Network Security IV

CSE 565: Fall 2024
Computer Security

Xiangyu Guo (xiangyug@buffalo.edu)

University at Buffalo

Disclaimer

- We don’t claim any originality of the slides. The content is developed heavily based on
  - Slides from Prof. Dan Boneh and Prof. Zakir Durumeric’s lecture on Computer Security (https://cs155.stanford.edu/syllabus.html)
  - Slides from Prof Nick McKeown’s lecture on Computer Network (https://vixbob.github.io/cs144-web-page/)
  - Slides from Prof Ziming Zhao’s past offering of CSE565 (https://zzm7000.github.io/teaching/2023springcse410565/index.html)
  - Slides from Prof Hongxin Hu’s past offering of CSE565

Announcement

- HW3 and Project3 **due Tue, Nov 12, 23:59 pm**.

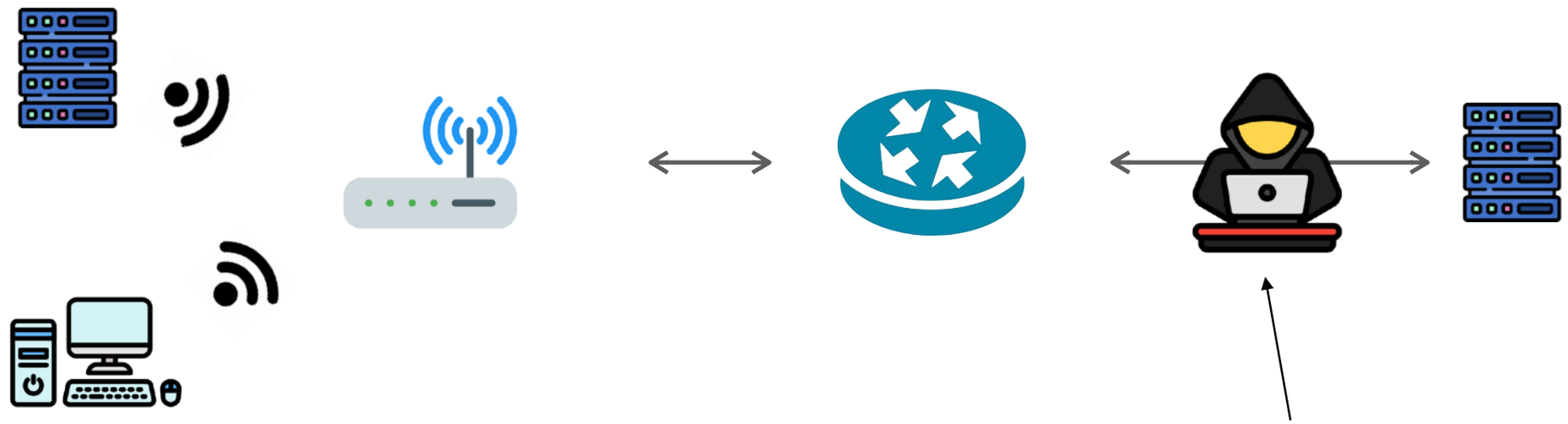
Review of Last Lecture

- Denial-of-Service Attack: overwhelming the victim with huge network traffic
  - DoS via Amplification
    - Exploits asymmetries in network protocols: DNS, NTP
  - DoS via Flooding (“DDoS”)
    - Generating traffic from many network devices (“botnet”)
    - Getting more and more powerful due to the huge number of IoT devices today
    - Example: Mirai
- DoS Defense

Today's Topic

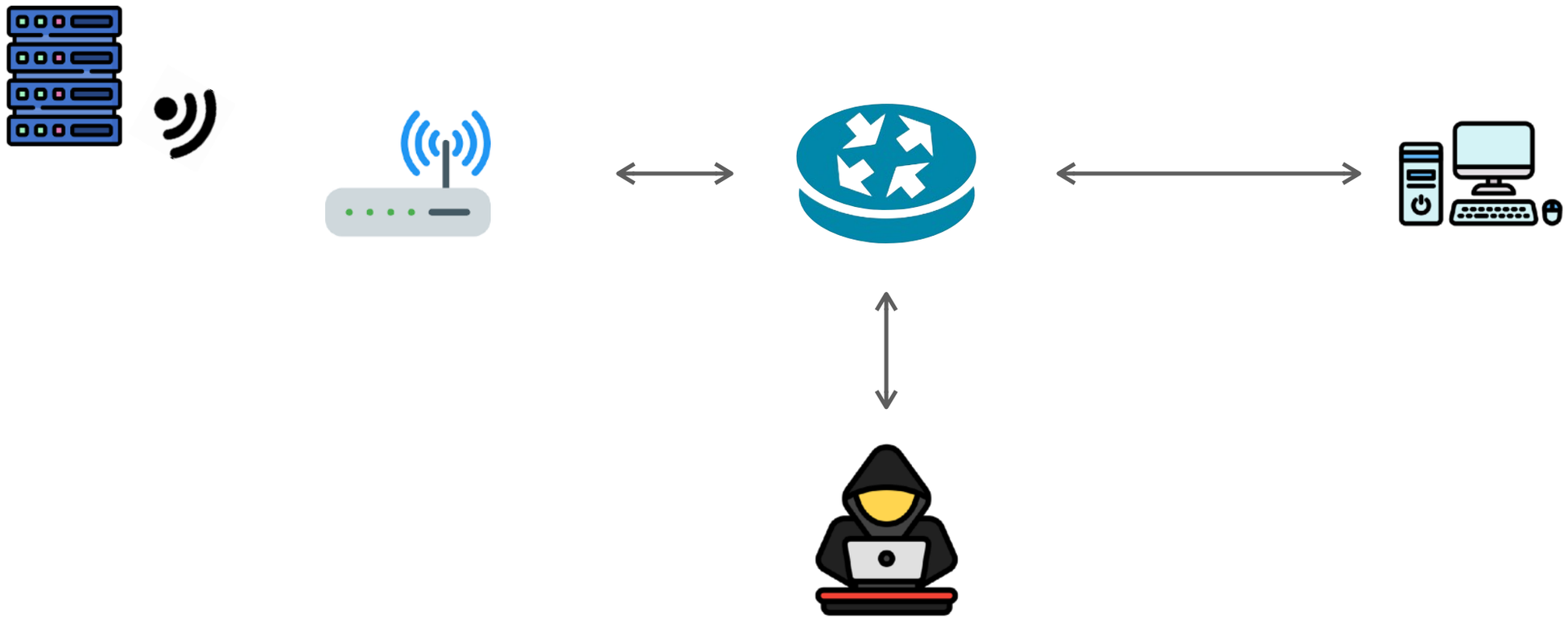
- Network Defense
  - IPSec
  - VPN
  - TLS: Transport Layer Security
    - QUIC: TLS + (multiplexed) UDP
  - Firewall

Notation: On Path Attacker



Attacker has access to read, manipulate, and drop traffic because they are on the path that the traffic takes across the Internet

Notation: Off Path Attacker



Attacker can inject traffic (including from fake source addresses), but can't read/modify traffic

No security guarantees

- **Confidentiality —** Ethernet, IP, UDP, and TCP do not provide any confidentiality. All traffic is in cleartext.
    - On-path attacker can do anything. ARP and BGP attacks allow an off-path attacker to become on-path and MITM connections.
- **Integrity —** No guarantees that attacker hasn't modified traffic. Ethernet, IP and UDP have no protection against spoofed packets. TCP provides *weak* guarantee of source authentication against off-path attacker
- **Availability —** Attackers can attempt to inject packets or launch "denial of service" attacks against services

Assume network is malicious

The network is out to get you.

Solution: Always use TLS if you want any protection against large-scale eavesdropping or guarantee that data hasn't been modified or corrupted by an on-path (or off-path since less strong) attacker

Note! HTTPS and TLS aren't just for sensitive material! There have been attacks where malicious Javascript or malware is injected into websites.

Building a network protocol

- **Don’t build network proto from scratch**
  - Never roll your own crypto
  - Many opportunities to mess up parsing network packets
- **gRPC**: HTTP/2 + TLS 1.3 RPC framework
  - Safe parsing in 11 languages
  - Exceptionally efficient
  - Streaming/Sync/Async
  - TLS-based authentication
- Or, **REST** on top of HTTP/2 + TLS 1.3

```
syntax = "proto3";

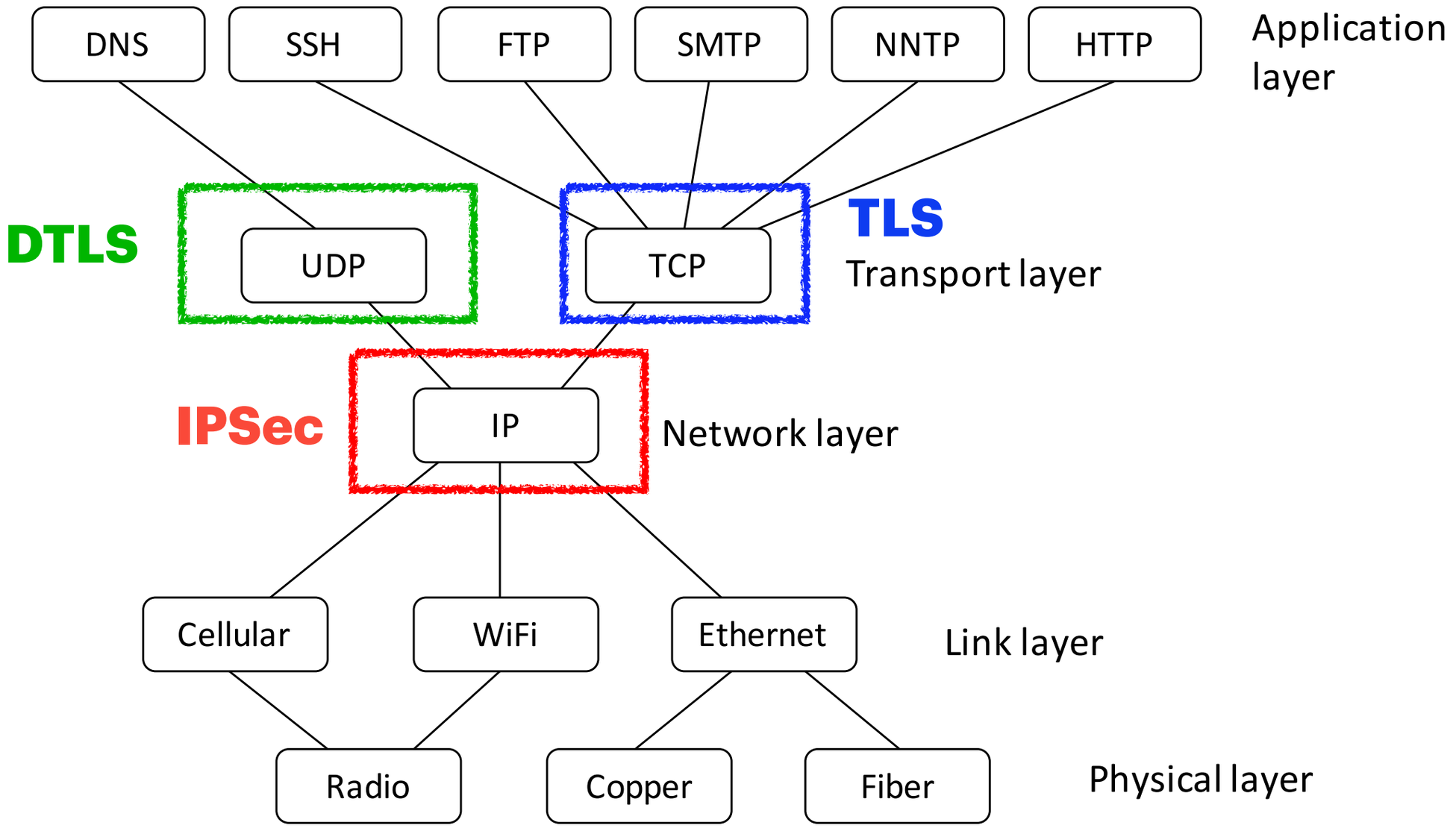
package calc;

message AddRequest {
  int32 n1 = 1;
  int32 n2 = 2;
}

message AddReply{
  int64 res = 1;
}

service Calculator {
  rpc Add(AddRequest)       returns
(AddReply) {}
  rpc Substract(SubRequest) returns
(SubReply) {}
  rpc Multiply(MultRequest) returns
(MultReply) {}
  rpc Divide(DivideRequest) returns
(DivideReply) {}
}
```

Where to put the cryptos?



Internet Protocol Security (IPSec)

IP is insecure

- It's *the* most important: the only Network Layer protocol
- Yet tragically *insecure*
  - No data integrity or confidentiality
    - No encryption to protect payload (TCP, UDP, User data)
  - Source spoofing
    - No host authentication
    - Leads to all sorts of spoofing attacks (for protocols above Lv 3)

Internet Protocol Security (IPSec)

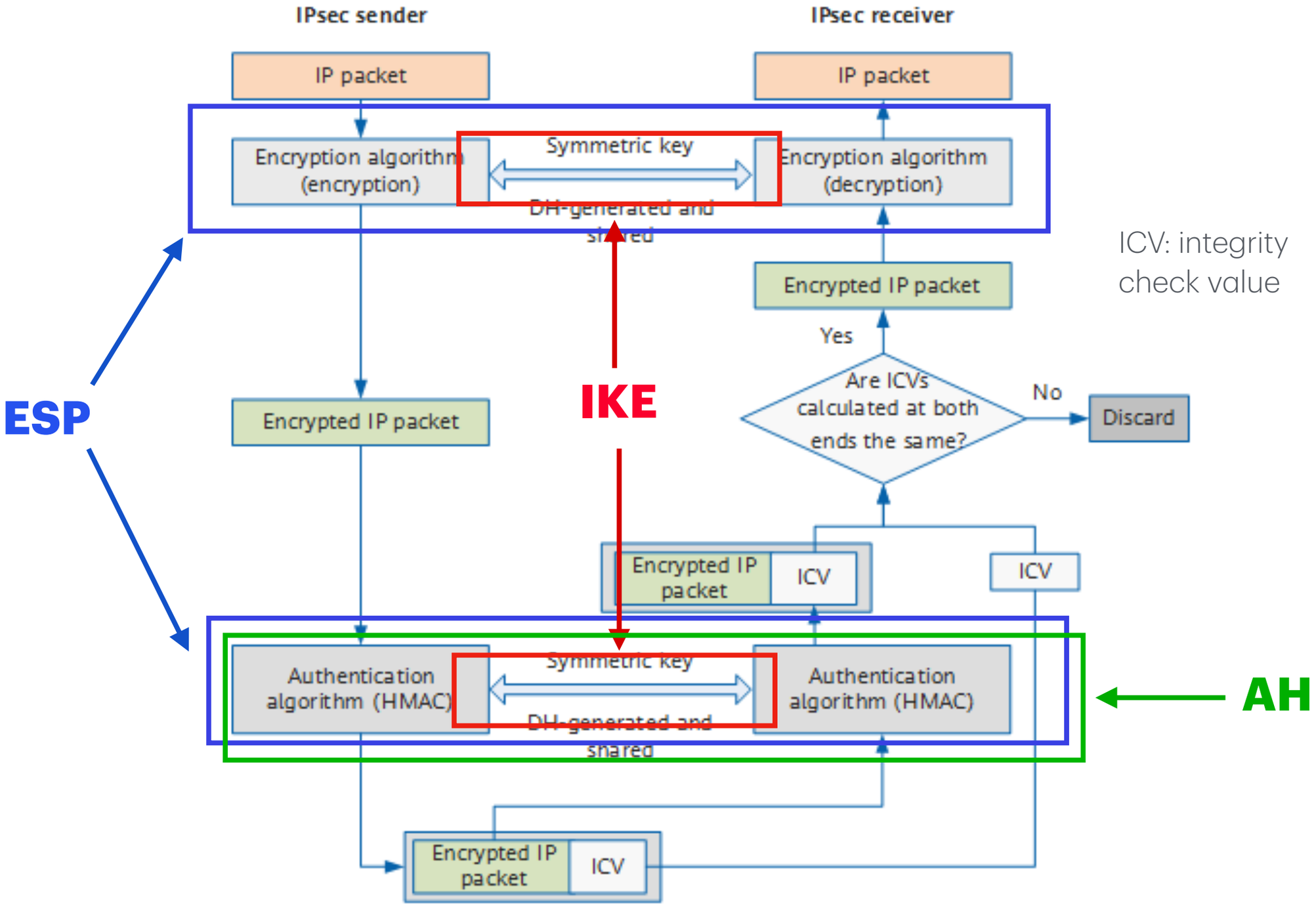
- **IPSec** is a set of protocols for Network Layer security
  - Below Transport Layer, hence transparent to applications
  - Can be transparent to end users
- Protects integrity and/or confidentiality of packets
- Authenticates sources of IP packets
- Applicable to use over LANs, across public & private WANs, and for the Internet
- Mandatory in IPv6, optional in IPv4

IPSec Main Components

- **Authentication Header (AH)** Protocol
  - Authenticates the whole IP packet, including the header
- **Encapsulating Security Payload (ESP)** Protocol
  - Encrypts the IP packet *payload*
  - Authenticates the (encrypted) payload (But not the header)
- **Internet Key Exchange (IKE)** Protocol
  - Session establishment: crypto algs to use, enc options, key exchange, etc.

IPSec Main Components

A simplified view



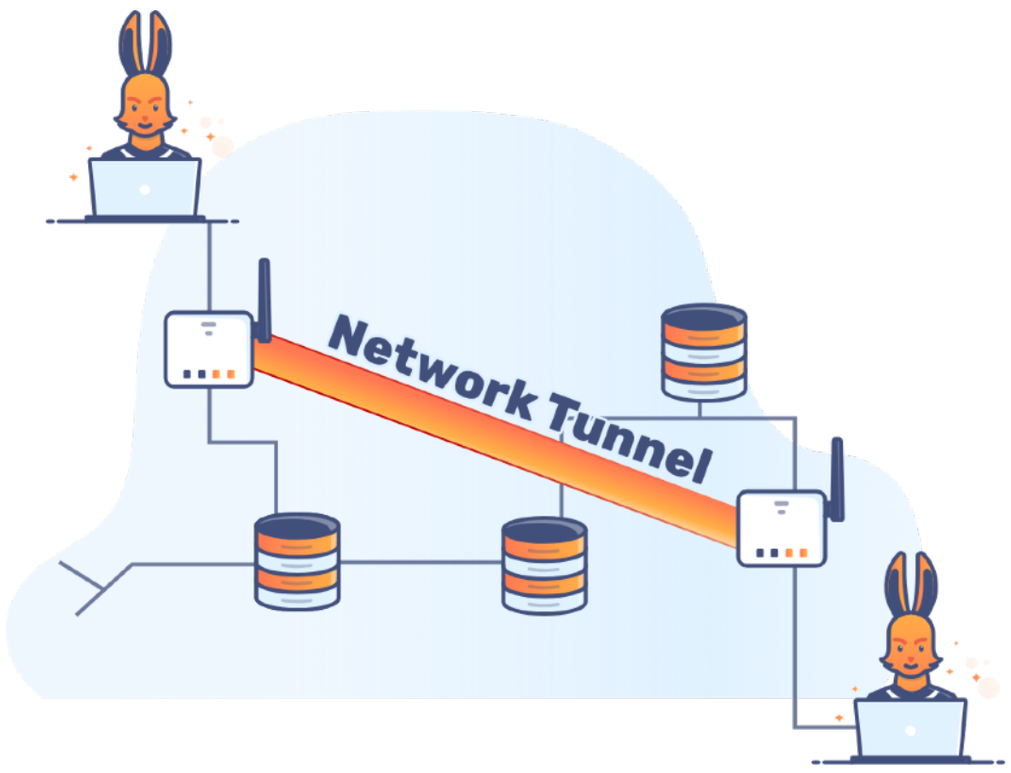
ICV: integrity check value

IPSec Modes

- Two modes for different scenarios
  - **Transport Mode**: for internal communications within a *secure* network
  - **Tunnel Mode**: securing traffic over *untrusted* networks, like the internet

Network Tunneling

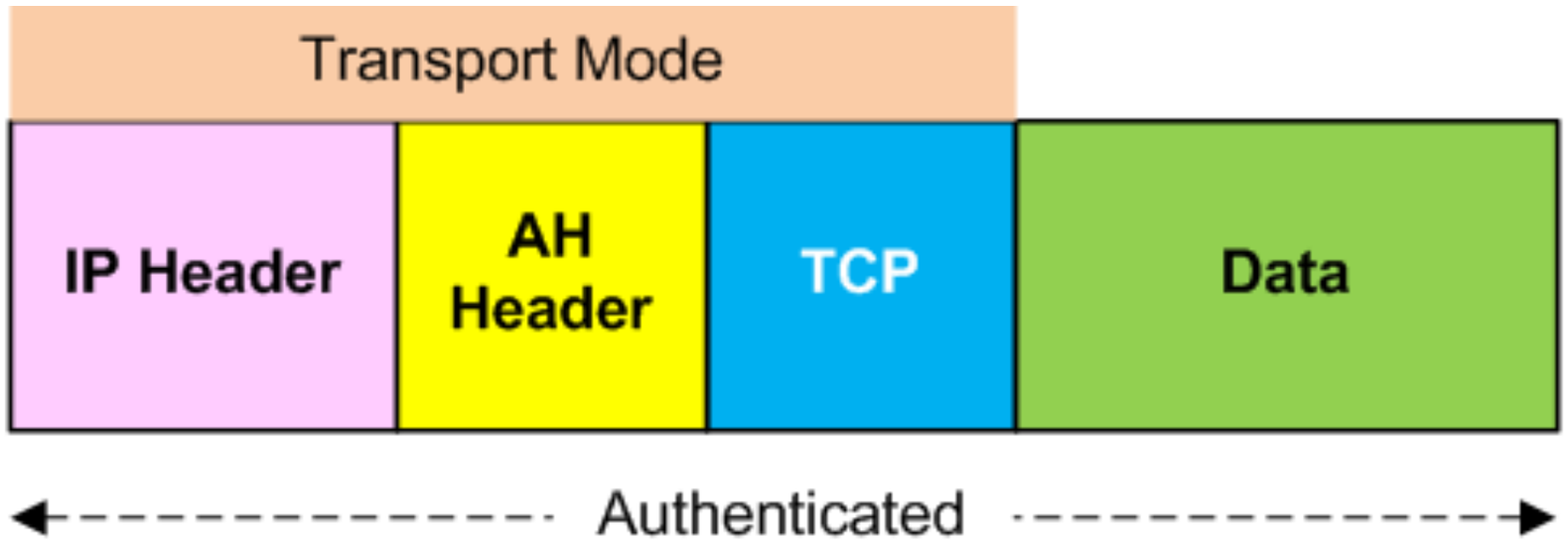
- Packets from the original protocol (the “payload”) are encapsulated within packets of a different protocol (the “carrier” or “transport” protocol).
    - Hides the payload protocol from intermediate devices on the network (e.g., routers or firewalls) that don’t need to understand it.
- Transports data across a network using protocols that are not supported by that network

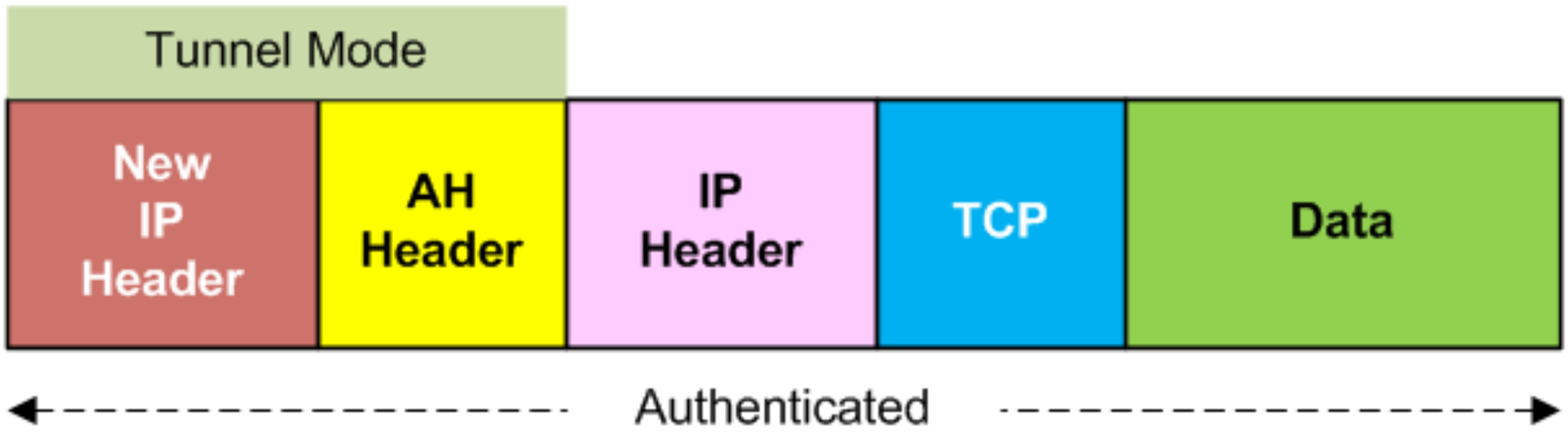


IPSec Modes

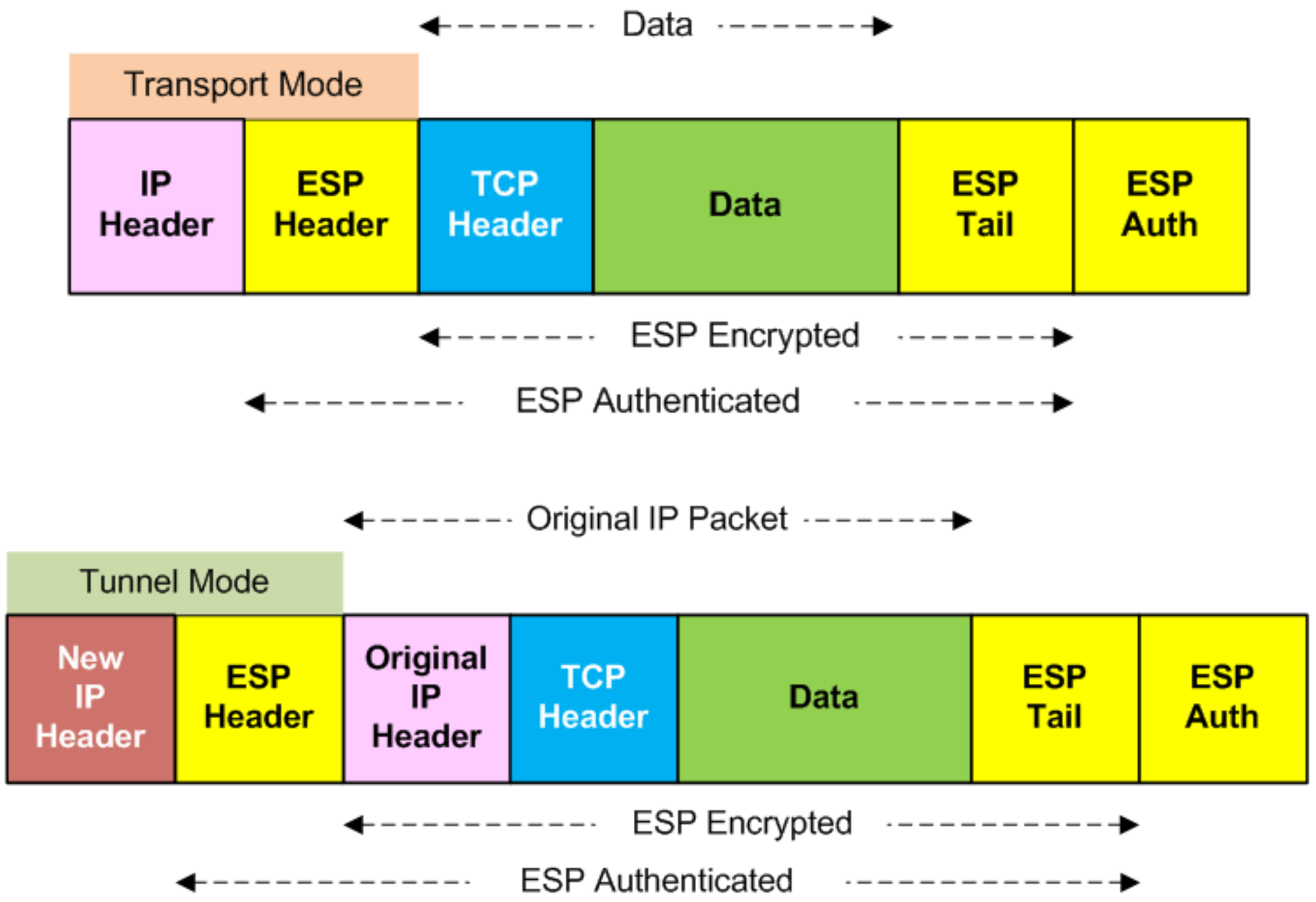
Feature	Transport Mode	Tunnel Mode
What is Encrypted	Only the IP payload (data)	Entire IP packet (header + payload)
Original IP Header	Visible to network	Hidden (encapsulated in new IP header)
Primary Use Case	Host-to-host communication in trusted networks	Site-to-site and remote-access VPNs across untrusted public networks
Security Level	Moderate (payload only)	High (entire packet, including IP header)

Authentication Header (AH)





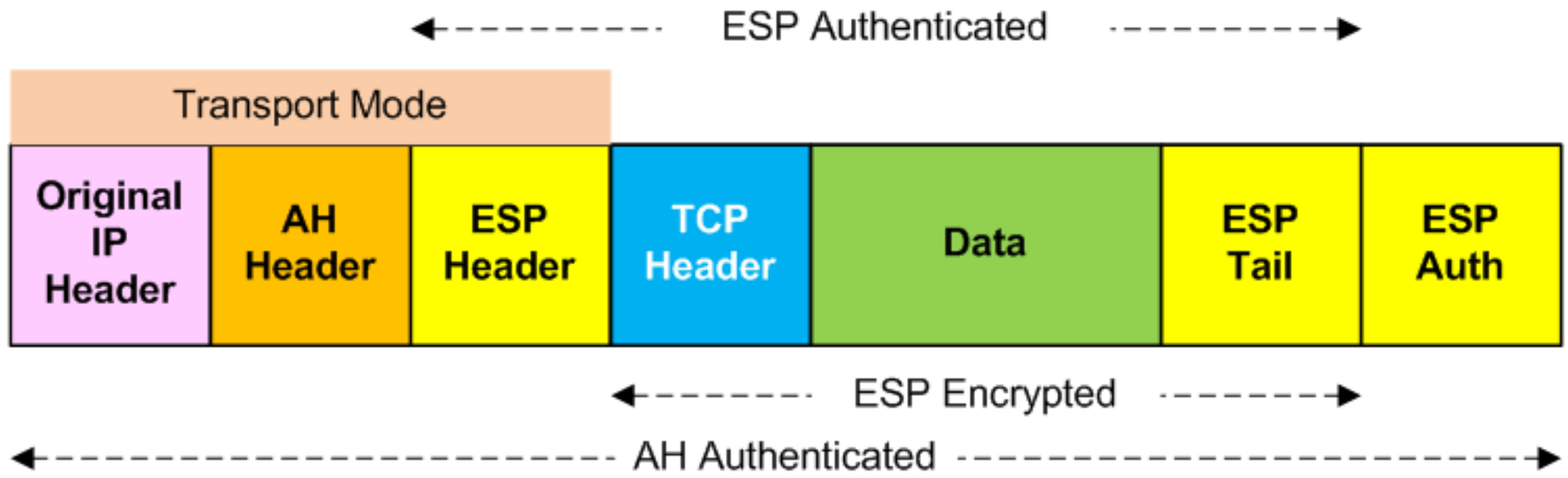
Encapsulating Security Payload (ESP)

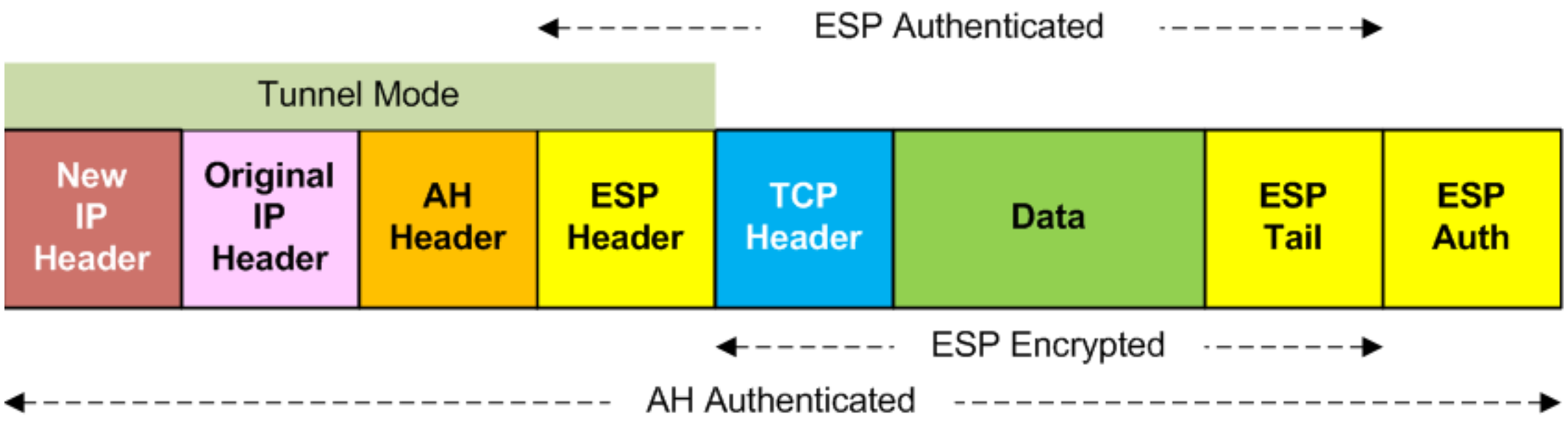


AH v.s. ESP

- AH provides **I**ntegrity, but no **C**onfidentiality.
- ESP provides **C**onfidentiality (and optionally **I**ntegrity), but only to the payload.
- AH and ESP can be combined freely.
- If you require NAT-Traversal, use ESP in Transport mode and do not use AH.
  - NAT will change IP Header, which causes AH integrity check to fail
    - If you really want both, use UDP Encapsulation (won't discuss here)

AH v.s. ESP


ESP Authenticated
Transport Mode
Original IP Header
AH Header
ESP Header
TCP Header
Data
ESP Tail
ESP Auth
ESP Encrypted
AH Authenticated


ESP Authenticated
Tunnel Mode
New IP Header
Original IP Header
AH Header
ESP Header
TCP Header
Data
ESP Tail
ESP Auth
ESP Encrypted
AH Authenticated

Internet Key Exchange (IKE)

- Includes 3 key-exchange protocols:Oakley, SKEME, ISAKMP
- Negotiate IPSec options
  - ESP and/or AH?
  - Encryption alg?
  - MAC alg?
  - Diffie-Hellman Key Exchange?
  - ...

Virtual Private Network (VPN)

Alice is Traveling

- Alice works for the Mergers and Acquisitions (M&A) department of **`abc.com`**

- She is on a business trip taking over a plant

- She wants to access the M&A server and other servers at her company (confidentially of course)

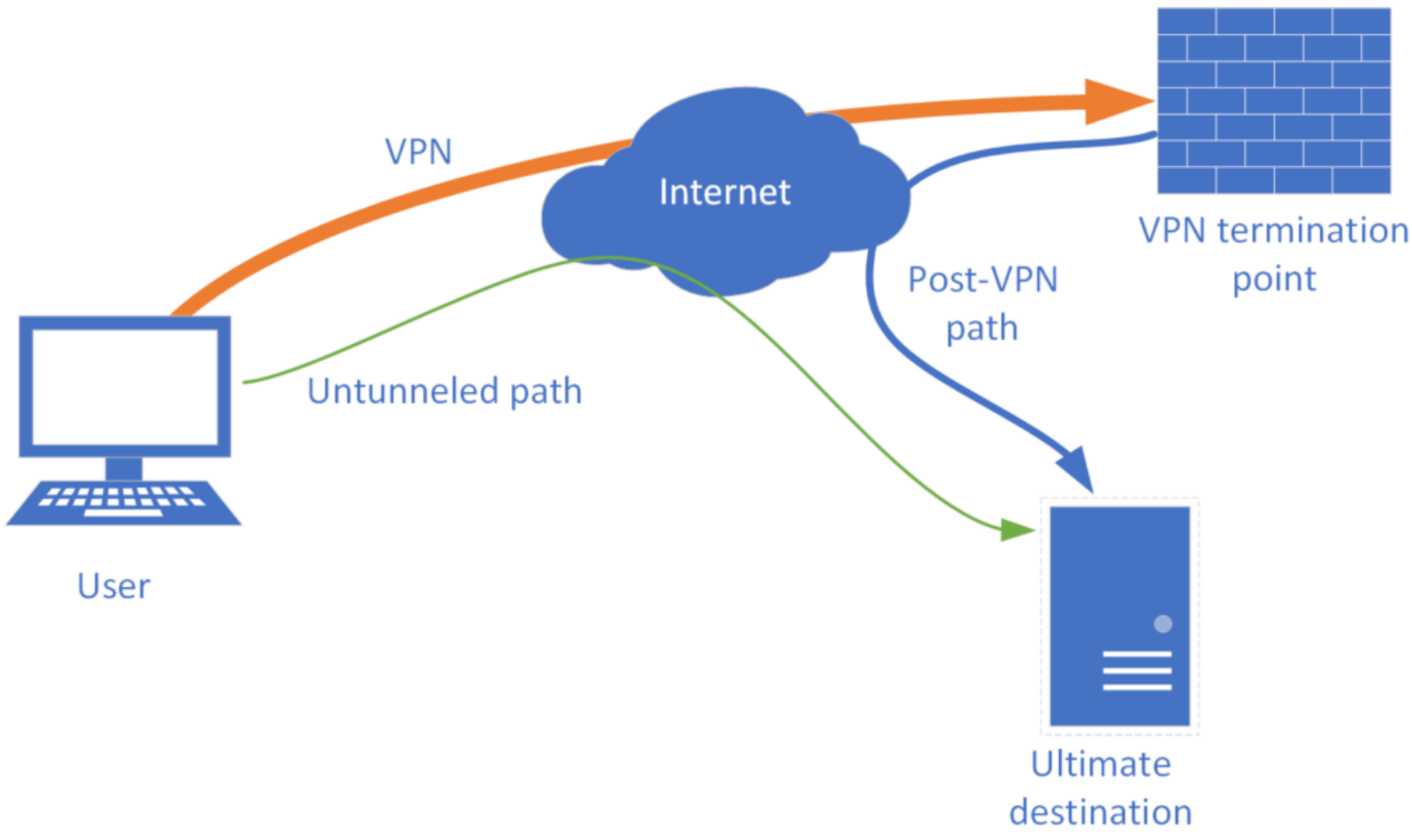
- **Problem**: How do you provide secure communication for protocols across the public Internet?

Virtual Private Network

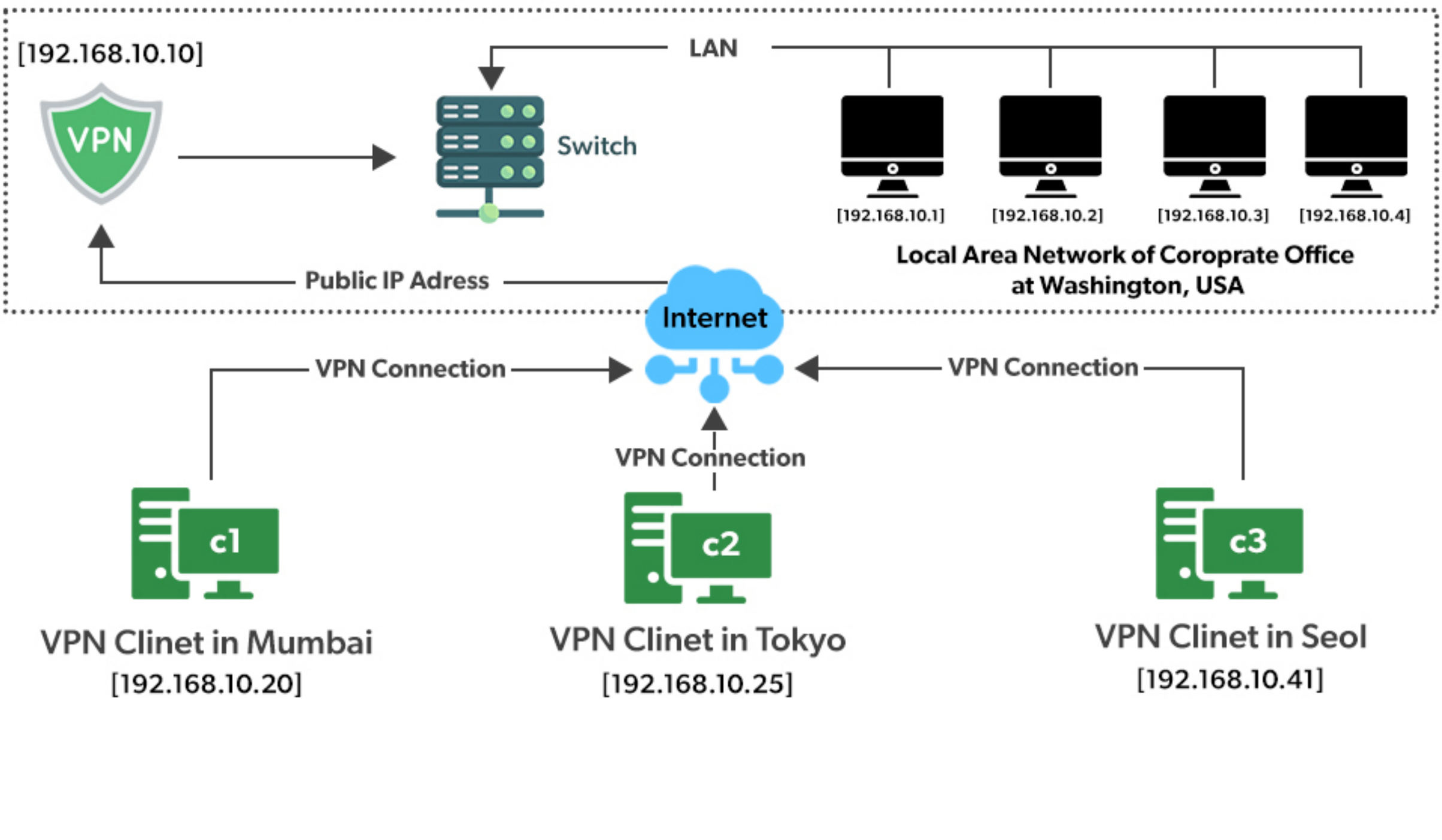
- VPNs create a fake shared private network
  - **Virtual**: It is not a physically distinct network
  - **Private**: Traffics are encrypted to provide confidentiality
- Two main purpose:
  - **Security**: Remote client (e.g., traveling Alice with laptop) to corporate network
  - **Privacy (Anonymity)**: Users try to hides their browsing activity, identity, location, etc.

VPN ≈ Secure Tunnel

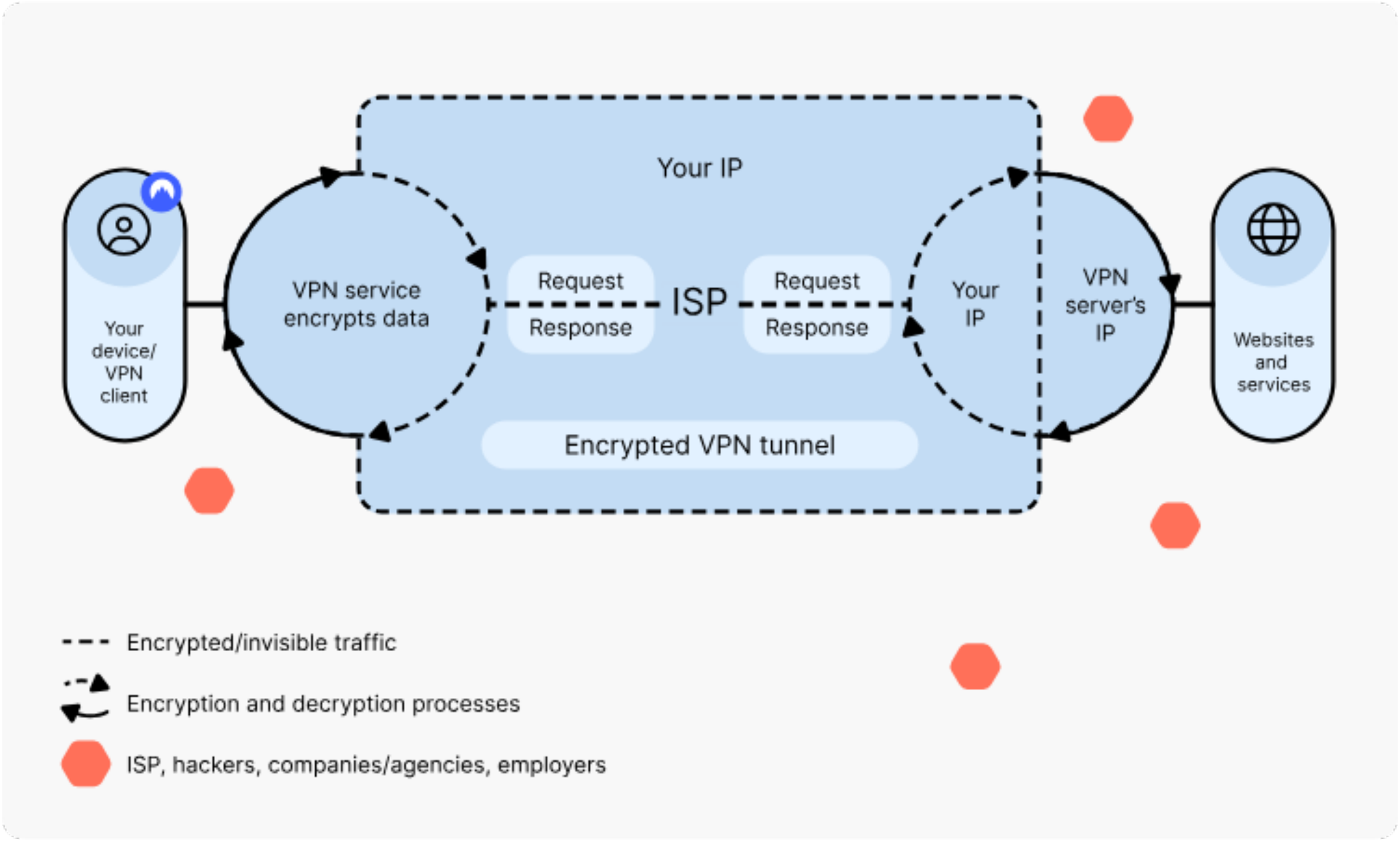
- Essentially builds a secure tunnel between its user and endpoints



VPN for a Traveling Alice



VPN for a privacy fanatic



VPN Protocols

- A **Protocol** describes how the VPN operates: connection establishment, data encryption methods, authentication, handshake, etc.

- **Common examples**

  - L2TP/IPSec: L2TP build an (unencrypted) tunnel for Link Layer, and the encryption is provided by IPSec (transport mode) in Layer 3.

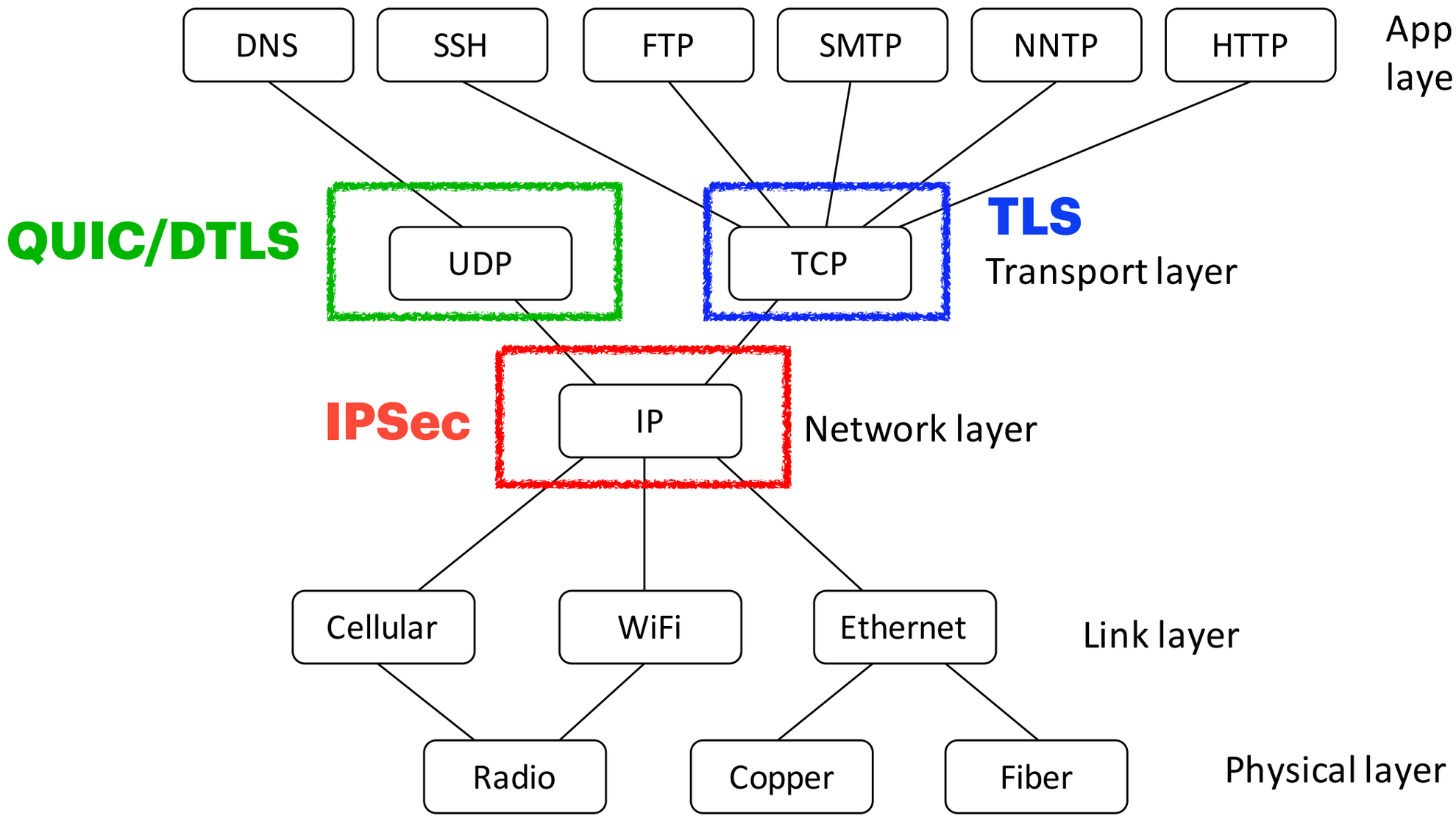
  - OpenVPN: Most widely used; Highly secure; TLS for handshake; Run over TCP/UDP.

  - Wireguard: Experimental; Simpler protocol; Utilizes modern cryptographic primitives

  - Cisco AnyConnect: Widely Used in corporates, governments, & universities. TLS for handshake, DTLS for data transport.

Transport Layer Security (TLS)

Why encryption at another layer?



Why encryption at another layer?

- **Application-Level Security**: Necessary for applications that need direct encryption and integrity at the Application Layer

- **End-to-End Protection** in Complex Networks: Additional setups are needed to make IPSec work with firewalls and NAT, while TLS traffic (especially HTTPS) can traverse firewalls and NAT easily.

Transport Layer Security (TLS)

- A crypto protocol on top of the Transport Layer that provides encryption, authentication, and (message) integrity.

- Some related terms

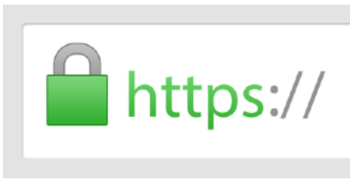
  - SSL: predecessor of TLS. People often use SSL/TLS interchangeably.

  - HTTPS: secure HTTP build on top of TLS.

  - DTLS: TLS adopted for UDP (“vanilla” TLS is designed to secure TCP).

Use cases

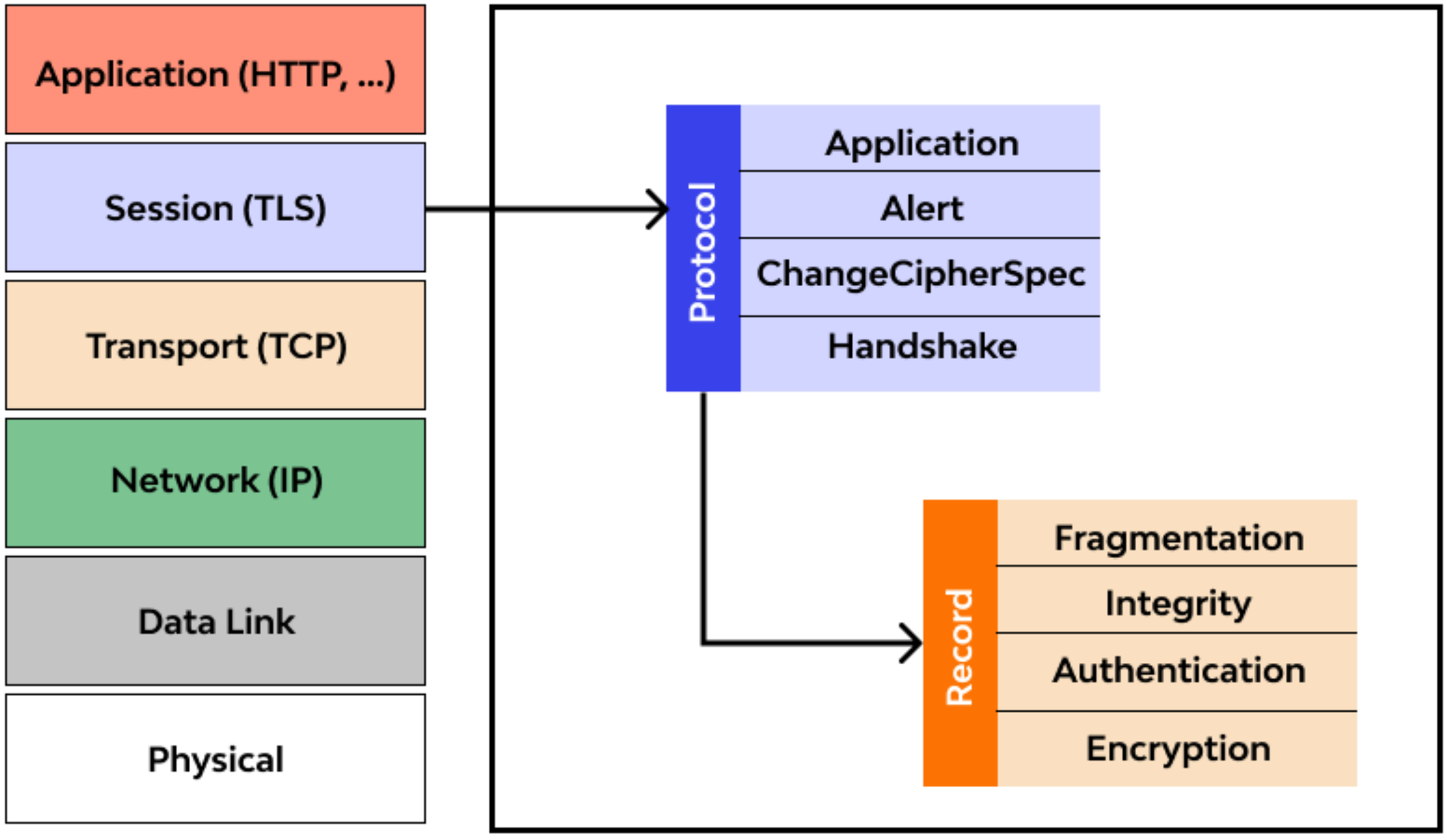
- **Web browsing**: HTTPS (HTTP over TLS)



- **Email**: SMTP over TLS (sending email), IMAP over TLS (receiving email). Now the standard practice of email providers.
- **VPN**: OpenVPN, Cisco Any Connect

- **Messaging and Chat Application**: WhatsApp, secure VoIP
- ...

Transport Layer Security (TLS)

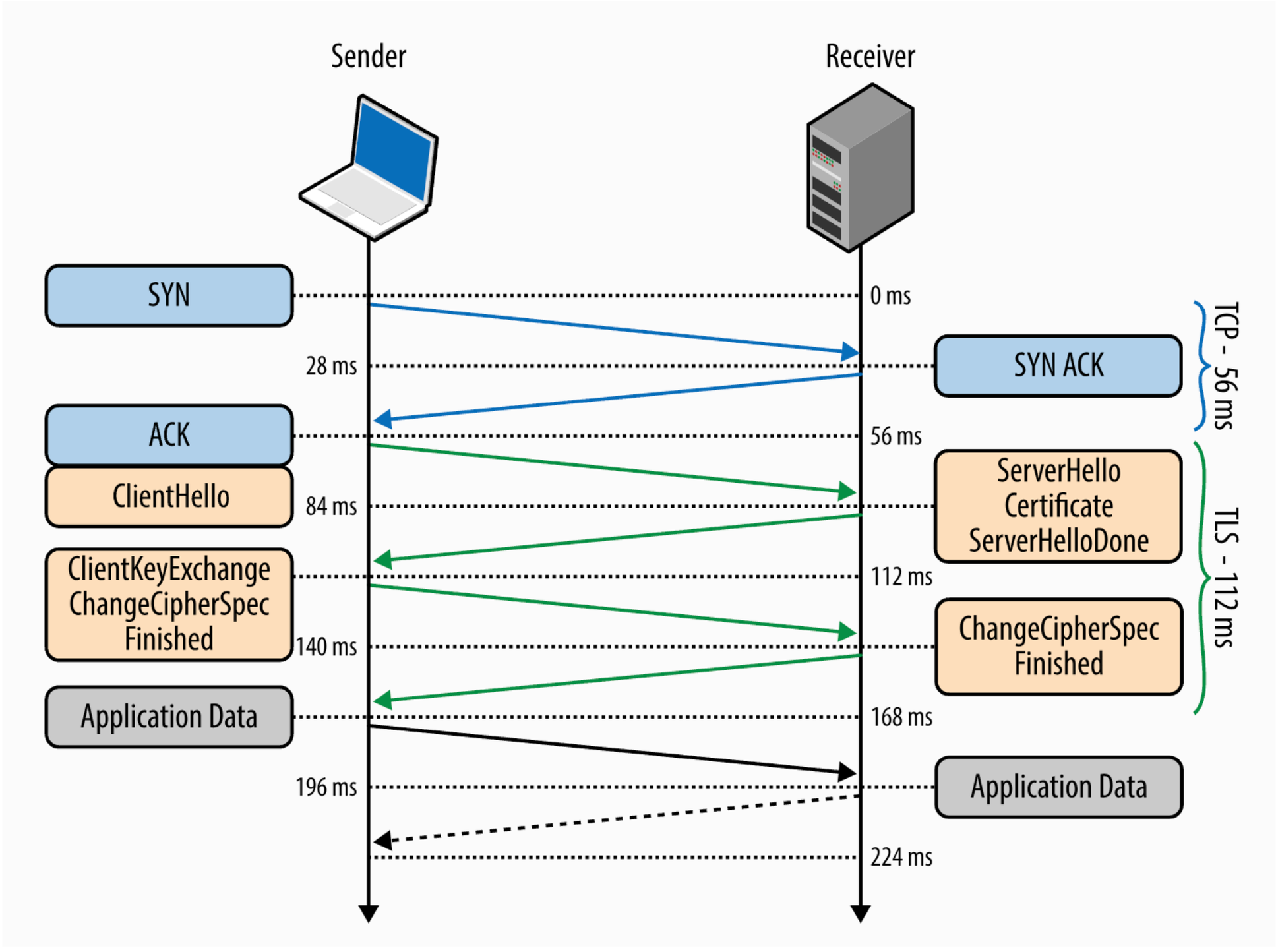


Transport Layer Security (TLS)

- **Protocol Layer**
  - Specifies how to establish (Handshake) and maintain (ChageCipherSpec & Alert) the secure channel, and how to receive/pass data to the applications on top.

- **Record Layer**
  - Specifies how to actually securing the application data (using encryption, authentication, fragmentation) after a secure connection has been established.

TLS Handshake



TLS Handshake

- Happened after one TCP handshake.
- **Main purpose**
  - Negotiate protocol options: e.g. what crypto alg & versions to use
  - Verify (server's) identity: client checks server's certificate
    - (optional) Server verify client's certificate
  - Encryption & MAC key exchange: Diffie-Hellman or RSA

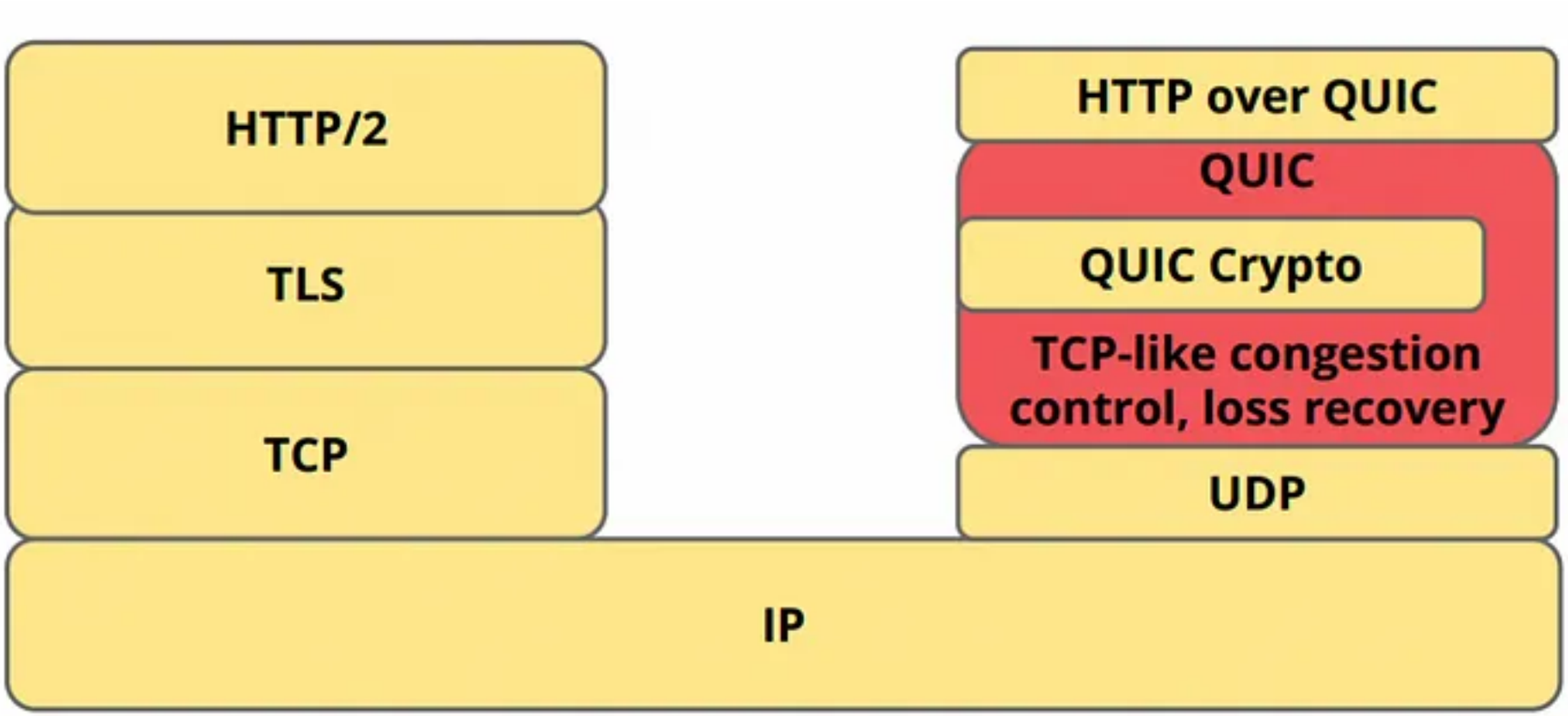
TLS Record

- The data exchanged in TLS is wrapped in a *TLS Record*
- Authentication & Encryption are handled by TLS, transparent to applications
- When sending a record, it will be passed down to the Transport Layer (TCP/UDP).

Byte	+0	+1	+2	+3
0	Content type			
1..4	Version		Length	
5..n	*Payload*			
n..m	MAC			
m..p	Padding (block ciphers only)			

QUIC

- A multiplexed transport over UDP with built-in security
  - The main change underlying HTTP/3
- Use TLS for handshake, encryption, and authentication



Attacks on TLS

- **BEAST (2011)**: exploits the predictable-IV vulnerability in TLS 1.0's usage of CBC mode.

- **CRIME (2012) and BREACH (2013)**: Infer plaintext info by observing changes in the compressed size of encrypted data.

- **Lucky Thirteen(2013)**: Timing attack on the padding of CBC-mode ciphers in TLS

- **FREAK(2014)**: Tricks the server to use a older/weaker version of TLS

- **DROWN (2016)**: exploits misconfigured servers that support both SSLv2 (insecure) and modern TLS versions

Why you should (still) use TLS

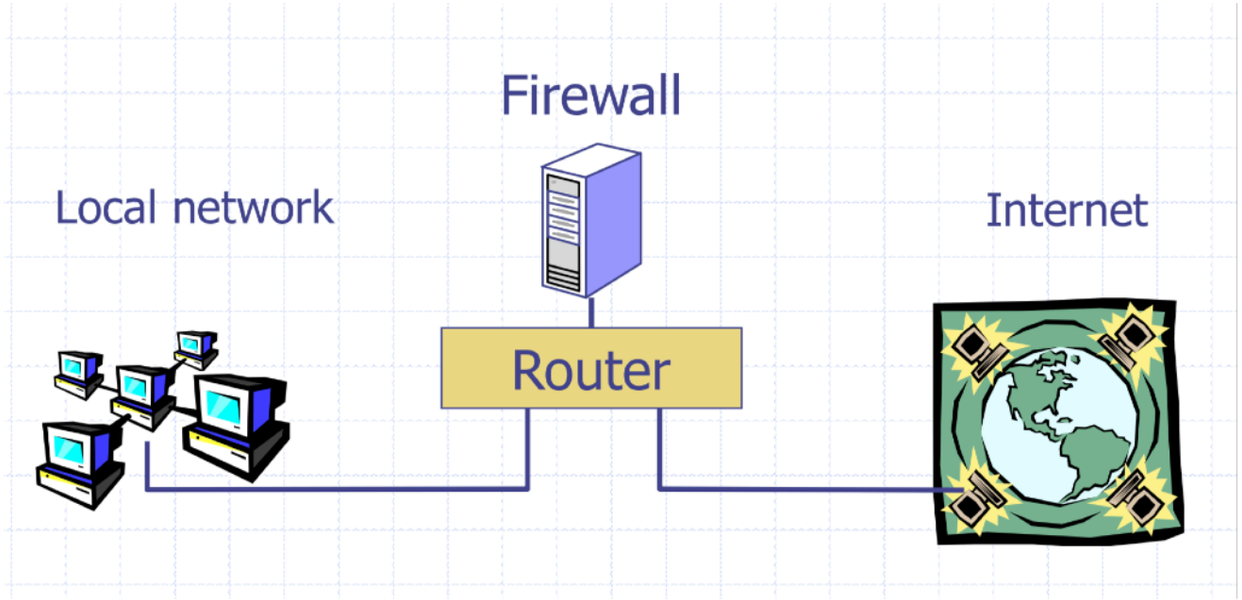
- Continuously improved and patched. The up-to-date version (**TLS 1.3**) is secure against all known vulnerabilities.
    - The security community actively researches TLS, and any vulnerabilities discovered are promptly addressed in newer protocol versions or through configuration changes.
- End-to-End Encryption and Authentication.
    - One of the most effective way to combat MiTM: it checks certificates by default
- Industry standard: supported by all major web browsers, applications, and servers

Firewalls

Firewalls

Separate local area network (LAN) from the Internet. Only allow some traffic to transit.

Sometimes rules on a router. Sometimes a standalone device.



Basic Packet Filtering

- Uses transport and IP layer information only
  - IP Source Address, Destination Address
  - Protocol (TCP, UDP, ICMP, etc.)
  - TCP and UDP source and destination ports
- **Examples**:
  - “Do not allow external hosts to connect to Windows File Sharing”
    -> `DROP ALL INBOUND PACKETS TO TCP PORT 445`

IANA Port Numbering

- **System or Well-Known Ports [1,1023]:**
  - Common services, e.g., HTTP -> 80, SSH -> 22
- **User or registered ports [1024, 49151]**
  - Less well-known services
- **Ephemeral/Dynamic/Private Ports [49152, 65535]**
  - Short lived connections

Blocklists and Allowlists

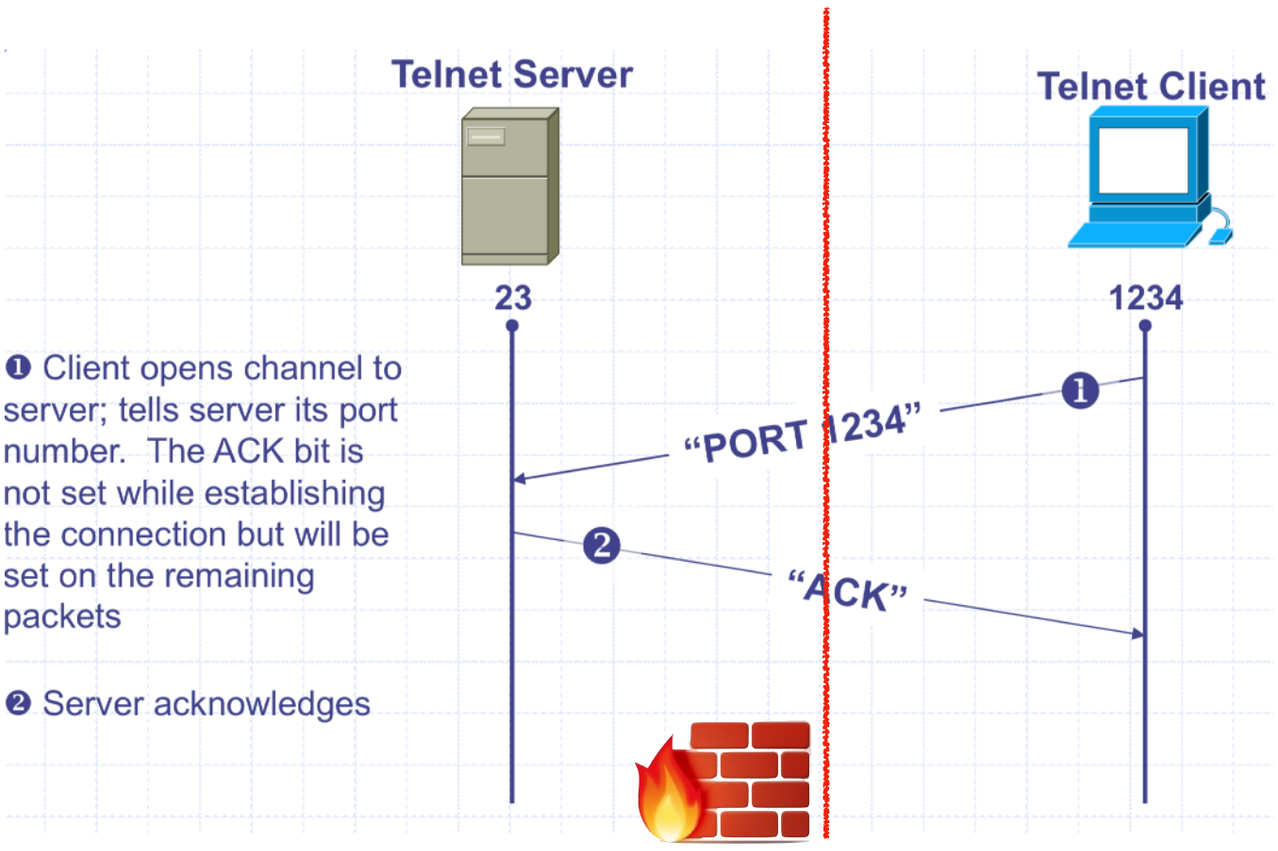
- Two fundamental approaches for firewall rulesets
- **Blocklists**: All packets are *allowed* through *except* those that fit the rules defined specifically in a blocklist.
    - Assumes the network administrator can enumerate all of the properties of malicious traffic.
- **Allowlists**: packets are dropped or rejected *unless* they are specifically allowed by the firewall

What's the rule?

- What if you have a network with lots of servers but only want outsiders to be able to access a web server?
  - `DROP ALL INBOUND PACKETS IF DEST PORT != 80`
- All outbound connections also have a source port! Their responses will blocked!

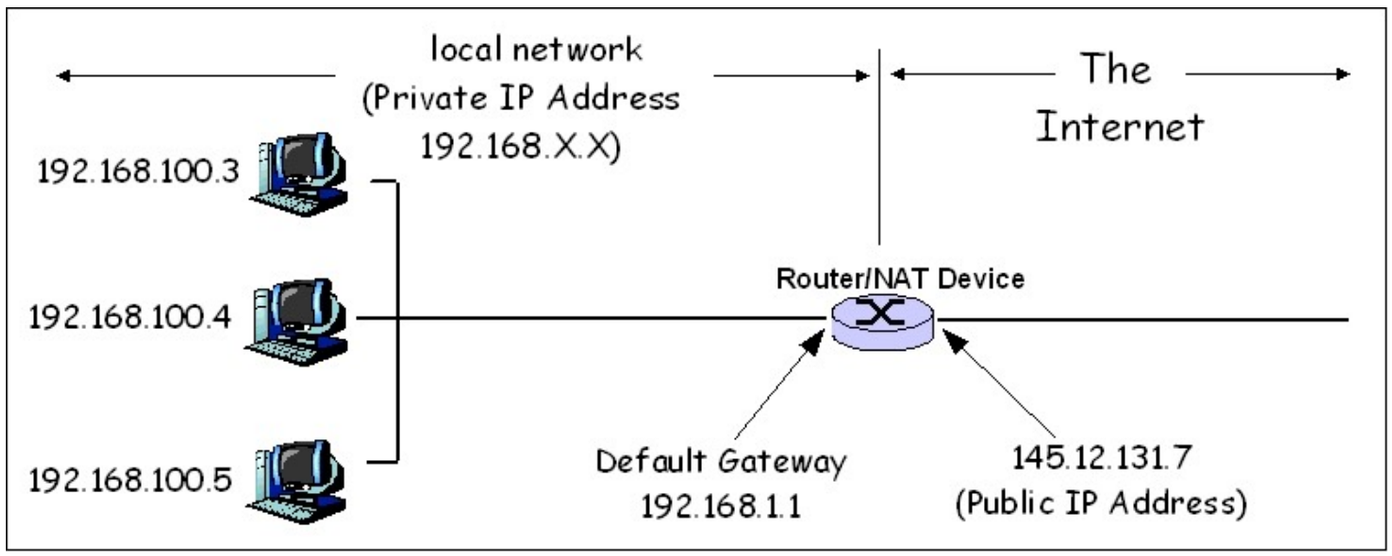
Stateful Filtering

Firewall tracks outgoing connections and allows associated inbound traffic back through



Network Address Translation (NAT)

NATs map between two different address spaces. Most home routers are NATs and firewalls.



Private Subnets

10.0.0.0 –
10.255.255.255

172.16.0.0 –
172.31.255.255

192.168.0.0 –
192.168.255.255

Local vs. Network Firewall

- Firewalls we've discussed so far have all been network firewalls. Most have lived at the edge of the organization.
- Firewalls also run on individual hosts. Linux servers use **iptables**.
- Typically have a combination of network and host firewalls
- ```
  sudo iptables -A INPUT -m conntrack --ctstate
  ESTABLISHED,RELATED -j ACCEPT
  ```
- `sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT`
  ```

Local vs. Network Firewall

- Organizations typically have a combination of network and host firewalls
  - Border (Network) Firewall will block malicious traffic *from the outside* and limit inbound traffic to accessing only servers intended to be accessed by the public
  - Host Firewalls protect hosts from other hosts (e.g., protect against internal compromise and malicious insiders)
- Think of firewall rules in terms of “Defense in Depth"

Next Generation Firewalls (NGFW)

- So far, firewalls operate by allowing access to a specific host or protocol — but what about malicious *application traffic*?

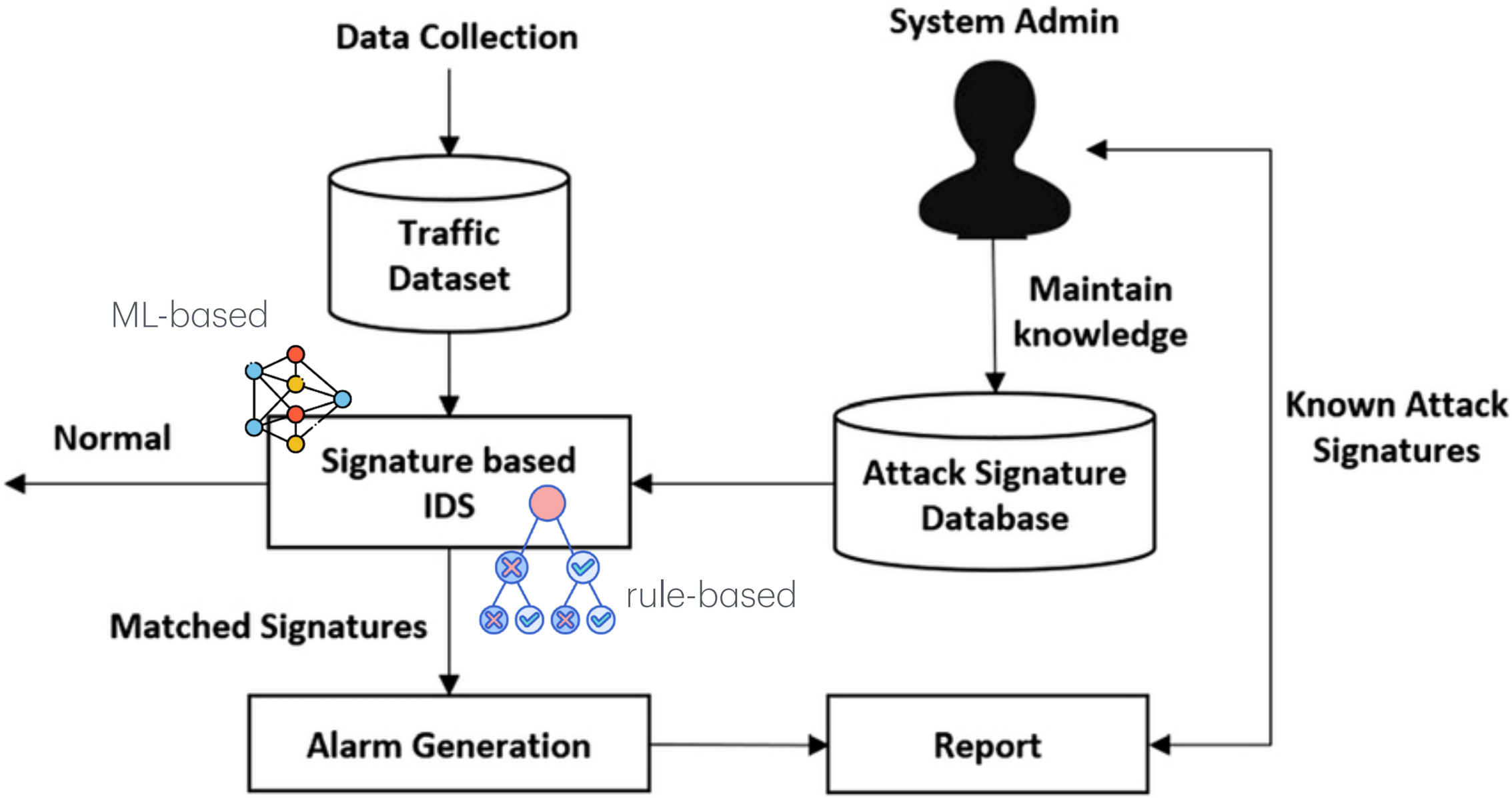
- Next Generation Firewalls (Industry term for Application-Layer firewall)s protect for attacks *within* L7 traffic

- For Example:
  - Virus scanning for SMTP
    - Need to understand protocol, MIME encoding, ZIP files, etc
  - Look for SQL injection attacks in HTTP POSTs
  - Look for a large number of authentication attempts or malformed requests

Intrusion Detection Systems (IDS)

- Software/device to monitor network traffic for attacks or policy violations
- Violations are reported to a central security information and event management (SIEM) system where analysts can later investigate
- **Signature Detection:** maintains long list of traffic patterns (rules) associated with attacks
- **Anomaly Detection:** attempts to learn normal behavior and report deviations

Signature Detection



Signature Detection

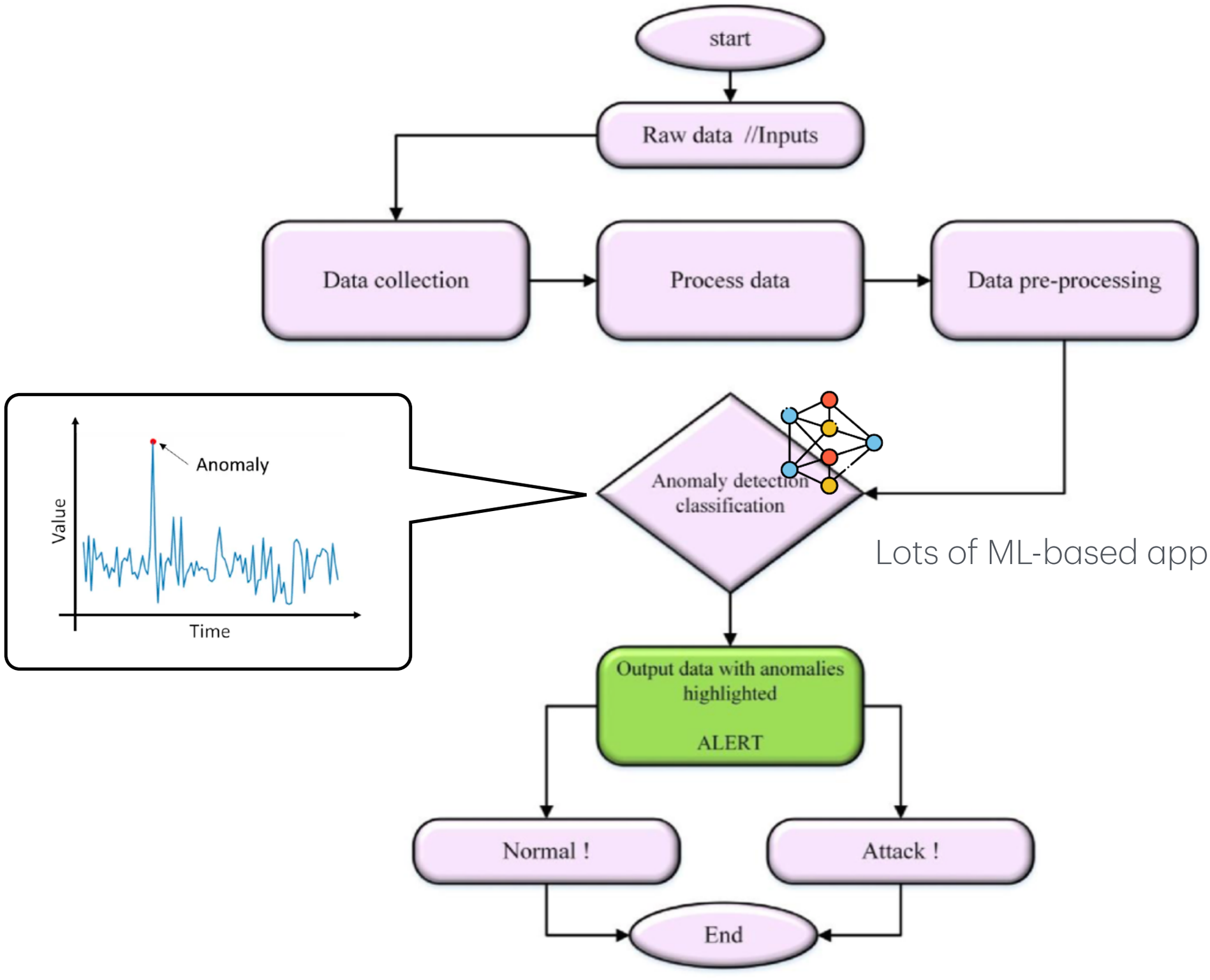
Examples

- Failed login attempts may indicate password cracking attack
- IDS could use the rule "N failed login attempts in M seconds" as signature
- If N or more failed login attempts in M seconds, IDS warns of attack
- Note that the warning is specific
  - Admin knows what attack is suspected
  - Admin can verify attack (or false alarm)

Signature Detection

- **Advantages**
  - Simple
  - Detect known attacks
  - Know which attack is happening at time of detection
  - Efficient (if reasonable number of signatures)
- **Disadvantages**
  - Signature files must be kept up-to-date
  - Number of signatures may become large
  - Can only detect known attacks
  - Variation on known attack may not be detected

Anomaly Detection



Lots of ML-based approaches

Anomaly Detection

- Anomaly detection systems look for unusual or abnormal behaviors
- There are (at least) two challenges
  - What is *normal* for this system?
  - How "far" from normal is *abnormal*?
- Statistics is obviously required here!
  - The mean defines normal
  - The variance indicates how far abnormal lives from normal

Anomaly Detection

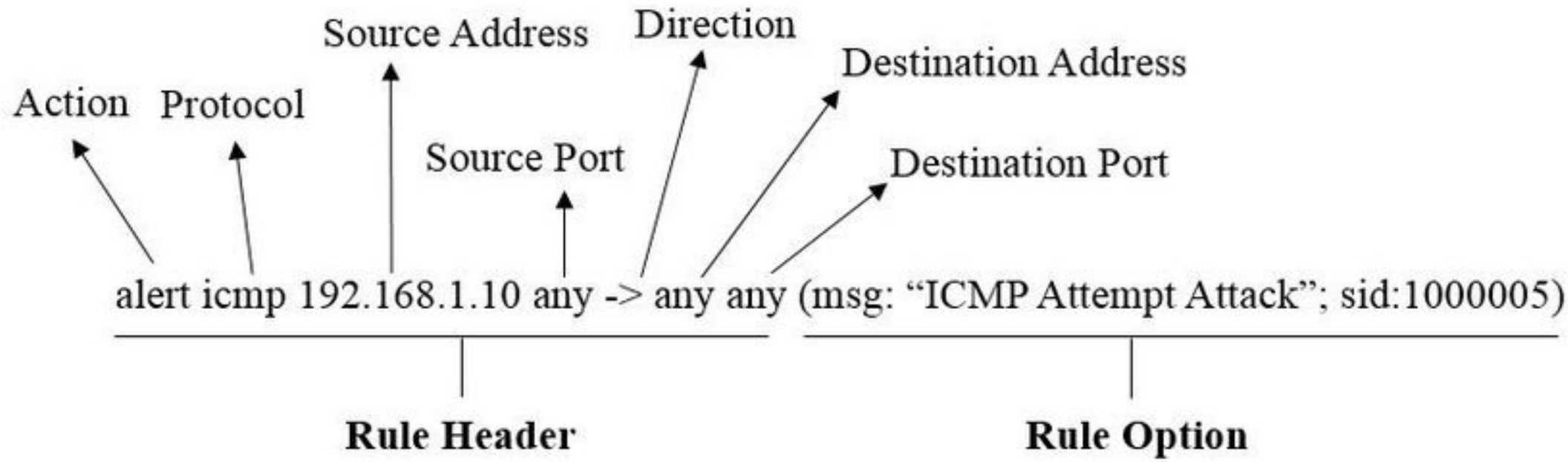
- **Advantages**
  - Chance of detecting unknown attacks
  - May be more efficient (since no signatures)
- **Disadvantages**
  - Reliability is unclear
  - High false positive/false negative
  - Anomaly detection indicates something unusual, but lack of specific info on possible attack!
- Should always be used *with* a signature detection system

Open Source IDS

Three Major Open Source IDS (and a *tremendous* number of commercial products)

- Snort
- ~~Bro~~ Zeek
- Suricata

Example Snort Rule



Outbound Too!

- Organizations will often inspect outbound traffic as well
  - Block access to sites with known malicious behavior
  - Prevent exfiltrating data
  - Block services like bit torrent
- Be careful on enterprise networks! Sometimes companies will even install their own root certificates on employee workstations to monitor TLS traffic.

Questions?