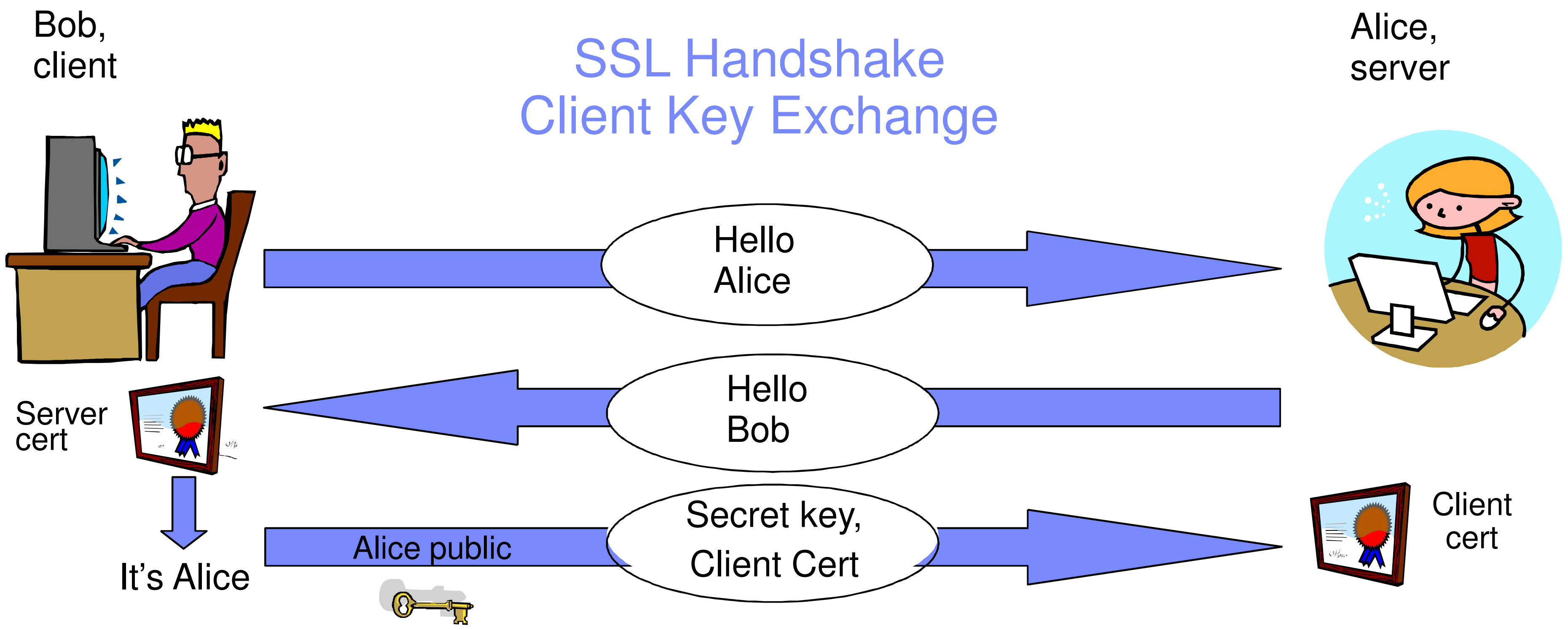
1. Alice (the server) responds with a "Hello"
2. Alice also sends Bob a signed Digital Certificate containing her public key
3. Bob will already have a CA certificate in a key database, against which to verify Alice's certificate
4. This response contains the CipherSuite that Alice has selected from the list Bob sent her

SSL/TLS Handshake (2a)



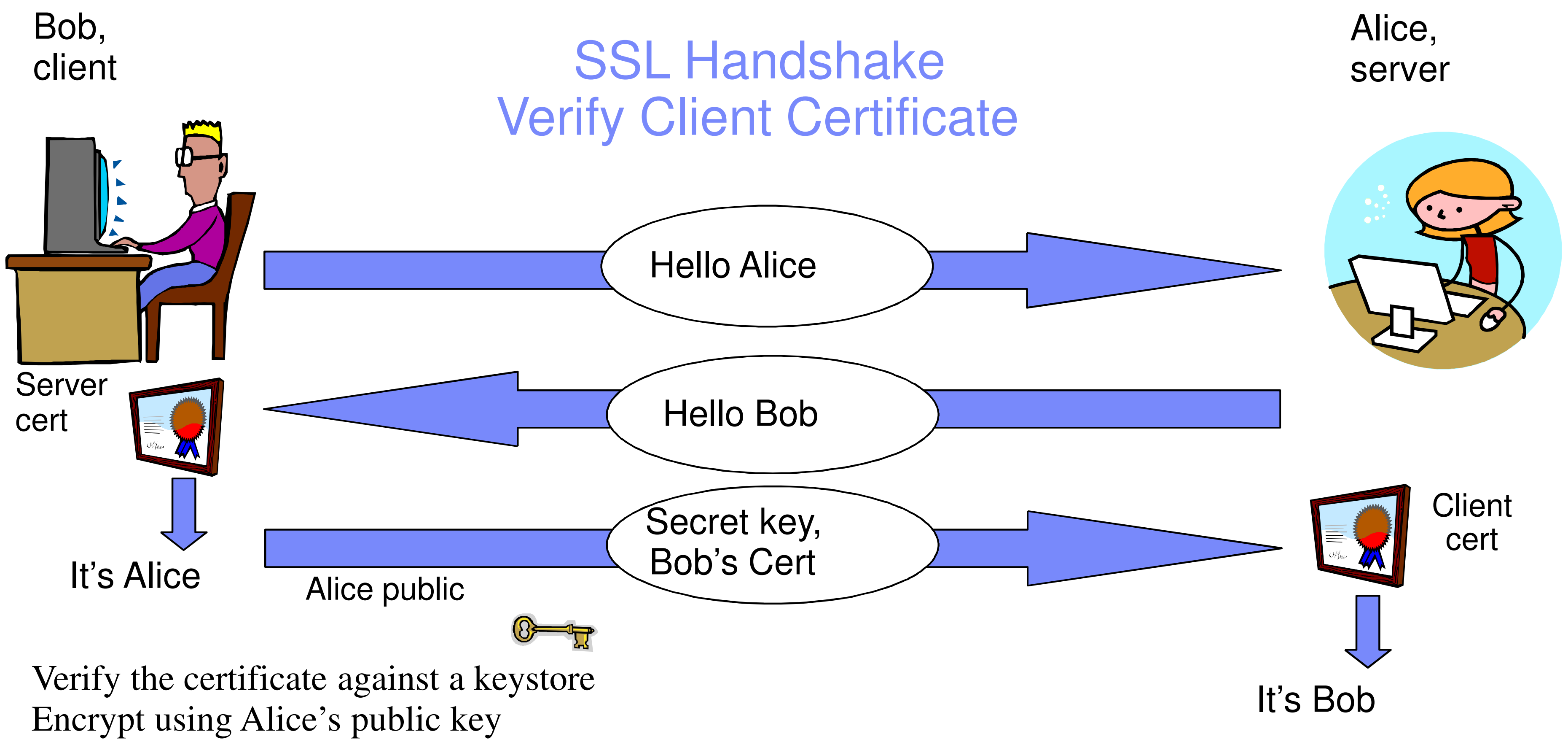
1. Receive Alice's public key
2. Check if the certificate has expired
3. Verify the certificate's signature against a CA certificate
4. If client authentication is being used, then Alice would request a digital certificate

SSL/TLS Handshake (3)



1. The client sends the server a secret symmetric key that is encrypted using the server's certificate and verify message
2. If client Authentication was requested, Bob would send a client public key.

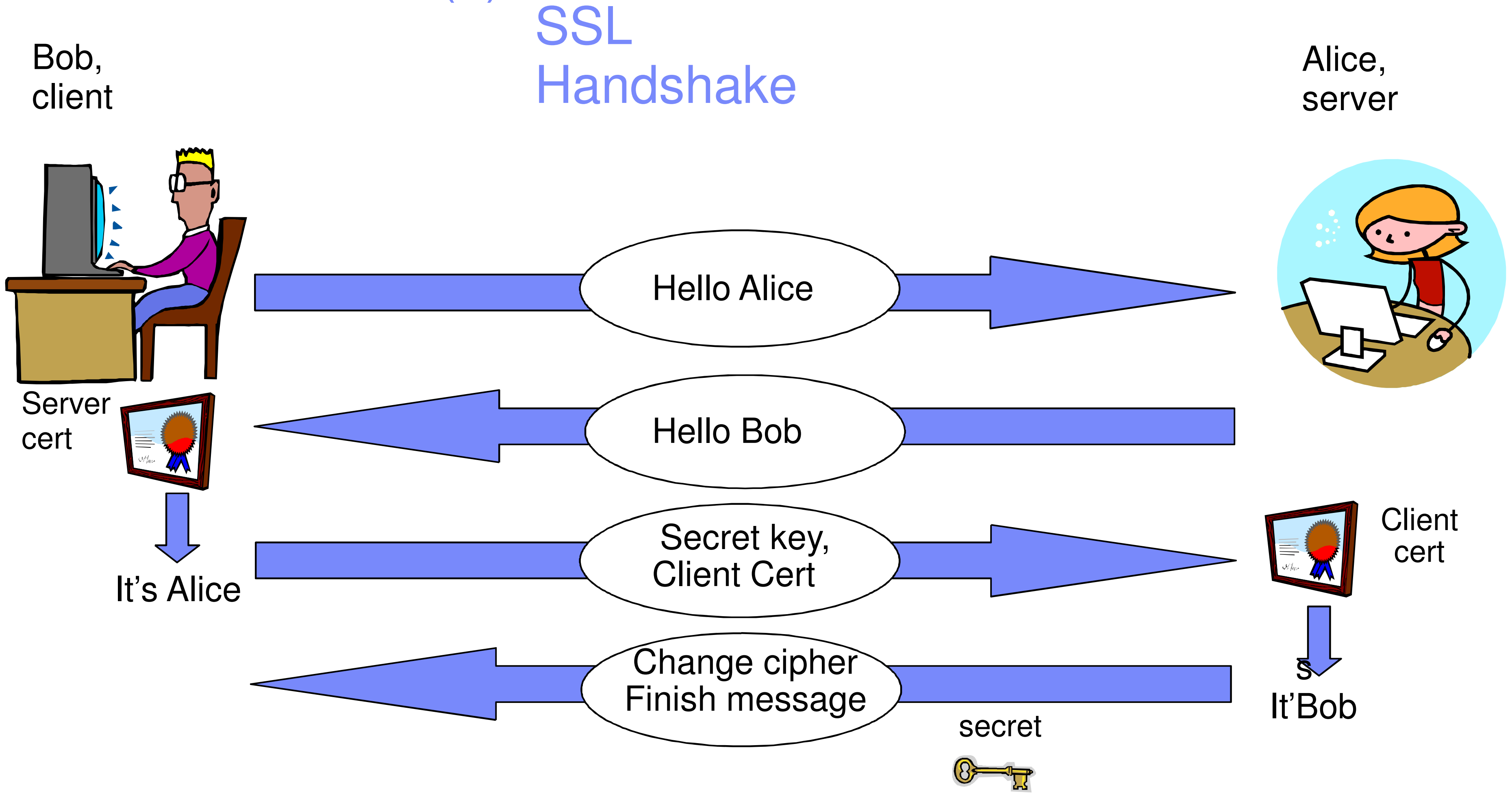
SSL/TLS Handshake (3a)



1. Verify the certificate against a keystore
2. Encrypt using Alice's public key
3. Client key exchange message
4. Change CipherSpec message
5. Finished message

Verify certificate against keystore

SSL/TLS Handshake (4)



1. Encrypted using secret key
2. Change CipherSuite message
3. Finish Message

END OF
PART ONE