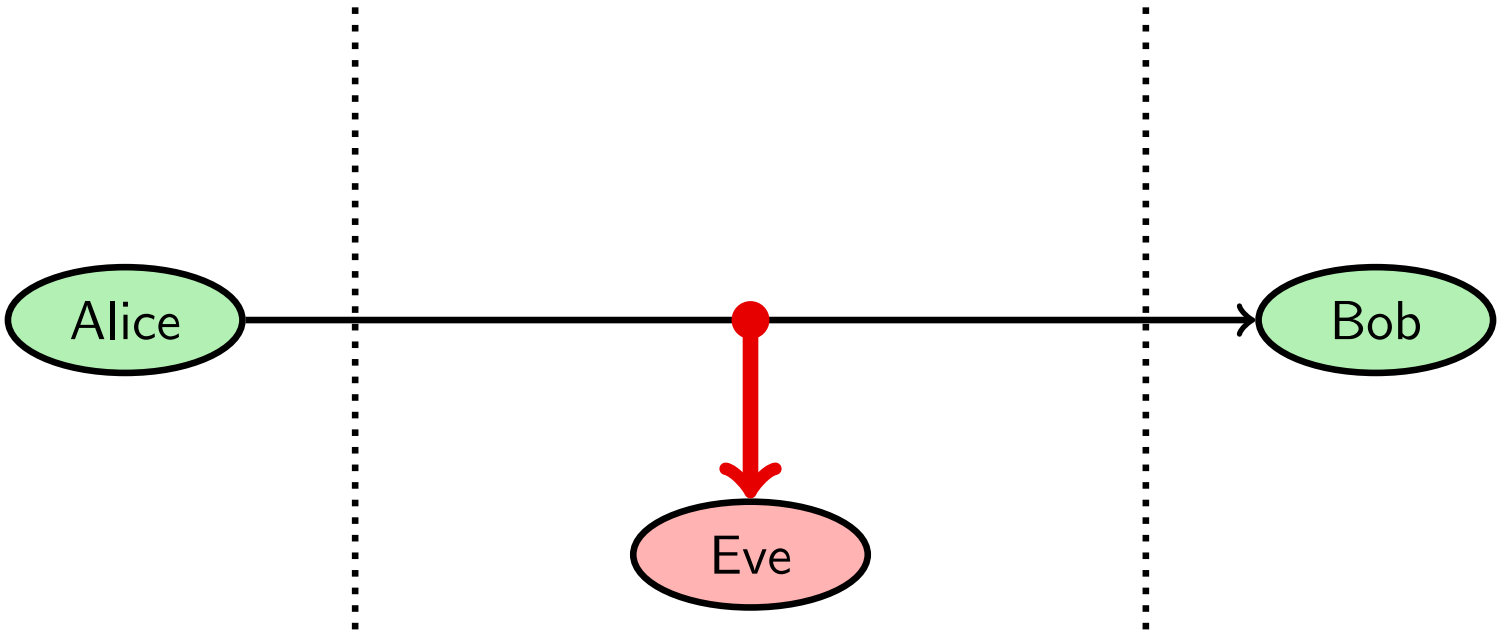
Internetworking Lecture 10

Communications and network security

LINKÖPING UNIVERSITY

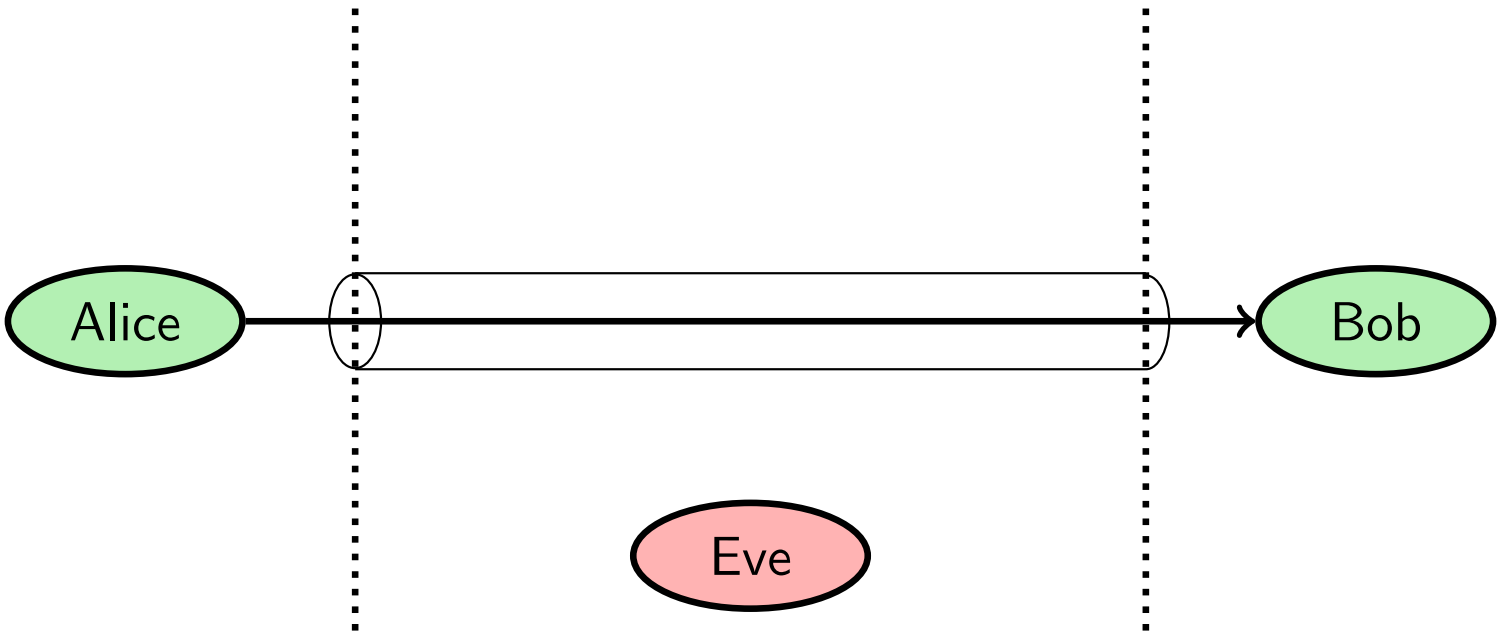
Communication and network security: Threat model

- Passive attacks: Eavesdropping, Wiretapping, Sniffing, and Traffic analysis



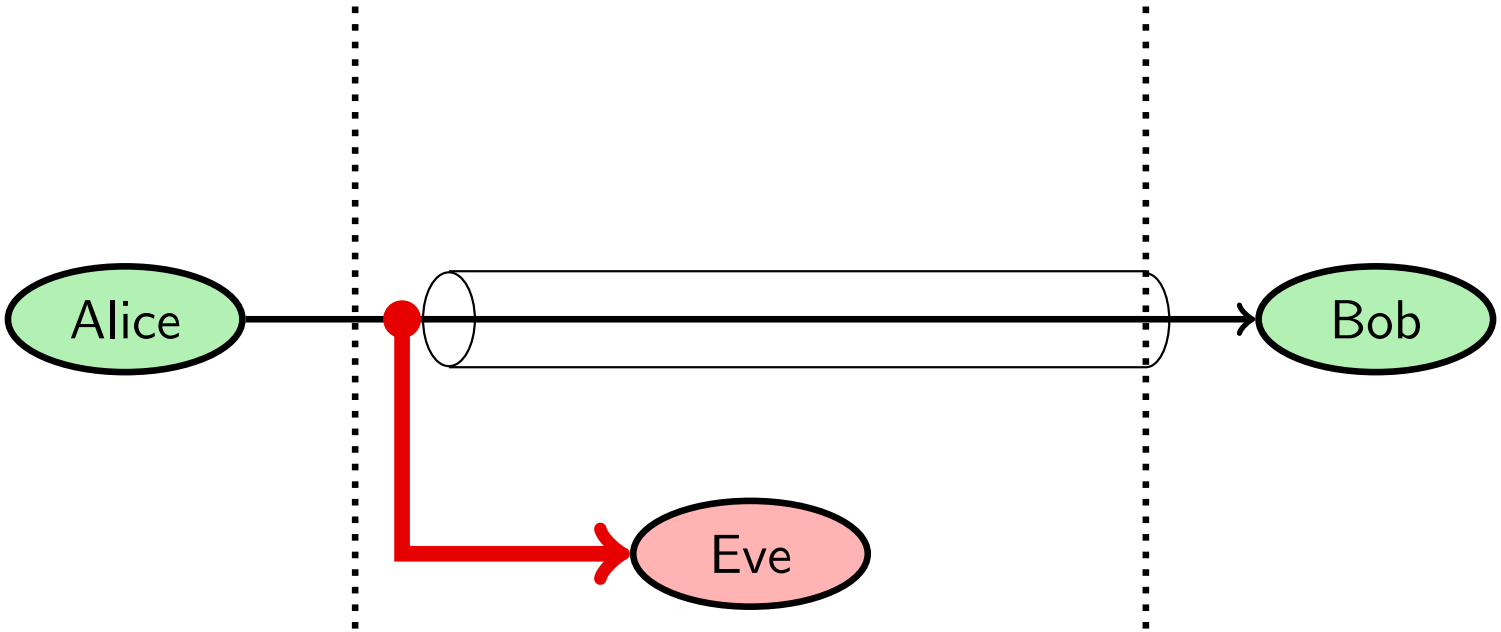
Communication security: Secure tunnels

- Typically provide Confidentiality, data Integrity, and data origin authentication
- End points may be machines or services on the local computer



Communication security: Secure tunnels

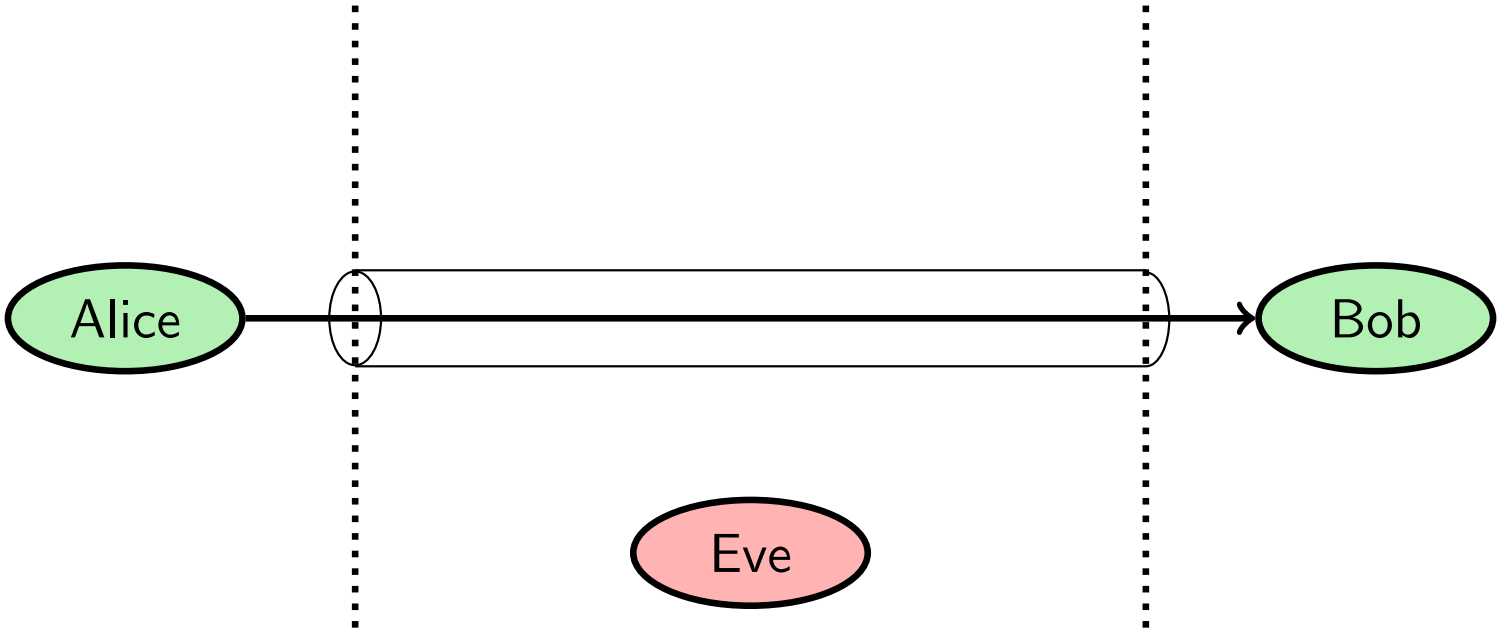
- Typically provide Confidentiality, data Integrity, and data origin authentication
- End points may be machines or services on the local computer
- The placement is important to achieve security



Communication security: Secure tunnels

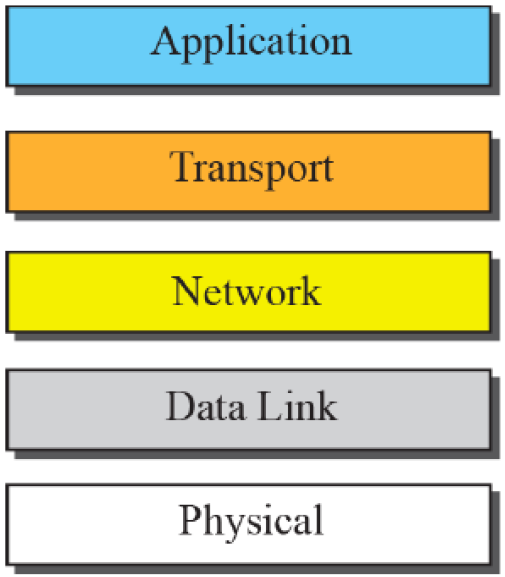
Steps to set up a tunnel

1. Authenticated key establishment ($\rightarrow$asymmetric key)
2. Key derivation ($\rightarrow$symmetric key)
3. Traffic protection through symmetric cryptography



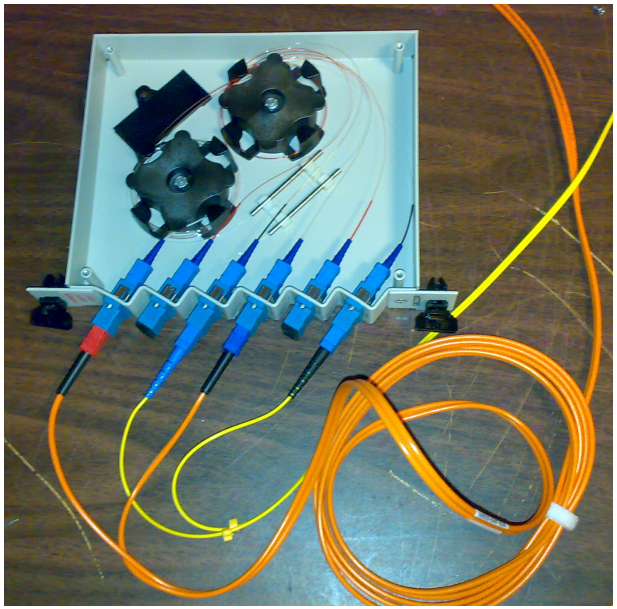
Security in the IETF layers of network protocols

- Security services at the top can be tailored for specific applications, but each application then needs a separate service
- Security services at the bottom can protect the upper layers transparently, but may not meet all requirements of specific applications



Physical layer: wiretapping

- Wiretapping is simple technology
- Direct electrical connection
- Fibre-optic beam splitter
- EM radiation
  - Radio
  - Landlines
  - Endpoint equipment
- Audio

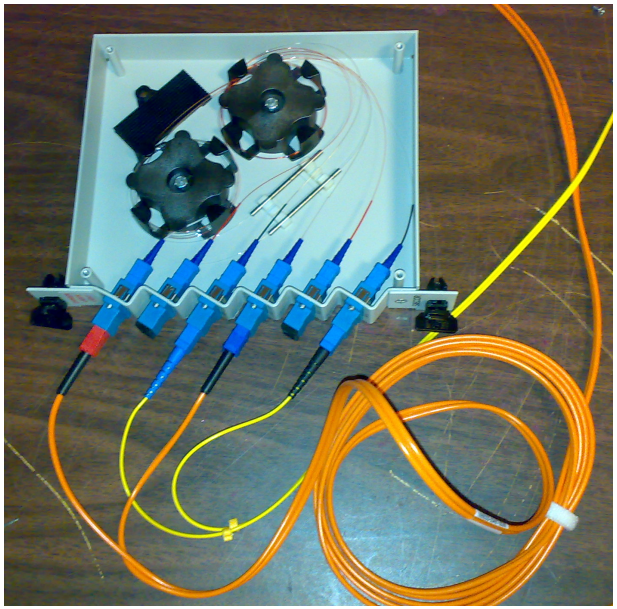


"Fiber optic tap" by Roens, GFDL

Physical layer: wiretapping countermeasures example

- A fibre-optic beam splitter will change intensity
- Also, the mode pattern of the laser light will change

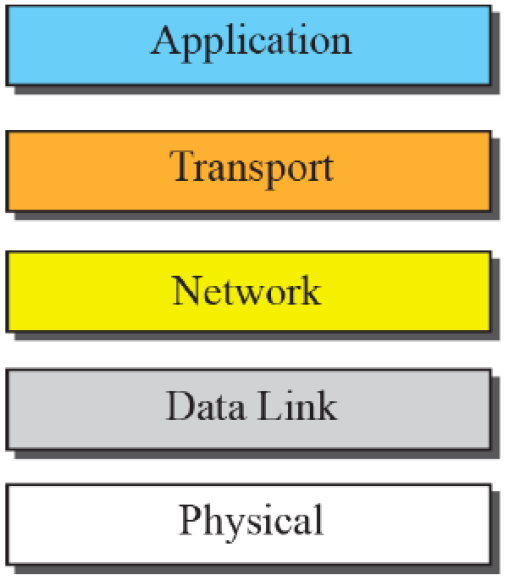
- EM radiation from equipment can be shielded
- 3mm stainless steel around your laptop will do the job



"Fiber optic tap" by Roens, GFDL

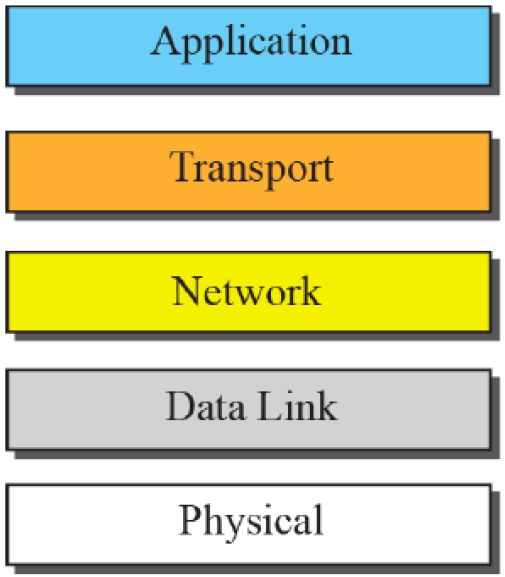
Link layer: MAC spoofing

- The hardware address can be changed from software
- Masquerade as different machine; avoid access control by MAC address
- There are reasonably legitimate uses, e.g., avoiding ISP or DRM MAC locking
- Two machines with the same MAC address will both receive all data
- A variation is “promiscuous mode”: disable the MAC filter



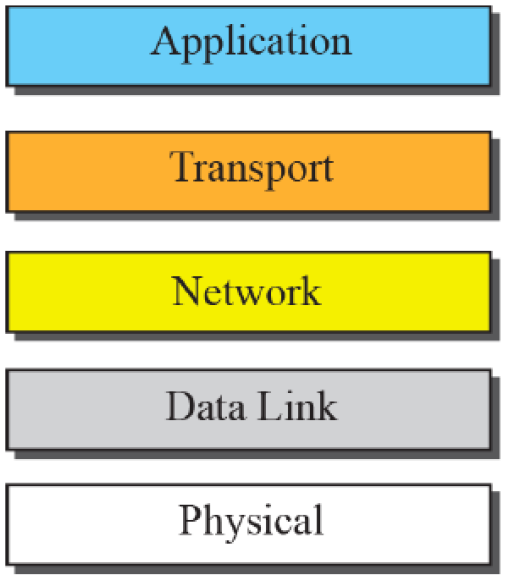
Link layer: ARP spoofing (poisoning)

- ARP is stateless: when ARP requests are broadcast, the host will not remember this
- All ARP replies are cached, even ones never requested
- These can change ("poison") the ARP table
- The packets can be forwarded to the real recipient so that the attack is not noticed



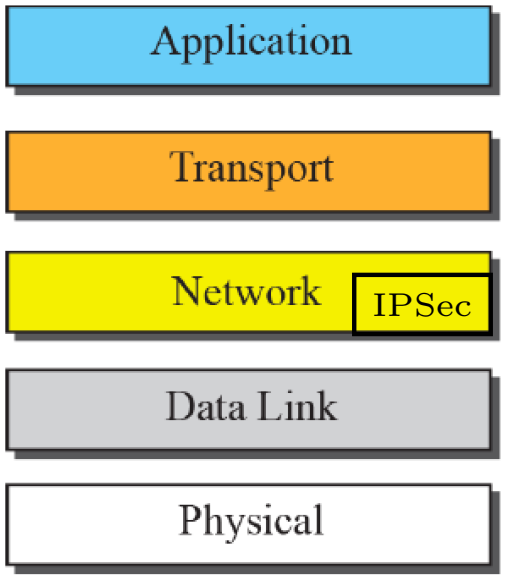
Link layer: MAC and ARP spoofing countermeasures

- Check ARP table for duplicate IPs
- Sticky ARP: router is configured to not accept changes in MAC address (this is high maintenance)
- Notify sysadmin of ARP changes
- MAC-aware switches (i.e., do not switch on IP only)
- Promiscuous mode can be detected too: send a ICMP ping to target IP with wrong MAC address; if there is an answer, the target's MAC filter is off



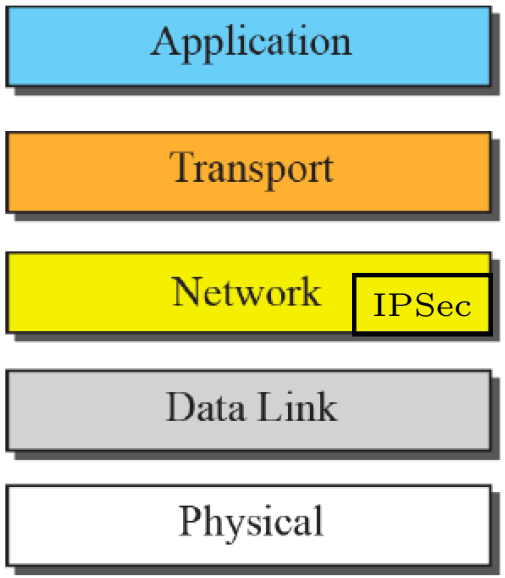
IPsec, IP protocol security

- Security architecture is in RFC 4301
- Optional for IPv4, mandatory for IPv6
- Two major security mechanisms: Authentication Header and Encapsulating Security Payload
- Authentication Header does not give Confidentiality; it was used to avoid export restrictions in the 90s



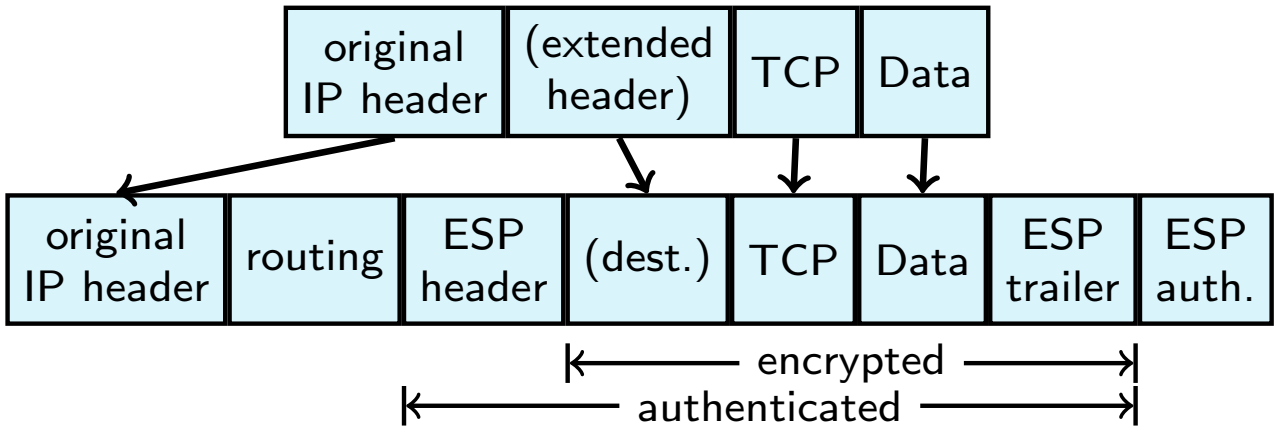
IPsec, Encapsulating Security Payloads

- ESP provides Confidentiality, data Integrity, data origin authentication, some replay protection, and limited traffic flow Confidentiality
- ESP can be run in two modes: Transport mode and Tunnel mode
- For transport mode, both nodes need to be IPsec-aware
- Tunnel mode, on the other hand, is transparent: IP-within-IPsec

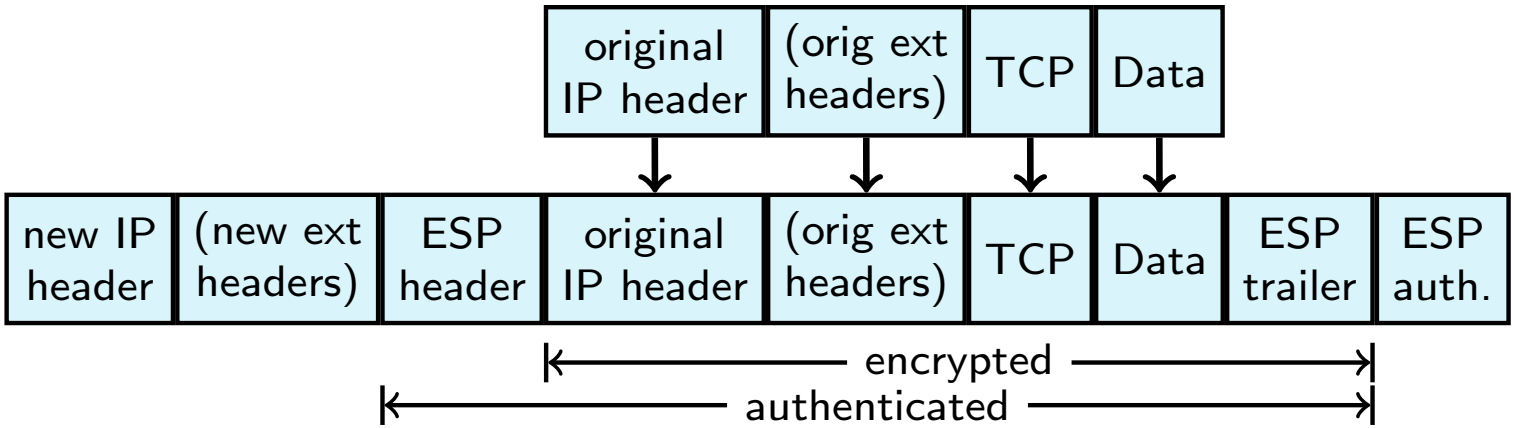


IPsec, Encapsulating Security Payloads

Transport mode

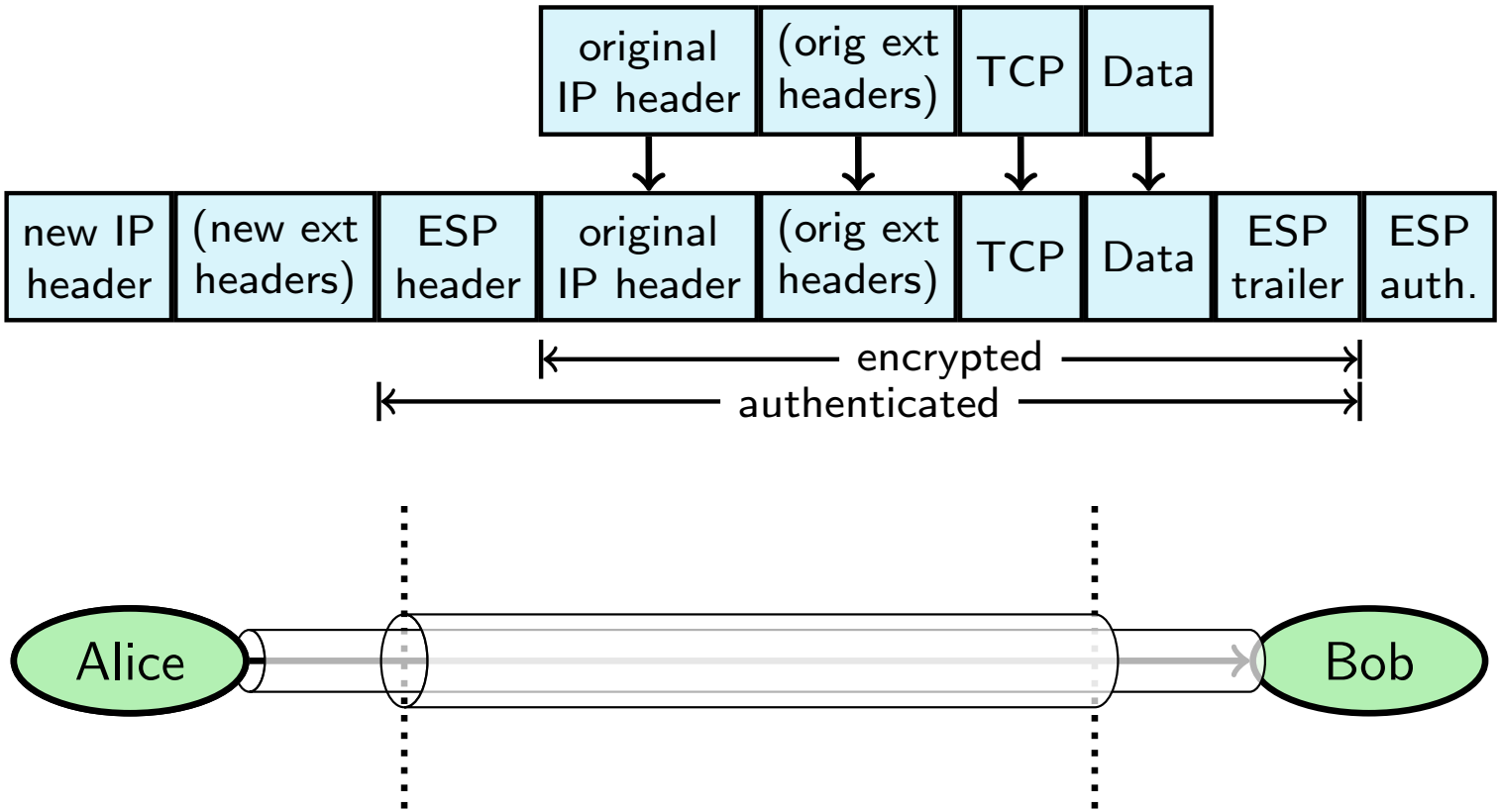


Tunnel mode



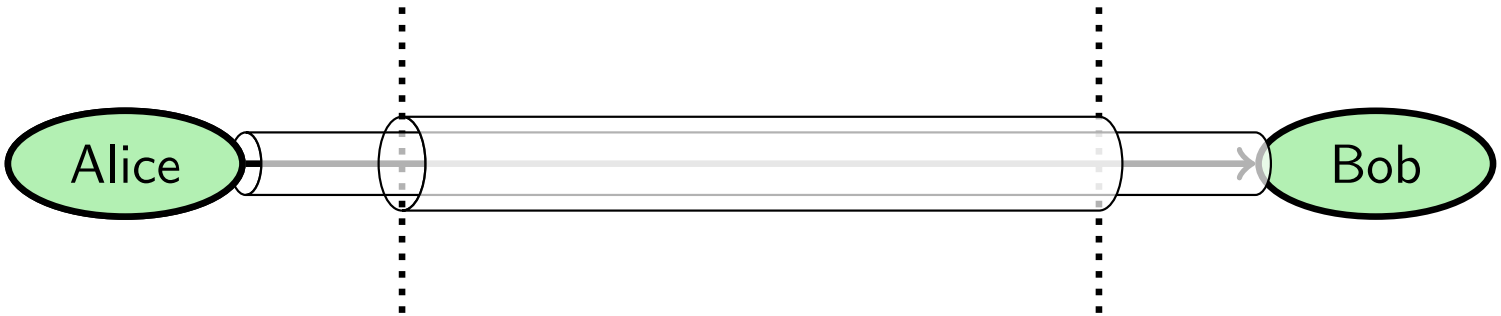
IPsec, Encapsulating Security Payloads

Tunnel mode



ESP headers, and Security Association

- The ESP header contains the ID of a Security Association (SA), and a sequence number
- The SA is a common state for communication from one node to another (so IPsec is not stateless)
- Usually created in pairs
- State includes source, destination, security protocol, cryptographic algorithms and keys, key lifetimes, IVs, sequence numbers and anti-replay windows
- SAs can be combined in multiple nesting levels

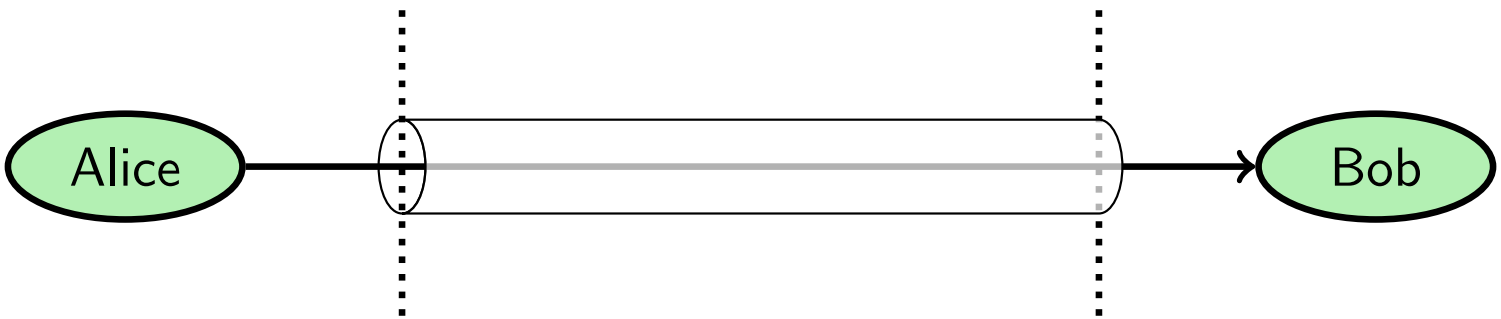


Establishing SAs through Internet Key Exchange (IKE, ISAKMP)

- Internet Key Exchange (v2) is in RFC 4306
- Uses two phases
- First phase:
  - establish a shared session key through public key techniques
  - a pre-shared secret or trusted public keys are needed to authenticate the nodes

Establishing SAs through Internet Key Exchange (IKE, ISAKMP)

- Internet Key Exchange (v2) is in RFC 4306
- Uses two phases
- First phase:
  - establish a shared session key through public key techniques
  - a pre-shared secret or trusted public keys are needed to authenticate the nodes
- Second phase establishes IPsec Security Association(s) in an encrypted session, that uses the key formed in the first phase



IPsec problems: DOS

- In the first phase, an attacker might send bogus responses
- The second phase would never complete
- The bogus keys will need storage

- Mitigation: use several responses
- Storing all responses for each connection will waste resources
- Solution: store them until negotiation completes
- Use the resulting SA

IPsec problems: DOS II

- Another DOS possibility is to flood a node with initiation requests from forged IPs
- This would make the node waste resources on calculating responses and storing session data

- Solution to this is to do no state storage first, but respond with a "cookie", expecting it to return from the initiator IP
- If the IP is bogus, the DOSer does not get the cookie, and cannot return it
- Cookie format is not standardized, the RFC suggests to hash IP, the request nonce, and a local secret that is changed regularly

IPsec and NAT

- Network Address Translation was invented to cope with the lack of available IPv4 addresses
- Nowadays it is sometimes seen as a security feature that internal IPs are not directly addressable
- This creates problems, especially for IKE since authentication is by IP address
- NAT-T ("IPsec passthrough") solves this by adding an extra header, triggering rewriting rules at the recipient after packet authenticity has been checked

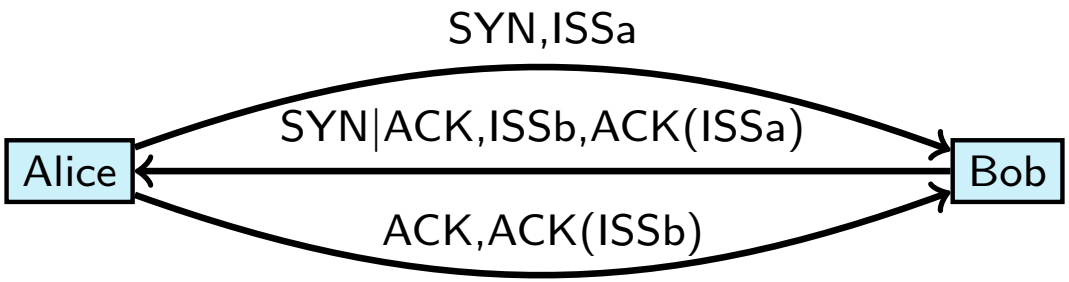
Pros and cons of IPsec

- IPsec provides security transparently
- Upper layers need not be aware that lower layers are more complicated to provide security

- Cannot be tuned for specific applications
- IPsec provides host-to-host (gateway-to-gateway) security, not user-to-user or application-to-application security
- IP is stateless and unreliable by construction, but IPsec is stateful
- IPsec packets need to be ordered, while IP should not be concerned with packet order or dropped packets

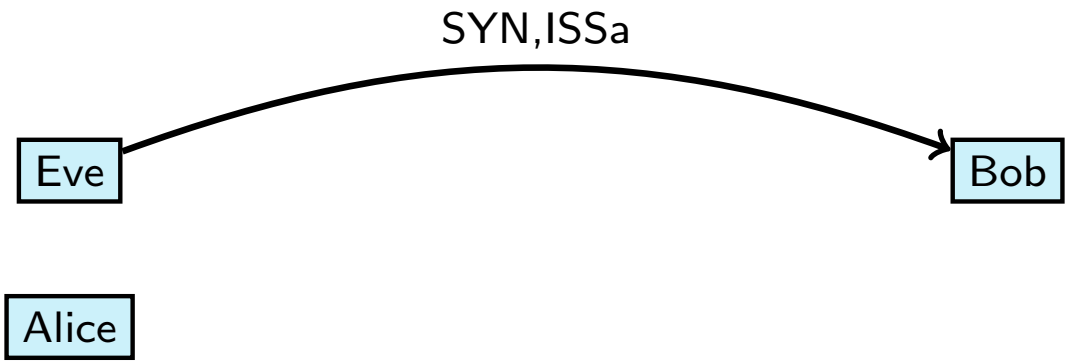
TCP session hijacking

- The below exchange starts a TCP session
- The acknowledgements are simple: ACK(ISSa)=ISSa+1



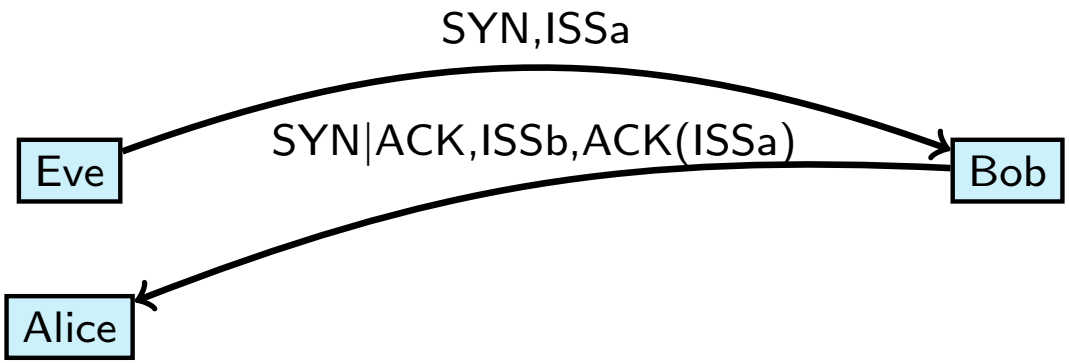
TCP session hijacking

- The below exchange starts a TCP session
- The acknowledgements are simple: ACK(ISSa)=ISSa+1
- Eve sends SYN,ISSa with Alice's response address



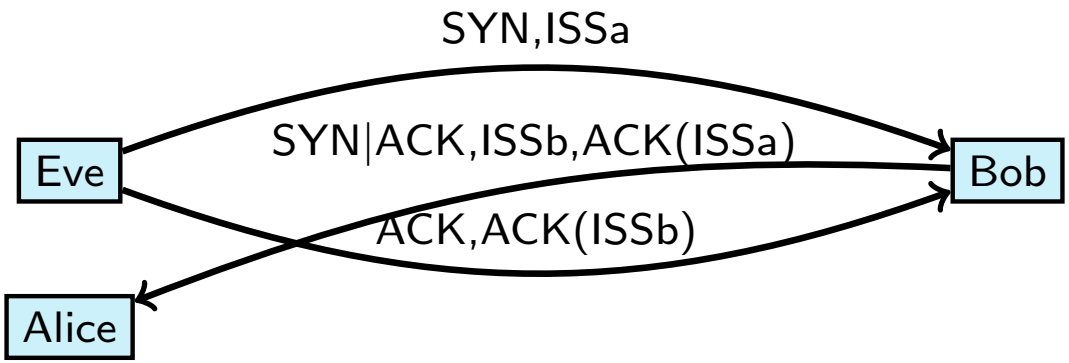
TCP session hijacking

- The below exchange starts a TCP session
- The acknowledgements are simple: ACK(ISSa)=ISSa+1
- Eve sends SYN,ISSa with Alice's response address
- She doesn't see the response, but ...



TCP session hijacking

- The below exchange starts a TCP session
- The acknowledgements are simple: ACK(ISSa)=ISSa+1
- Eve sends SYN,ISSa with Alice's response address
- She doesn't see the response, but ...
- If Eve can guess ISSb, she can hijack the session

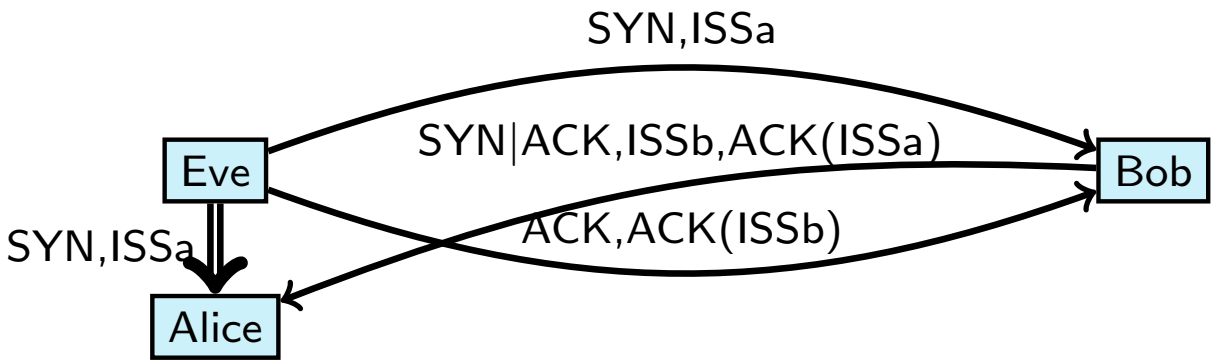


TCP session hijacking

- Eve has established a (blind) session through session hijacking
- Certain protocols use no more authentication than this
- For these, Eve can use Alice's credentials at Bob
- Solution: firewall, or don't use services with address-based authentication

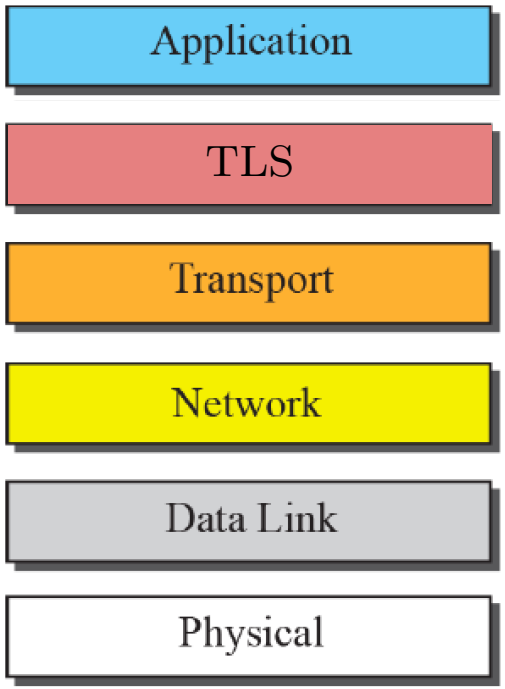
TCP SYN flooding

- To stop Alice from tearing down the faulty (to Alice) session, Eve can mount a SYN flood attack against Alice
- This is to exhaust Alice's resources
- An Availability attack here results in possible Integrity damage



SSL/TLS

- Placed between “normal” TCP and application
- Handshake phase uses asymmetric encryption and certificates to exchange the session key
- The server (but not the client) is authenticated (by its certificate)
- Session key is for a symmetric algorithm
- Many different algorithms can be used, the set is not standardized

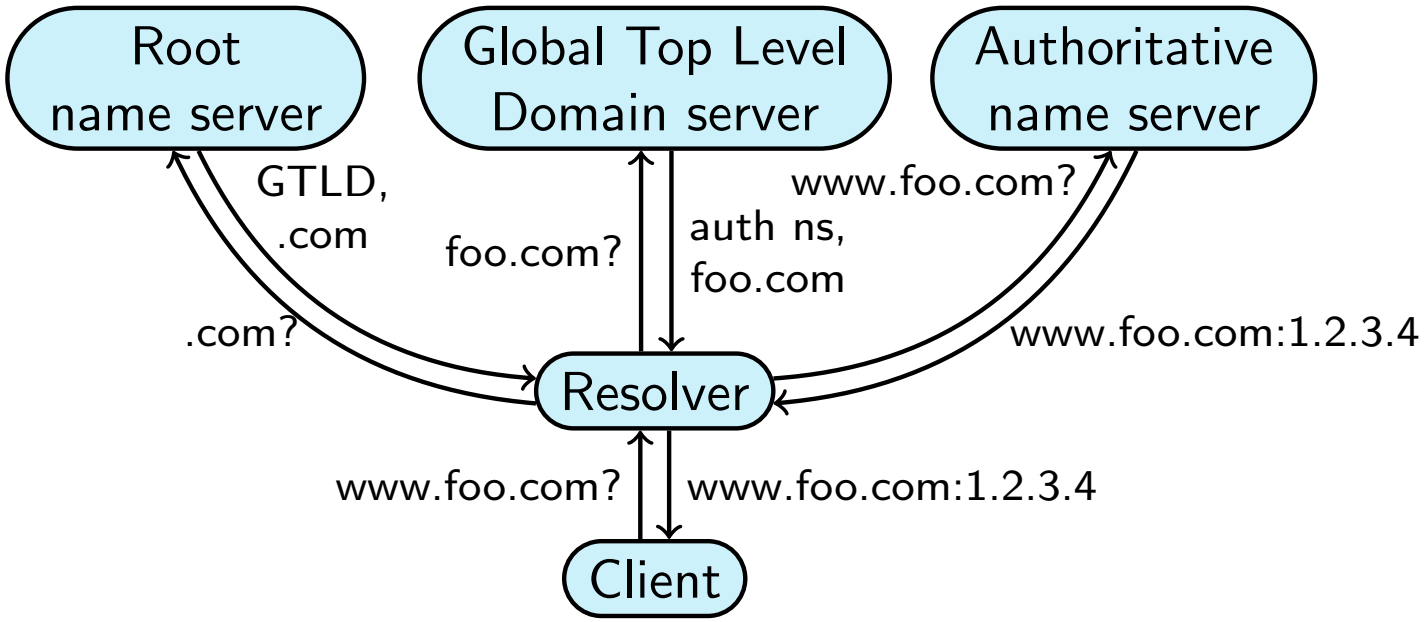


Pros and cons of TLS/SSL

- TLS provides user-to-user or application-to-application security
- Useful even in specific applications
- Sits above TCP, so can use benefits of stateful TCP packet handling

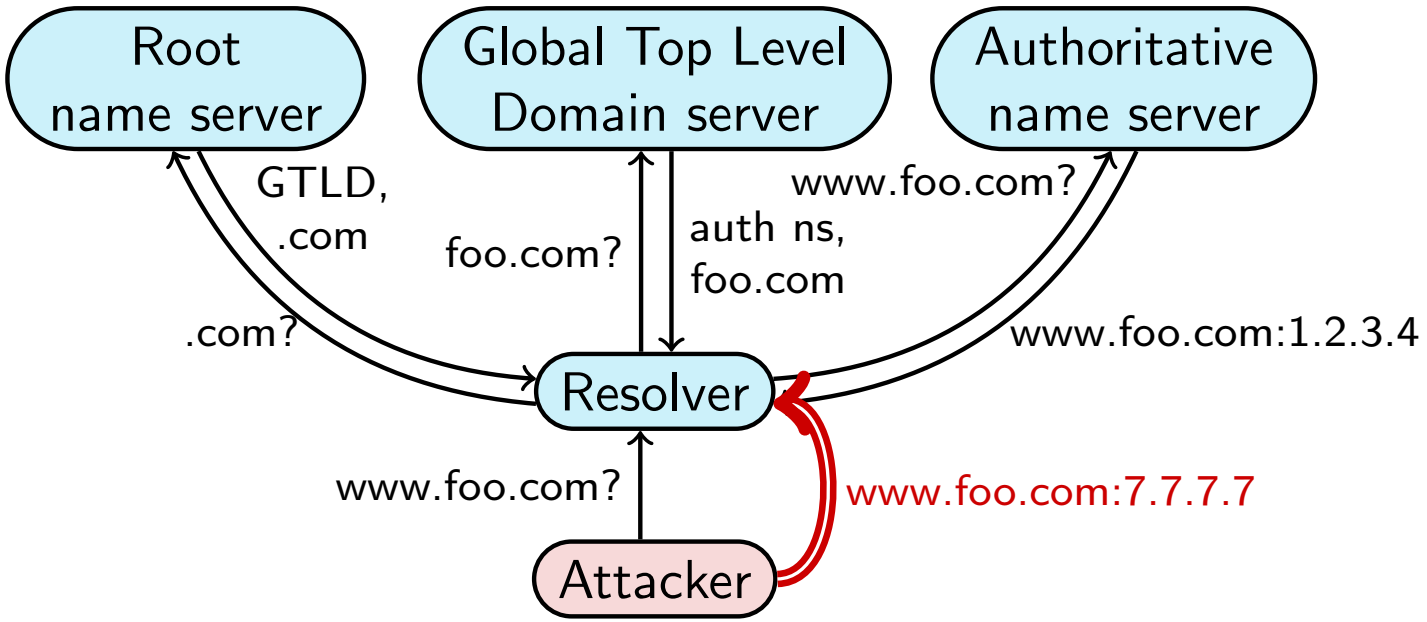
- Applications need to be security-aware
- They must explicitly ask for security
- Changes are not that large, use TLS connect instead of TCP connect
- The security state sits closer to the application/user, and may be more vulnerable

Domain Name System, DNS



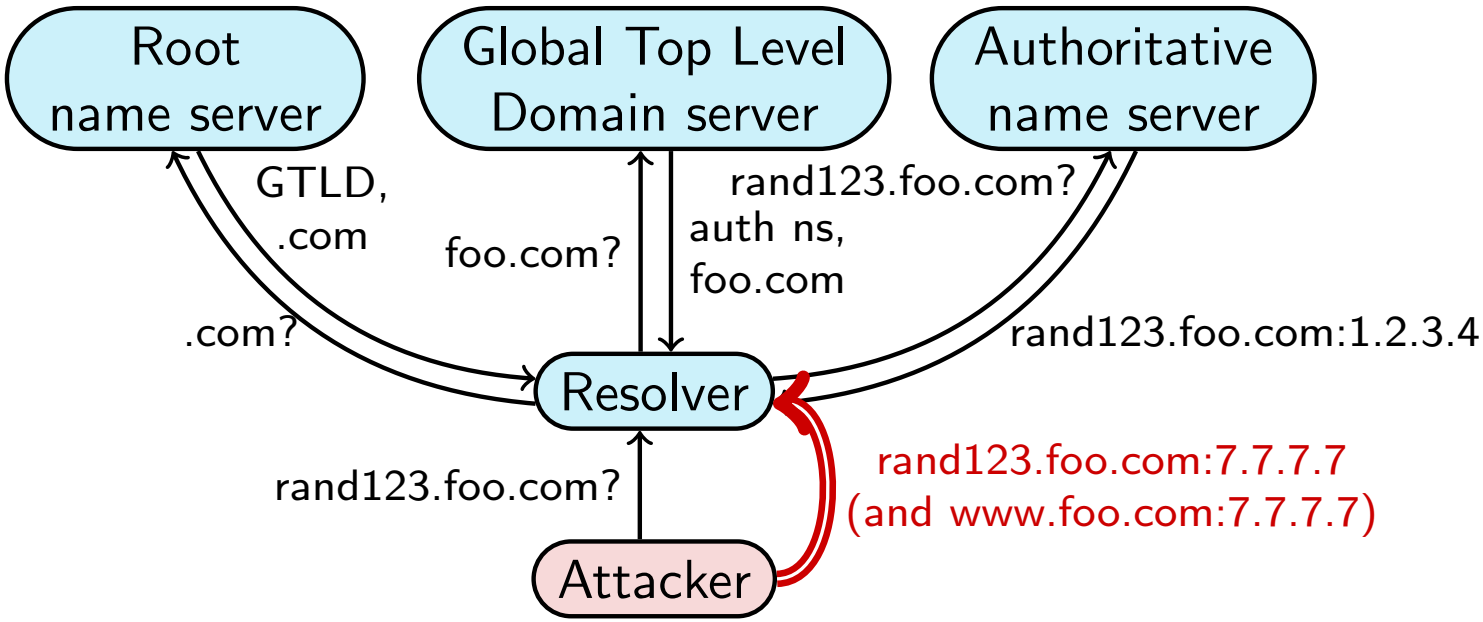
- DNS uses "Lightweight authentication", a 16-bit QID and a UDP response port that the answering server should use

DNS Cache poisoning



- Attacker asks for IP for target, then immediately floods the resolver with guessed QIDs at guessed UDP ports
- If successful, the attacker gets to decide Time To Live for the record

Dan Kaminsky's attack



- Attacker asks for IP for random host in target domain, then immediately floods the resolver with guessed QIDs at guessed UDP ports
- The attacker can include Additional Resource Records in the spoofed responses
- He can now try again without waiting for TTL expiry

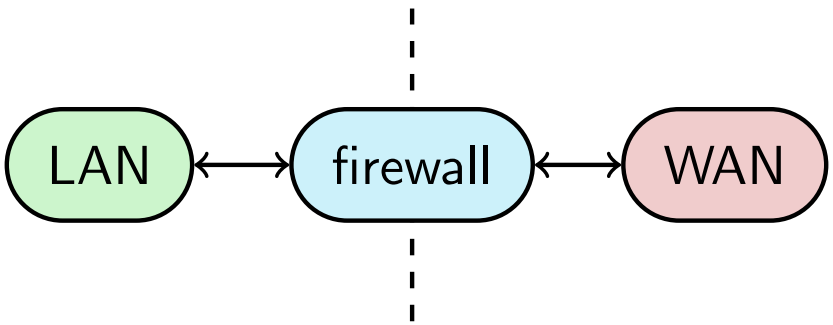
DNSSec

- DNS Security Extensions uses digital signatures to protect DNS records
- The DNS root is the trusted party
- The signature chain is built from the DNS root, through the TLD, and down to the current subdomain

- Not so easy to design a backward-compatible standard that can scale to the size of the Internet
- Worries of "zone enumeration", many feel their DNS info is confidential
- Disagreement among implementers over who should own the top-level domain root keys
- DNSSEC deployment is thought to be complex

Firewalls

- Main function: Filter traffic according to IP address and TCP port
- Do Network Address Translation to hide internal network
- Application proxies can do more, like filtering email for viruses and spam



Firewalls: packet filters

- Based on IP address and port (the Internet and Transport layers)
- Typical rules specify source and destination IP and TCP/UDP port number
- . . . and of course ALLOW or DENY
- In its simplest incarnation only allows static rules, say FTP to certain servers, or HTTP to others

Firewalls: stateful packet filters

- These can change the filtering rules depending on the session history
- For example, can handle the TCP SYN, SYN|ACK, ACK
- Another example are more complicated patterns like an inbound HTTP response from a server after an outbound HTTP request
- Filtering is limited by what is available in the packet headers

Firewalls: policy types

- There are two possible choices: Permissive policies or restrictive policies
  - Permissive policies (blacklisting) allow all connections except those known to be dangerous
  - Restrictive policies (whitelisting) deny all connections except those known to be secure (and useful)
- The latter is the more secure option
  - if you forget to allow something that people need, you will hear about it
  - if you forget to block a known attack path, you might lose your job

Firewalls: application proxies

- A proxy implements both server and client roles for a given protocol
- When a client connects to the proxy, the proxy checks if the request should be allowed
- If so, it acts like a client and connects to the destination server
- Responses come back to the proxy and also gets filtered before being passed on to the client

- An example is email virus filtering
- Proxies can be seen as performing controlled invocation

Firewalls: application proxies

- Application proxies typically run on hardened PCs
- These give a high level of control on filtering, so are secure from that point of view
- But they are more work-intensive to maintain
- Configuration is more complicated
- Often, one server is needed per service

Firewalls: Perimeter networks, or DMZ

- DMZ stands for De-Militarised Zone
- Some services must be accessible from the outside and from the inside
- One way to solve this is to place it on the border, with different access rules from the other machines on the protected network
- Sometimes this is done through two firewalls, an inner and an outer
- Some firewalls have a special interface for the DMZ clients

Firewall problem: protocol tunneling

- Since ports other than HTTP are often blocked, services tend to tunnel through port 80
- Even worse, some tunnel through SSH (or HTTPS), and these cannot be monitored
- Creating a proxy will not be popular, since end-to-end security will be lost

- Some think that separate firewalls soon will be a thing of the past
- The personal firewall will move network security back onto the end system

Mobility

- Mobile networks create a whole new set of problems:
  - How do we authenticate users?
  - How do we authenticate roaming users?
  - How do we hand over sessions from one base station to another?
  - Do we share a new session key, or transfer the old one?
  - Can the user distinguish a false base station from a real one?

Analogue phones

- First generation cellphones provided no confidentiality (some obfuscation)
- Criminals used them to create alibies through call forwarding from land lines
- Authentication was sent through a challenge-response protocol, but in the clear, resulting in charge fraud

GSM politics

- At the time (the eighties, early nineties), there were strong restrictions on use of cryptography
- Law enforcement wanted to be able to perform “wiretaps”
- GSM consortium consisted of mainly national post, phone, and telegraph operators

GSM

- Design goals: good voice quality, cheap end systems, low running costs, international roaming, hand-held devices(!), new services such as SMS
- Security goals: protect against charge fraud, eavesdropping, and track stolen devices (but this was not always implemented)

GSM components

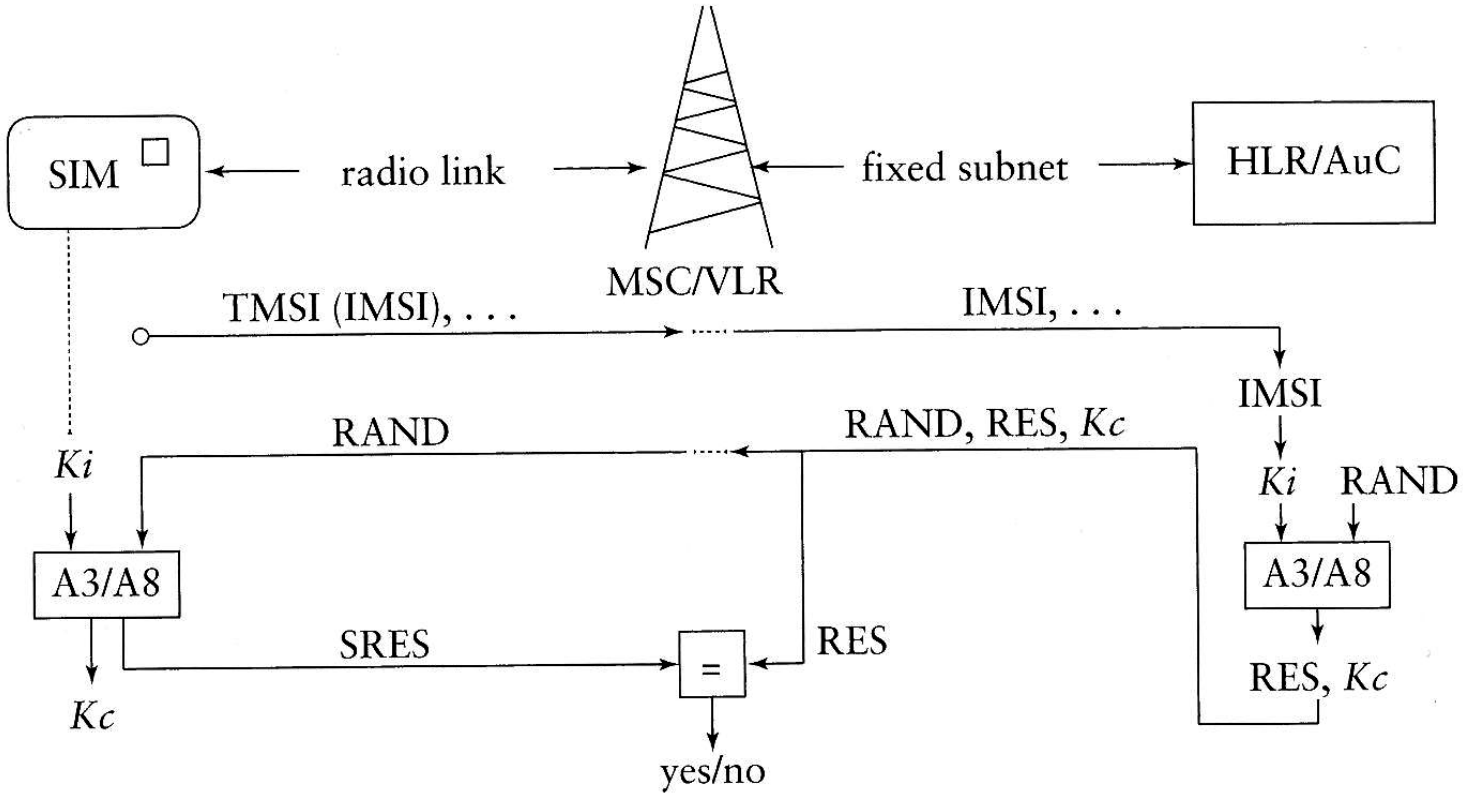
- Each GSM user has a subscription in a "home network"
- Users can use other "visited network"
- A phone consists of "mobile equipment", and a Subscriber Identity Module (SIM) that contains cryptographic hardware, cryptographic keys, and other info

- The network side has base stations and a few different servers that handle user data

GSM IMSI, TMSI

- The identifier for a GSM phone is the International Mobile Subscriber Identity, IMSI
- The phone and home network share a 128-bit authentication key K_i
- To protect user privacy, the IMSI is sent in the clear only in the initial connection to the GSM network, after that a Temporary MSI, TMSI is used
- The TMSI changes when phone moves between networks

GSM session key, Subscriber Identity Authentication



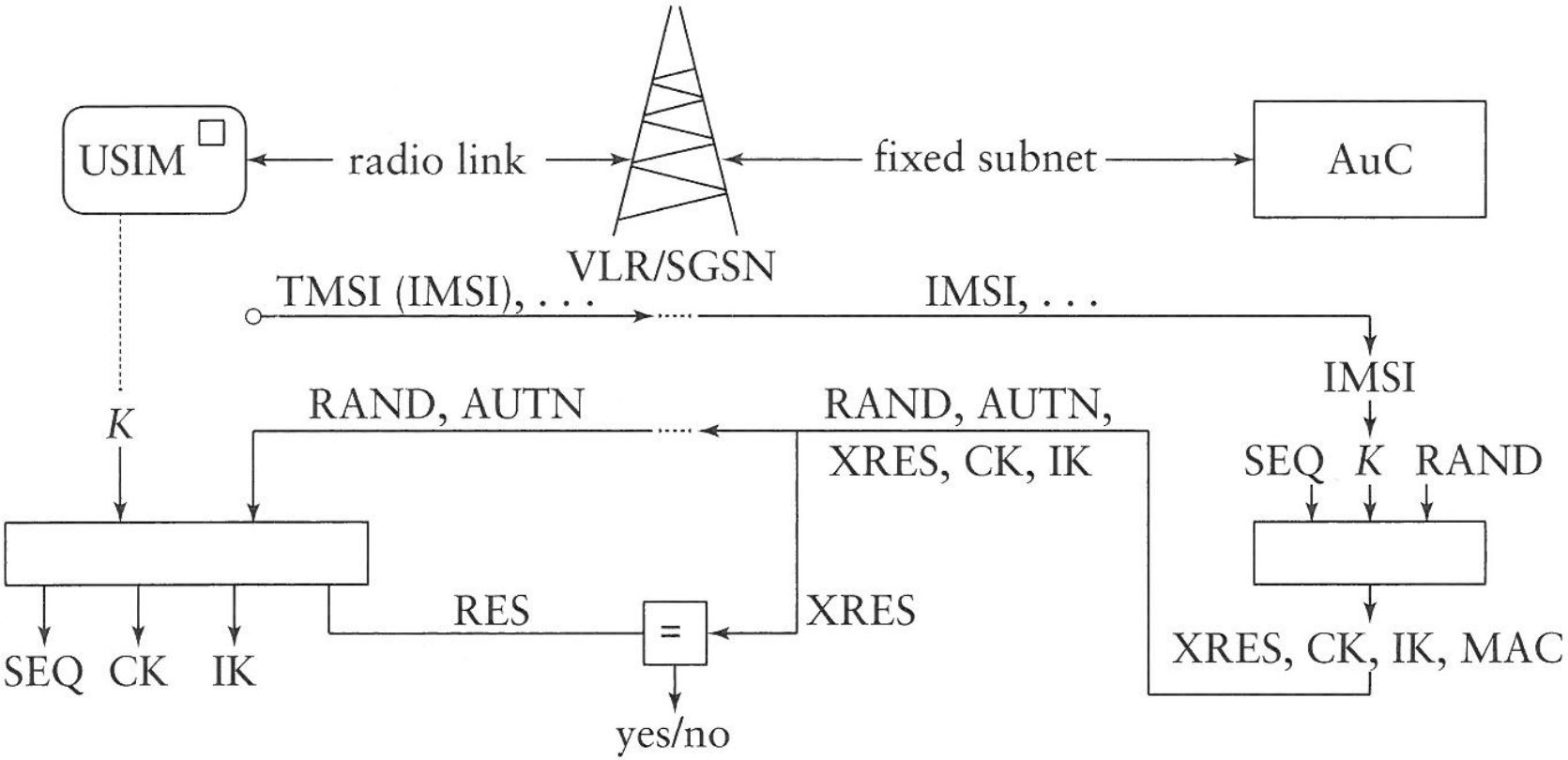
False base stations

- Phone is authenticated, but network is not
- Furthermore, the network can ask the phone for the IMSI, breaking privacy
- And the network can even ask the phone to switch off encryption altogether
- Suddenly, network authentication is needed

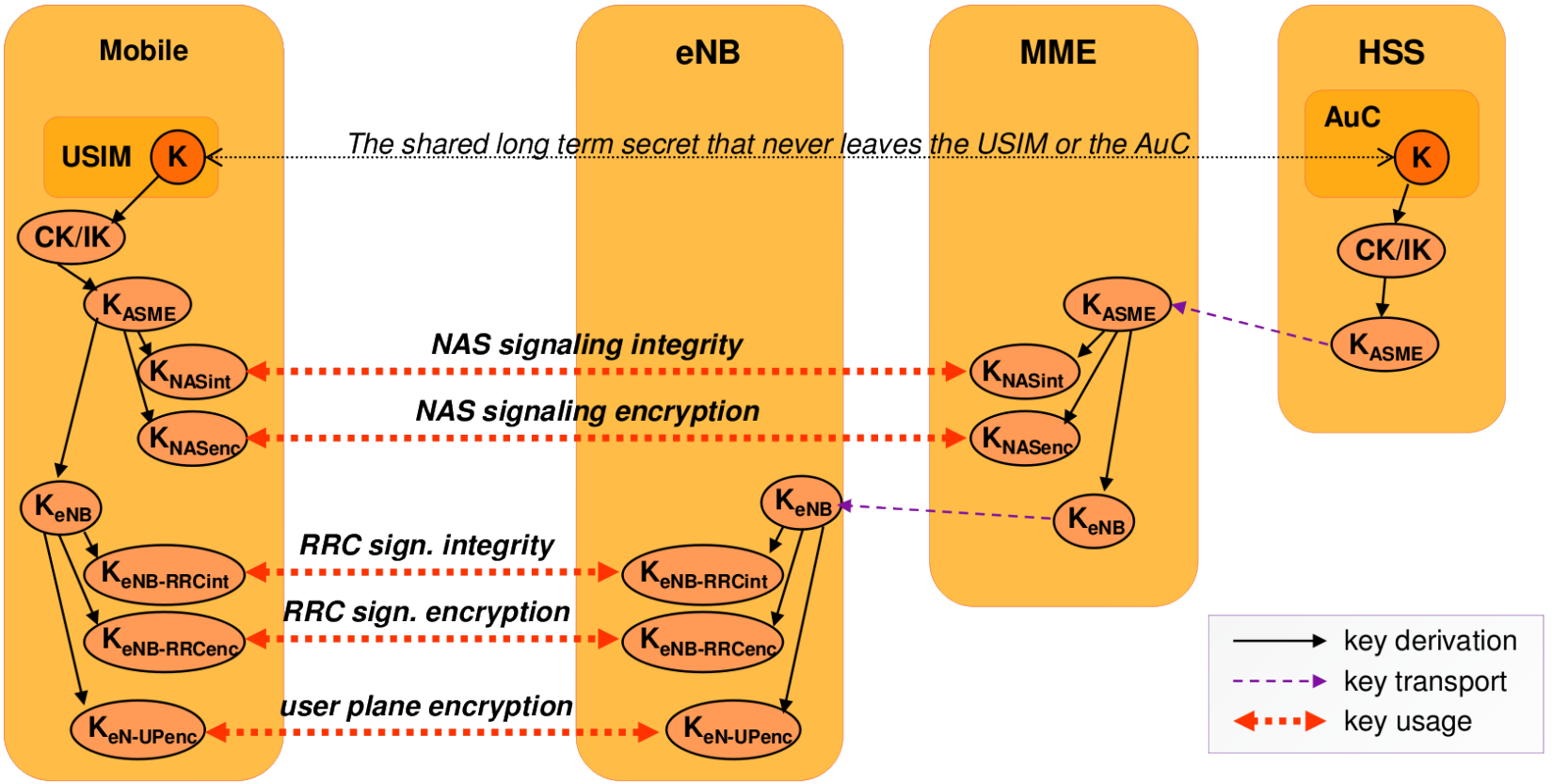
UMTS (3G)

- UMTS is one 3G standard
- Higher speed, better security
- Above all, network authentication
- But also better crypto

UMTS Authentication and Key Agreement



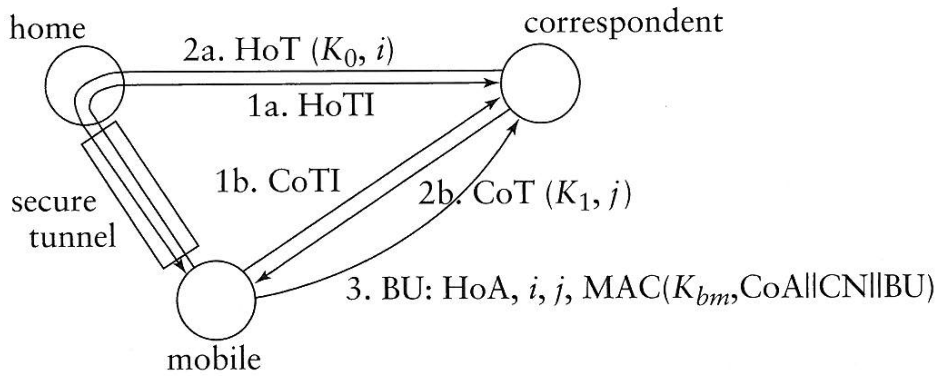
LTE Authentication and Key Agreement



ASME	Access Security Mgmt. Entity	eNB	Evolved Node B	NAS	Non Access Stratum
AuC	Authentication Centre	IK	Integrity Key	RRC	Radio Resource Control
CK	Cipher Key	MME	Mobility Management Entity	USIM	UMTS Subscriber Identity Module

Mobile IPsec

- Mobile phone security is much about preventing fraudulent billing
- Internet mobility should meet other threats
  - Redirecting traffic to get another user's messages
  - Redirecting traffic to crowd another users reception
- In Mobile IPsec a user announces the current location to the intended correspondent directly and via the home agent
- The correspondent returns keygen tokens through both links
- The location is "proved" by knowing data from both links



WLAN

- WLAN security is not so much mobility but wireless network access
- Hiding the Service Set ID (SSID) is common but not so helpful
- The same applies to restricting access to a list of Medium Access Control (MAC) addresses
- The problem is that these are used before security can be set up, and can be sniffed in transit

WEP

- WEP is flawed
- Uses a stream cipher RC4 with a short IV; when IV repeats, so does the key stream, and this is really really bad for a stream cipher
- Checksum is CRC-32, and this is suited for finding random errors, not intentional modifications
- Finally, RC4 has been broken
- URGH

WPA

- This is a quick-fix designed to run on (mostly) the same hardware as WEP
- Generates a new encryption key for each packet through the Temporal Key Integrity Protocol
- Also changes the CRC checksum to a message integrity check algorithm called “Michael”
- There are still weaknesses that remain from WEP and some limitations of Michael that makes it possible to retrieve the keystream from short packets

WPA2

- A new design
- Can be used in Transitional Security Network mode to allow older security modes (but. . .)
- Robust Security Network mode is not backwards compatible
- Encryption is AES used in a stream cipher mode (CTR)
- Message integrity is ensured by CBC-MAC

Bluetooth

- Short range ad hoc networks
- Pairing may be by simple keypresses, close in time, or by PIN
- Main protection is physical proximity
- Uses the "E0" stream cipher
- Security is better since Bluetooth 2.1 (2007), and includes "Simple Secure Pairing"
- Even more recently, Bluetooth 4.2 (2014) provides even better security, especially against device tracking

Security in the IETF layers of network protocols

- Security services at the top can be tailored for specific applications, but each application then needs a separate service
- Security services at the bottom can protect the upper layers transparently, but may not meet all requirements of specific applications