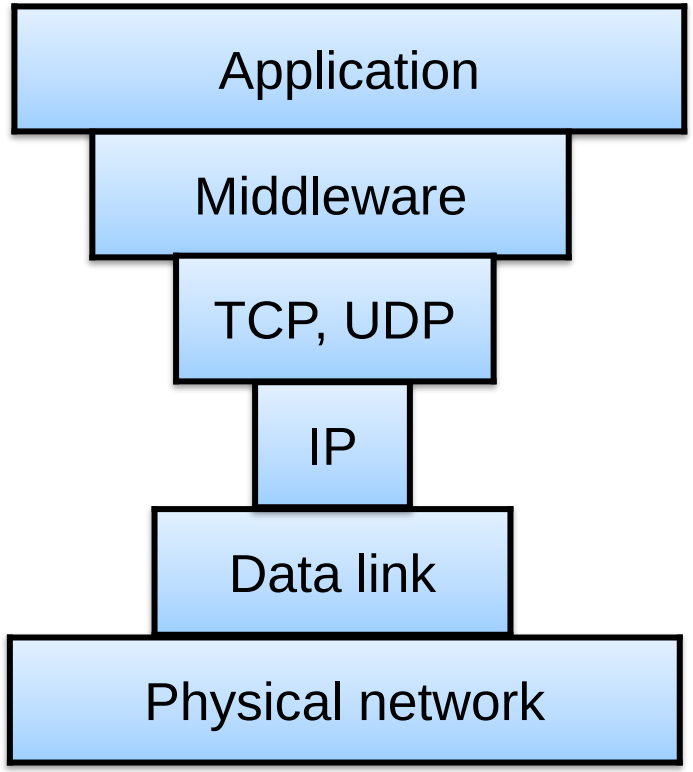
Aalto University

Network Security:
Security and the network protocol stack

Tuomas Aura, Aalto University

CS-E4300 Network security

Protocol Stack and Security



- Which layer in the protocol stack should implement security mechanisms, esp. encryption and authentication?

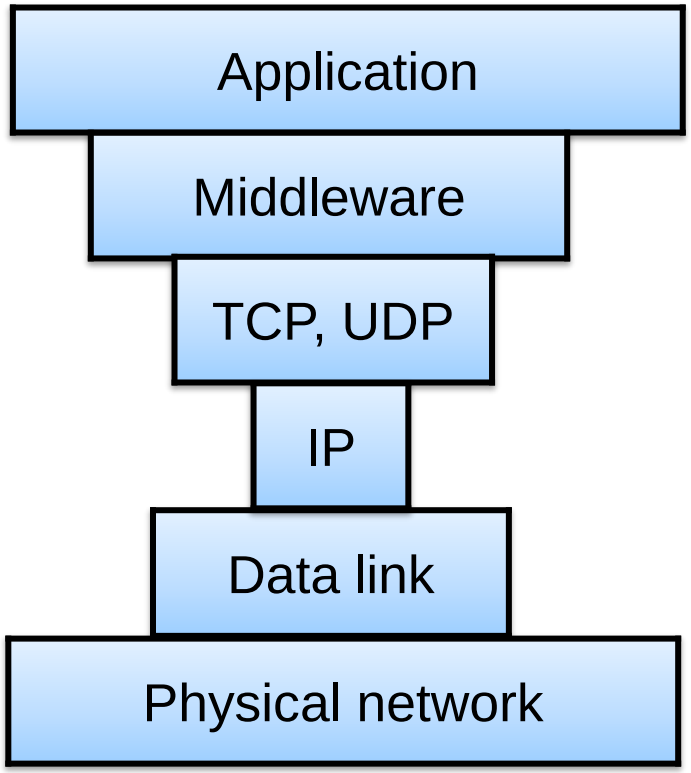
Which layer security?

- Reasons to implement cryptographic security in lower layers:
  - Security provided by physical, link or network layer is a service to the higher layers
  - Lower-layer security protects all higher-layer data: all connections, both payload and signaling or metadata
  - Security in lower layers is transparent to higher layers. No changes to applications needed
  - Lower-layer security protects the lower layer, too

Which layer security?

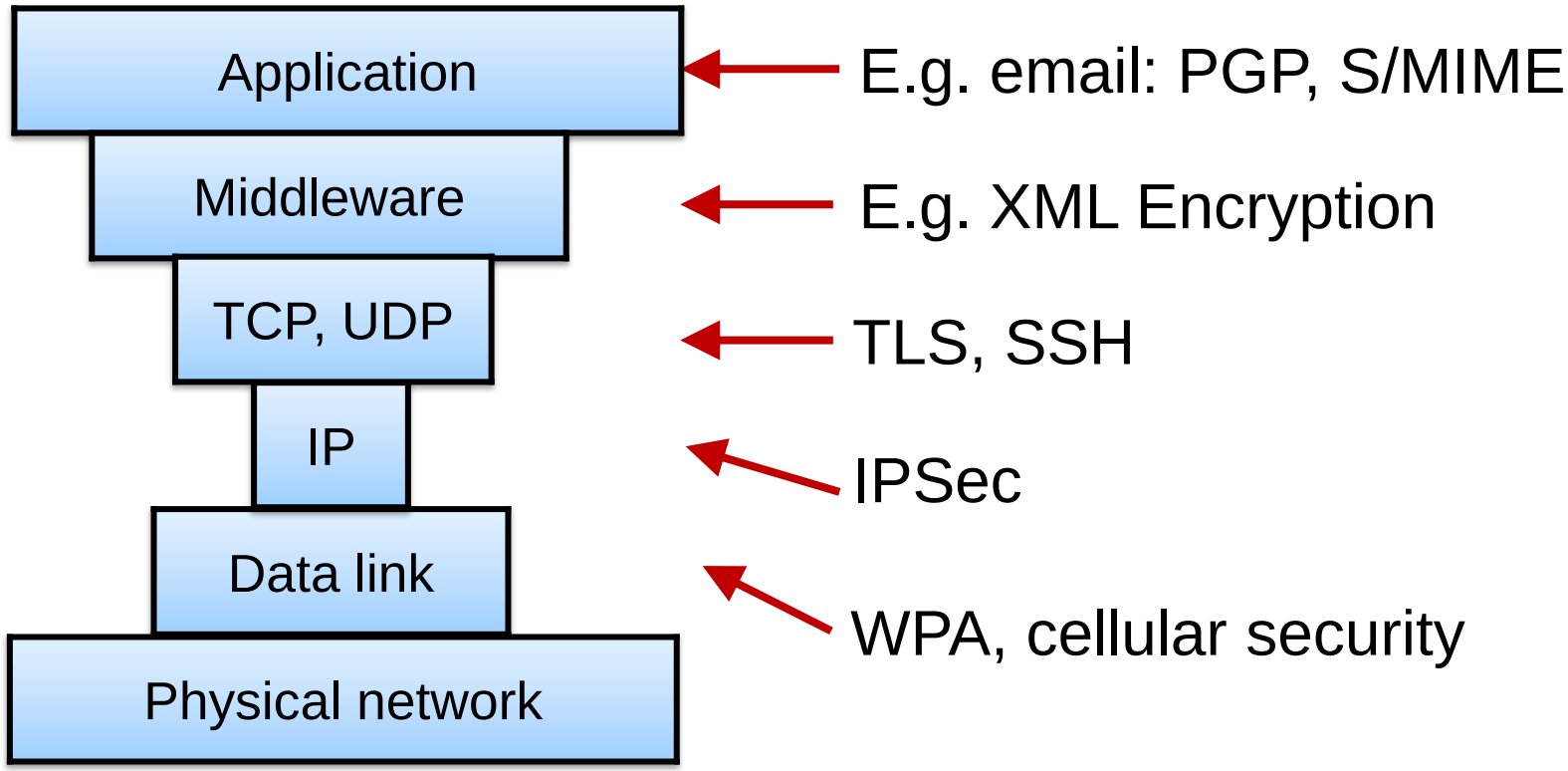
- Reason to implement security in higher layers:
  - Security implemented in the application or middleware will fit exactly to the application requirements
  - Authentication of lower-layer identifiers may not be meaningful to higher layers
  - Application developers can deploy security mechanism faster

IP layer



- Hourglass-shaped TCP/IP protocol stack → any service should be in the IP layer:
  - Implement only once (or twice for IPv4+IPv6)
  - Works over any data link layer
  - Works for any application

Protocol Stack and Security



- Security solutions exist for every protocol layer
- Layers have different security and performance trade-offs, trust relations and endpoint identifiers

End-to-end security

- Security should be implemented between the endpoints of communication.
  - All intermediaries are part of the untrusted network
- End-to-end security only depends on the end nodes
  - Hop-by-hop (link-layer) security assumes trusted and secure intermediate nodes
- End-to-end mechanism are independent of the link technology
  - Link-layer security is different for each link type
- Confidentiality and authentication are usually user or application requirements
  - Network or link layer does not know application-level requirements
- Link and network layer infrastructure and signalling need protection, too

Host as endpoint

- Traditionally, host i.e. computer is the security endpoint
  - OS is trusted to isolate apps running as processes and their connections from other processes
  - OS must be trusted because it has access to software memory and controls execution
- Increased communication inside the host:
  - Inter-process communication, VMs, containers, microservices, APIs
  - More nuanced trust models: web apps, mobile and cloud computing
  - Trusted execution environments isolate software form the host