Network Security

Learning Objectives

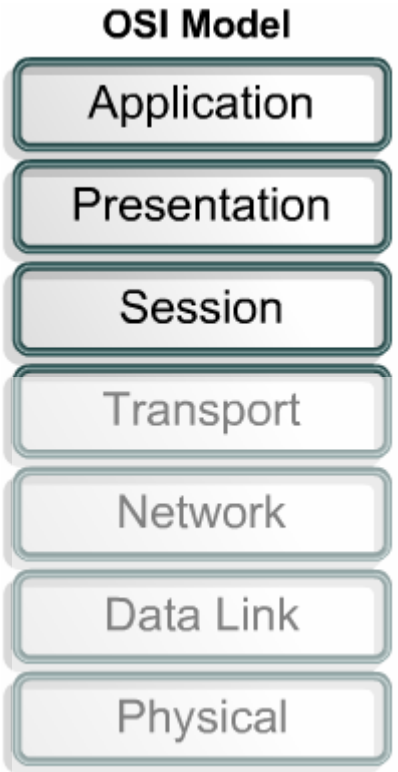
- Learn about network fundamentals
- Understand the network layer and traffic engineering

Network fundamentals

The OSI model

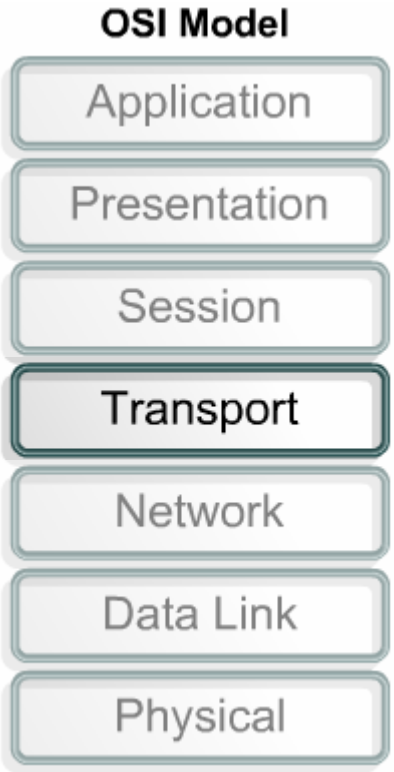
- Open Systems Interconnection (OSI) Reference Model
- Developed by ISO
  - Attempt to prompt interoperability
- A protocol defines rules
- Protocol suites define a series of protocols and interactions.
- Layers talk to each other using peer communications
- Each layer has its own protocol data unit (PDU)
  - Protocol specific information and user data

Upper layers



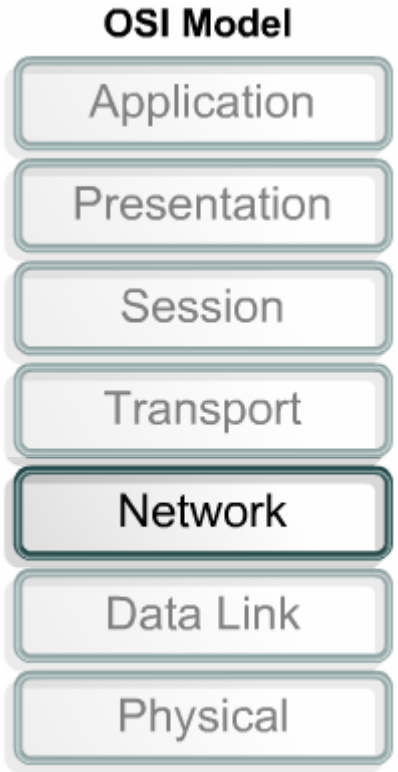
- Application: handle file transfer, virtual terminals and fulfilling networking request of applications
  - Defines what data should be transmitted
  - FTP, HTTP, Telnet, etc. The protocols that are implemented within
- Presentation: handle translation into standard format, data compression/decompression, data encryption/decryption
  - Specifies the way that data should be represented (ASCII, UNICODE, GIF, JPEG)
- Session: setup connection between applications; maintain control, negotiate, establish and close communication channel
  - Manages multiple transport layer connections to provide a session (consider FTP), remote procedure call (RPC)

Transport layer



- Transport: handles end-to-end transmission and segmentation of a data stream
  - Transport services (connection and connectionless)
  - Application access points (ports)
- Main protocols
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)
  - Secure Sockets Layer (SSL)
  - Sequence Packet Exchange (SPX)

Network layer

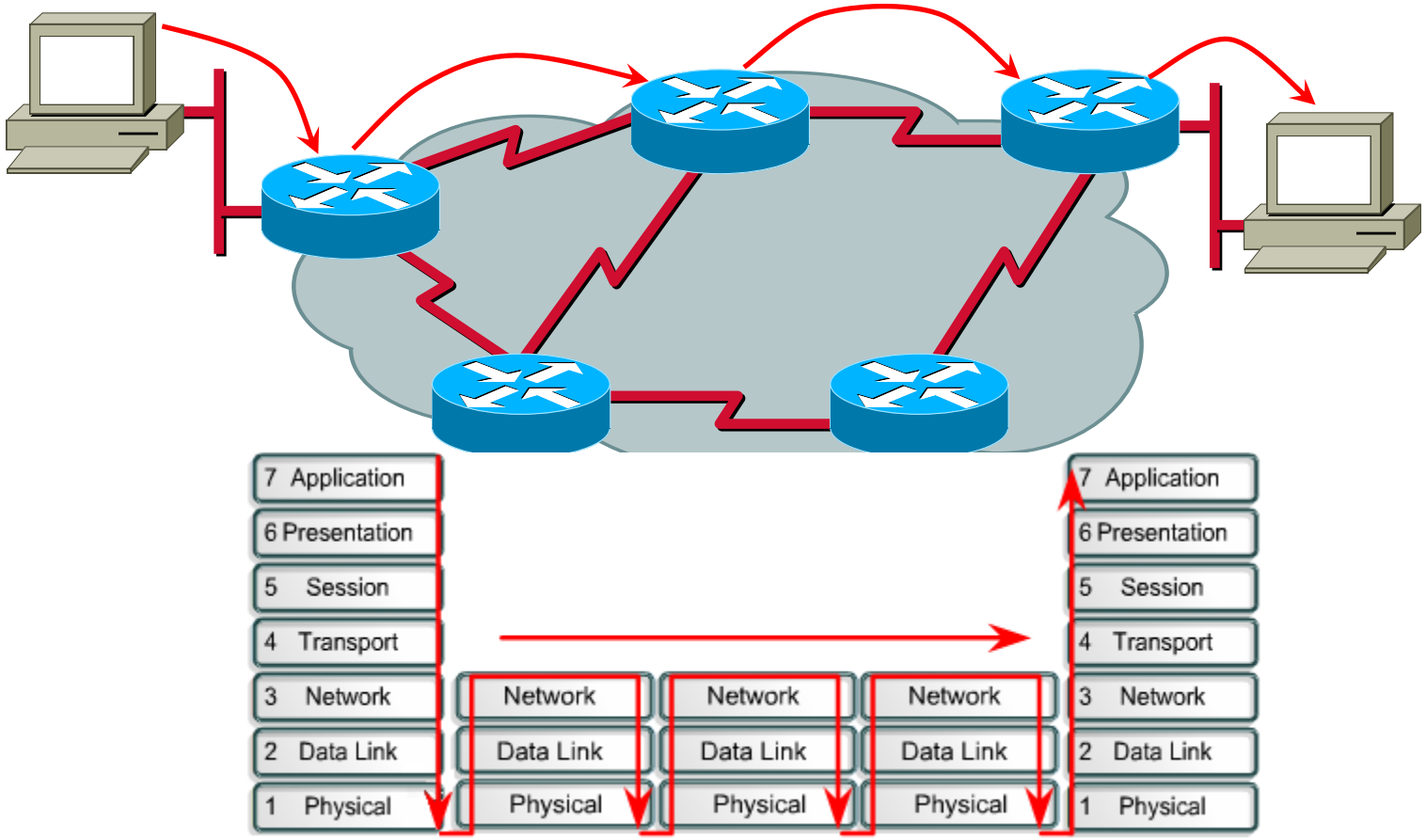


- Insert information into the packet's header to be properly addressed and routed
- Network layer defines
  - Logical network addressing
  - Packet formats
  - Logical network structure
- Common protocols
  - Internet Protocol (IP)
  - Internet Control Message Protocol (ICMP)
  - Routing Information Protocol (RIP)
  - Open Shortest Path First (OSPF)
  - Internet Group Management Protocol (IGMP)

Lower layers

- Data Link: convert data into LAN or WAN frames for transmission and define how a computer accesses a network
  - Logical topology
  - Framing format
  - Protocols: Address Resolution Protocol (ARP), reverse ARP, point to point protocol (PPP).
- Physical layer converts bits into voltage for transmission
  - Physical topology
  - Electrical signals
  - Signalling methodologies
  - Data rates

Routers and routing



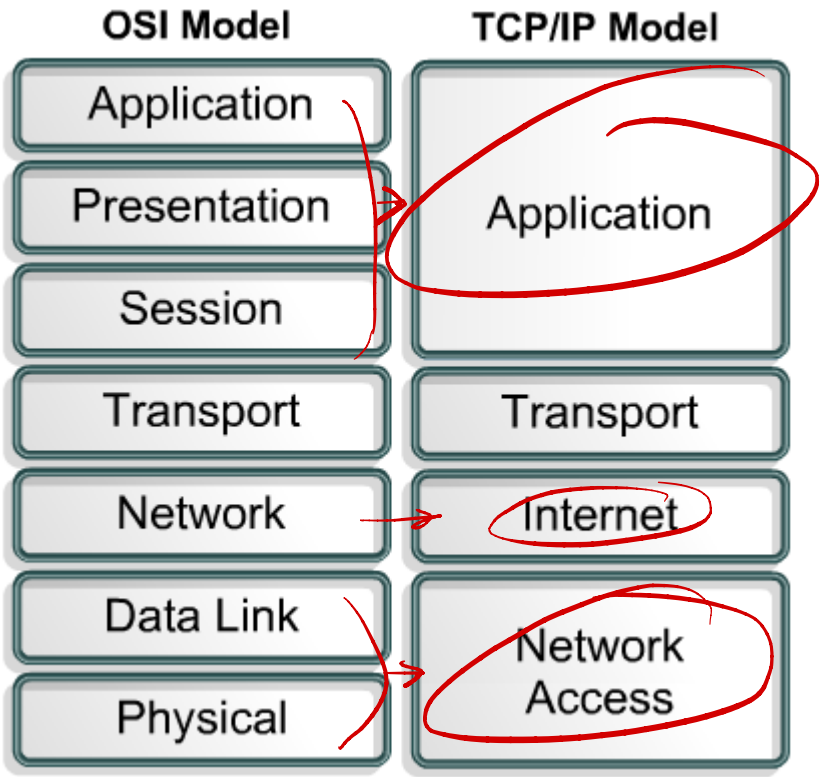
signals -> frames -> packets (router checking where packets go)

TCP/IP Model

- TCP/IP proposed in 1978
- Network and transport for ARPANet
  - Official protocol in 1981
- Design Goals
  - Hardware independence
  - Built in failure recovery
  - Fault tolerant
  - Efficient
  - Expandable
- TCP/IP became popular through distribution of Berkley Software Distribution (BSD) of UNIX.
- Open specification through the IETF and RFCs

TCP/IP layers

- Internet Layer
  - IP
  - ICMP
  - Routers and Routing
- Transport Layer
  - User Datagram Protocol
  - Transmission Control Protocol
  - Provides end to end transport services



Firewalls

- Packet Filtering
- Stateful
- Proxy
- Dynamic packet filtering

Packet filtering

- First generation
- Access Control Lists
- Based on network layer information
- Examine the header
- Weaknesses:
  - Cannot prevent attacks using the layers above, e.g. application,
  - Logging typically limited
  - Prone to improper configuration
- Example, permit udp host 10.1.1.3 host 172.16.1.2

Stateful

- Considered third generation
- Works at the transport and network layers
- Maintains a state table of communications channels → Keeps track of connections
- Checks if connection had been established first. If not, it uses its ACL
- Example: iptables -A INPUT -i eth0 -m state –state \ ESTABLISHED,RELATED -j ACCEPT

• skips checking through ACL → efficiency

Proxy firewalls

Special software needed that is compatible w/ the server being used

- Second generation firewalls
- Resides between a trusted and an untrusted network
- Application level:
  - Inspects the packet up to the application layer
  - Must understand the protocols
- Circuit level:
  - Creates a circuit at the session layer between the client and proxy server
  - Makes decisions based on source and destination.
- Degrades traffic performance → due to additional overhead

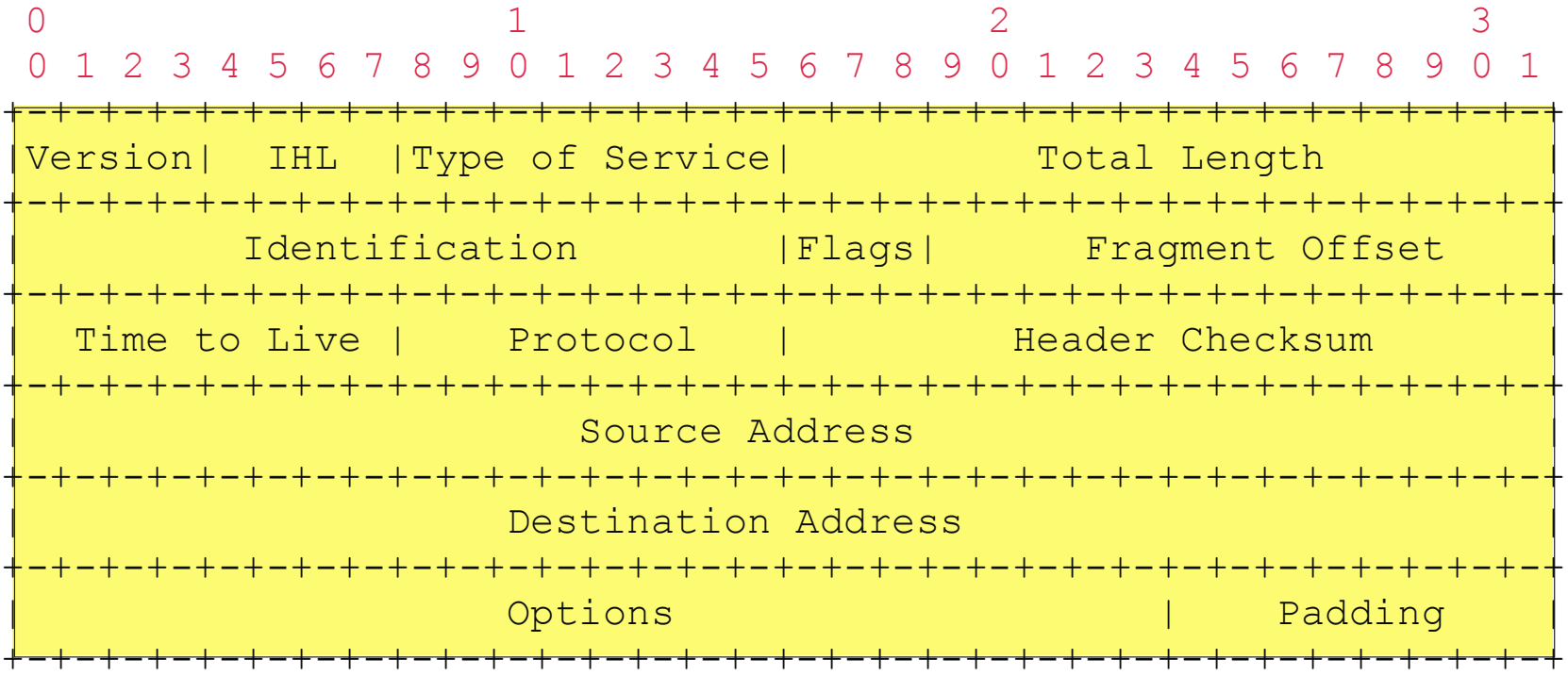
Dynamic packet filtering

an ACL created to connect local system (ports) to external services

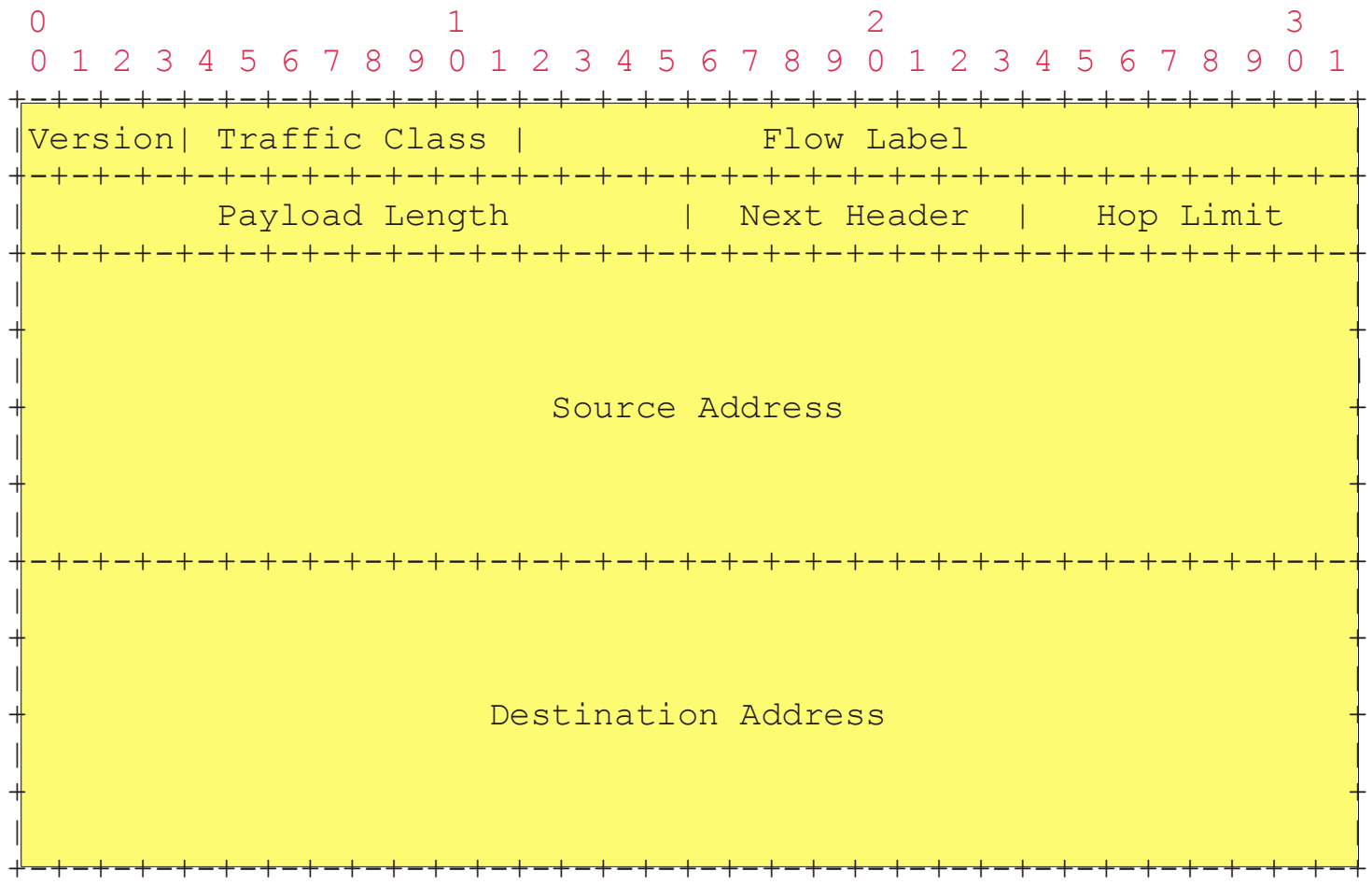
- Fourth-generation firewall
- Ports 0-1023 allocated to server side services
- Need for client to use higher ports
- How do we cope with the dynamic selection of ports?
- Allow any type of traffic outbounds and permitting only response traffic inbound

Network layer

IP Head to Head: v4



IP Head to Head: v6



Some of the differences

- IPv6 increases IP address size from 32 bits to 128 bits
- Some IPv4 headers have been dropped or made optional to reduce processing cost of packet handling and limit bandwidth
- New type of address called 'anycast address' is defined, which is used to send a packet to any one of a group of nodes
- New capability to enable the labelling of packets belonging to particular traffic flows – QoS or 'real-time' service
- Extensions to support authentication, data integrity, and (optional) data confidentiality Security

IPv6 and Security

- The good (almost):
  - IPSec was mandatory for all IPv6 stack implementation
    - Has been downgraded to a recommendation
- The bad:
  - Immature protocols = increased vulnerability
  - Unfamiliarity causes problems / misconfigurations
  - Automatic addressing may pose privacy concerns
  - IPv6 security controls lagging behind hacking arsenal / tools

IPSec

- Industry standard
- Main components
  - Security protocol
    - Authentication Header (AH)
    - Encapsulation Security Payload (ESP)
  - Security Association (SA)
  - Key management: Internet Key Exchange (IKE)
  - Algorithm

Authentication Header

AH protocol

Verify the identity of the sender

- AH can provide
  - Data integrity
  - Origin authentication
  - Anti-replay protection (optional)

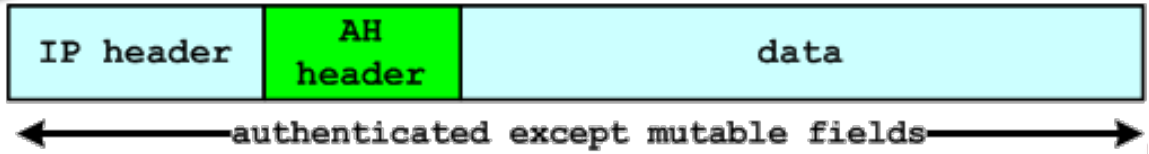
not encrypting anything

- AH does not provide
  - Data confidentiality
- Use for data integrity in cases where data is not secret but must be authenticated

prevents attackers from intercepting & retransmitting packets to gain unauthorised new valid access to systems

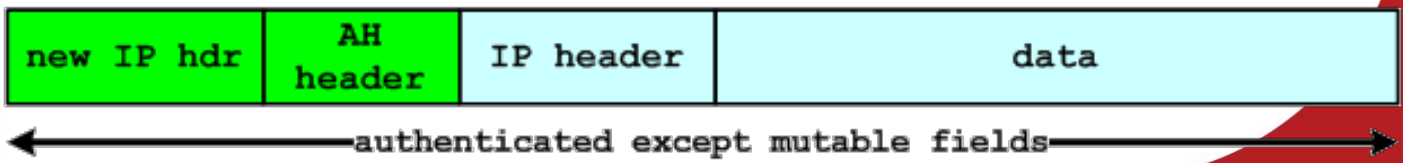
Provides integrity and data origin authentication only for the payload of an IP packet.

transport mode



Provides integrity and data origin authentication for the entire IP packet including the header.

tunnel mode



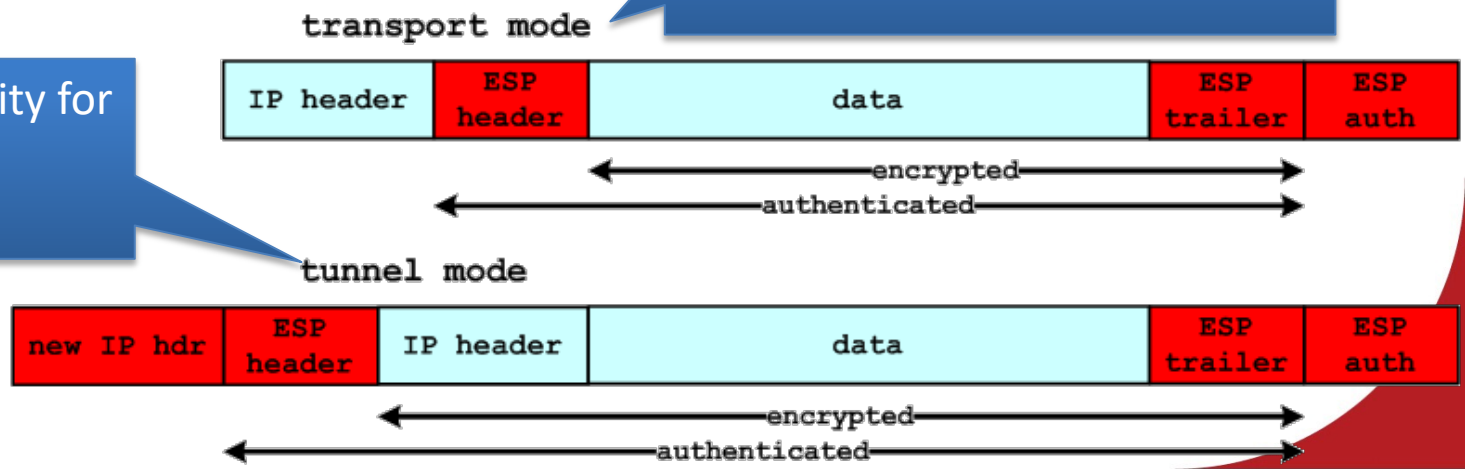
ESP protocol

- ESP can provide
  - Data confidentiality through encryption
  - Data integrity
  - Origin authentication
  - Anti-replay

- ESP does not protect the IP header → Router has to read & understand IP header
- Use when data must be kept secret.

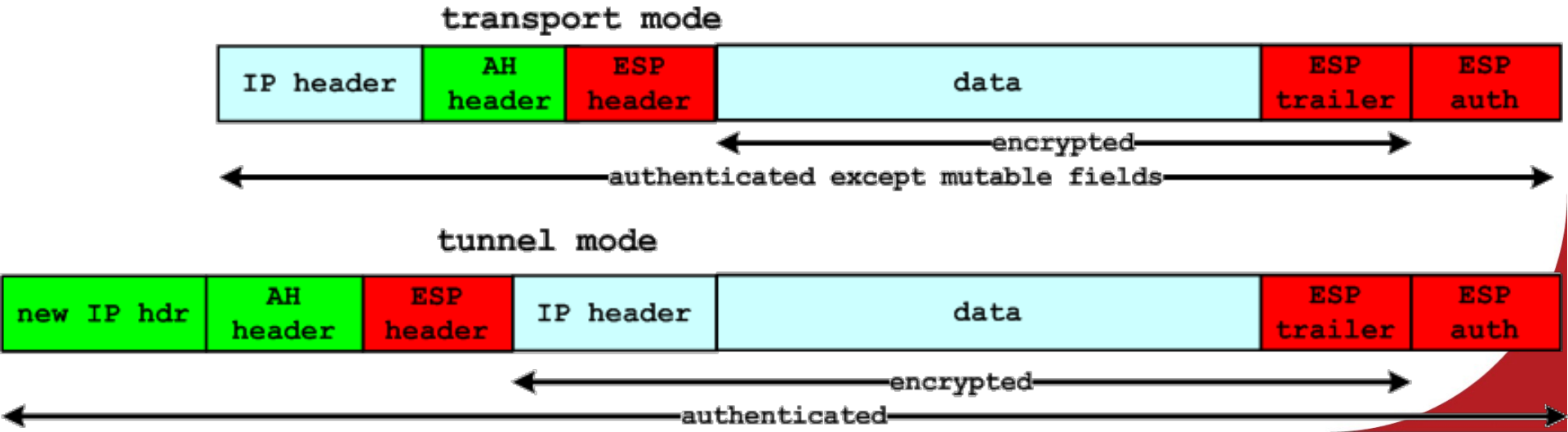
Provides confidentiality for the payload of an IP packet.

Provides confidentiality for the entire IP packet including the header.



AH and ESP

- Using both AH & ESP can provide protection for the IP header and encrypt data.
  - Rarely used because of overhead incurred by AH.
- Use for the highest security.



Traffic engineering

- Network engineering
  - Engineer the topology to fit the traffic
- Traffic engineering
  - Engineer the traffic to fit the topology
  - Given a fixed topology and a traffic matrix, what set of explicit routes offers the best overall network performance?
- Refers to the ability to control where traffic flows in a network, with the goal of reducing congestion and getting the most use out of the available facilities

Purpose of traffic engineering

- Network Performance
  - Maximize utilization of links and nodes throughout the network
  - Spread the network traffic across network links, minimize impact of single failure
  - Ensure available spare link capacity for re-routing traffic on failure
  - Deal with network design issues when the longer-range solution has not been deployed

Purpose of traffic engineering

- Service Levels
  - Offer premium services
  - Offer secure VPNs
  - Meet policy requirements imposed by the network operator

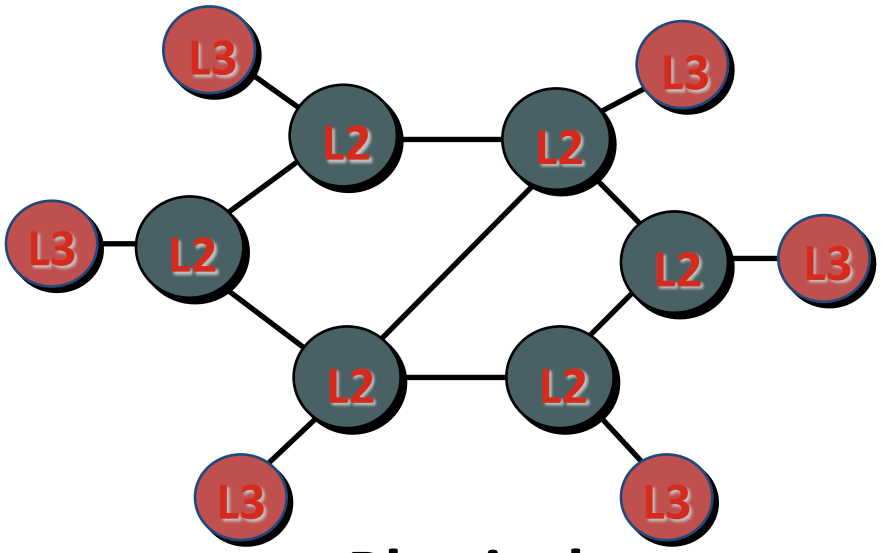
Current methods of traffic engineering

- Need mechanisms to explicitly manage the traffic inside a network
- Manipulating routing metrics → Attempting optimisation

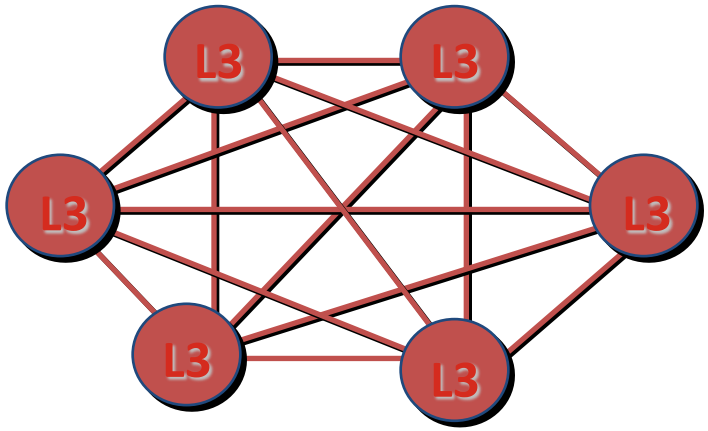
  → Difficult to manage
- Over-provision bandwidth

  → Not economical → Spending too much
- Overlay model?

The overlay solution



Physical



Logical

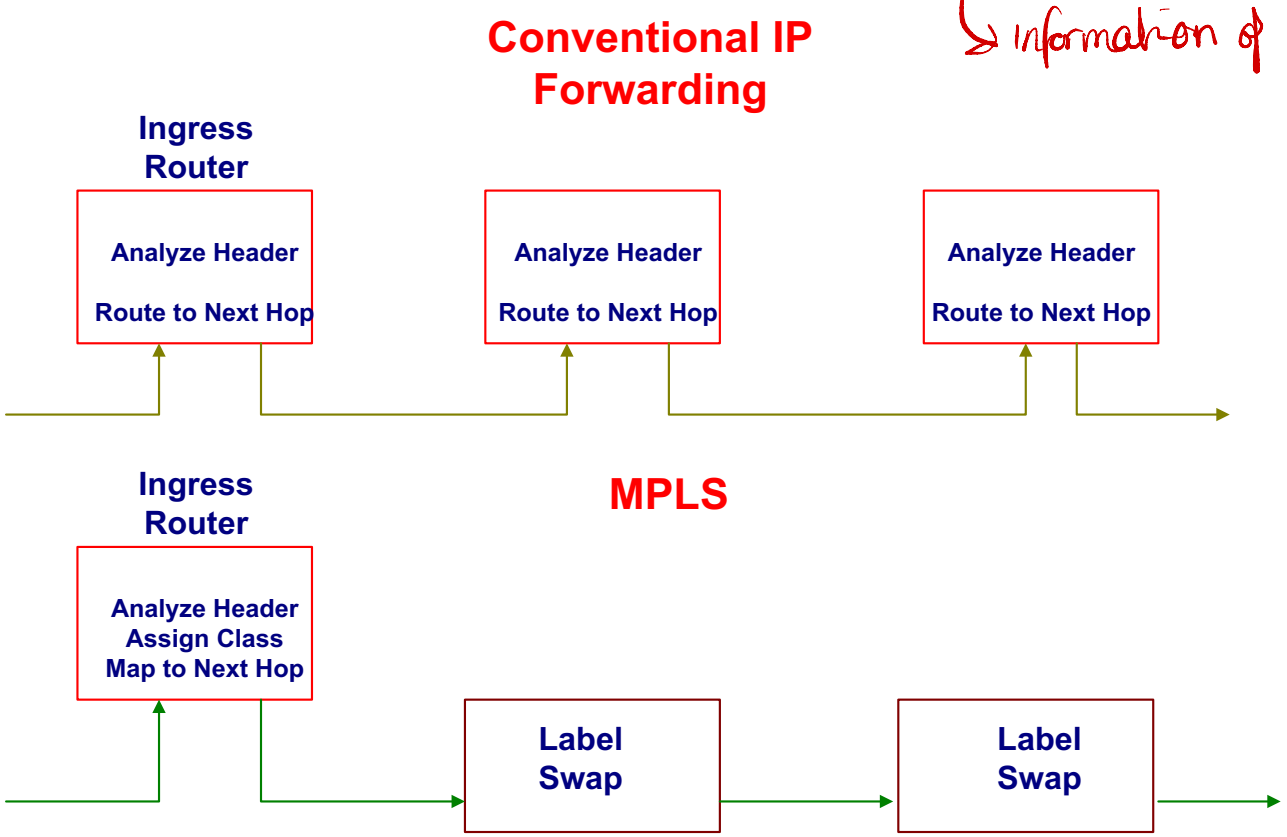
- Layer 2 network used to manage the bandwidth
- Layer 3 sees a complete mesh

Overlay drawbacks

- More complex network management
  - Two-level network without integrated management
  - Additional training, technical support, field engineering → Human factor
- *Is there a better approach to integrate layer 2 and layer 3?*

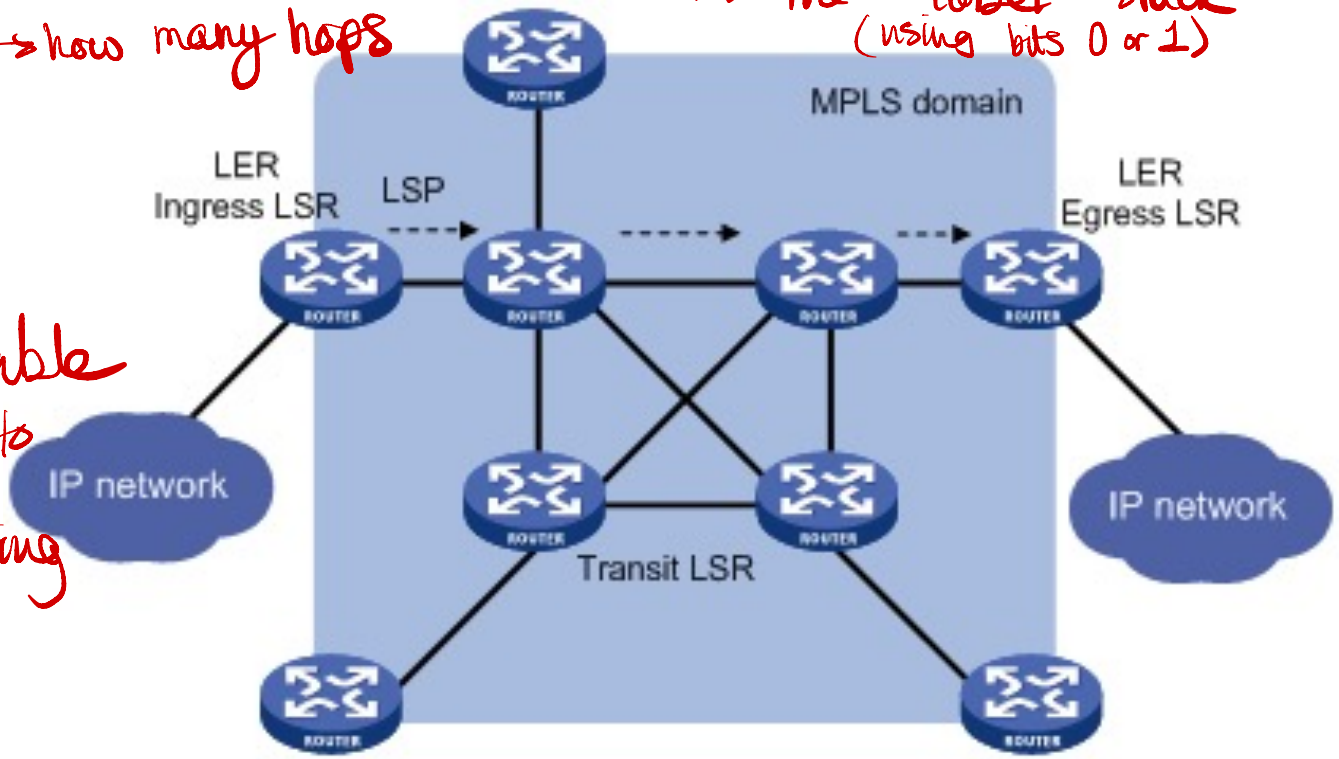
MPLS overview

- Multiprotocol Label Switching (MPLS)
- Packet forwarding method using labels (labels on packet) → information of next hop



MPLS overview

- Labels → Information of packets (where to go next)
- Experimental → Quality of service properties
- Bottom of stack → indicates whether a label is the last one in the label stack (using bits 0 or 1)
- Time to live → how many hops



Has a forwarding table mapping incoming labels to next-hop routers & outgoing interfaces

MPLS from the Perspective of an ISP

- MPLS is used by leading corporate ISPs worldwide

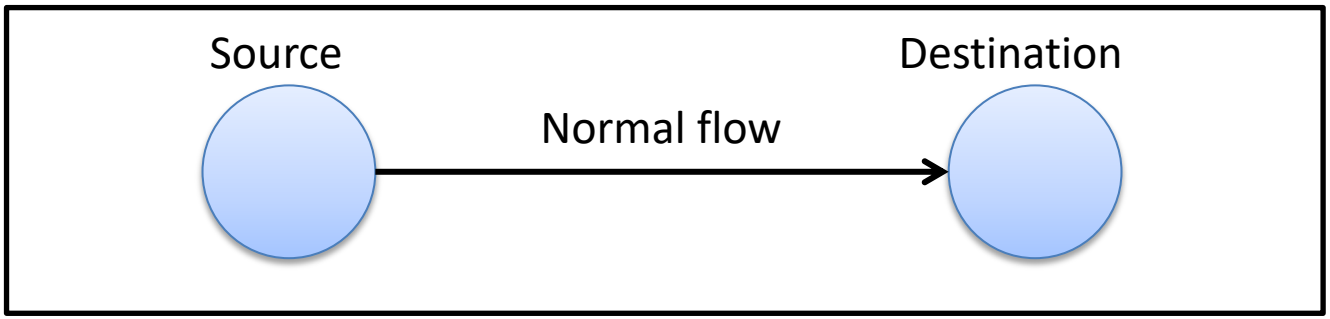
- Often used to:
  - Provide tunnels across a complex internal network → protection
  - Support traffic management / engineering
  - 'Hide' internal infrastructure → makes it hard for attackers to understand the infrastructure
  - MPLS tunnels can be configured on a 'per-customer' basis

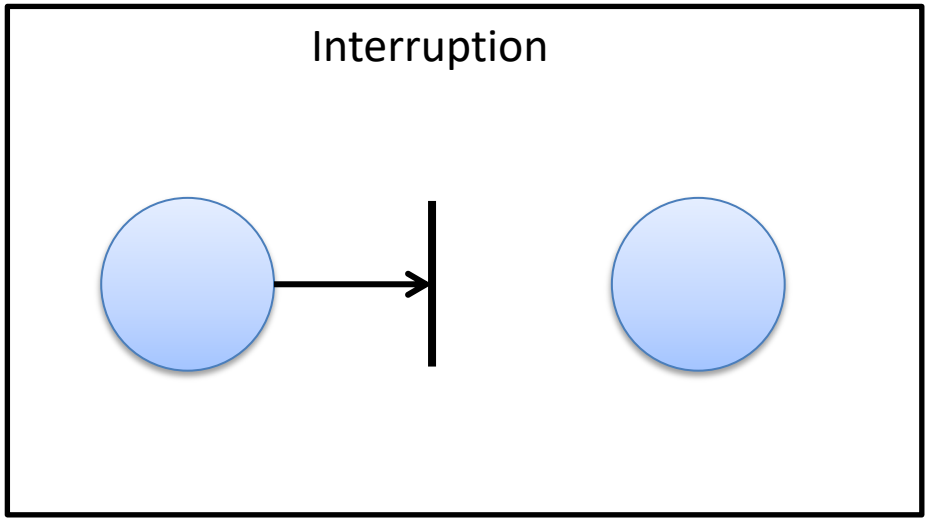
↓

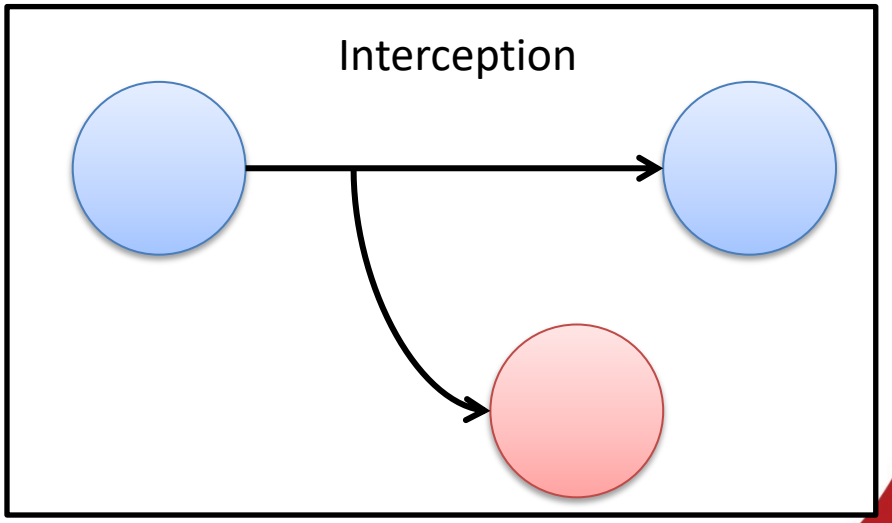
Business model

Network security threats

Type of network security threats

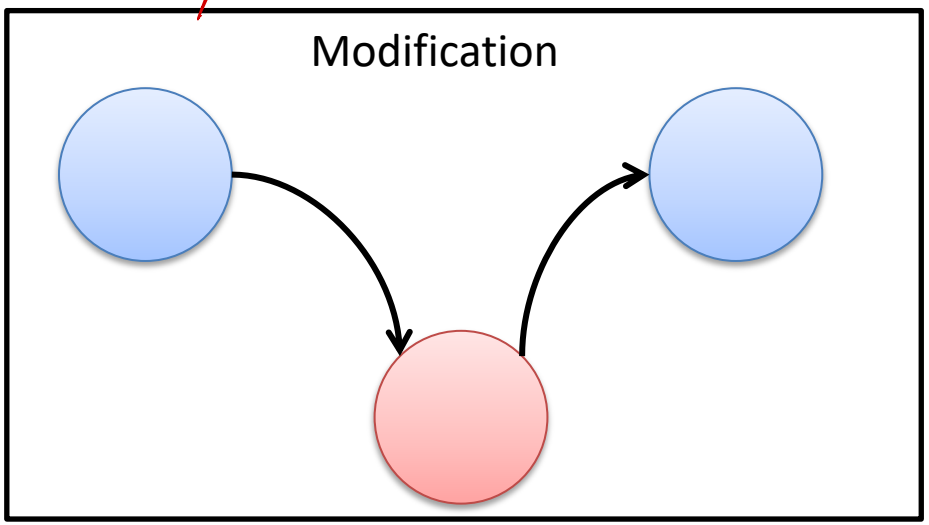




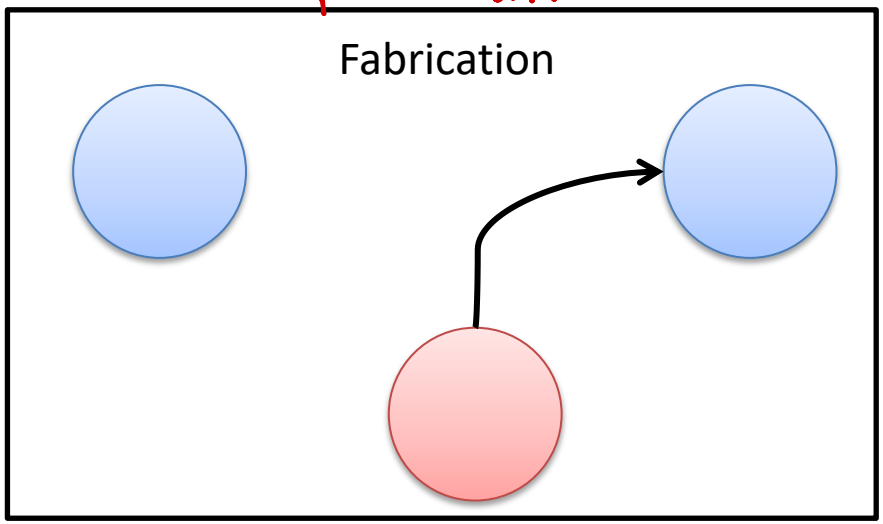


Type of (network) security threats

Man in the middle



Impersonation



Type of (network) security threats

Attack	On	Solution
Interruption	Availability	No effective solutions for interruption, denial of service (DoS) attacks
Interception	Confidentiality, Privacy	Encryption, decryption
Modification	Integrity	Attacks in integrity can be solved by applying digital signatures on every message.
Fabrication	Authenticity	Authentication

Questions?

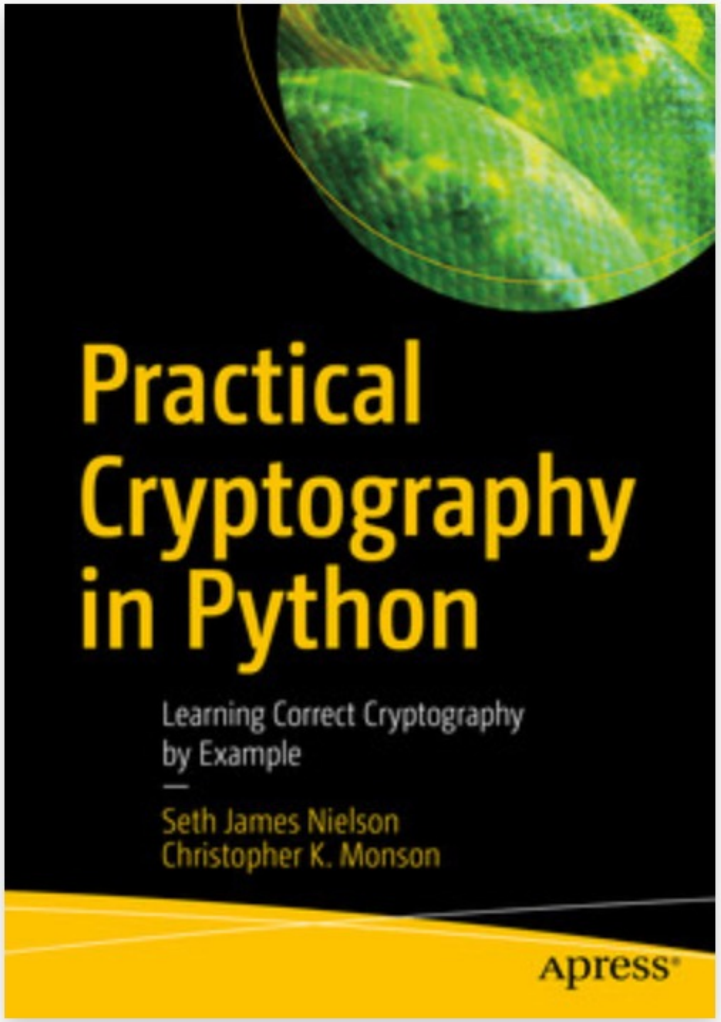
References

- CISSP All in One. Chapter 7 Telecommunications and Network Security.
- Internet Protocol RFC 791
- Transmission Control Protocol RFC 793

Systems Security Group | Lancaster University

Week 14 Asymmetric encryption

Recommended reading



The book is available to you via the library

Technology stack

- Python 3
  Link to a Python Cheat Sheet
- cryptography.io
  Link to the library

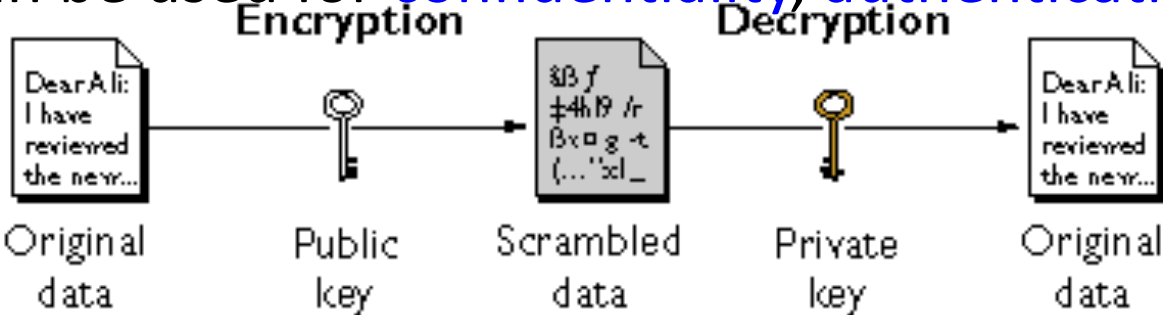
Topics

- Asymmetric encryption
- RSA
- Diffie-Hellman

Recommended reading: Chapter 4 from the book of "Practical Cryptography in Python"

Asymmetric cryptography

- A cryptographic scheme is called asymmetric if *d* <> *e* *and* it is computationally infeasible in practice to compute *d* out of *e*.
- In asymmetric cryptography *e* goes public and *d* is kept as a secret.
  - Anybody can use *e* to encrypt a plaintext and only the one that has *d* can decrypt it.
  - Public key cryptographic schemes.
  - Can be used for confidentiality, authentication or both



RSA#0

- Ron **R**ivest, Adi **S**hamir, Len **A**dleman (RSA)
- General-purpose approach to public-key encryption

RSA#1

- Plaintext is encrypted in blocks
  - Each block has a binary value less than *n*
  - Block size is *i* bits $2^i < n \leq 2^{i+1}$
- M is the plaintext and C the ciphertext

$$C = M^e mod\ n$$

$$M = C^d mod\ n = (M^e)^d mod\ n = M^{ed} mod\ n$$

RSA#3

- In reality e = 2^16+1 = 65537

- Order of generating keys
  - First create a private key object
  - Than get a public key object corresponding to the values of the private key

https://cryptography.io/en/latest/hazmat/primitives/asymmetric/rsa/

Serialization of keys

- Encode keys using PEM: Privacy Enhanced Mail; it simply indicates a base64 encoding with header and footer lines
- Set format for private key to PKCS8: A more modern format for serializing keys which allows for better encryption. Choose this unless you have explicit legacy compatibility requirements.
- Set format for public key to SubjectPublicKeyInfo: This is the typical public key format. It consists of an algorithm identifier and the public key as a bit string. Choose this unless you have specific needs.
- Use or not an encryption algorithm

Converting key objects to/from PEM

Key object to PEM

```
.private_bytes(select encoding, etc)
.public_bytes(select encoding, etc)
```

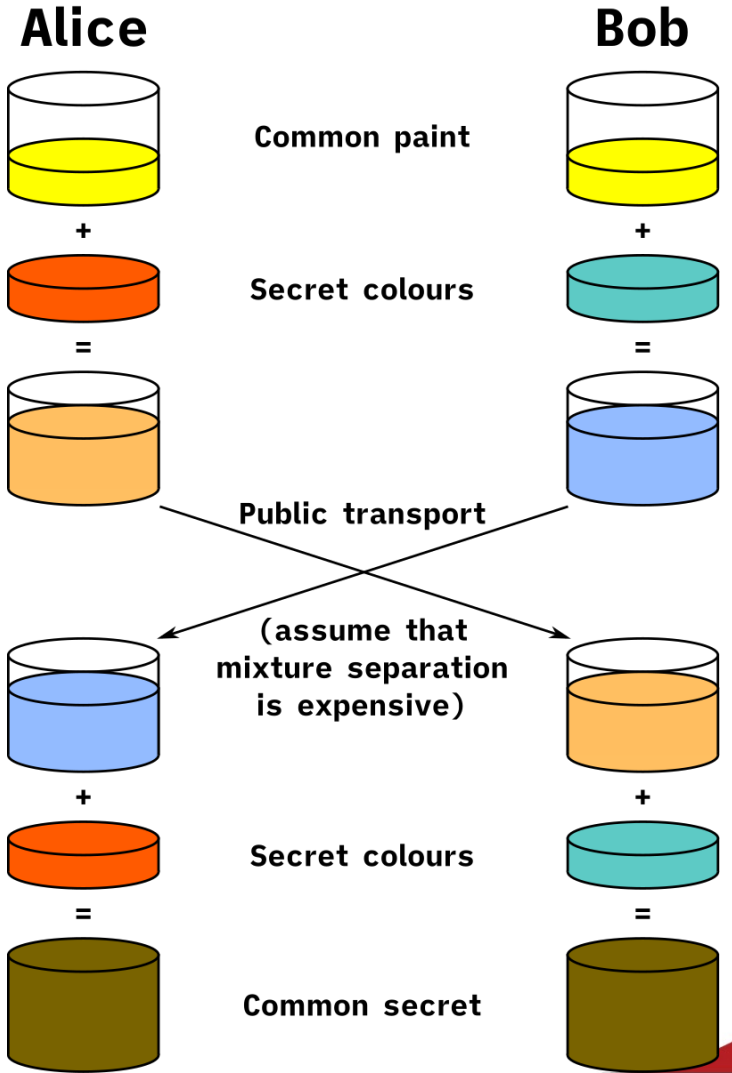
PEM to key object

```
.serialization.load_pem_private_key
.serialization.load_pem_public_key
```

Diffie-Hellman key exchange

- Used to securely exchange a key
- Based on discrete logarithms

Known difficult mathematical problem
- Involves trying to find the exponent that solves a mathematical equation
- hard for larger exponents



[source: 3]

Diffie-Hellman

Two parties choose secret numbers known only to themselves

- Alice selects: Must be prime
  - $X_A < q$ and computes $Y_A = \alpha^{X_A} \, mod \, q$
- Bob selects:
  - $X_B < q$ and computes $Y_B = \alpha^{X_B} \, mod \, q$
- X values are kept private and Y are sent away.
- Alice computes the key:
  - $K = (Y_B)^{X_A} \, mod \, q$
- Bod computes the key:
  - $K = (Y_A)^{X_B} \, mod \, q$

Diffie-Hellman

- The calculated key is the same:

$$K = (Y_B)^{X_A} mod\ q$$

$$= (\alpha^{X_B} mod\ q)^{X_A} mod\ q$$
$$= (\alpha^{X_B})^{X_A} mod\ q$$
$$= (\alpha^{X_A})^{X_B} mod\ q$$
$$= (\alpha^{X_A}\ mod\ q)^{X_B} mod\ q$$
$$= (Y_A)^{X_B} mod\ q$$

Structure of your code…

Modules you want to import

```
import XYZ
```

List of functions you implement

```
def  myFunction():
     # TODO

     return # TODO
```

Have a main section to call your functions

```
if __name__ == "__main__":
     x = myFunction()
```