Security

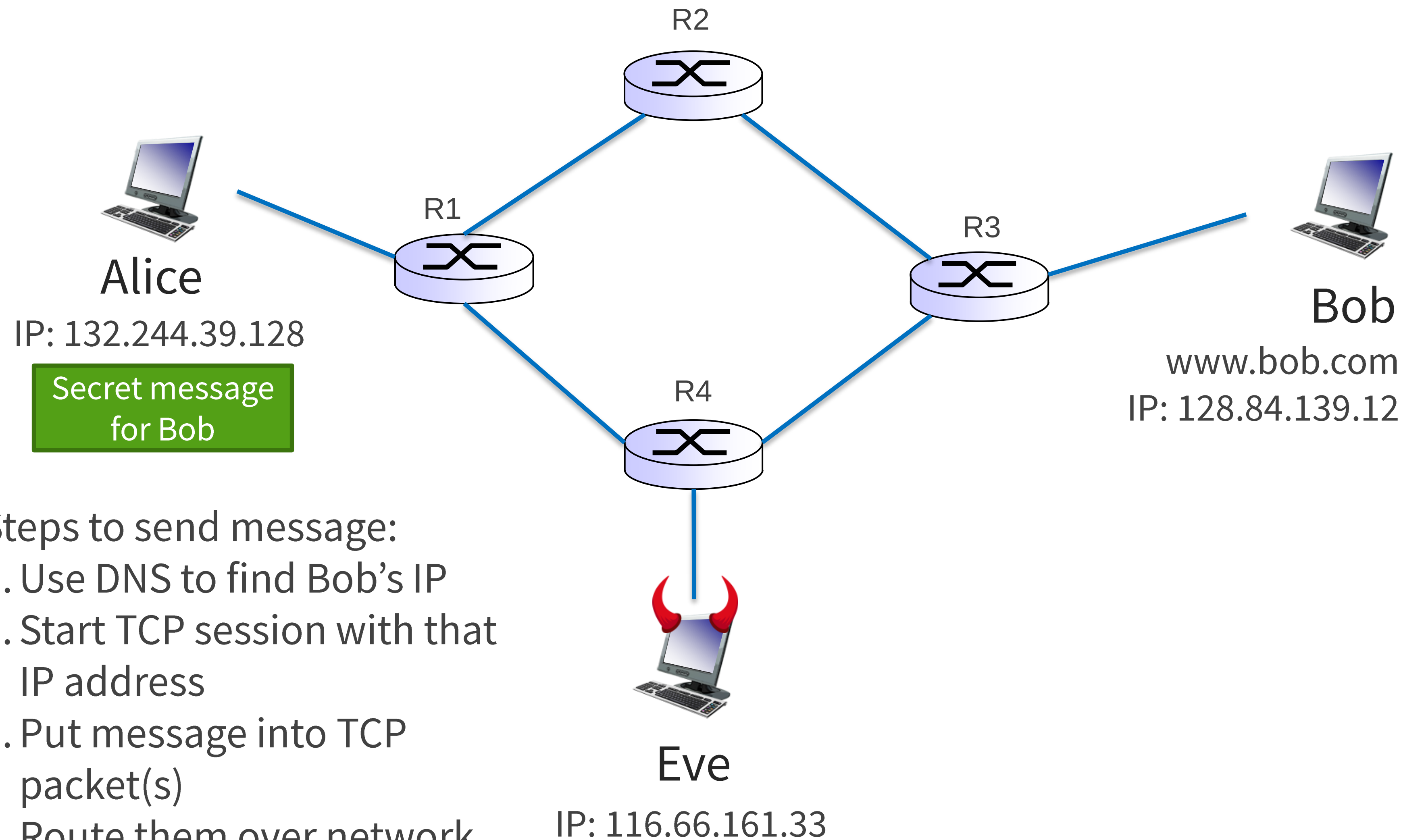
CS 4410
Operating Systems

Cornell University Founded A.D. 1865
Cornell CIS
Computing and Information Science

Security in Networking

- Network Vulnerabilities
- Secure Sockets (Encryption)
- Secure Naming (Certificates)

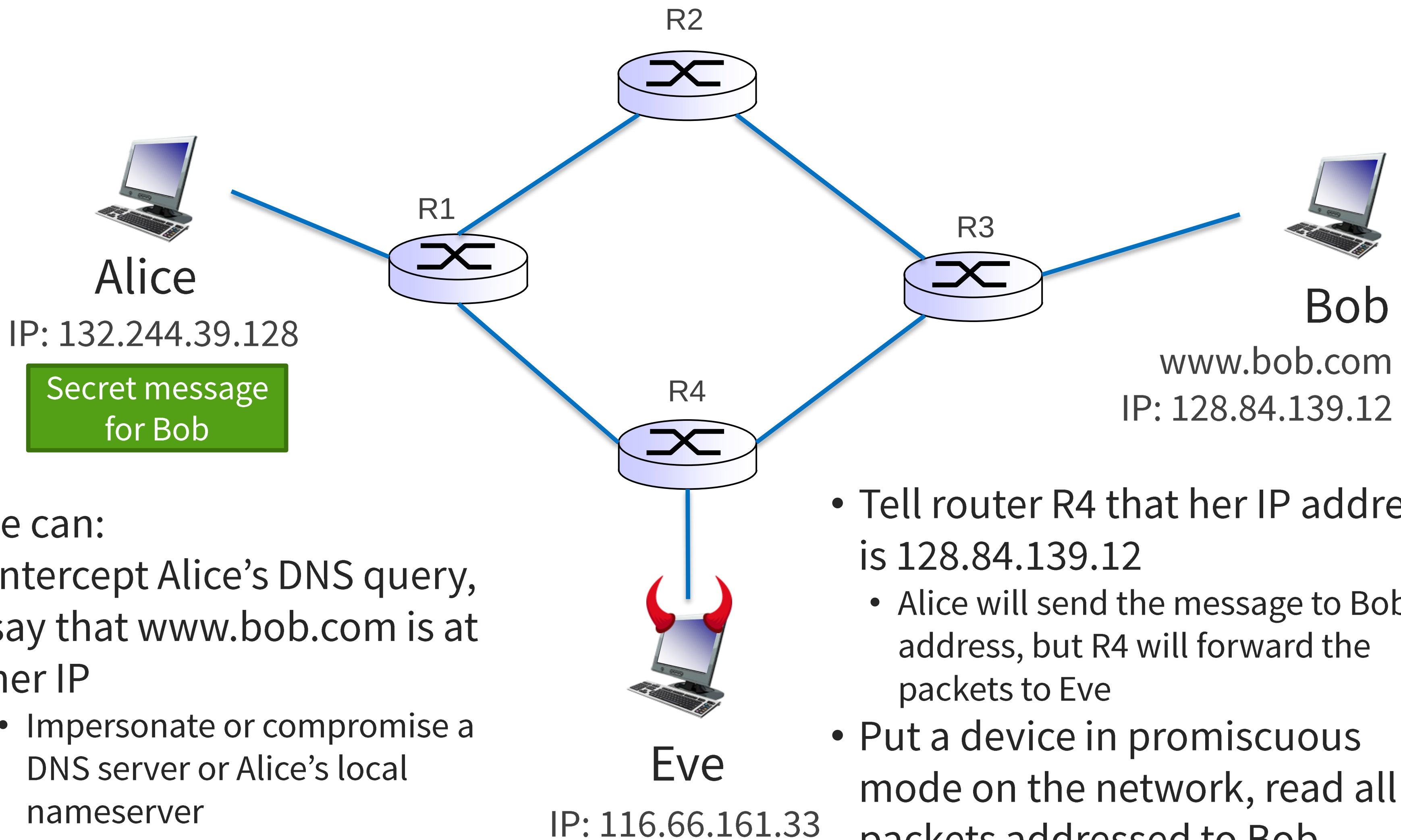
What Could Go Wrong?



Steps to send message:

1. Use DNS to find Bob's IP
2. Start TCP session with that IP address
3. Put message into TCP packet(s)
4. Route them over network

What Could Go Wrong?



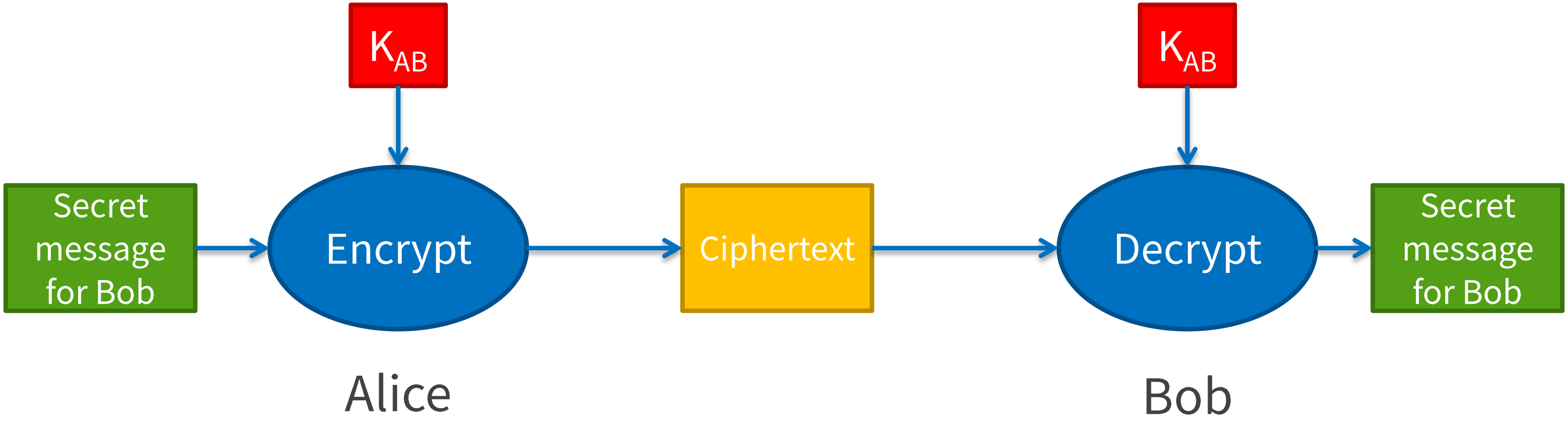
Eve can:

- Intercept Alice's DNS query, say that www.bob.com is at her IP
  - Impersonate or compromise a DNS server or Alice's local nameserver
  - Alice will send the message to 116.66.161.33 thinking it's Bob
- Tell router R4 that her IP address is 128.84.139.12
  - Alice will send the message to Bob's IP address, but R4 will forward the packets to Eve
- Put a device in promiscuous mode on the network, read all packets addressed to Bob
  - Bob gets the message, but Eve gets to read it too

Communicating Securely

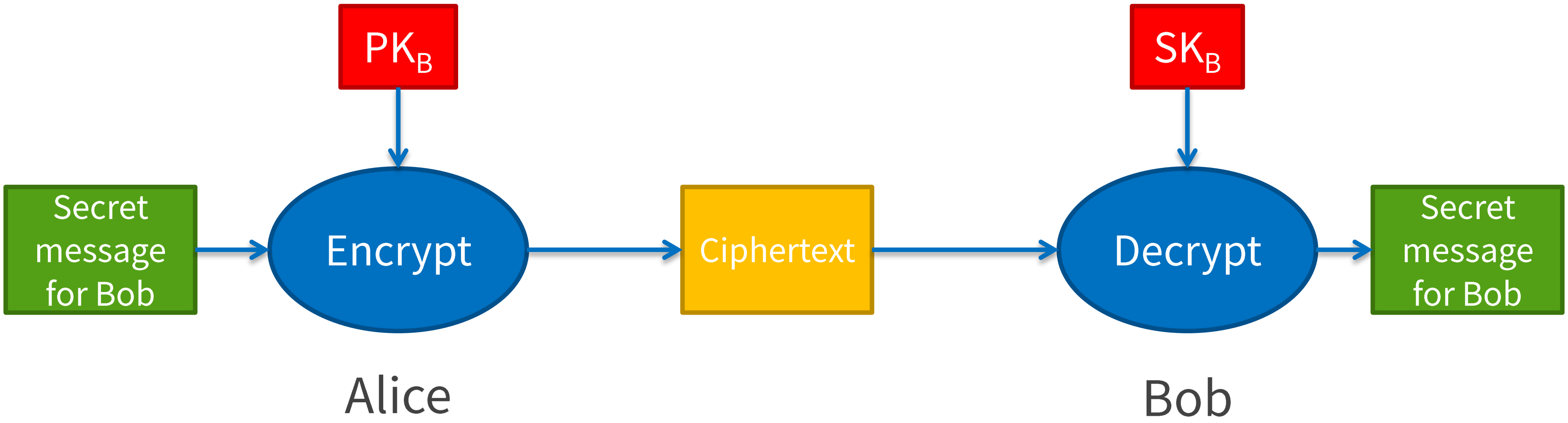
- Verify that the recipient is who they say they are (**Authentication**)
- Prevent anyone other than intended recipient from reading the message (**Authorization**)

First Step: Encrypt the Message



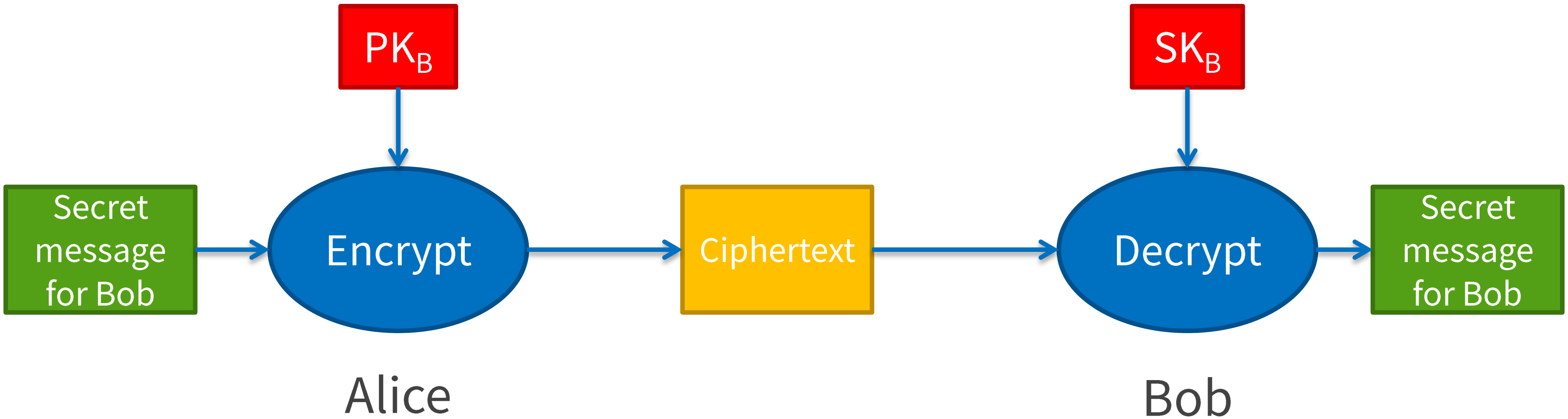
- Symmetric-Key Cryptography: Alice and Bob have a shared secret key
- Pro: Eve can't read the message even if she intercepts it
- Con: How does Alice share the key with Bob? Send it over the network?

Encrypting the Message



- Public-Key Cryptography
  - Bob tells everyone his public key (PK_B), Alice uses it to encrypt the message for him
  - Only Bob knows his private key (SK_B), which is necessary to decrypt the message

Encrypting the Message



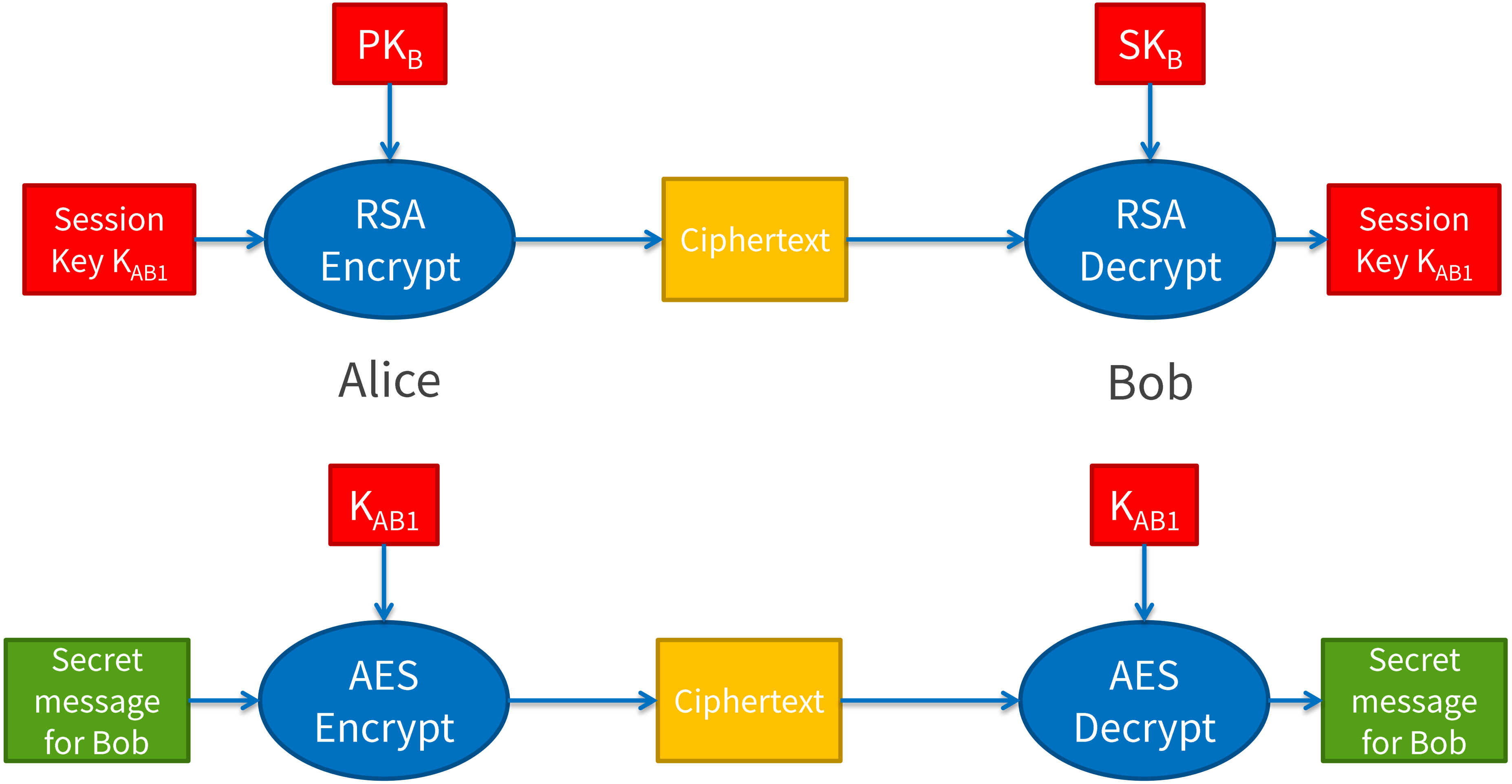
- Public-Key Cryptography
- Pro: No need to share a secret over the network
- Con: Public-key encryption (RSA) is **much** slower than symmetric encryption (AES)

Session Keys

- Combine public-key and symmetric encryption to get benefits of both

1. Use public-key encryption to safely send a **session key**: secret key for a symmetric cipher
2. Use symmetric encryption with the session key for subsequent messages

Session Keys

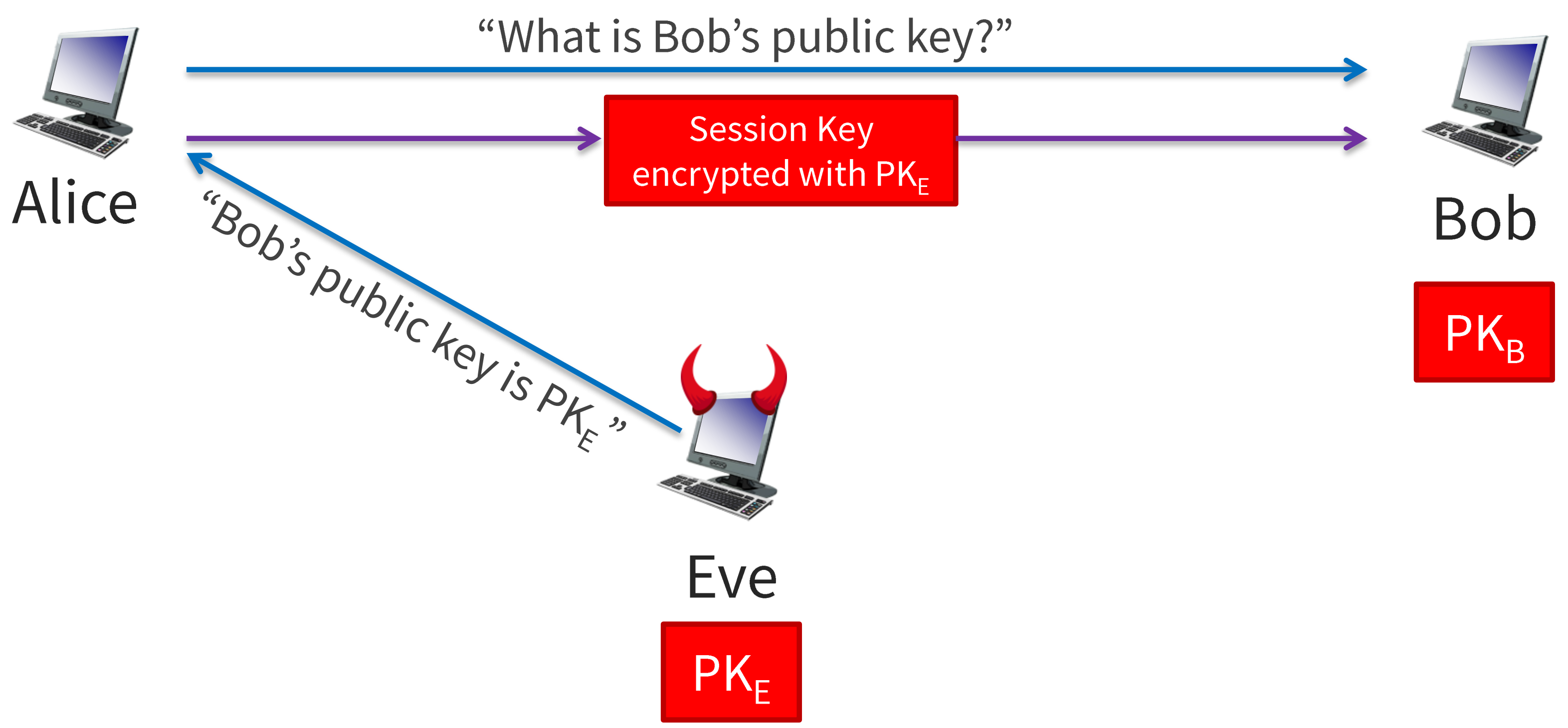


TLS: Transport Layer Security

- Originated with SSL (Secure Sockets Layer), now obsolete
- Runs on TCP connection
- Initial handshake to establish identity of server and create session key
- Subsequent TCP segments have data encrypted with session key

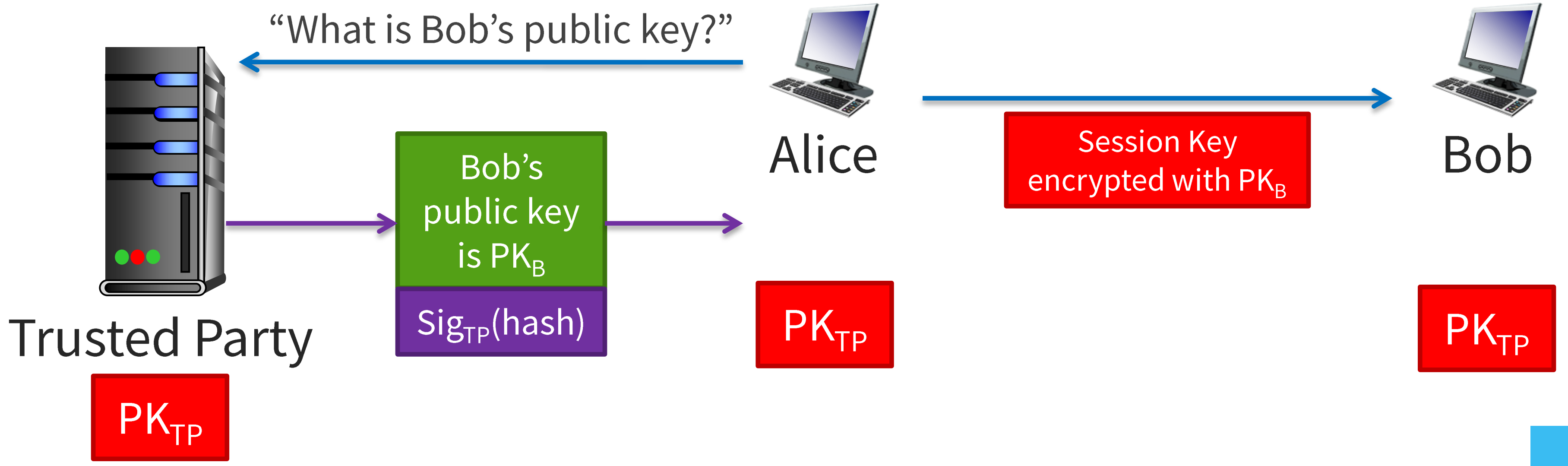
Establishing Identity

- How does Alice learn Bob's public key?
- What if Eve pretends to be Bob and presents her own public key?



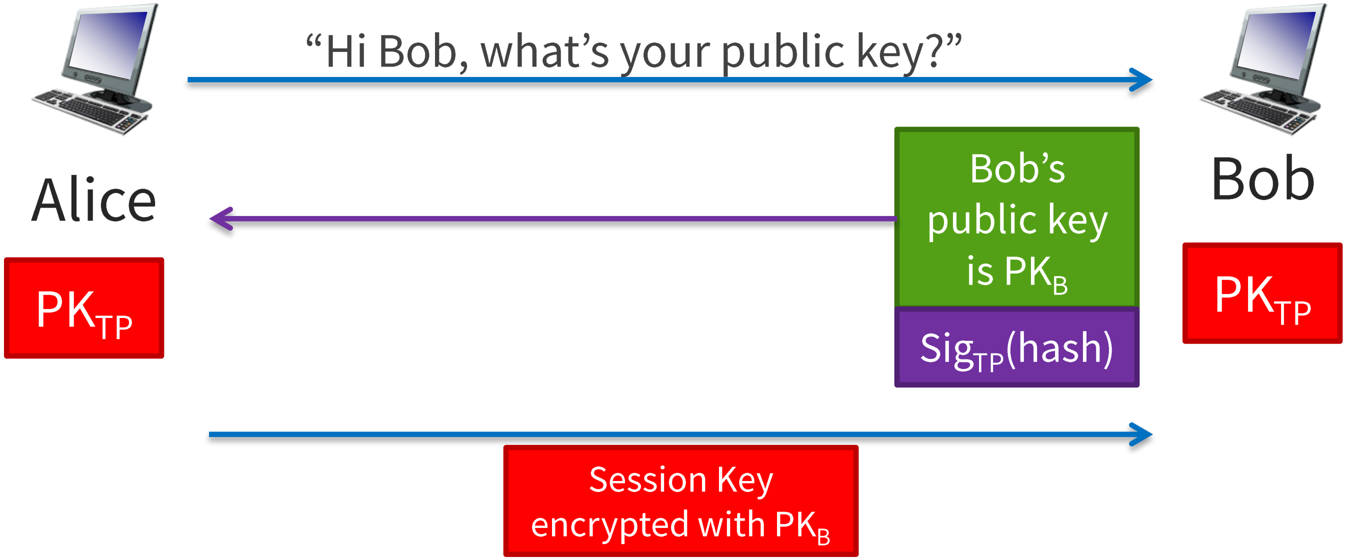
Digital Certificates

- Use a trusted third party to provide Bob's public key
- Pre-distribute or hard-code third party's public key (easier problem)
- Use digital signatures to ensure Eve can't impersonate trusted third party



Digital Certificates

- Signed message from trusted party is a certificate proving that this is Bob's key
- Bob can provide this certificate himself, no need to contact third party

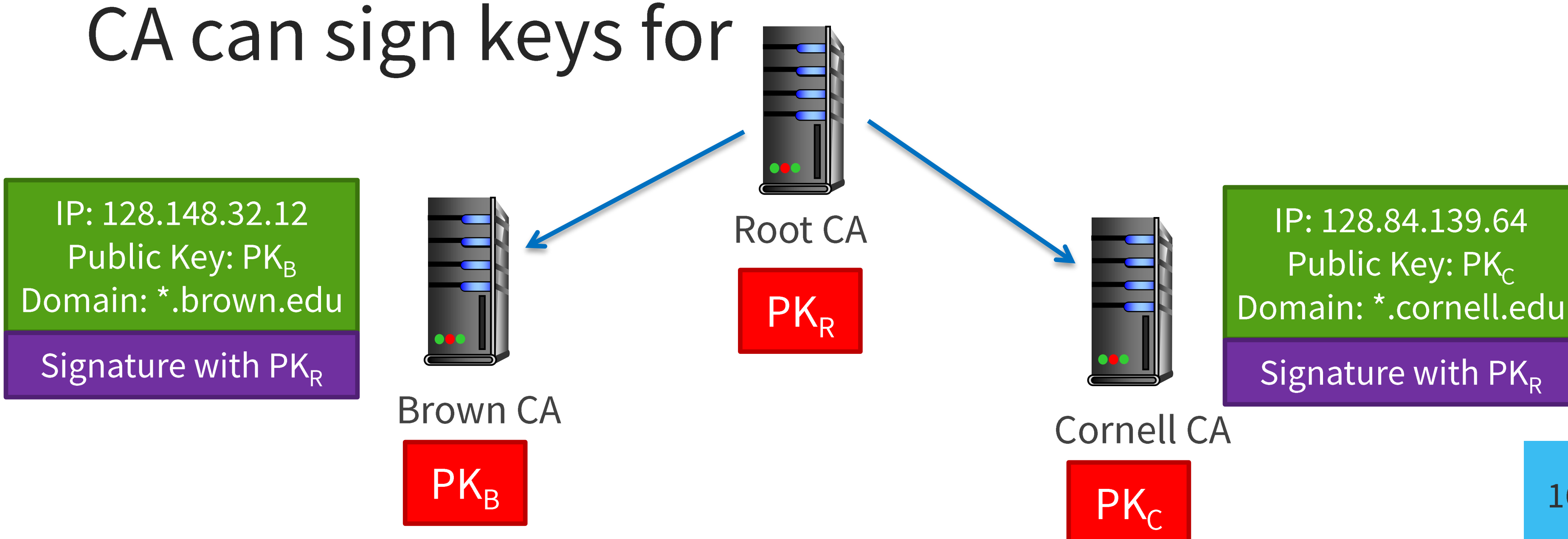


Certificate Authorities

- Trusted party that issues key certificates is a **certificate authority** (CA)
- CA's job is to verify Bob's identity, then sign a certificate for his public key
- Public keys for CAs must be manually installed

Certificate Authority Hierarchy

- Small number of hard-coded “root” CAs
  - Just like DNS root servers
- A CA can sign certificates for other CAs
  - Once you know one CA’s public key, you can use it to discover others (just like DNS)
- CA certificate describes what domains this CA can sign keys for



TLS in More Detail