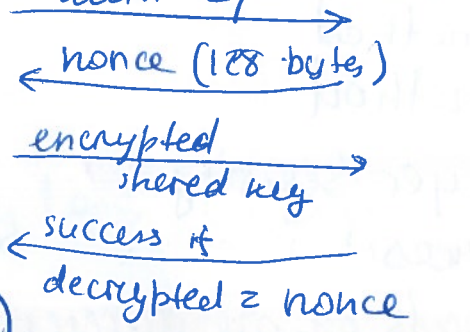
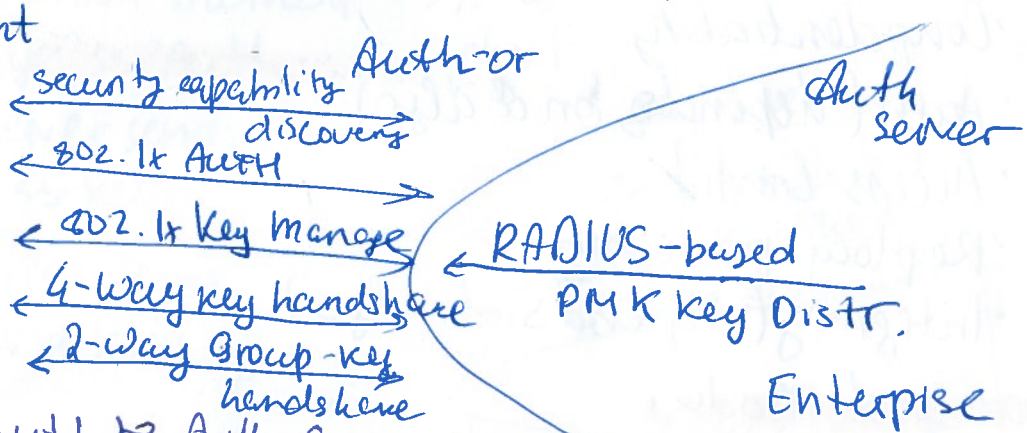
Lecture 15

HTTPS = 443

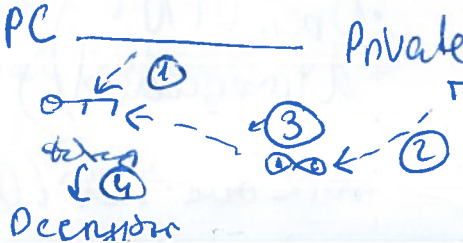
- Physical layer security
  - Wired Equivalent Privacy (WEP)
    - 40 → 104 bit key
    - considered broken



- ~~Wi-Fi Protected Access (WPA)~~
- Wi-Fi Protected Access (WPA 2)
  - Personal — auth to Access Point with Pre-shared key
    - requires testing and certification by the Wi-Fi Alliance
    - AES-based (CCMP) encryption mode
    - Pairwise Master Key (PMK) is generated from the SSID and the PSK

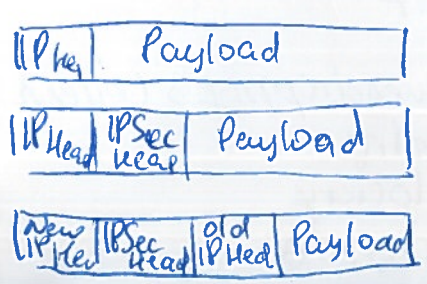


  - Enterprise — auth to Auth Server with credential
    - Auth is tunnelled to a RADIUS server
    - PMK is returned by RADIUS server for that session (renewed per session)
- WPA3
  - uses forward secrecy ciphers
  - freezers shows 6 access points with support



1. Eavesdropping & Collecting
2. Stealing
3. Unlocking
4. Decrypting

- Wi-Fi Protected Setup (WPS)
  - Several modes
    - ~~PIN~~ PIN method ← flaws in early implementation
    - Push button method
    - NFC method
    - USB method
- Network Layer Security
  - IPSec Services
    - Two protocols offering different service
      - Auth Header (AH) ← incompatible with NAT (changes IP addr cause does not know the key
        - Auth
        - Access control
        - Replay protection
        - Integrity
      - Encapsulated Security Payload (ESP)
        - Confidentiality
        - Auth (depends on algo)
        - Access Control
        - Replay protection
        - Integrity (depends on algo)
  - Transport Mode
    - End-Hosts are ~~do~~ IPSec aware
    - Protects upper level protocols
    - IPSec datagram emitted and received by end-system
  - Tunneling mode ← can be used for VPN
    - Routers are IPSec aware (both)
    - One Router and one end-system is IPSec aware



- Open VPN
- Wireguard

work over TCP/UDP

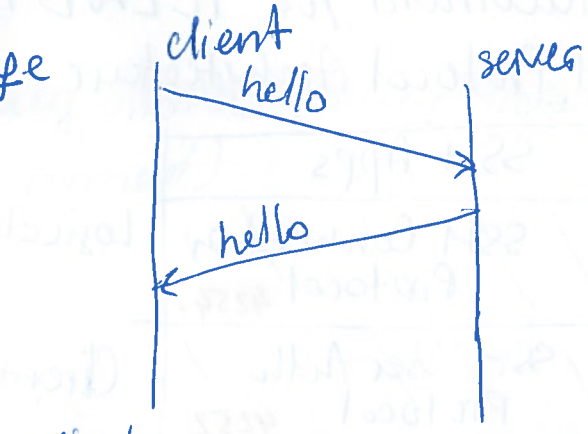
Transport Layer Security

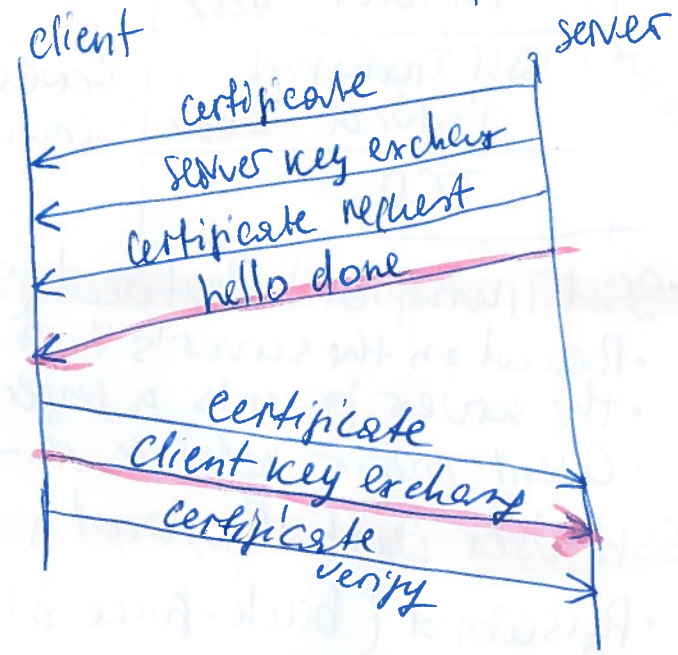
TLS (upgrade from SSL)

- Certificates
  - create certificate for your website and ask CA to sign it
    - Chain of trust
    - Let's Encrypt
  - used in HTTPS
  - Certificate Transparency Logs (CT Logs) (after attack)

TLS Phases

- Peer negotiation
  - client and server exchange
    - TLS version
    - Session ID
    - Random
    - Cipher suite
    - Compression method
- Key exchange ss Auth
  - if asked server sends its auth ss keys
  - if asked client sends its auth ss keys





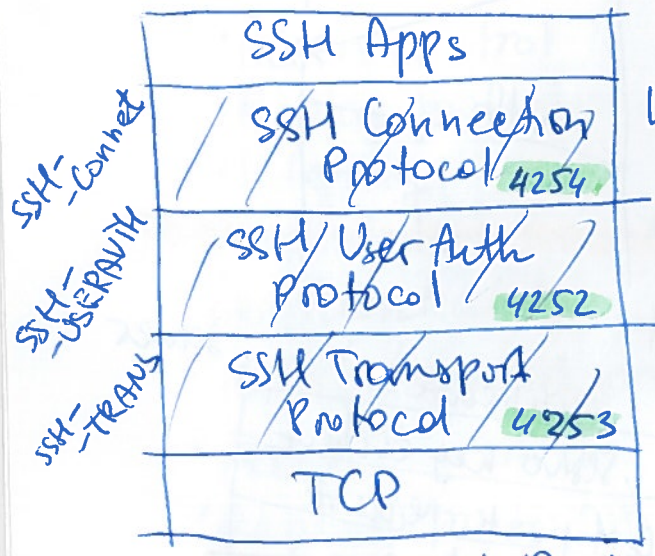
- Data exchange

TLS 1.3

Only forward secrecy ciphers → ↑security, ↓ supported ciphers
Ability to detect a downgrade → ↑ security
Support for 1RTT handshake → ↑ performance than TLS 1.2
Initial support for 0-RTT handshake →

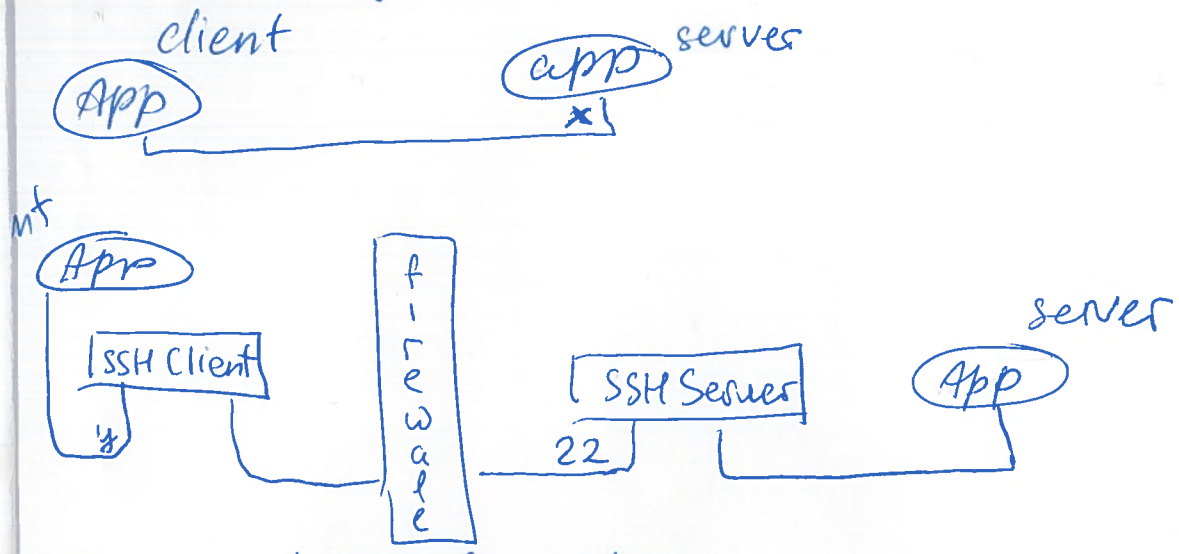
- QUIC
  - UDP-based protocol with security-by-default
  - ~~TCP+TLS+HTTP~~ UDP+TLS+HTTP/2
  - All connections encrypted and authenticated
  - Connection setup in 1RTT
  - gQuic — purely for HTTP/2
  - IETF QUIC — generic transport protocol (HTTP/3)
- Secure Shell (SSH) — port 22
  - replacement for TELNET, rlogin, rsh
  - SSH Protocol Architecture

RFC 4251



- SSH Transport Protocol (server Auth)
  - Based on the server's host key
  - the server presents a fingerprint of this key on every connection
  - Client must validate it → if not then warning
- SSH User Auth Protocol
  - Password (brute-force attacks)
  - Public key (automated logins; more secure)
- SSH Connection Protocol (Multiplexing protocol)
  - Opens and closes "channels"
  - multiple "channels" mapped on a single SSH Transport Connection
  - Each app uses 1+ dedicated channels
  - Std channel types:
    - shell
    - direct-tcpip
    - forwarded-tcpip

Port Forwarding



Application Layer Security

PGP/GPG (Pretty Good Privacy / GNU Privacy Guard)

- RFC 4880 (OpenGPG msg format)
- Allows to sign and encrypt:
  - email
  - files
  - git commits
- uses the private/public key mechanism