Computer and Network Security: Secure Network Protocols

Kameswari Chebrolu

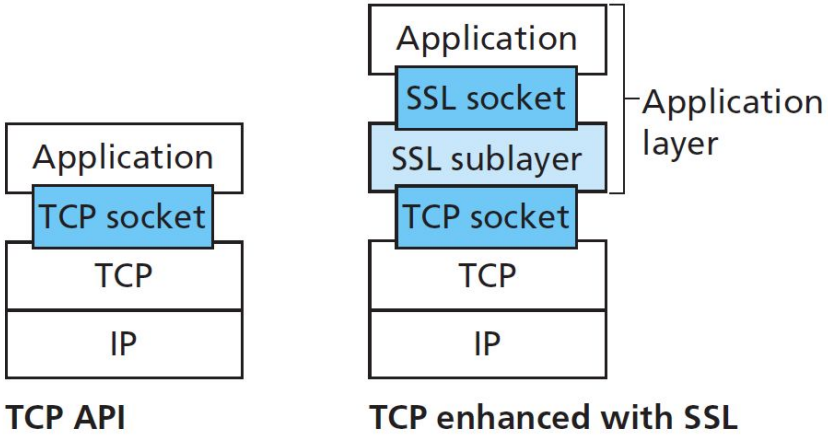
Outline

- Application Layer: SSH, ~~DNSSEC~~
- ~~Transport Layer: TLS/SSL (Done)~~
- Network Layer: IPSec
- Link Layer: WPA, WEP

Security at Different Layers

- Application Layer:
  - Developer has to implement all security mechanisms
  - Security operations (encryption/integrity) over application payload

- Transport Layer
  - Many applications can leverage provided functionality
  - Security operations (encryption/integrity) over application payload
  - No changes to the OS but applications need to change
  - Can be subject to DOS
    - Inject malicious data integrity checks fail close connection



- Network Layer
  - Secure the Internet (protect every packet)
    - Better protection against DOS
  - Changes to OS; all applications are protected without any changes
    - End point protection is IP address not user (changes to application, end point can be a user)
  - Security operation over transport header and application payload

- Link Layer:
  - Protection over only the link (very local)
    - Security can be compromised at other points (e.g. routers or other links or end host) along the path
  - Specific network device driver will implement it
    - All applications can benefit without changes
  - Security operation over network/transport header and application payload

Secure Shell (SSH)

- Very useful to remotely administer a machine
- Prior protocols: Telnet, rlogin, ftp (no security)
- SSH: secure shell; goes hand in hand with scp (secure copy for file transfer)

Secure Shell (SSH)

- Steps:
  1. Client connects to server via TCP
  2. Both parties exchange supported encryption methods, protocol version
  3. Both parties initiate a secret key exchange to establish shared key (for encryption, not authentication)
     - Based on Diffie-hellman key exchange

4. Server sends a list of acceptable authentication mechanisms which client will try in sequence
   - Password based: Client passes the password encrypted with shared key
   - Public key based:
     - Client sends server its public key
     - Server checks if this key is authorized (this is pre-configured)
     - Server sends challenge to client using the client's public key
     - Client decrypts with its private key and responds to server
5. Once client is authenticated, server lets it access resources (e.g. command prompt)

– Server authentication? Leap of faith (as covered before)

IPSec

- Why IPSec?
- IPSec Architecture
  - Authentication Header (AH)
  - Encapsulating Security Payload (ESP)
- Internet Key Exchange (IKE)

IP is not Secure!

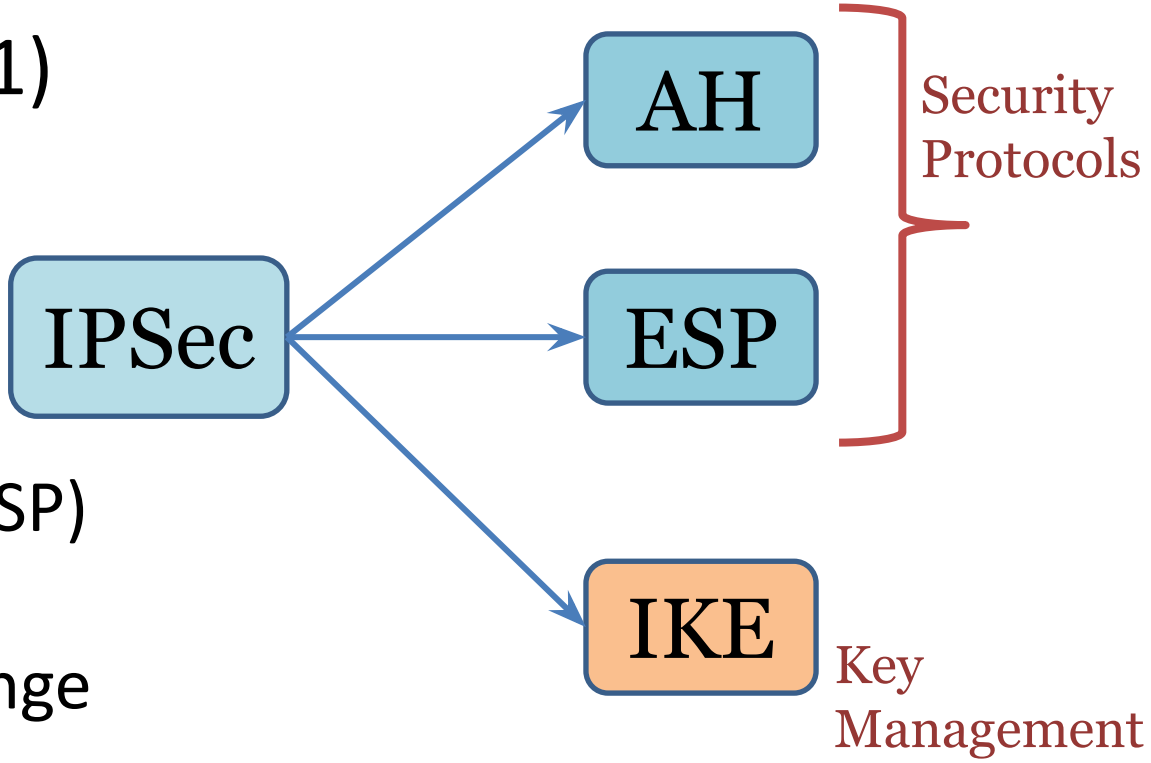
- IP: Designed without security in mind
  - IP addresses can be spoofed
  - No confidentiality
  - No integrity (crypto based; not checksum)
  - Packets can be replayed

Goals of IPSec

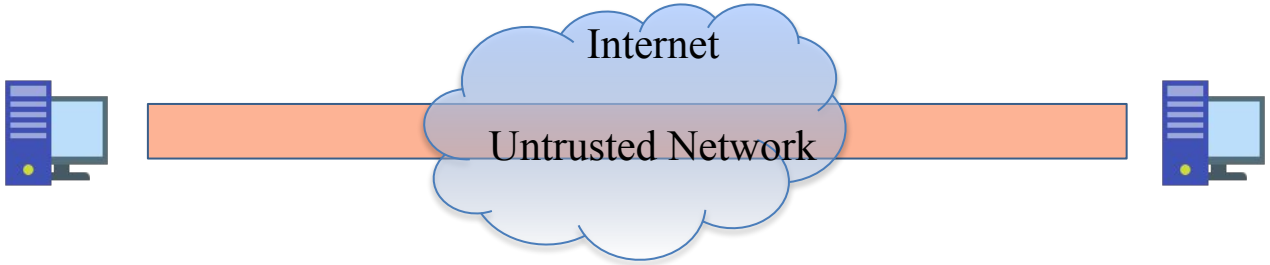
- Source authentication (to prevent spoofing)
- Provide data encryption/integrity
- Prevent replay of old packets

IPSec

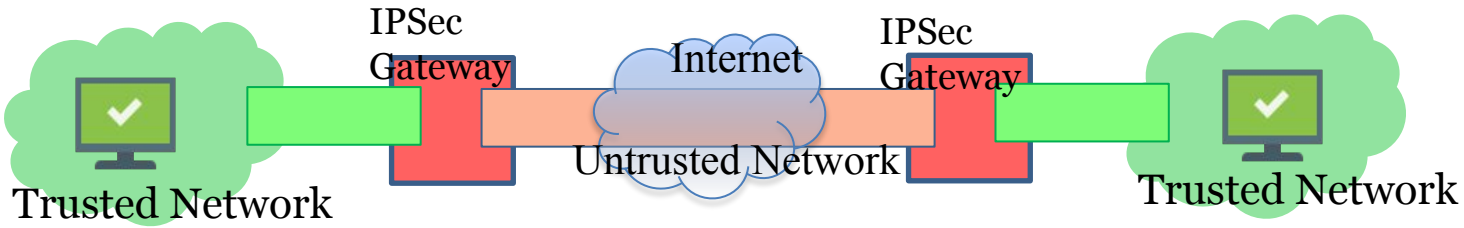
- A collection of protocols (RFC 4301)
  - Authentication Header (AH)
    - RFC 4302
  - Encapsulating Security Payload (ESP)
    - RFC 4303
  - Internet Key Exchange (IKE)
    - RFC 7276



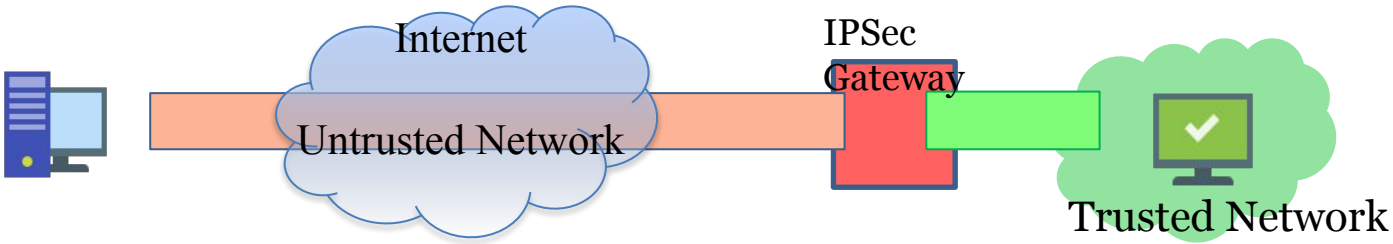
- IPSec provides security in three situations:
  - Host-to-host, gateway-to-gateway, host-to-gateway (see Fig in next slide)
- IPSec operates in two modes:
  - *Transport mode* (for end-to-end)
  - *Tunnel mode* (for VPN)



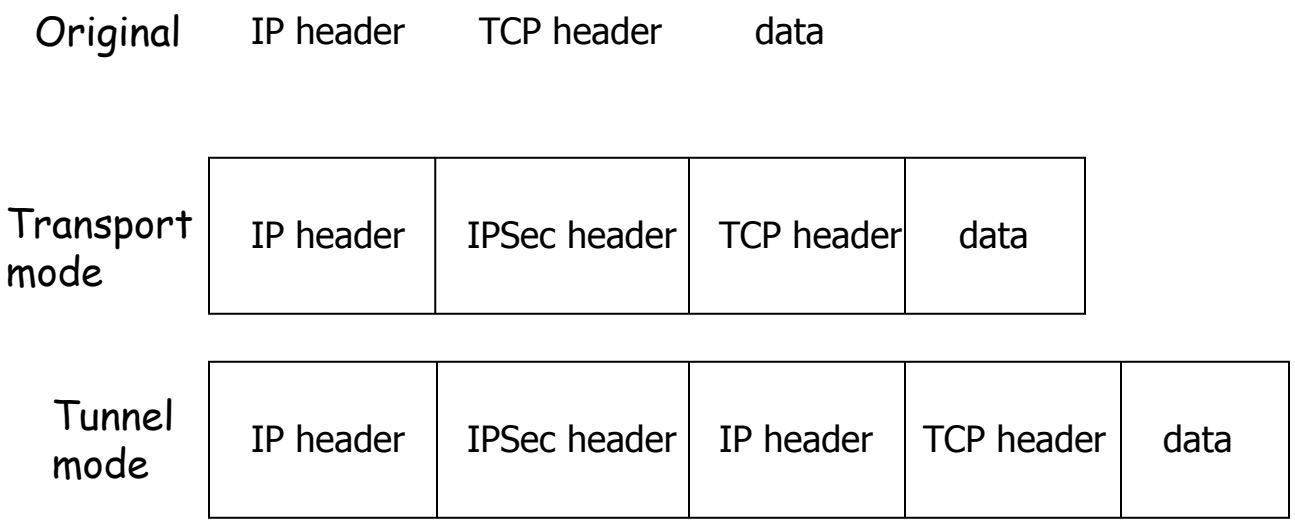
Host-to-host (not VPN)



“Gateway-to-Gateway” VPN model: between IPSec gateways

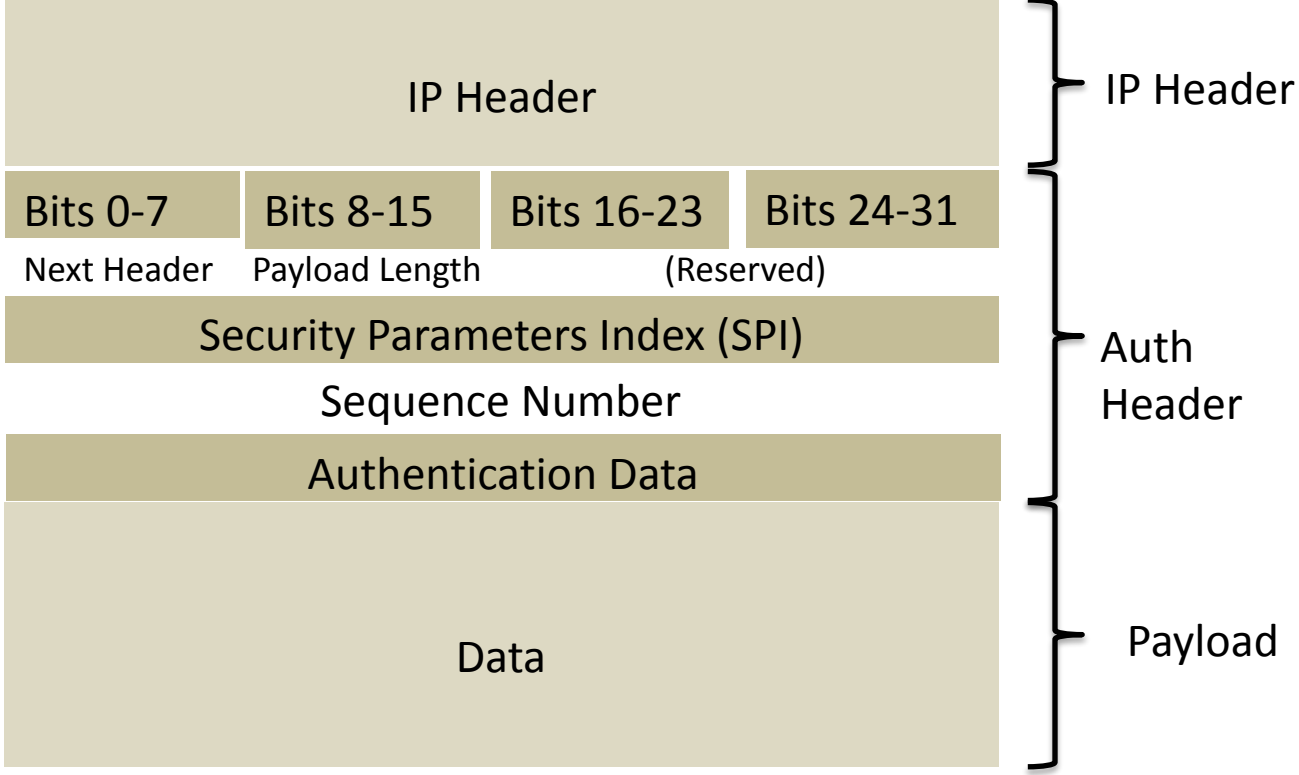


“Remote access” VPN model: host to gateway


Original
IP header
TCP header
data
Transport mode
IP header
IPSec header
TCP header
data
Tunnel mode
IP header
IPSec header
IP header
TCP header
data

Authentication Header (AH)

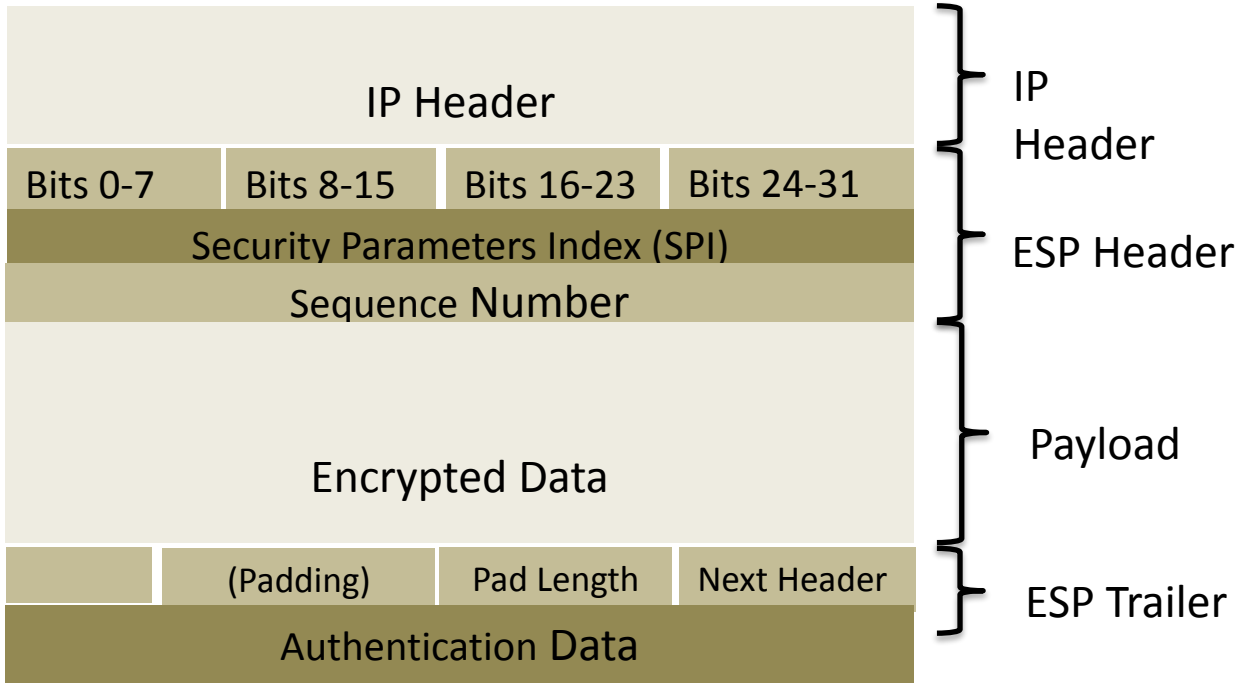
- Henceforth, assume end-nodes have shared session keys
  - Configured manually or obtained via IKE
- Provides source authentication and data integrity
  - Via hash based MAC
- Protection against replay attacks
  - Use monotonically increasing sequence numbers
- NO confidentiality!



- Authentication Data (Integrity check): calculated over entire packet
  - includes Auth header
  - Excludes fields that change during routing and Authentication data
  - Integrity check is a MAC mostly SHA-256 based
- Does not work with NAT. Why?

Encapsulating Security Payload (ESP)

- Provides confidentiality and/or integrity
  - Via symmetric key crypto
- If you want encryption, you must use ESP
- If you want integrity only, you could use ESP or AH
- If you want both encryption and integrity, you could use both ESP and AH, or just ESP



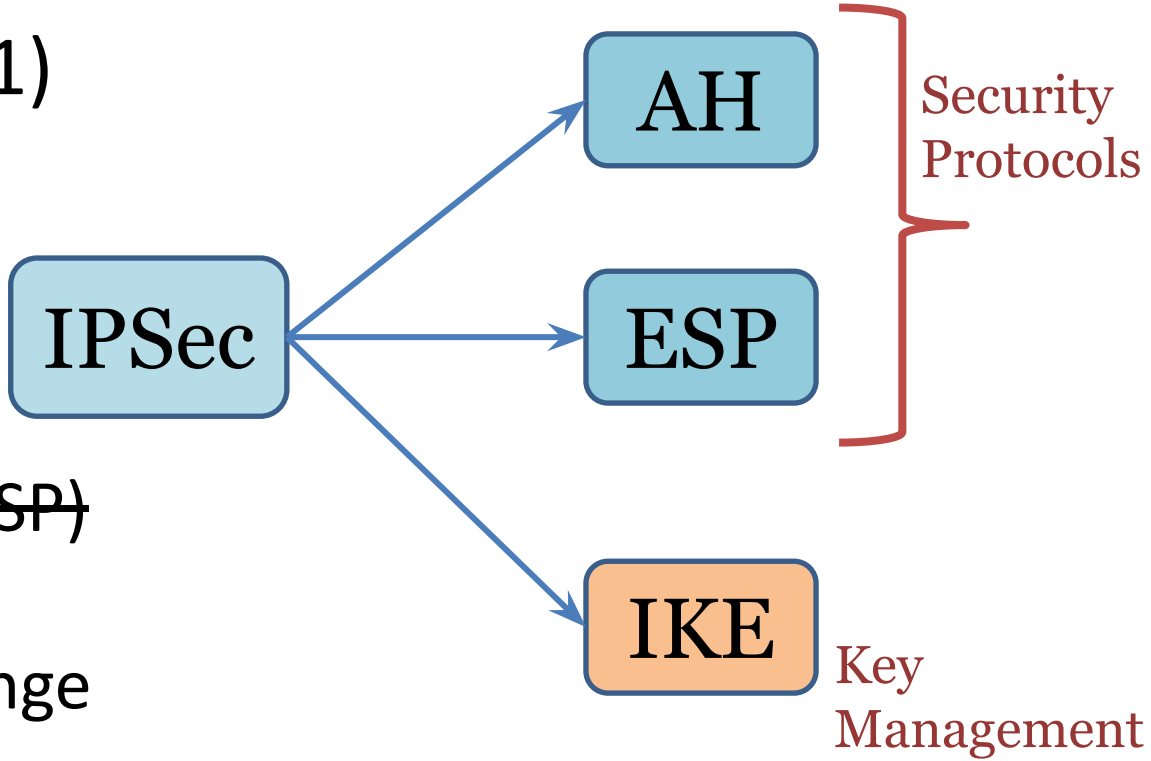
- ESP authenticates the ESP header and payload, but not the IP header.
- Allows NAT but TCP port numbers are no longer visible to NAT devices

AH vs ESP

- ESP provides confidentiality and/or integrity. Why AH?
  - Some think AH is totally unnecessary and exists due to politics
- AH protects the IP header
  - ESP can also protect it in tunnel mode
  - Many don't see why it should be protected given the MAC
- ESP does not expose layer 4 info which firewalls require; AH exposes it
  - Note: even when no encryption is used in ESP; firewalls have no way of knowing this and hence cannot look at layer 4
  - Exposed info can be altered; intermediate nodes like firewalls cannot integrity check it; so why bother?

IPSec

- A collection of protocols (RFC 4301)
  - ~~Authentication Header (AH)~~
    - ~~RFC 4302~~
  - ~~Encapsulating Security Payload (ESP)~~
    - ~~RFC 4303~~
  - Internet Key Exchange (IKE)
    - RFC 7276



Internet Key Exchange (IKE)

- Protocol for mutual authentication and establishment of a shared secret key; creates IPsec Security Associations (SAs)
  - Can be used outside IPsec as well
- An end-node will receive IPsec protected packets from many sources
- How can it know how to process them?
  - Need to know which key, algorithm to use

Solution

- IPSec header in the packet (at network layer)

- Concept of security association (SA)
  - A conversation between A and B will have two SAs, one for each direction
  - Each SA is associated with
    - Identity of the other end point
    - Seq number# (for preventing replay)
    - Cryptographic service (integrity, and/or confidentiality)
    - Which algorithms for above service

- SPI (Security Parameter Index) included in IPsec header
  - Identifies the security association
  - SPI value is chosen by the destination; can ensure that the SPI is unique with respect to all the sources
- SA is defined by the triple <SPI, destination address, flag for whether it's AH or ESP>.

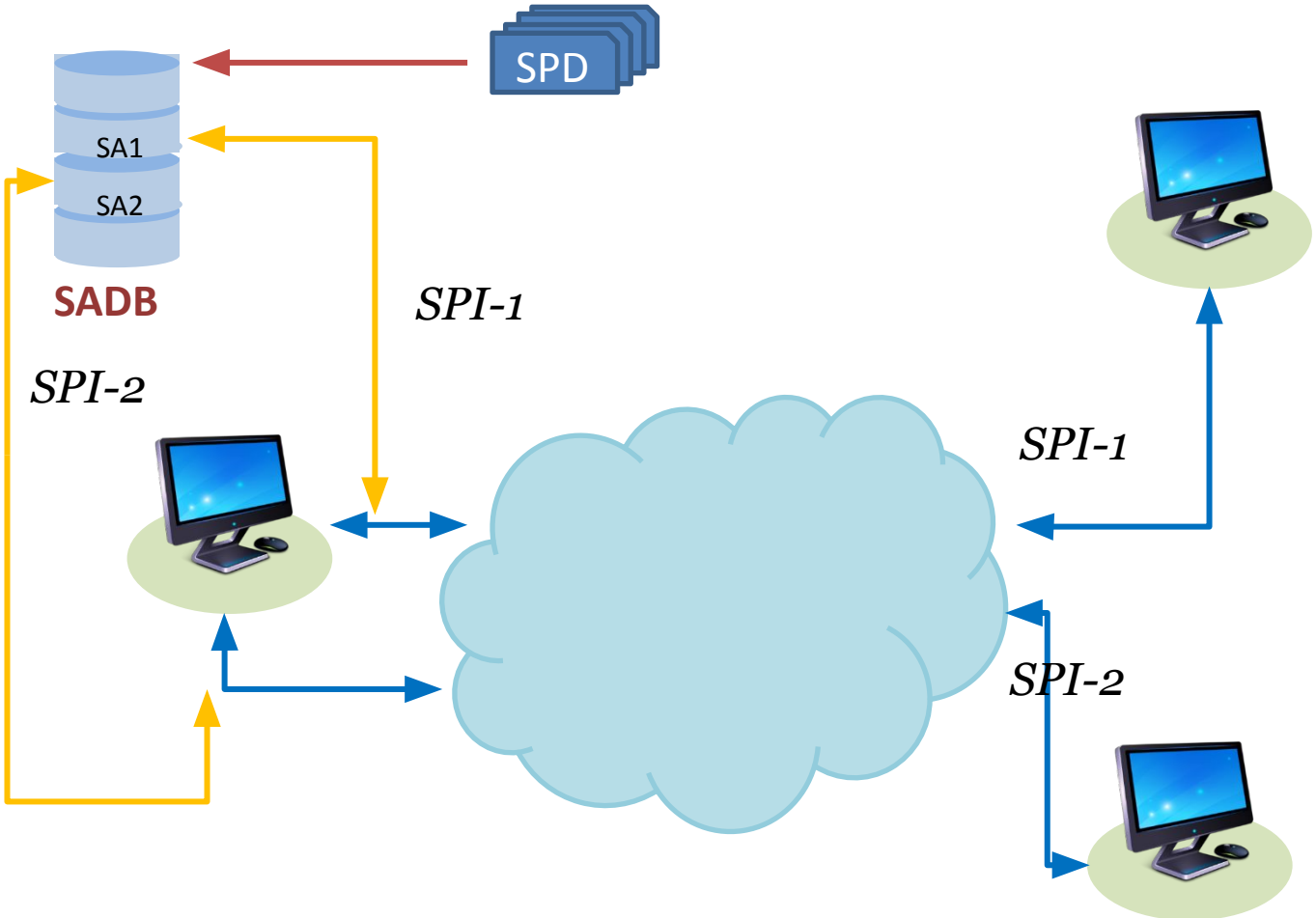
SA Database

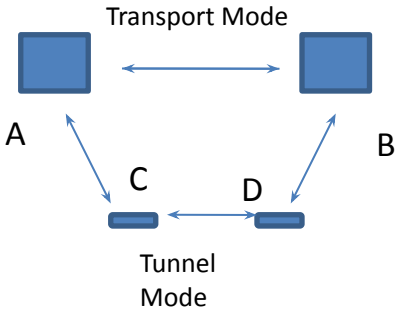
- Security Association (SA) Database: When transmitting to IP destination *X*, transmitter looks up *X* in SA database
  - Lookup will provide SPI, the key, the algorithms, the sequence number, etc
- When receiving an IP packet, the SPI of the received packet is used to find the entry in the SA database
  - Lookup will tell key, sequence number etc to use to process the packet.

Security Policy (SP) Database

- Similar to firewall tables
- SP database specifies
  - which types of packets should be dropped completely
  - which should be forwarded or accepted without IPsec protection
  - which should be protected by IPsec, and if so how (AH or ESP?)
- Decisions could be based on layer3/4 headers

How They Fit Together





A's SPD

From	To	Protocol	Port	Policy
A	B	Any	Any	AH[HMAC-MD5]

From	To	Protocol	SPI	SA Record
A	B	AH	12	HMAC-MD5 key

A's SADB

From	To	Protocol	Port	Policy	Tunnel Dest
A sub	B sub	Any	Any	ESP [3DES]	D

C's SPD

From	To	Protocol	SPI	SA Record
A sub	B sub	ESP	14	3DES key

C's SADB

IKE Phases

- Protocol used to set up a SAs between two nodes
- Phase 1: mutual authentication and establishment of session keys
  - Shared key generated based on diffie-hellman key exchange
  - Authentication based on pre-shared secret or public key crypto;
- Phase-2: Using keys established in phase 1, multiple SAs between the same pair of nodes established
- Why two phases?
  - Many SAs may have to be established (for different traffic flows)
  - Symmetric based session keys of phase 1 make it faster

Example

- All packets sent to
  - canara-bank.com must be encrypted using AES with HMAC-MD5 integrity check
  - www.iitb.ac.in must use HMAC-SHA1 integrity check (no encryption)

Reference

- https://briolidz.wordpress.com/2012/01/23/ipsec-made-simple/

Outline

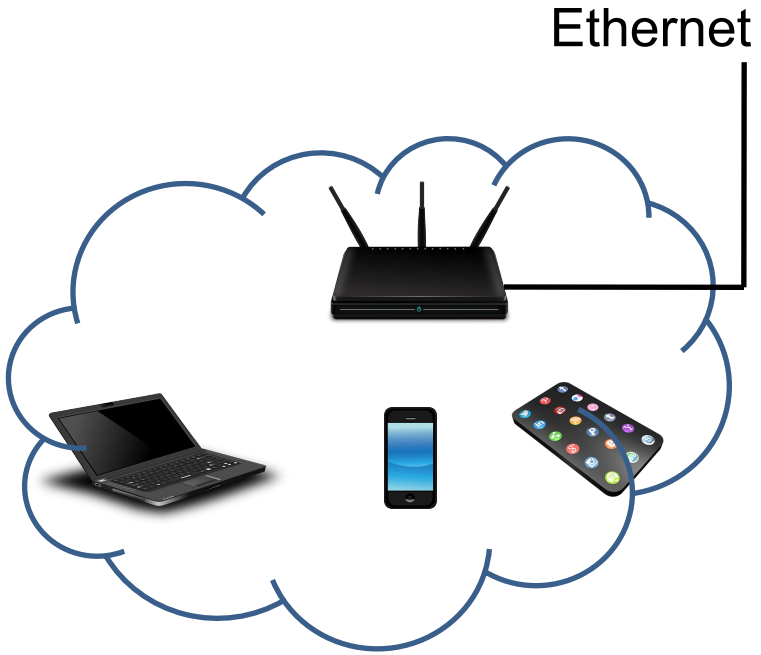
- ~~Application Layer: SSH~~
- ~~Transport Layer: TLS/SSL (Done)~~
- ~~Network Layer: IPSec~~
- Link Layer: WPA, WEP

Wireless

- Link layer encryption a must in wireless
  - Wired is lot tougher to eavesdrop; often no security mechanisms employed
- Challenges to overcome
  - Eavesdropping
  - Session Hijacking
  - Interloping (unauthorized user using some one else's AP)
    - Funny SSID names: "I read your email", "Get off my lan", "my-fi not your-fi" ☺️
- Focus: 802.11 (WiFi) Link layer security mechanisms

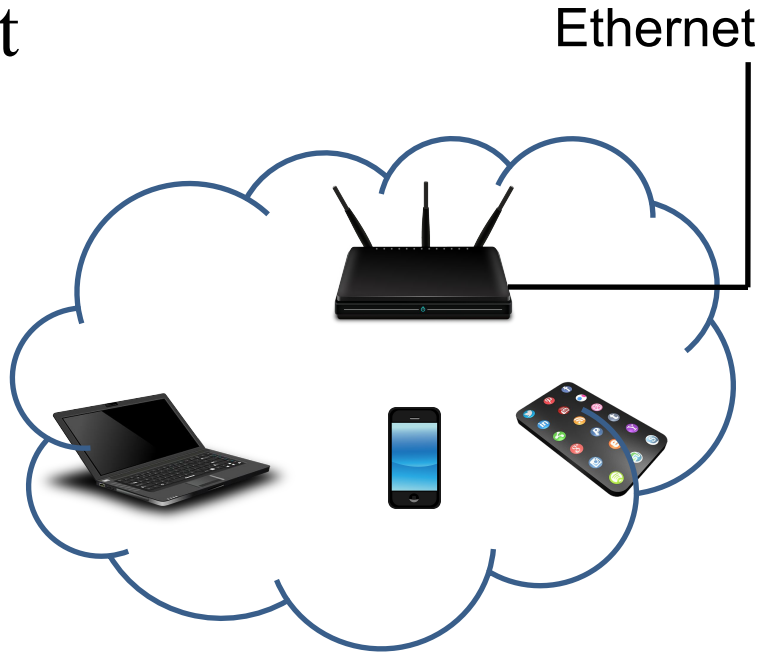
Working of WiFi: SSID -- 1

- Every AP configured with an SSID
- SSID broadcast via periodic beacons
  - Beacons carry other information: AP capabilities, time-stamp etc
  - Typically sent once every 100ms



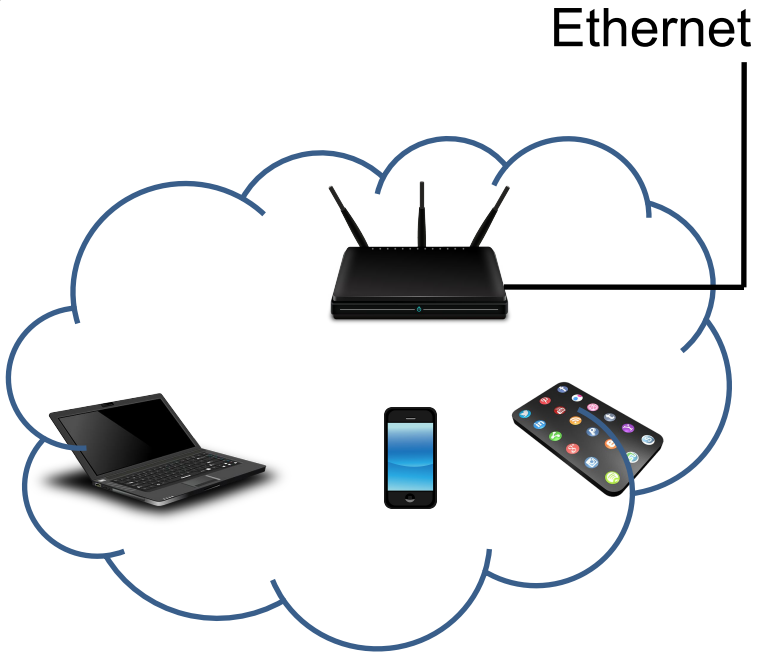
Working of WiFi: Scanning -- 2

- Client can be in coverage area of many APs operating over different channels
- Passive Scanning: Scan channels and simply listen to beacons
- Active Scanning: Probe request from client elicits probe response from AP
  - Scanning all channels time consuming; can save time



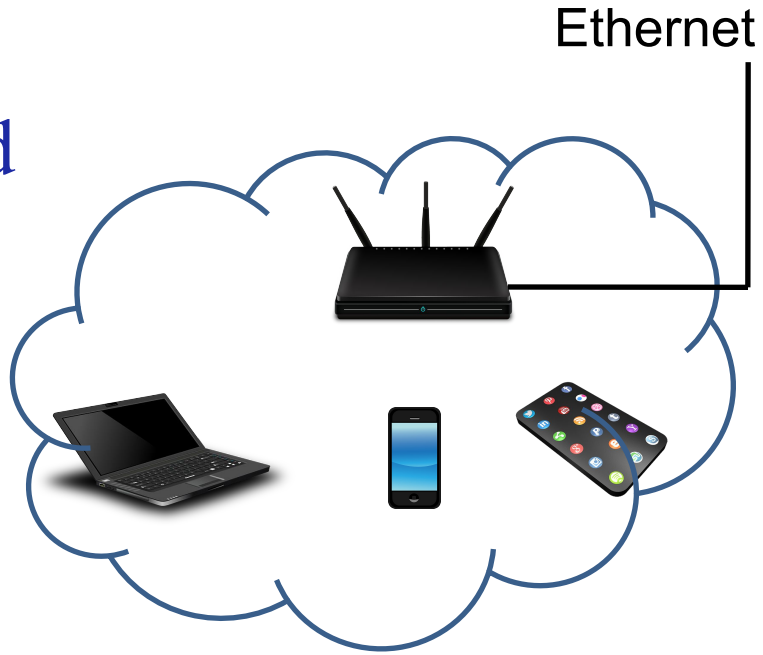
Working of WiFi: AP Selection -- 3

- Client acquires a list of APs via scanning
- Select “best” one
  - Based on signal strength
  - User preferences
  - Trust
  - Free or payment based



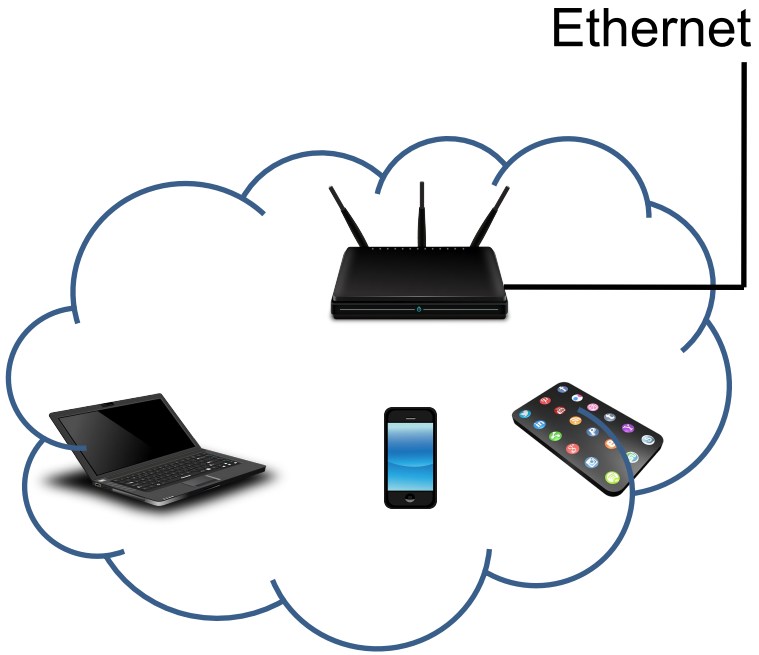
Working of WiFi: Authentication -- 4

- Allow only authorized clients to connect to AP
- Network security features defined by 802.11i
  - Apart from authentication, also provides data confidentiality
- A client can authenticate with multiple APs
  - Speeds up roaming



Working of WiFi: Association -- 5

- Any client must associate with an AP before data transfer
  - Can associate with only one AP at any time
  - Client packets are effectively routed
- Association request from client specifies its capabilities and SSID
- Association response from AP specifies accept or reject
- After association, **data transfer** can begin



Types of Frames

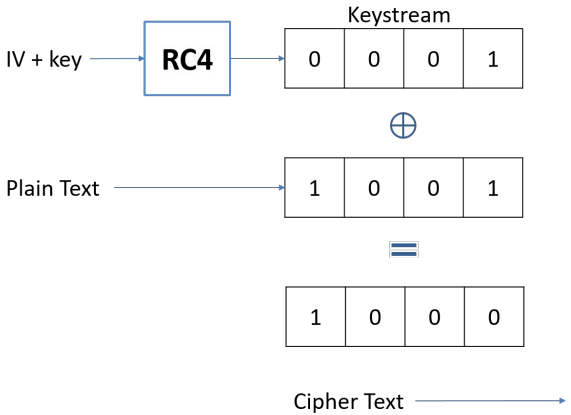
- Management: Help maintain communication
  - Authentication, Association, Beacons, Probe request/response
- Data: Carry higher layer data (email, web traffic etc)
- Control: Facilitate exchange of data frames
  - ACK, RTS (Request to Send), CTS (Clear to Send)

Wired Equivalent Privacy (WEP)

- Goal: Provide confidentiality, integrity and access control in a wireless LAN
  - Part of the 802.11 standard
- All clients and AP have a pre-shared secret key (same for all; derived from the password you normally type to access home WiFi APs)

Confidentiality

- Based on RC4 stream cipher
  - Original key was 40 bits, later extended to 128, then 256 bits
  - Seed to cipher: 24 bit IV + WEP key
  - Cipher generates keystream
  - Plain-text xor'ed with keystream to get cipher-text
- (Ciphertext, IV) sent over air
- Vulnerability: IV should not be reused but receiver does not check and reject reused IVs (more later)



Integrity

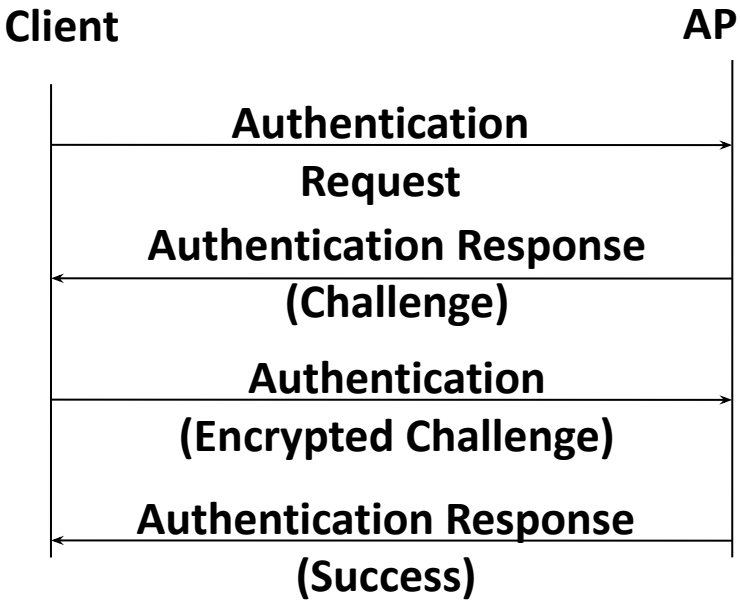
- Provided by CRC-32 checksum
  - Not a cryptographic hash function
  - Protects against transmission errors
  - Some attacks exploit this weakness of CRC-32 (more later)

Access Control

- Handled by authentication frames
- Two methods: **Open system** and **Shared key** authentication
- **Open system**: Client request (unecrypted) always successful
  - No need to provide any credentials
  - Note client still needs pre-shared key to send/receive data frames which are encrypted
  - If correct key not used, AP drops data frames

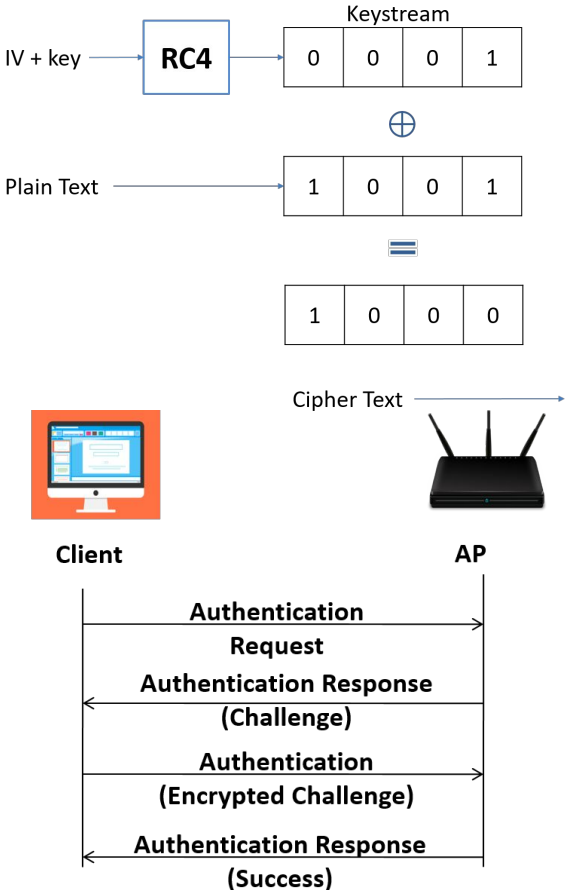
Access Control

- **Shared key**: client needs to prove possession of the WEP key before associating
- Based on challenge-response
  - Encryption based on RC4 algorithm
- Vulnerability (for both): Only client is authenticated
- Which method Open or Shared key more secure?



Shared Key mode: Vulnerability

- Sniffing challenge-response, attacker can get keystream and associated IV
- The IV and keystream can be reused for authenticating the attacker or injecting packets
- Note: AP does not check reuse of IV



Open System: Vulnerability

- This is also insecure; RC4 cipher is weak
- Ciphert-text only attack: 40,000 encrypted data packets and corresponding IVs; 50% chance of recovering the WEP key
- Want to get 40,000 packets fast?
  - Capture an ARP request (encrypted) and repeatedly transmit it to AP
  - AP will rebroadcast it with different IV each time

- Can’t capture an ARP request?
  - Pretend to be an AP and de-authenticate a client
  - Client with re-authenticate and issue a new ARP (ARP table is often flushed on de-authentication)

Other Attacks

- Decrypt packet via chop-chop attack
  - Vulnerability: CRC-32 checksum; AP drops checksum failed pkts
  - Attacker truncates data by one byte; guesses the dropped byte and corrects checksum (nature of checksum calculation permits this even when data encrypted)
  - If guess is correct, AP will reply; else guess again till last byte guessed correctly
  - Repeat until all bytes guessed correct, one at a time
  - See: https://www.aircrack-ng.org/doku.php?id=korek_chopchop

- Caffe-latte attack: Get holds of WEP key
  - Interacts only with client; no need to be near AP
  - Vulnerability: AP not authenticated; weak checksum
  - Attacker setups a honeypot AP (with same SSID whose key needs to be recovered) and makes the client associate with it
  - Client sends an encrypted gratuitous ARP after association
  - Attacker modifies this ARP request from client into valid ARP request to client (using chop-chop theory)
  - Repeated sending of valid ARP request (same one) will result in many ARP replies (with different IVs)
  - Use these to break the key (cipher text only attack)
- See https://www.aircrack-ng.org/doku.php?id=cafe-latte

WPA and WPA2

- WiFi Protected Access (WPA)
- WPA: 2003, WPA2: 2004
  - WPA a stop gap measure till proper standard implemented
  - WPA needs no hardware changes

WPA

- RC4 cipher + 128 bit key + 48 bit IV
- Per packet key
  - no longer a concatenation of IV and key
  - Mixing function of IV and root key
- MAC based on Michael algorithm (64 bit integrity check)
  - Not secure but better then CRC
- Seq. no in the packets to avoid replay

WPA2

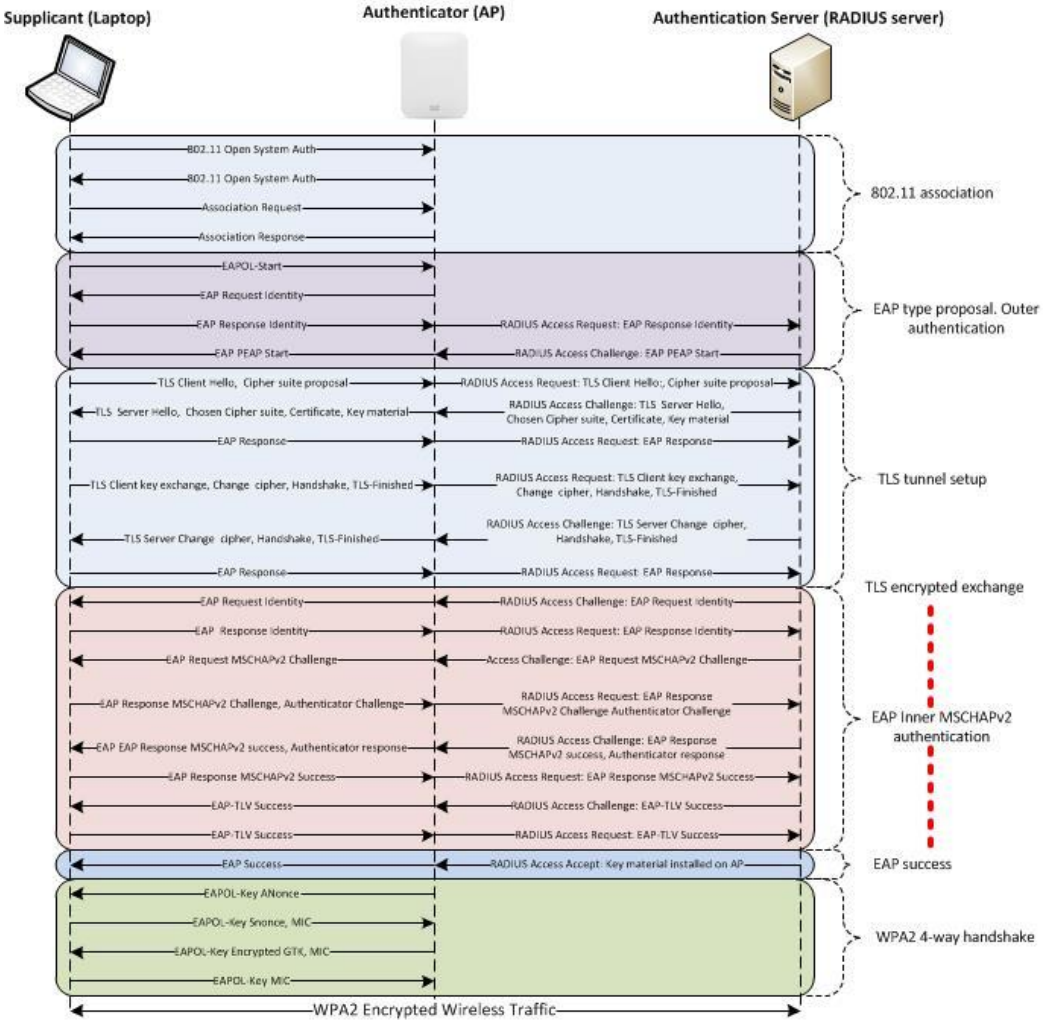
- Based on AES for integrity and confidentiality
- Seq no for preventing replay
- Key management and authentication via EAP (Extensible Authentication Protocol)
- In 2017, some vulnerabilities found in WPA2 standard (See https://www.krackattacks.com/)
  - Can decrypt packets due to nonce reuse (does not get the key/password)
  - Patch available to prevent this; no need for a new standard ☺️

Authentication Modes

- Pre Shared Key (PSK)
  - Suitable for home networks
  - Key derived from pass phrase
  - Drawbacks:
    - Same passphrase for all users
    - Stored on device stolen device issues
    - Key change difficult
    - Vulnerable to dictionary attack

Authentication Modes

- 802.1x (EAP/Radius based; WPA enterprise)
  - For large organizations and secure applications
  - Login credentials for users (can be revoked)
  - User never deals with key (keys created and assigned per user session after login)
  - Drawback: Lot more complex to setup


Supplicant (Laptop)
Authenticator (AP)
Authentication Server (RADIUS server)
802.11 Open System Auth
802.11 Open System Auth
Association Request
Association Response
802.11 association
EAPOL-Start
EAP Request Identity
EAP Response Identity
RADIUS Access Request: EAP Response Identity
EAP PEAP Start
RADIUS Access Challenge: EAP PEAP Start
EAP type proposal. Outer authentication
TLS Client Hello, Cipher suite proposal
RADIUS Access Request: TLS Client Hello:, Cipher suite proposal
TLS Server Hello, Chosen Cipher suite, Certificate, Key material
RADIUS Access Challenge: TLS Server Hello, Chosen Cipher suite, Certificate, Key material
EAP Response
RADIUS Access Request: EAP Response
TLS Client key exchange, Change cipher, Handshake, TLS-Finished
RADIUS Access Request: TLS Client key exchange, Change cipher, Handshake, TLS-Finished
TLS Server Change cipher, Handshake, TLS-Finished
RADIUS Access Challenge: TLS Server Change cipher, Handshake, TLS-Finished
EAP Response
RADIUS Access Request: EAP Response
TLS tunnel setup
TLS encrypted exchange
EAP Request Identity
RADIUS Access Challenge: EAP Request Identity
EAP Response Identity
RADIUS Access Request: EAP Response Identity
EAP Request MSCHAPv2 Challenge
Access Challenge: EAP Request MSCHAPv2 Challenge
EAP Response MSCHAPv2 Challenge, Authenticator Challenge
RADIUS Access Request: EAP Response MSCHAPv2 Challenge Authenticator Challenge
EAP EAP Response MSCHAPv2 success, Authenticator response
RADIUS Access Challenge: EAP Response MSCHAPv2 success, Authenticator response
EAP Response MSCHAPv2 Success
RADIUS Access Request: EAP Response MSCHAPv2 Success
EAP-TLV Success
RADIUS Access Challenge: EAP-TLV Success
EAP-TLV Success
RADIUS Access Request: EAP-TLV Success
EAP Inner MSCHAPv2 authentication
EAP Success
RADIUS Access Accept: Key material installed on AP
EAP success
EAPOL-Key ANonce
EAPOL-Key Snonce, MIC
EAPOL-Key Encrypted GTK, MIC
EAPOL-Key MIC
WPA2 4-way handshake
WPA2 Encrypted Wireless Traffic

Summary

- Many new network protocols developed from scratch to handle security concerns
- Looked at a small sample across layers
  - SSH, TLS/SSL, IPSec and WEP/WPA