TSIT01 Datasäkerhetsmetoder

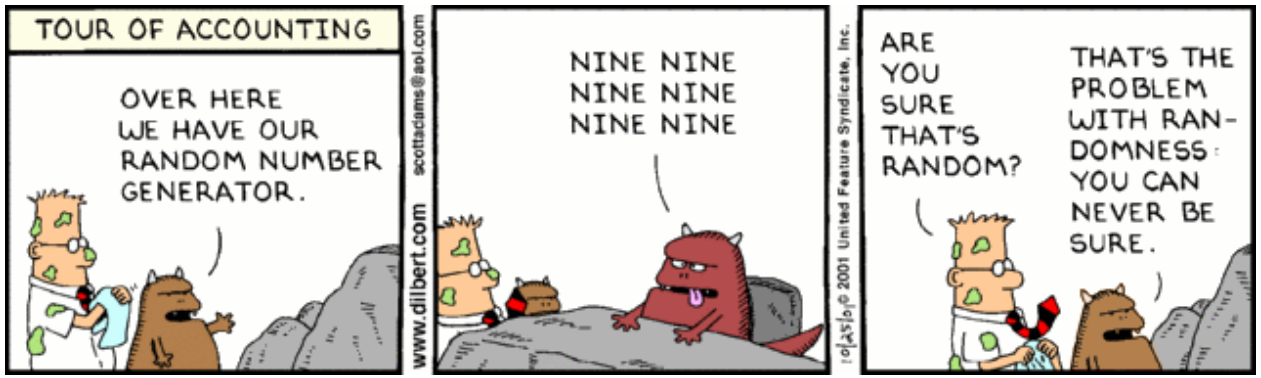
Föreläsning 6: Key management and network security

Guilherme B. Xavier

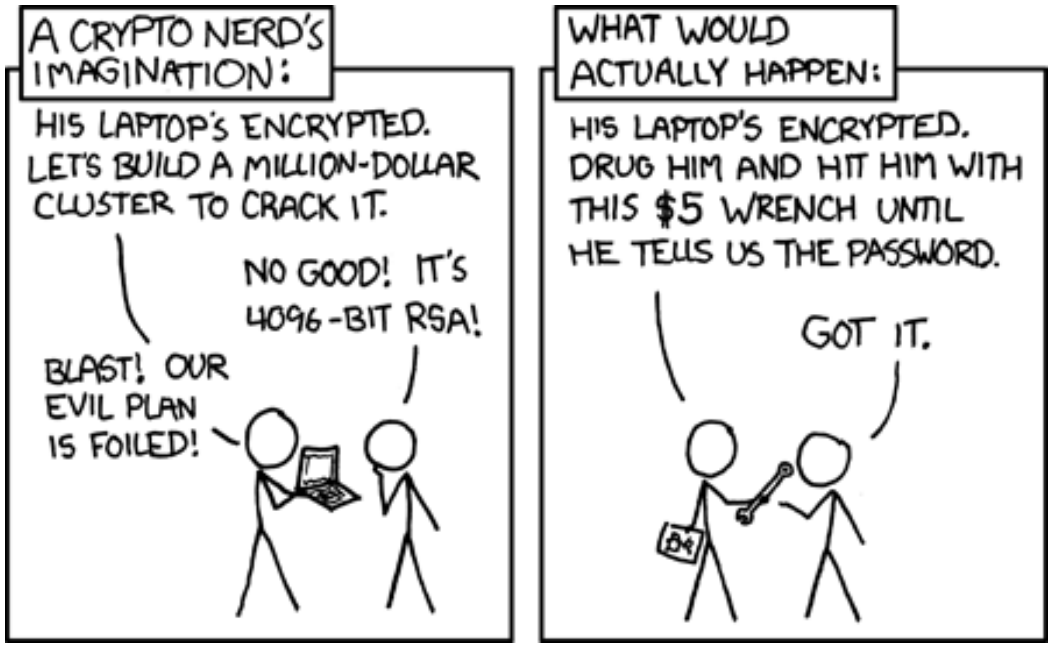
LINKÖPING UNIVERSITY

Key generation

- The key size decides how many different keys you can have, the search space for exhaustive key search
- If keys are not chosen at random, the attacker can first try more likely keys
- If all bit combinations are not used, security is given by the number of possible keys, not the size in bits
- If keys are generated from a known random seed, the size of that seed decides the security



Key length

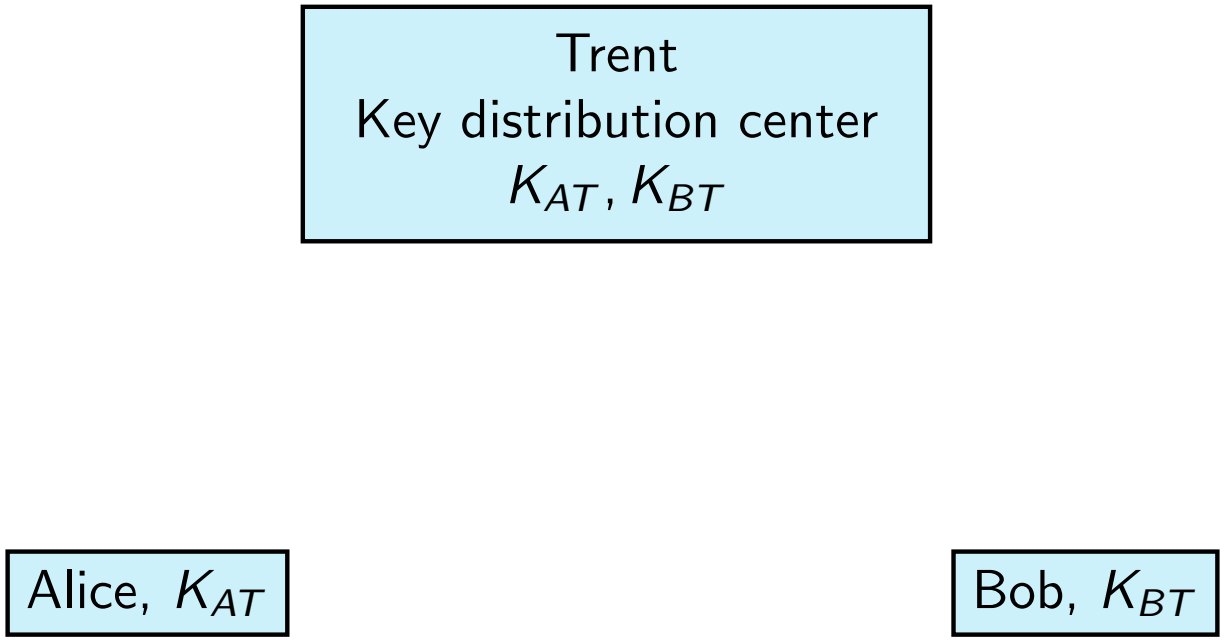

A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Key Management

- The first key in a new connection or association is *always* delivered via a courier
- Once you have a key, you can use that to send new keys
- If Alice shares a key with Trent and Trent shares a key with Bob, then Alice and Bob can exchange a key via Trent (provided they both trust Trent)

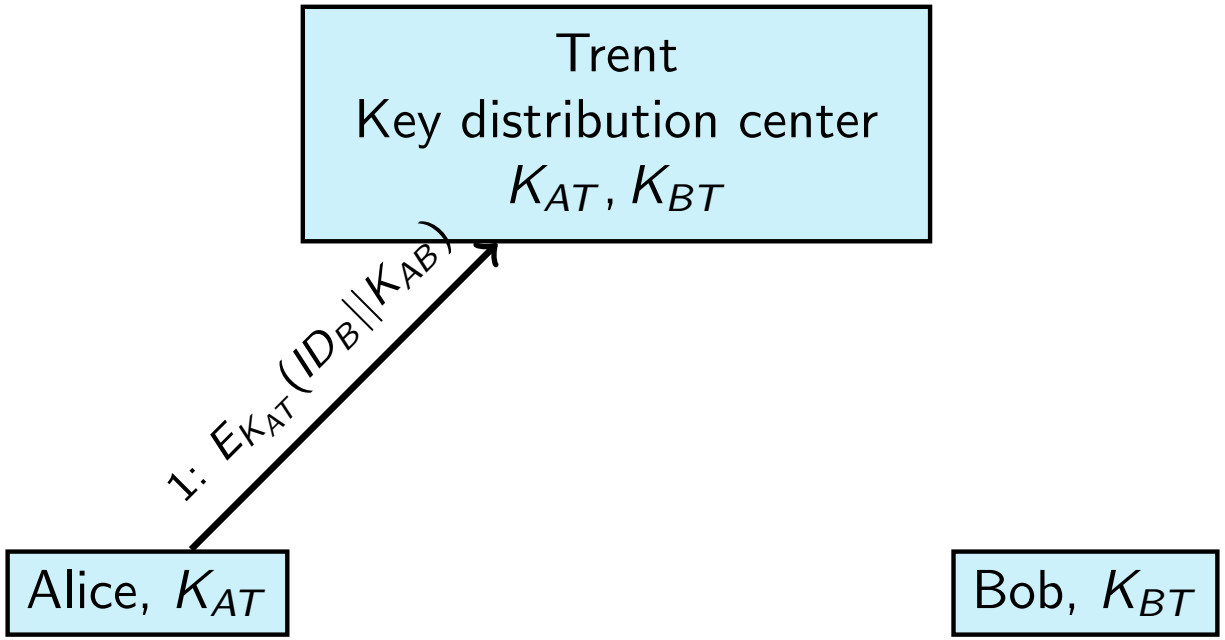
Key distribution center

- If Alice shares a key with Trent and Trent shares a key with Bob, then Alice and Bob can exchange a key via Trent (provided they both trust Trent)



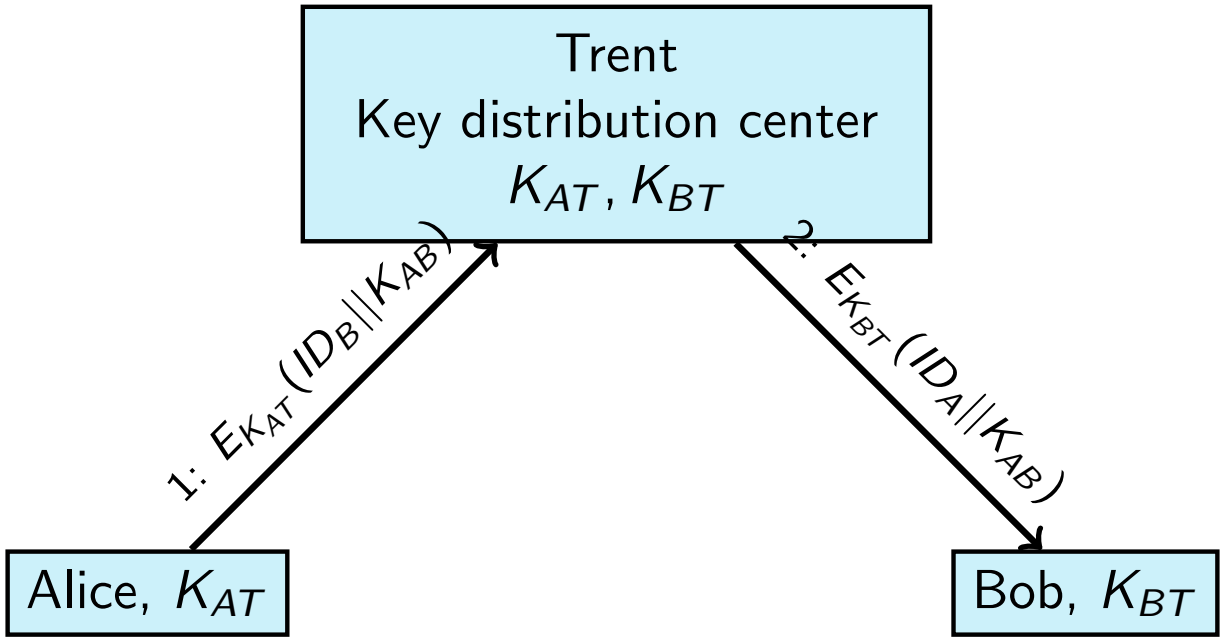
Key distribution center

- If Alice shares a key with Trent and Trent shares a key with Bob, then Alice and Bob can exchange a key via Trent (provided they both trust Trent)



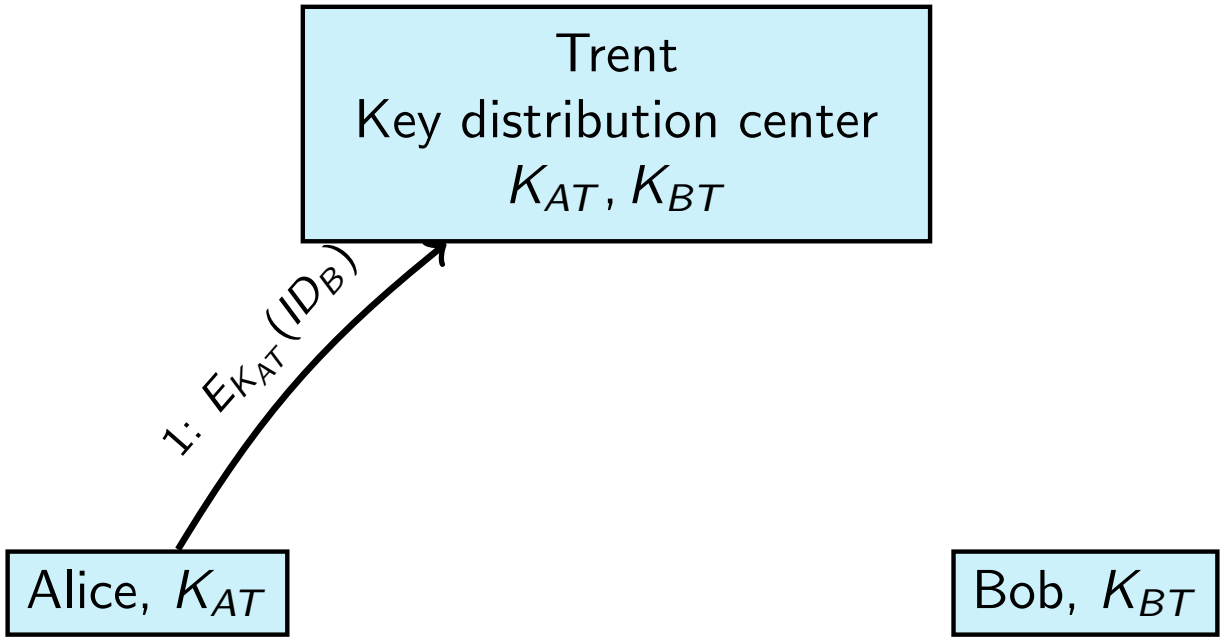
Key distribution center

- If Alice shares a key with Trent and Trent shares a key with Bob, then Alice and Bob can exchange a key via Trent (provided they both trust Trent)



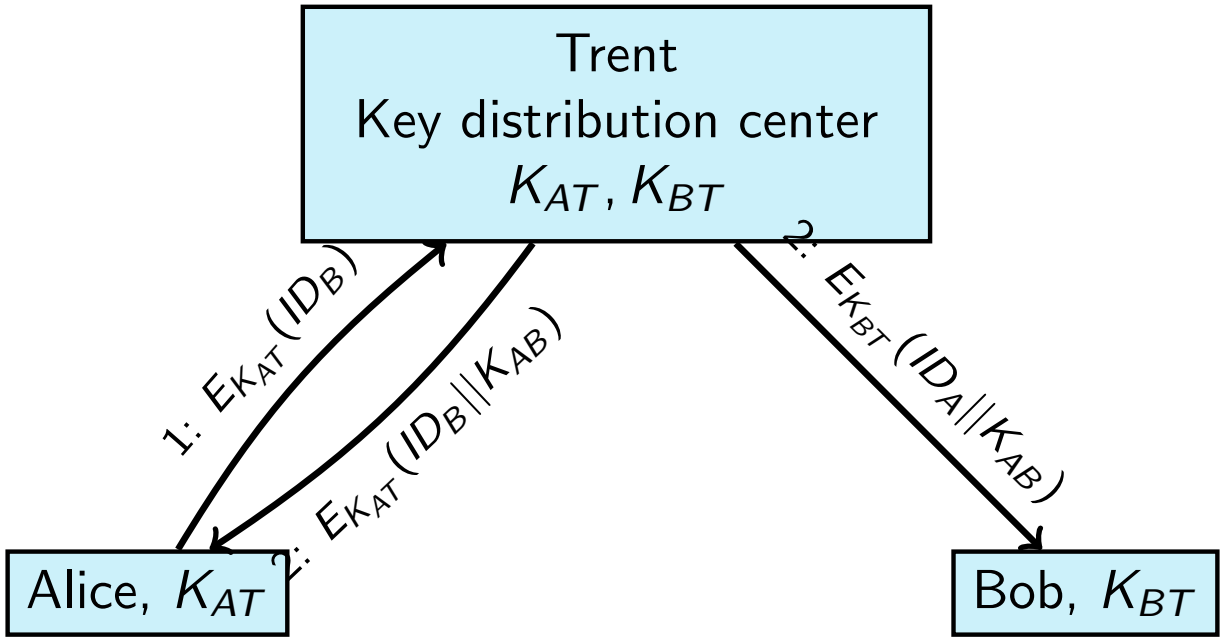
Key distribution center, key server

- If Alice shares a key with Trent and Trent shares a key with Bob, then Alice and Bob can receive a key from Trent (provided they both trust Trent)



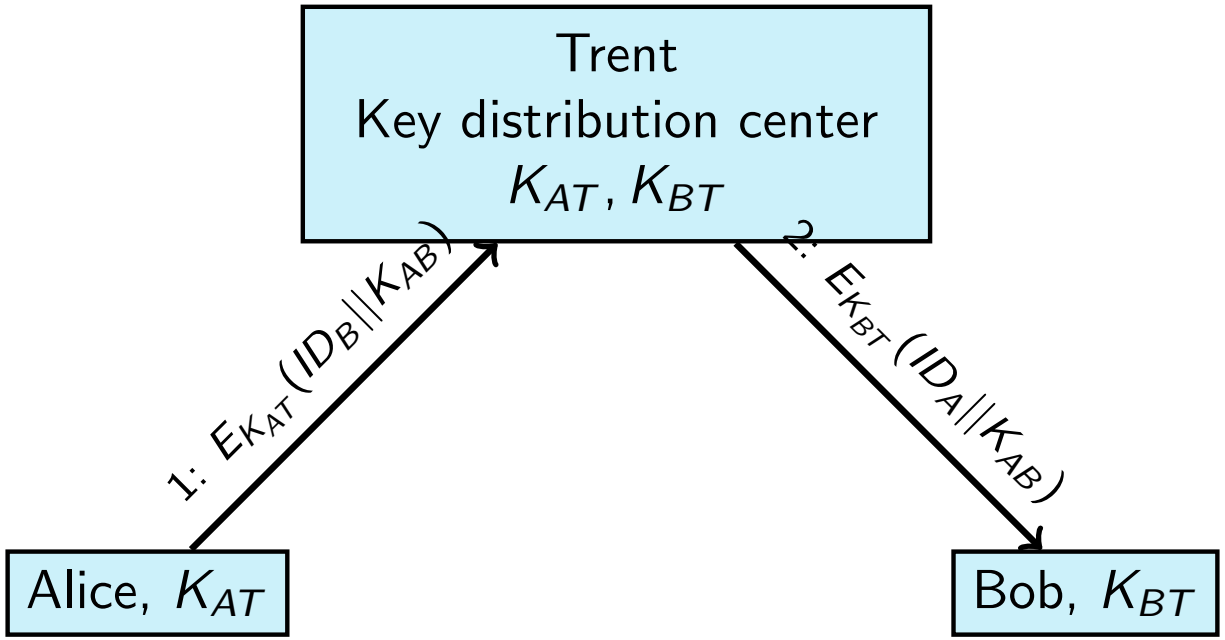
Key distribution center, key server

- If Alice shares a key with Trent and Trent shares a key with Bob, then Alice and Bob can receive a key from Trent (provided they both trust Trent)



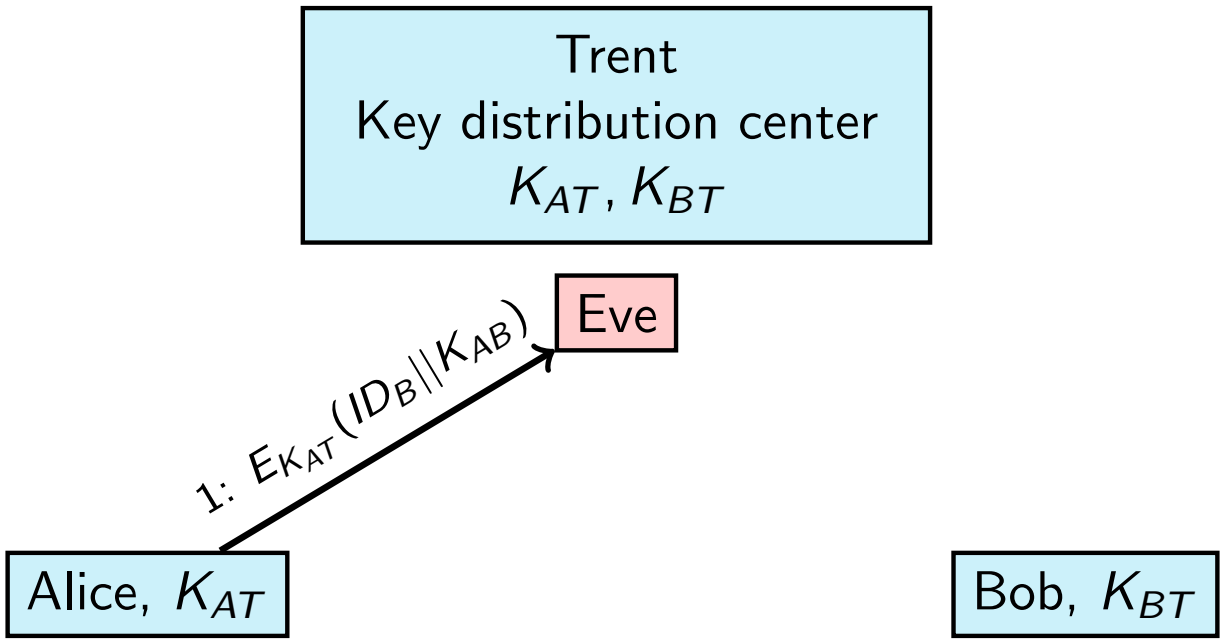
Key distribution center

- If Alice shares a key with Trent and Trent shares a key with Bob, then Alice and Bob can exchange a key via Trent (provided they both trust Trent)



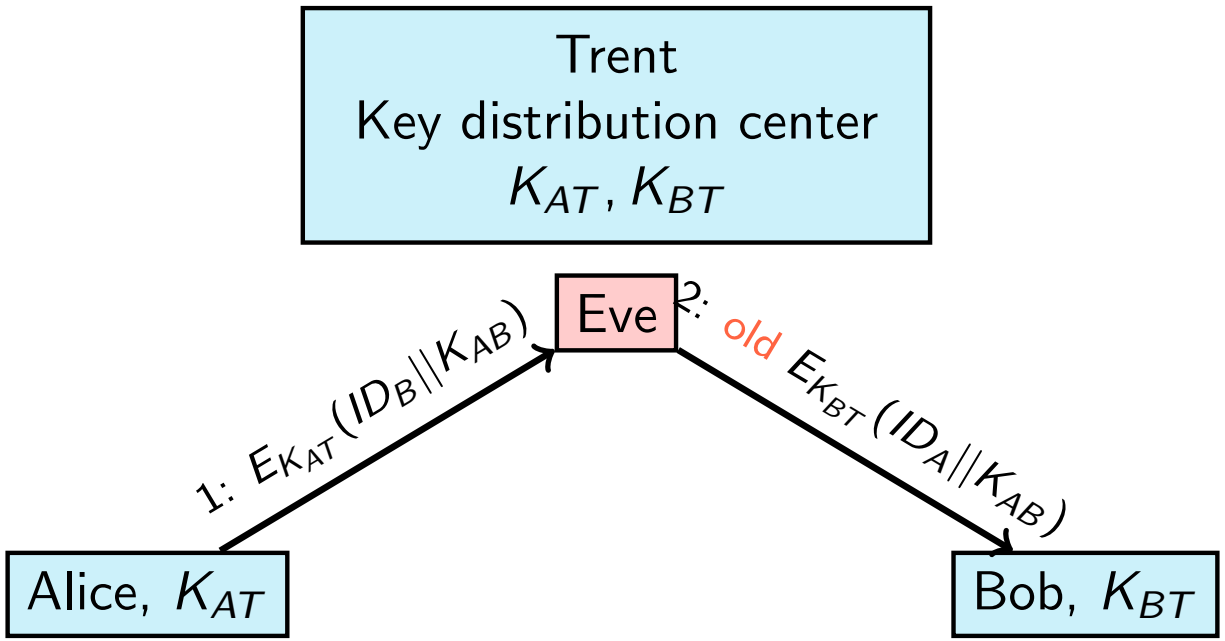
Key distribution center, replay attacks

- Eve intercepts Alice's request



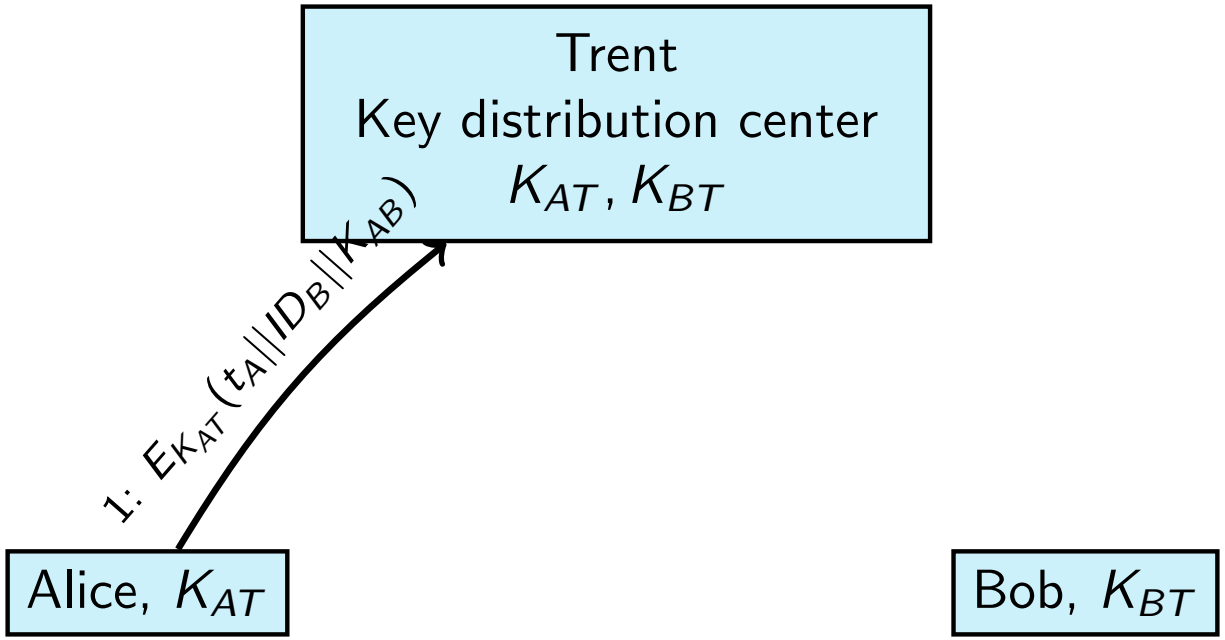
Key distribution center, replay attacks

- Eve intercepts Alice's request
- Then she can fool Bob into communicating with her



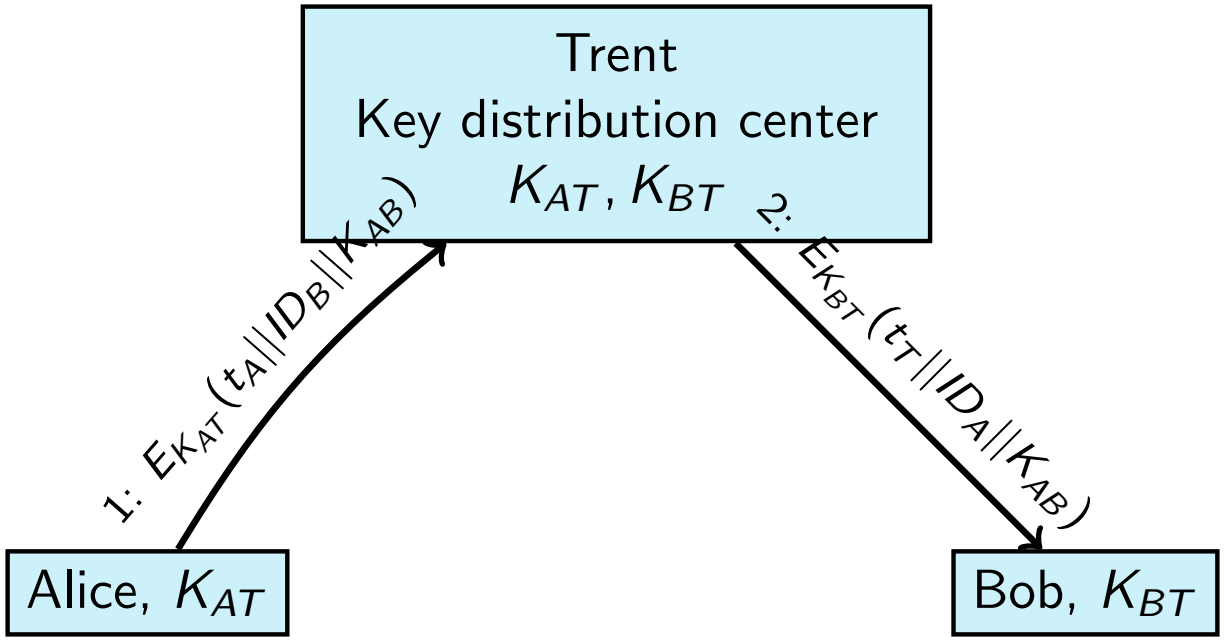
Key distribution center, wide-mouthed frog

- Alice and Trent add time stamps to prohibit the attack



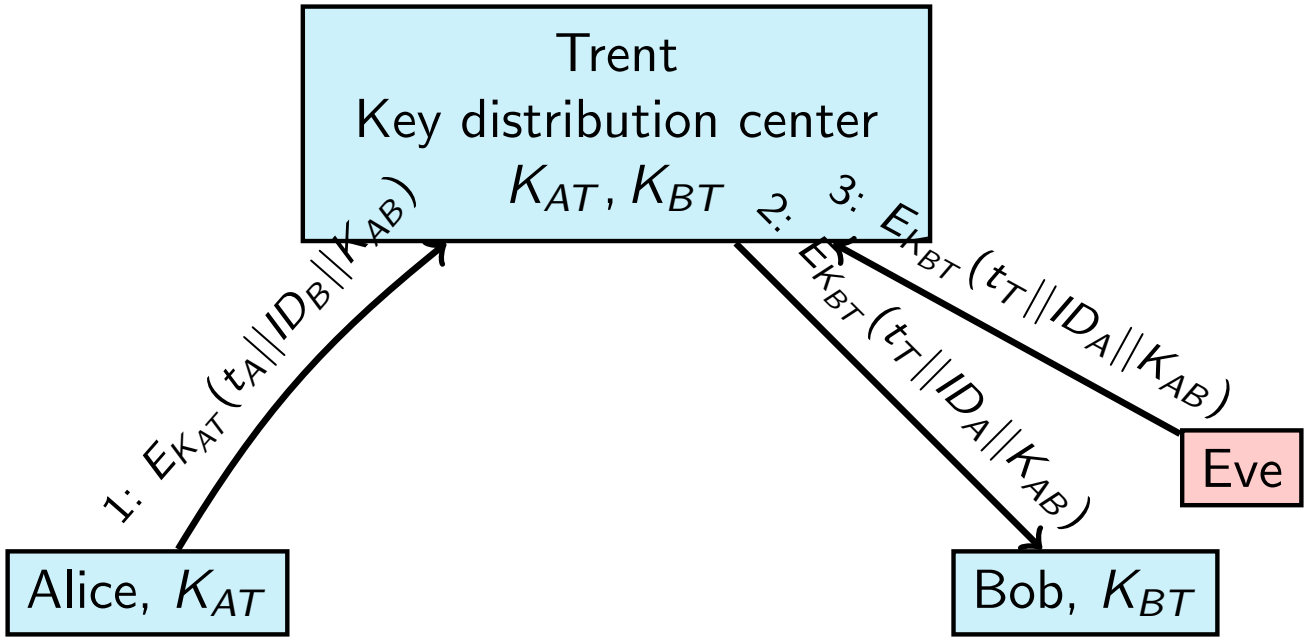
Key distribution center, wide-mouthed frog

- Alice and Trent add time stamps to prohibit the attack



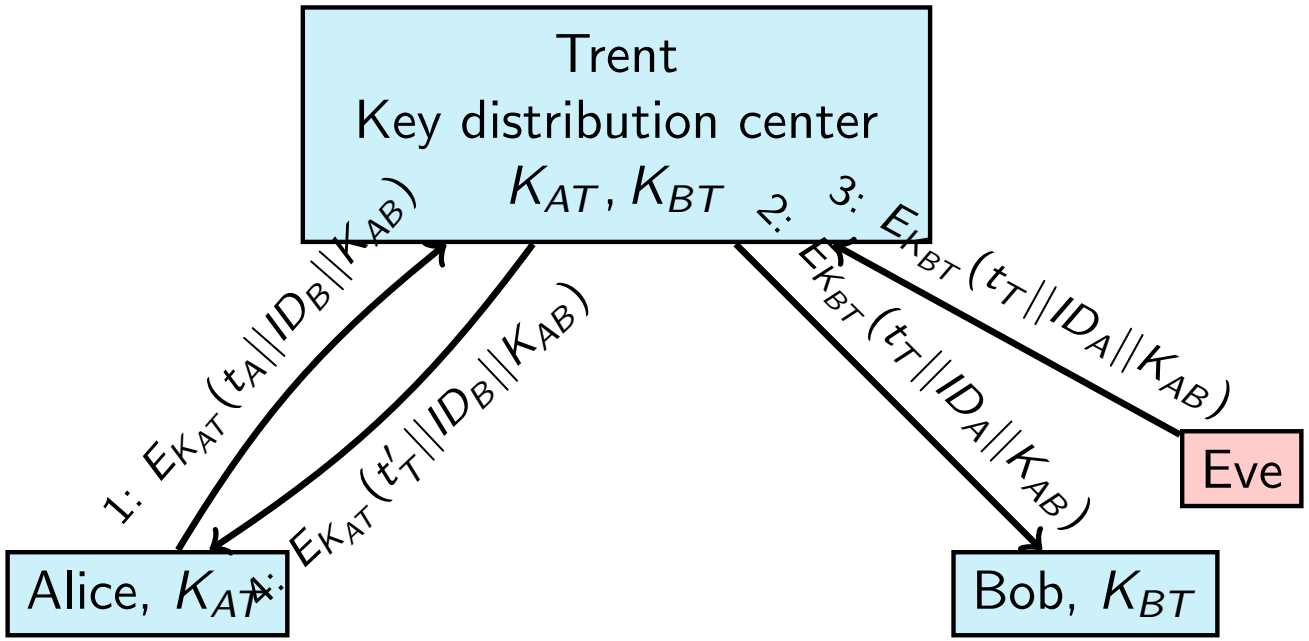
Key distribution center, wide-mouthed frog

- Alice and Trent add time stamps to prohibit the attack
- Eve can now only pretend to be Bob and make a request to Trent for a short period of time



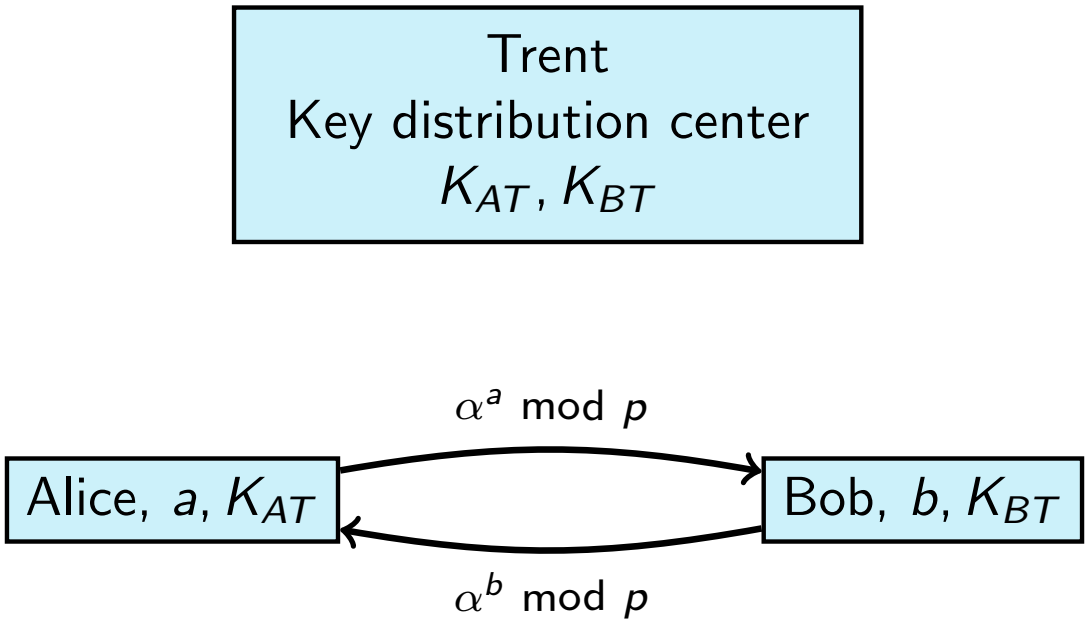
Key distribution center, wide-mouthed frog

- Alice and Trent add time stamps to prohibit the attack
- Eve can now only pretend to be Bob and make a request to Trent for a short period of time



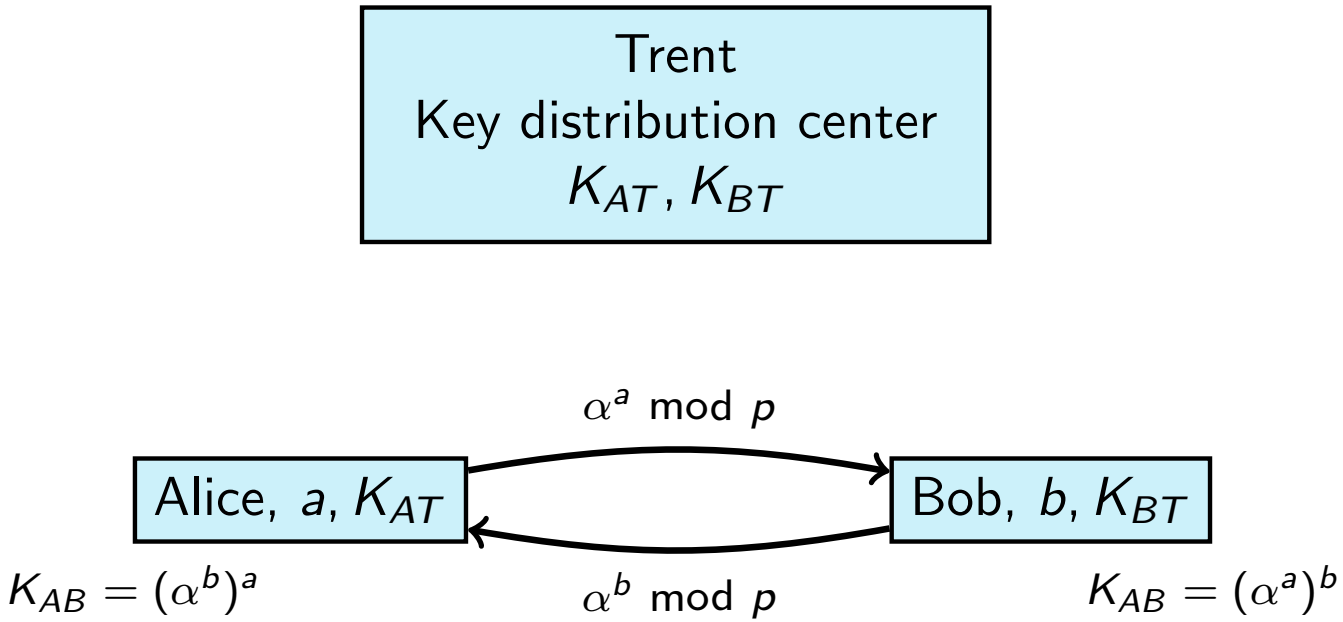
Public key distribution, Diffie-Hellmann

- Diffie-Hellman key exchange is a way to share key
- Alice and Bob create secrets a and b
- They send α^a mod p and α^b mod p to each other



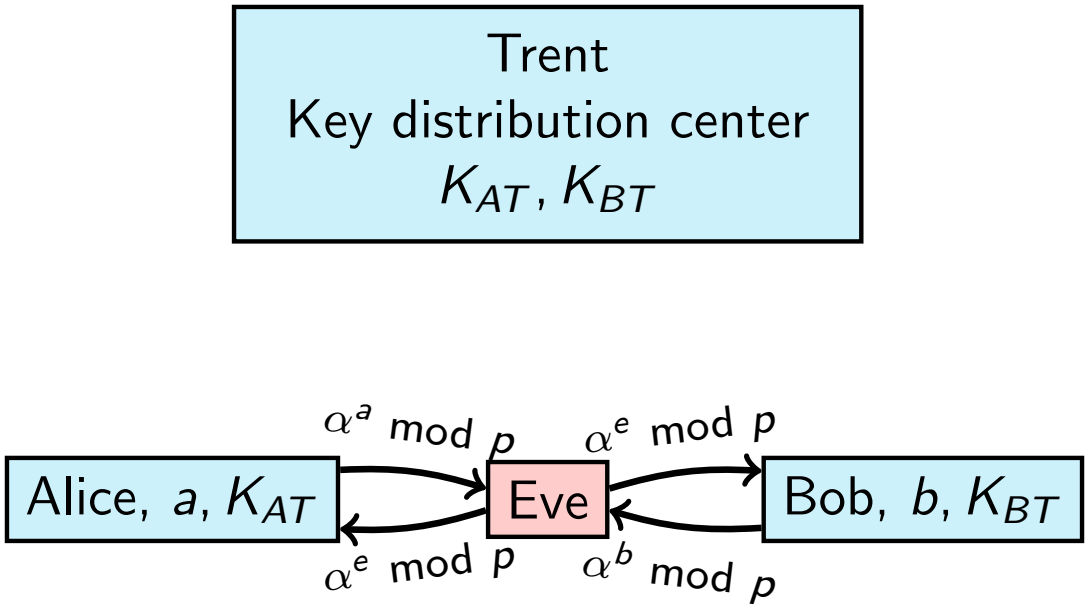
Public key distribution, Diffie-Hellmann

- Diffie-Hellman key exchange is a way to share key
- Alice and Bob create secrets a and b
- They send α^a mod p and α^b mod p to each other
- Both calculate $K_{AB} = (\alpha^a)^b = (\alpha^b)^a$ mod p



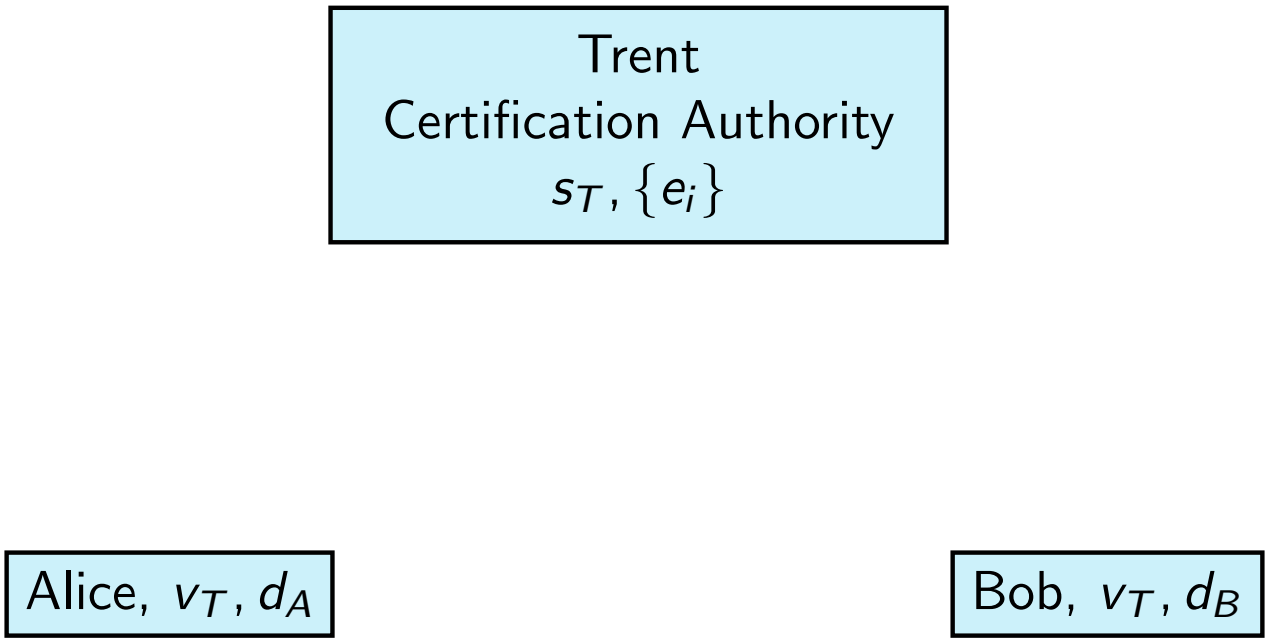
Public key distribution, Diffie-Hellmann

- Diffie-Hellman key exchange is a way to share key
- However, Eve can do an "man-in-the-middle"



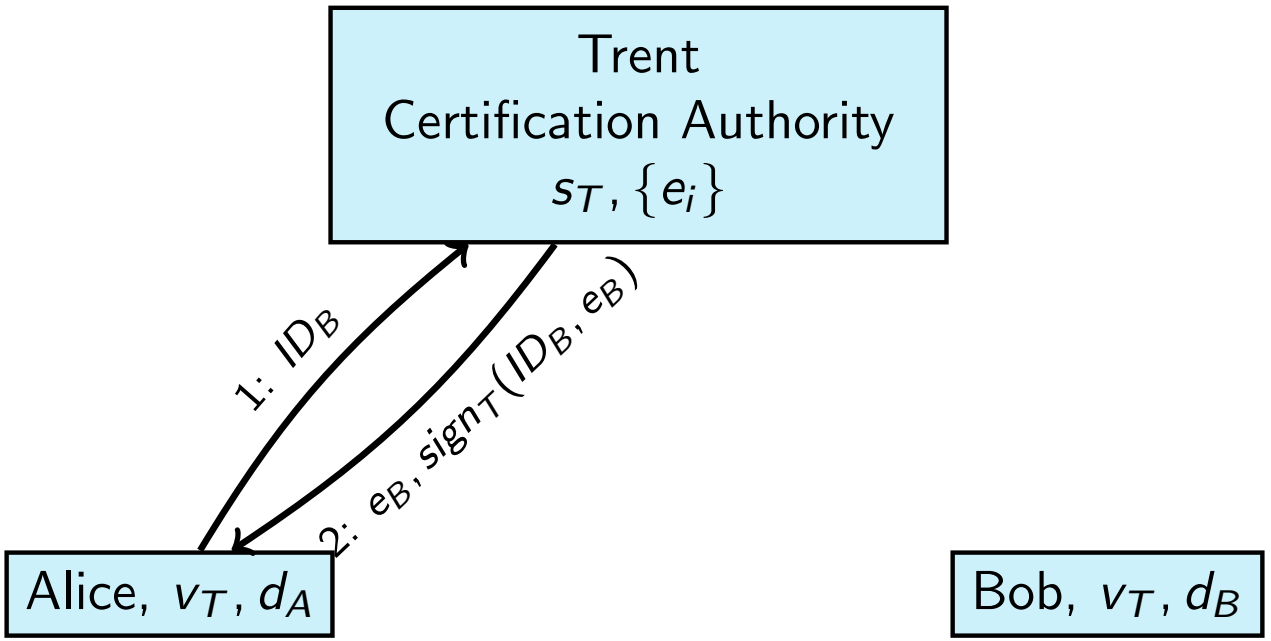
Public key distribution

- Public key distribution uses a Public Key Infrastructure (PKI)



Public key distribution, using Certification Authorities

- Public key distribution uses a Public Key Infrastructure (PKI)
- Alice sends a request to a Certification Authority (CA) who responds with a certificate, ensuring that Alice uses the correct key to communicate with Bob



Public key distribution, using X.509 certificates

- The CAs often are commercial companies, that are assumed to be trustworthy
- Many arrange to have the root certificate packaged with the web browsers
- They issue certificates for a fee
- They often use Registration Authorities (RA) as sub-CA for efficiency reasons
- This creates a “certificate chain”

The content of a X.509 certificate

Version (v3)
Serial Number
Algorithm ID
Issuer
Validity Period
Subject Name
Subject Public Key Info (Algorithm, Public Key)
Issuer Unique Identifier (optional)
Subject Unique Identifier (optional)
Extensions (optional)
Certificate Signature Algorithm
Certificate Signature

Revocation

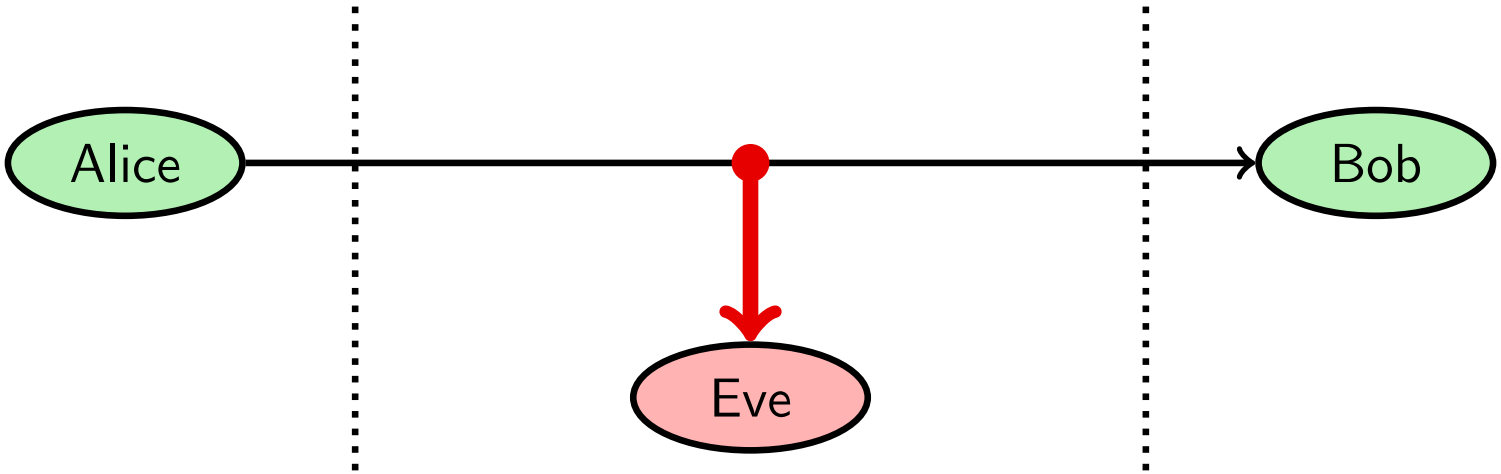
- Certificate Revocation Lists distributed at regular intervals is the proposed solution in X.509
- On-line checks are better, but can be expensive
- Short-lived certificates are an alternative, but needs frequent certificate changes
- And the CAs themselves are not the best examples of trustworthy organizations

Key revocation

- Cease issuing new tickets
  - Does not invalidate issued tickets
- Short validity of tickets (Kerberos often one day, X.509 could be years)
  - Frequent renewal
  - Bandwidth/processing limitation
  - Clock skew problem
  - Availability of key server
- Revocation lists
  - Needed, even with time limits
  - Are these properly updated?
  - There will be a delay
  - Availability of revocation list server
  - Bandwith limitation

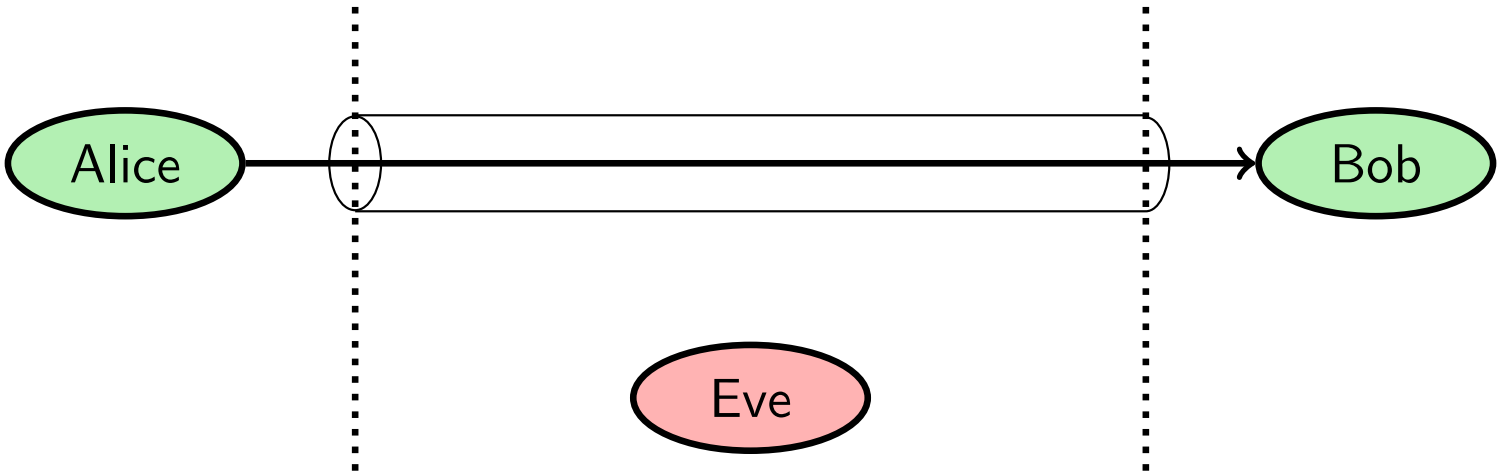
Communication and network security: Threat model

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



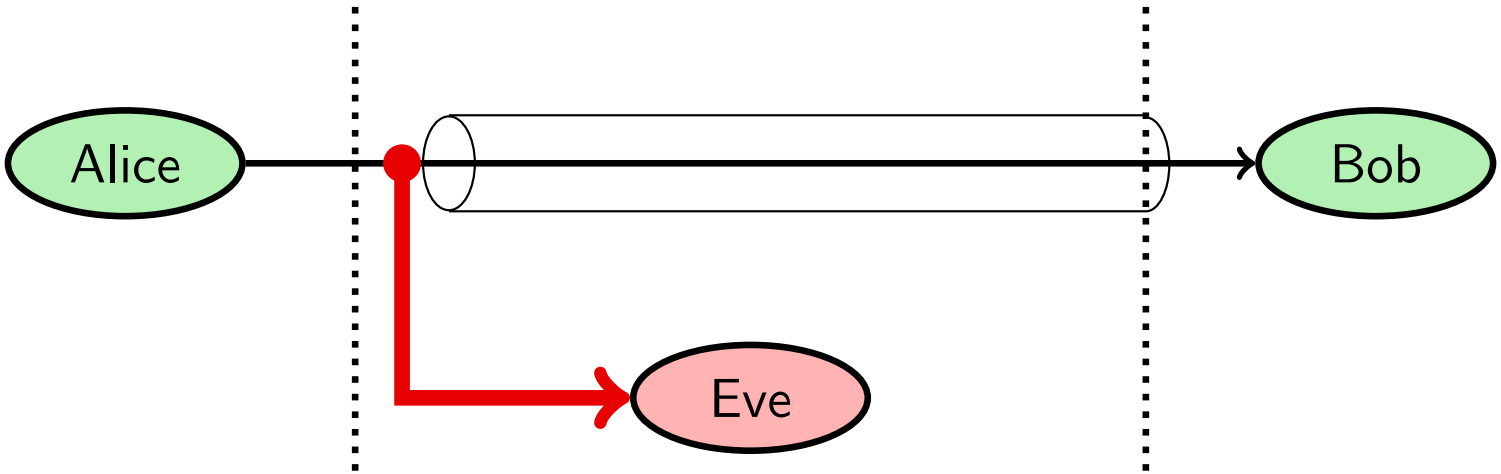
Communication security: Secure tunnels

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



Communication security: Secure tunnels

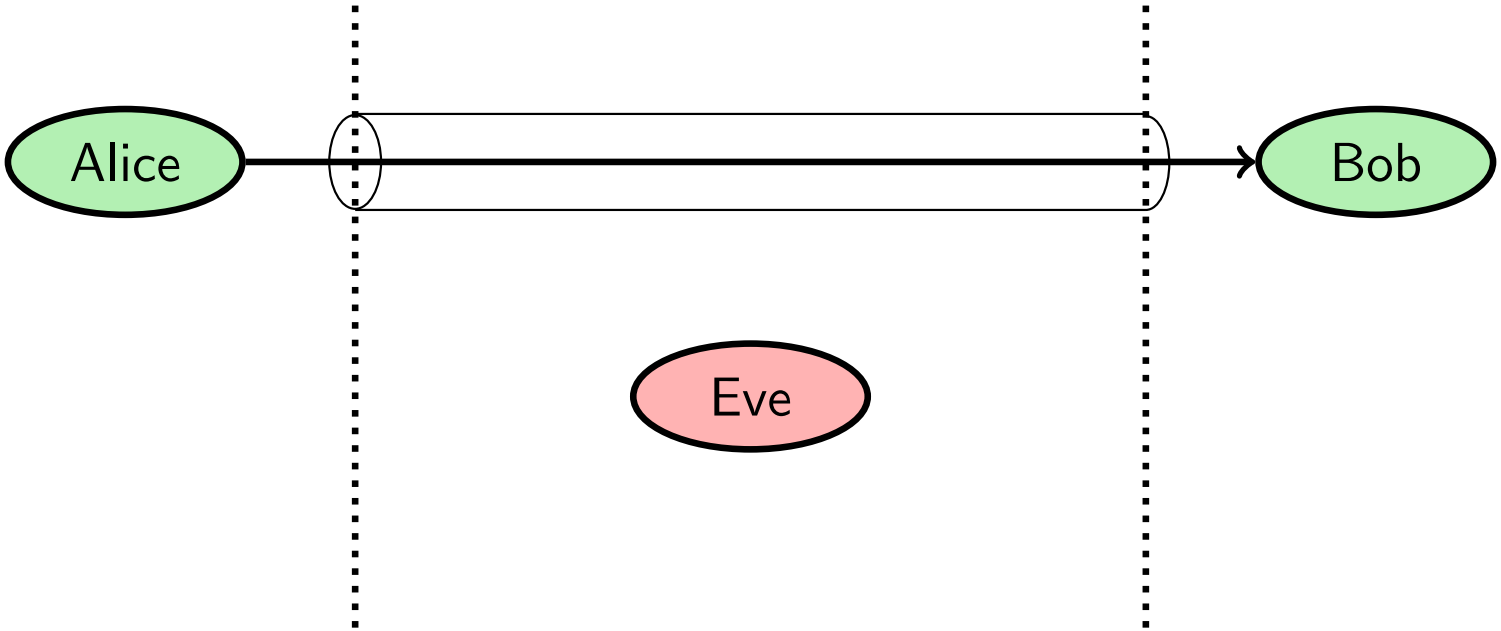
- Typically provide Confidentiality, data Integrity, and data origin authentication
- End points may be machines or services on the local computer
- The placement is important to achieve security



Communication security: Secure tunnels

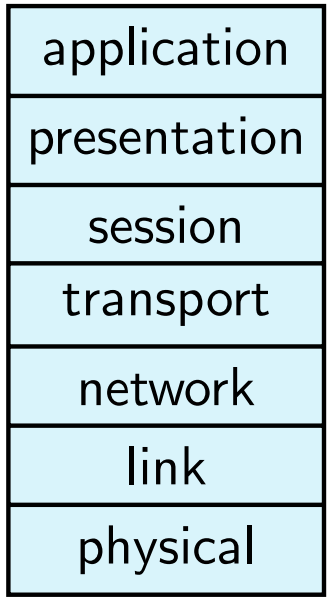
Steps to set up a tunnel

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

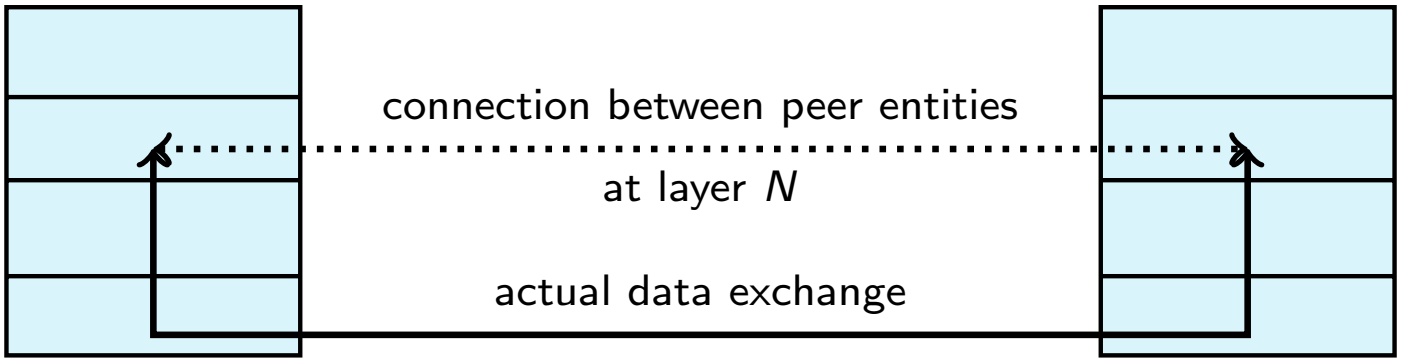


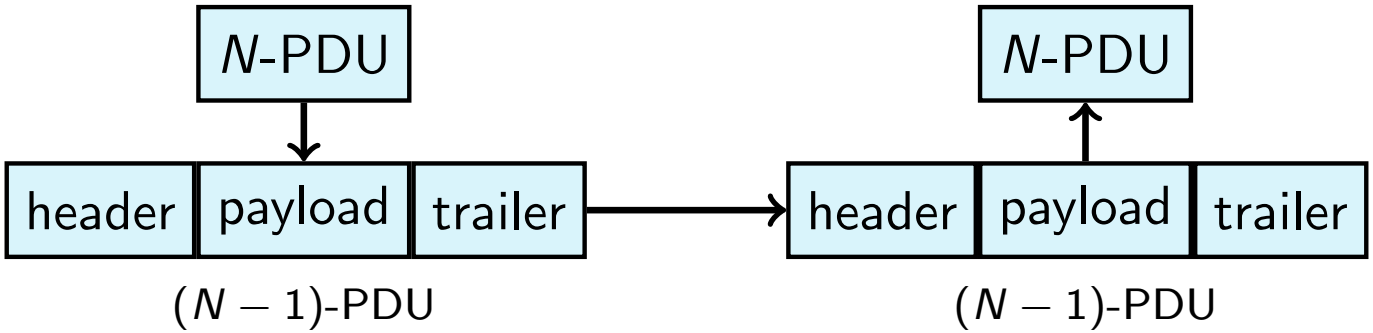
Layered model of network protocols

- *The* abstraction of network structure, the ISO/OSI seven-layer model
- 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

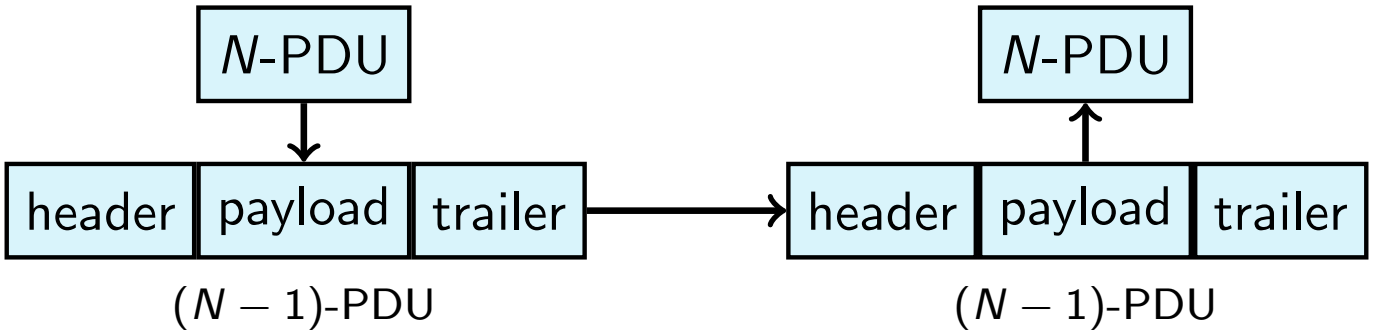


Briefly on the layered model





Briefly on the layered model

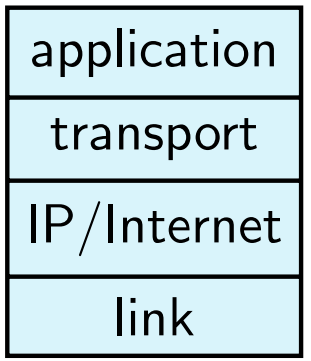


There are two options for security services in the $N-1$ layer

- The upper layer can be *aware* of the security services at the lower layer
- The lower layer security services can be *transparent*

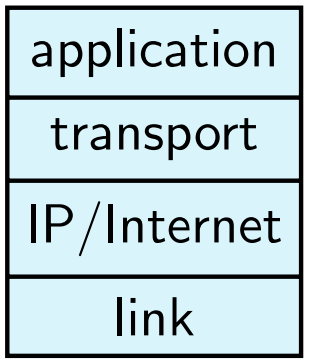
The internet protocol stack

- The application layer has Telnet, FTP, HTTP, SMTP, SET, ...
- The transport layer has the protocols TCP and UDP, and applications connect to *ports* in this layer
- The internet layer has the IP protocol, nodes are identified through their *IP address*
- The link (and the physical) layer are specific to the tech used
- There are security services both in the transport and Internet layers



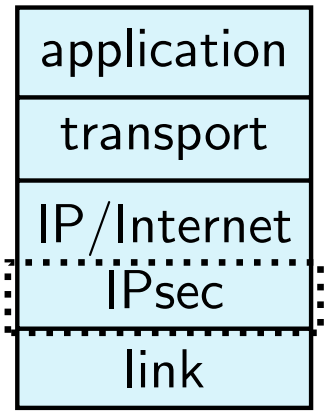
The IP protocol

- The IP protocol is stateless and does not keep track of connections
- Each packet is independent
- No guaranteed delivery
- Order is not preserved
- No security mechanism



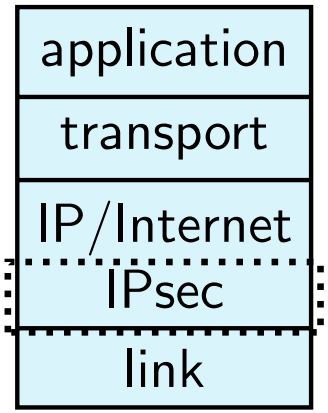
IPsec, IP protocol security

- Optional for IPv4, mandatory for IPv6
- Two major security mechanisms: Authentication Header and Encapsulating Security Payload (ESP)
- 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 and some replay protection
- 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



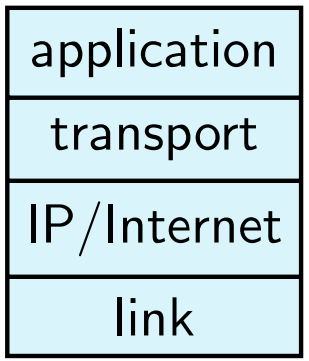
Pros and cons of IPsec

- 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

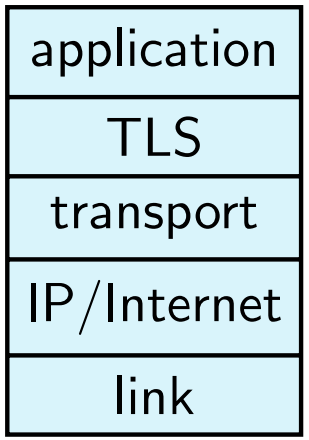
SSL/TLS

- 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



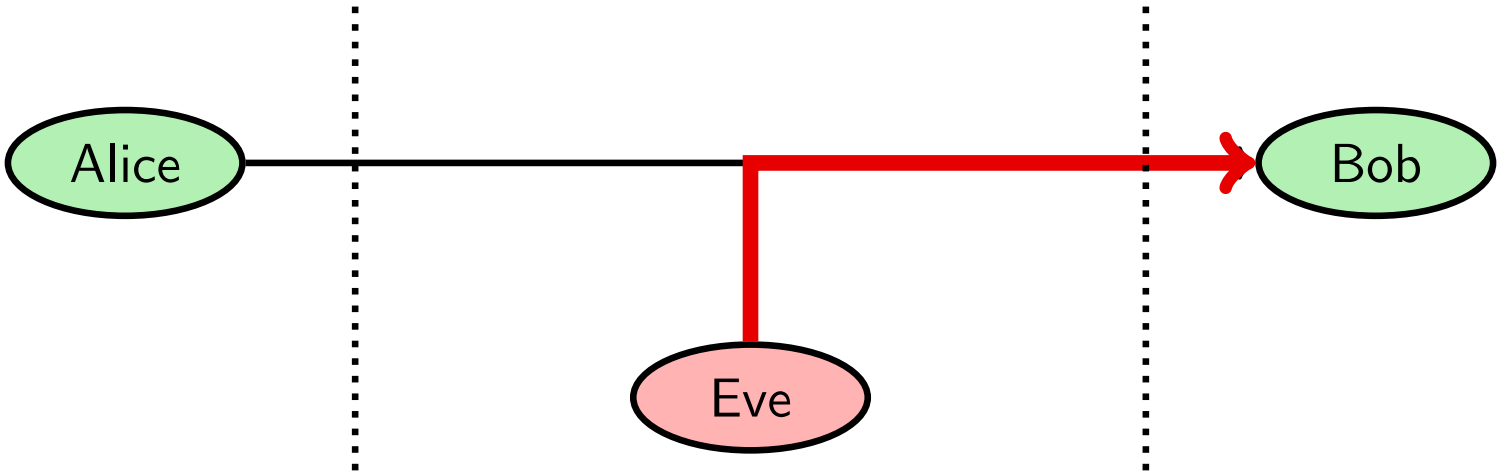
SSL/TLS

- 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



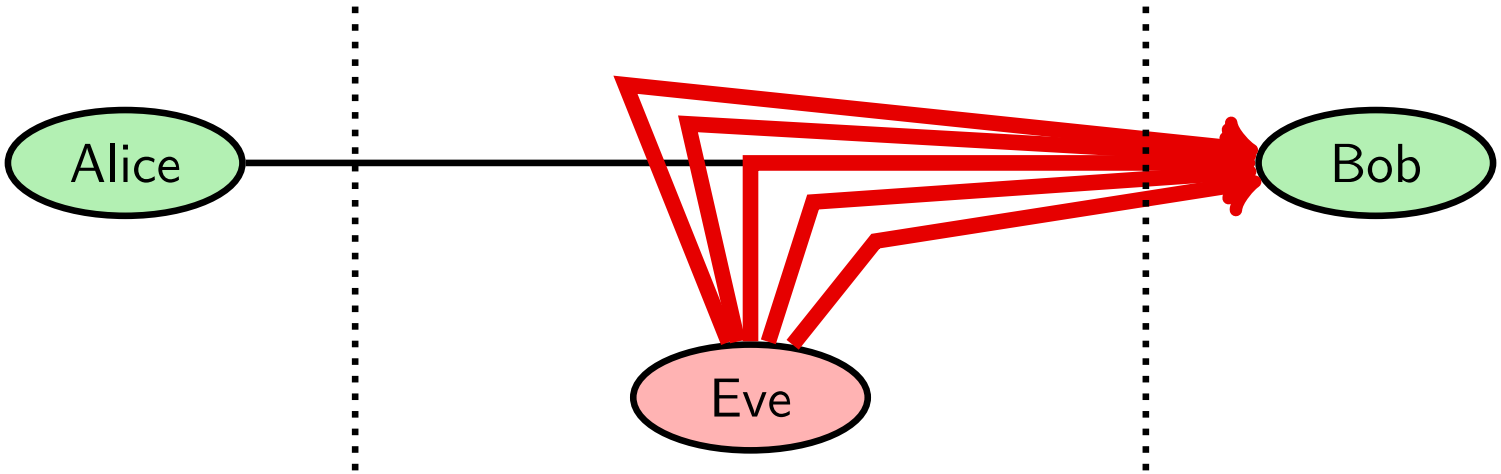
Communication and network security: Threat model

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



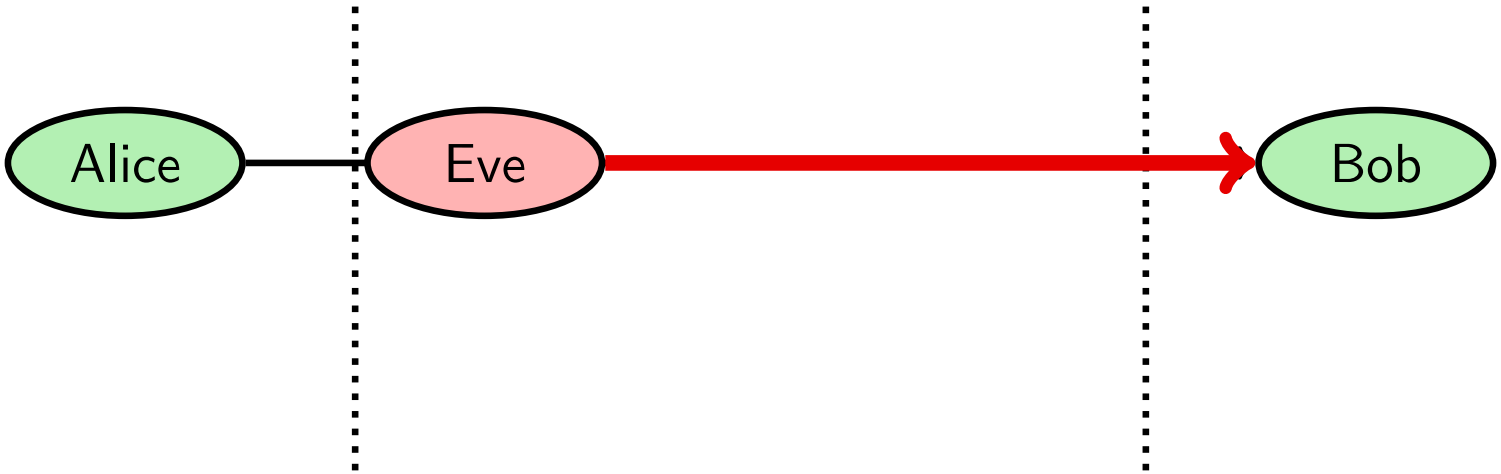
Communication and network security: Threat model

- Passive attacks: Eavesdropping, Wiretapping, Sniffing, and Traffic analysis
- Active attacks: Spoofing, Flooding,



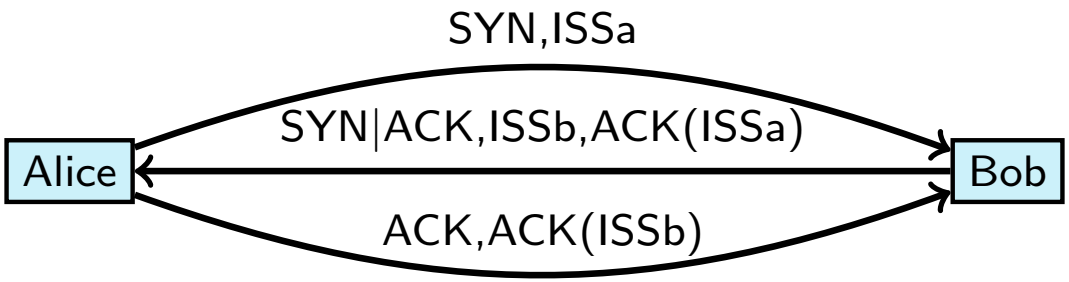
Communication and network security: Threat model

- Passive attacks: Eavesdropping, Wiretapping, Sniffing, and Traffic analysis
- Active attacks: Spoofing, Flooding, Squatting



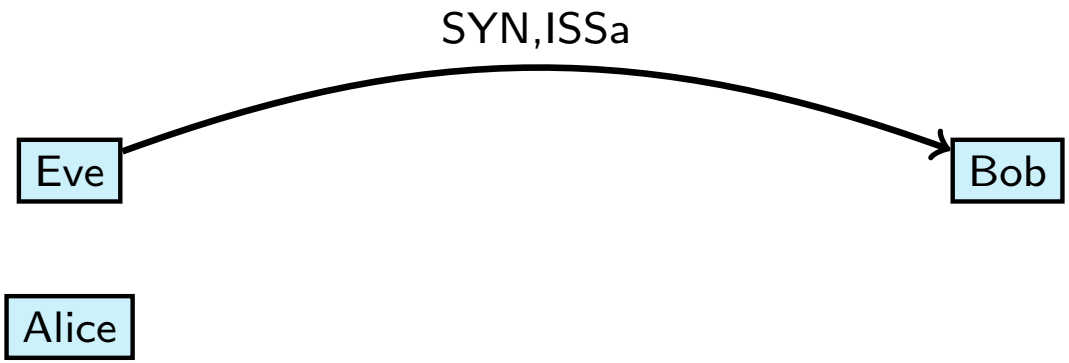
TCP session hijacking

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



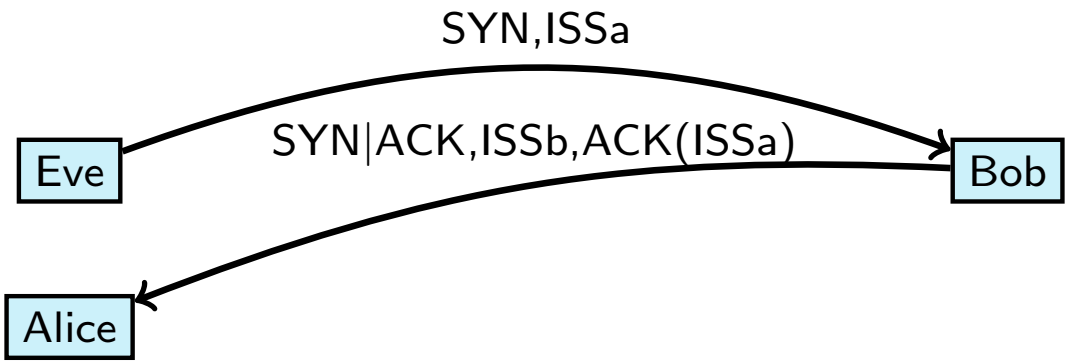
TCP session hijacking

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



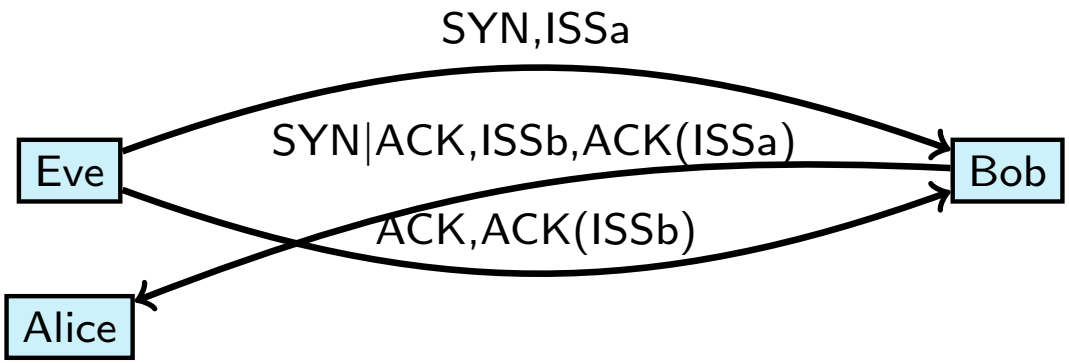
TCP session hijacking

- 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

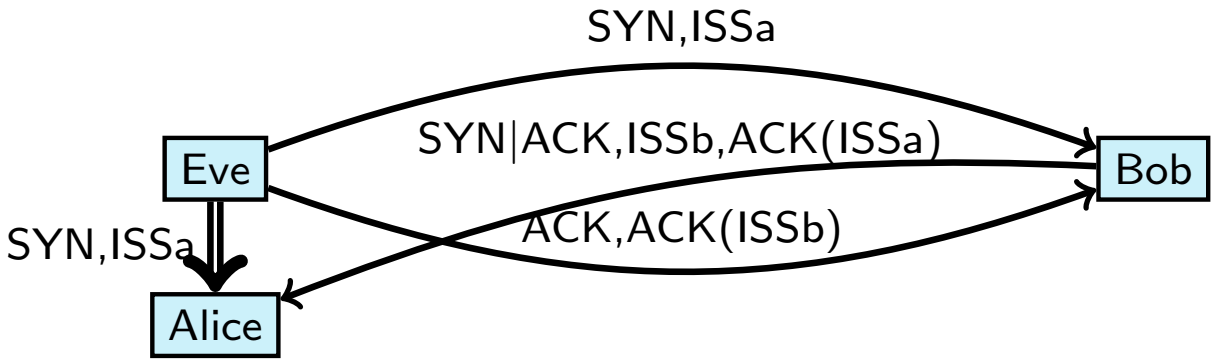


TCP session hijacking

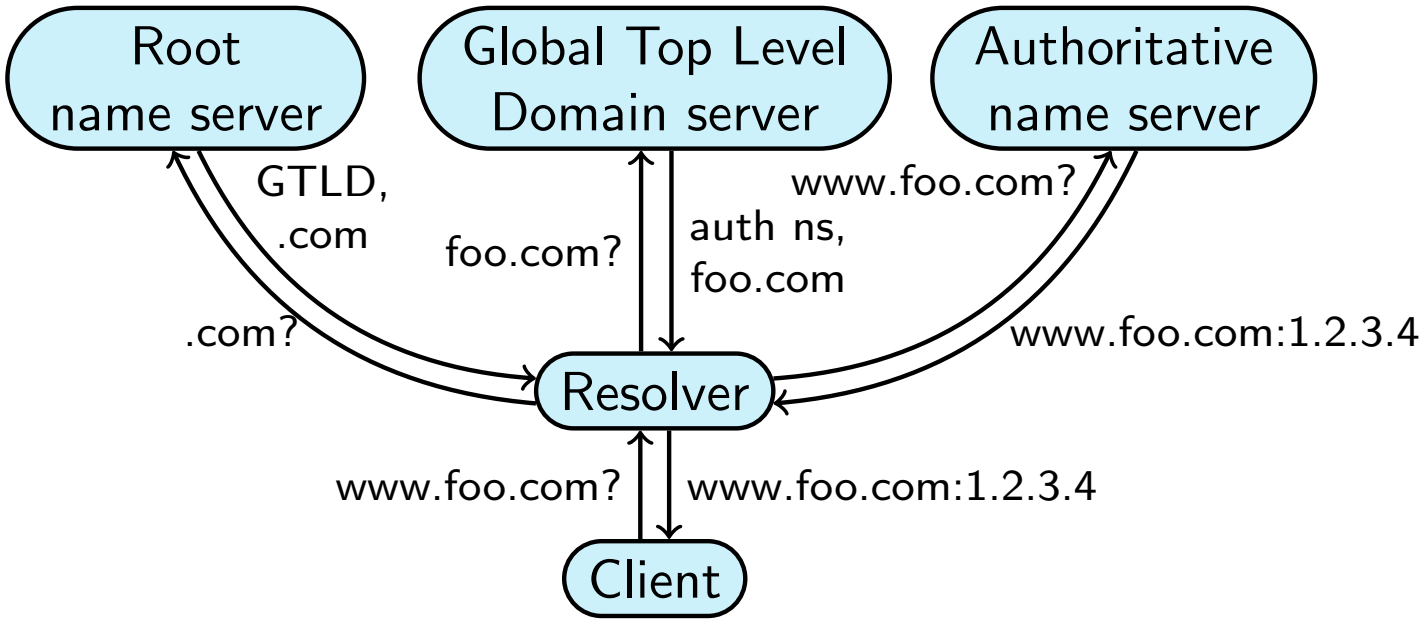
- Eve has established a (blind) session through session hijacking
- Certain protocols use no more authentication than this
- For these, Eve can use Alice's credentials at Bob
- Solution: firewall, or don't use services with address-based authentication

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

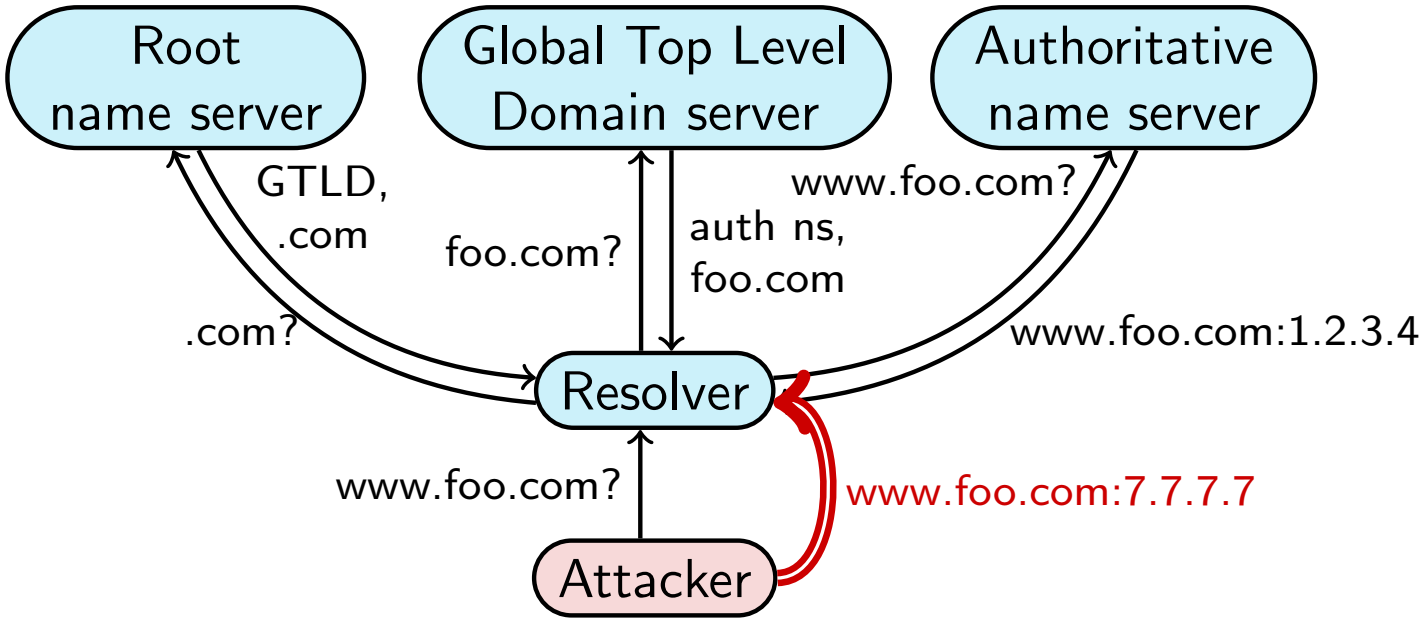


Domain Name System, DNS



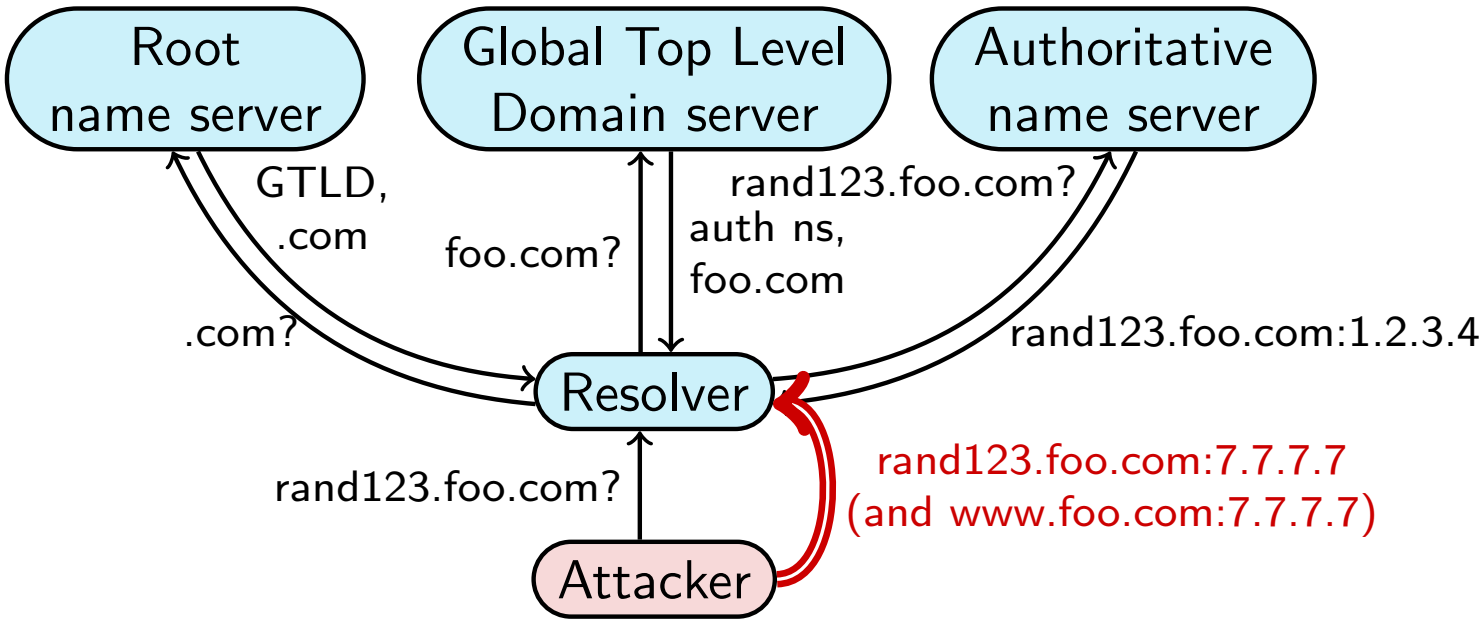
- DNS uses "Lightweight authentication", a 16-bit query id (QID) and a UDP response port that the answering server should use

DNS Cache poisoning



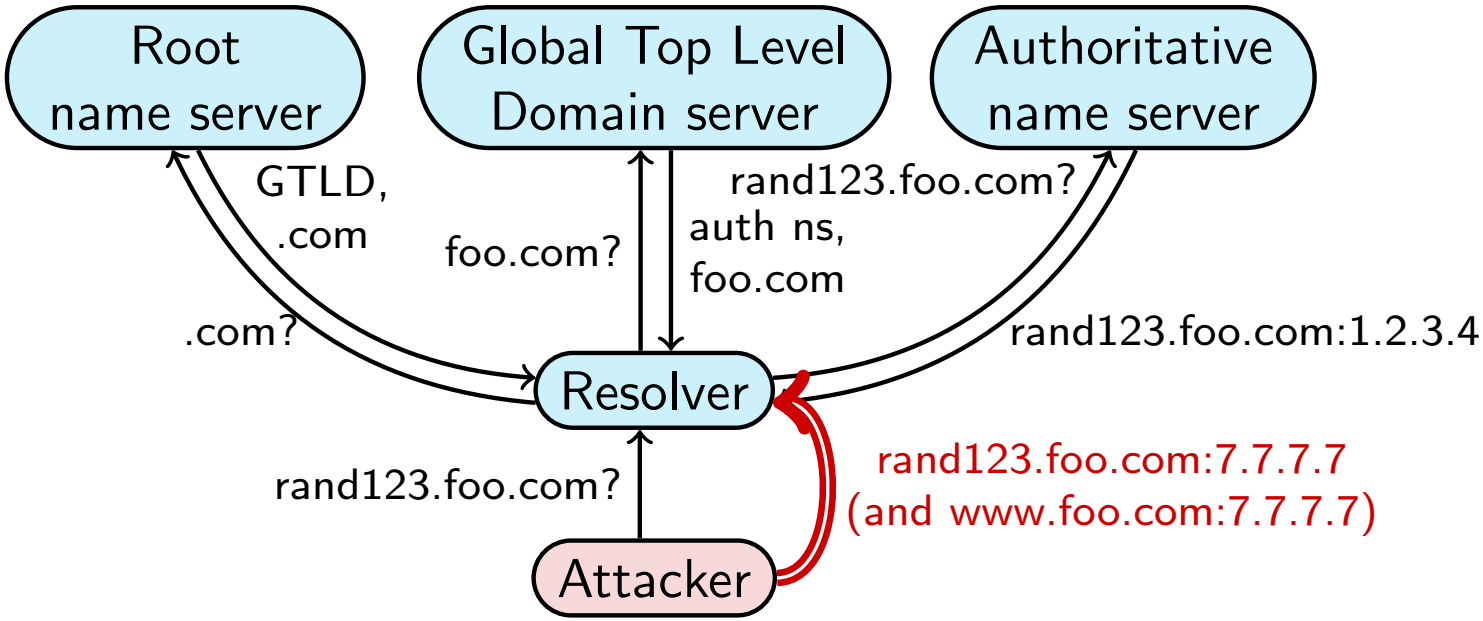
- Attacker asks for IP for target, then immediately floods the resolver with guessed QIDs at guessed UDP ports
- If successful, the attacker gets to decide Time To Live for the record

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

Dan Kaminsky's attack



- The attacker 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
- 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