Security: Securing Protocols, HTTPS, IPSec, Operational Security, Firewalls

Michael Kirkedal Thomsen

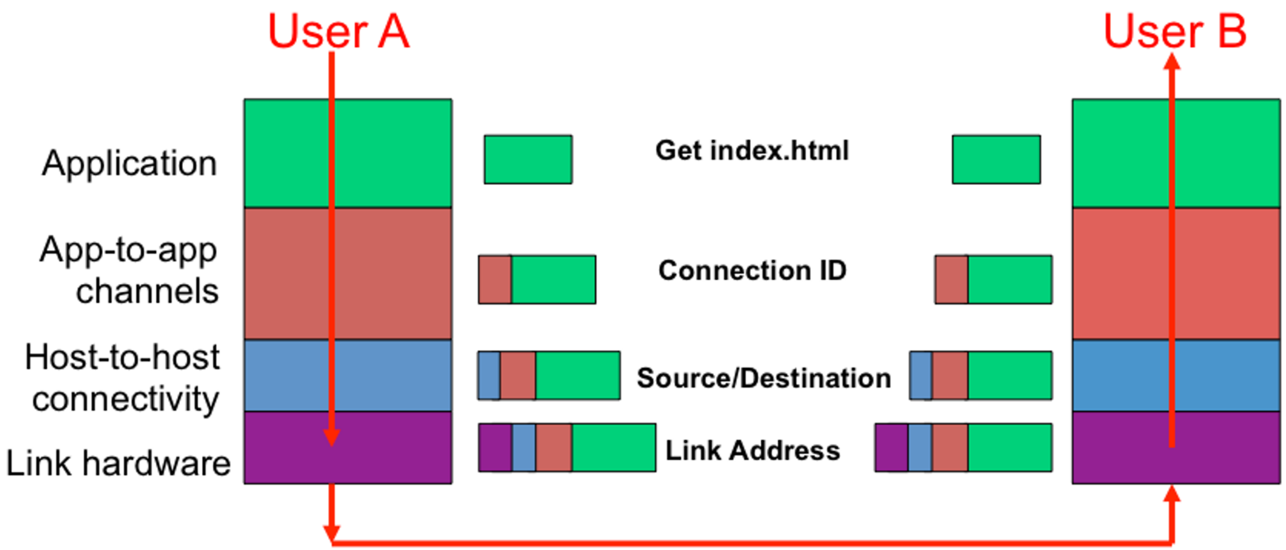
Based on slides compiled by Marcos Vaz Salles with modifications by Vivek Shah

What should you be able to do after today?

- List security properties and related attacks
- Relate basic cryptographic schemes to their use in network protocols
- Explain the main mechanisms of HTTPS
- Explain the motivation and uses of IPSec
- Discuss operational security concerns and solutions, such as firewalls

Do-It-Yourself Recap: HTTP

- What transport protocol does HTTP use? Can an unauthorized party read the content of HTTP requests/responses?
- Which two types of performance optimizations are common with HTTP and web applications?



Source: Freedman (partial)

Nothing is secure forever

"You have 1 minute to design a maze that takes 2 minutes to solve" – some scriptwriter

Source: Freedman (partial)

Internet's Design: Insecure

- Designed for simplicity
- "On by default" design
- Readily available zombie machines
- Attacks look like normal traffic
- Internet's federated operation obstructs cooperation for diagnosis/mitigation

Source: Freedman (partial)

Security Properties

- **Confidentiality:** Concealment of information or resources
- **Authenticity:** Identification and assurance of origin of info
- **Integrity:** Trustworthiness of data or resources in terms of preventing improper and unauthorized changes
- **Availability:** Ability to use desired info or resource
- **Non-repudiation:** Offer of evidence that a party indeed is sender or a receiver of certain information

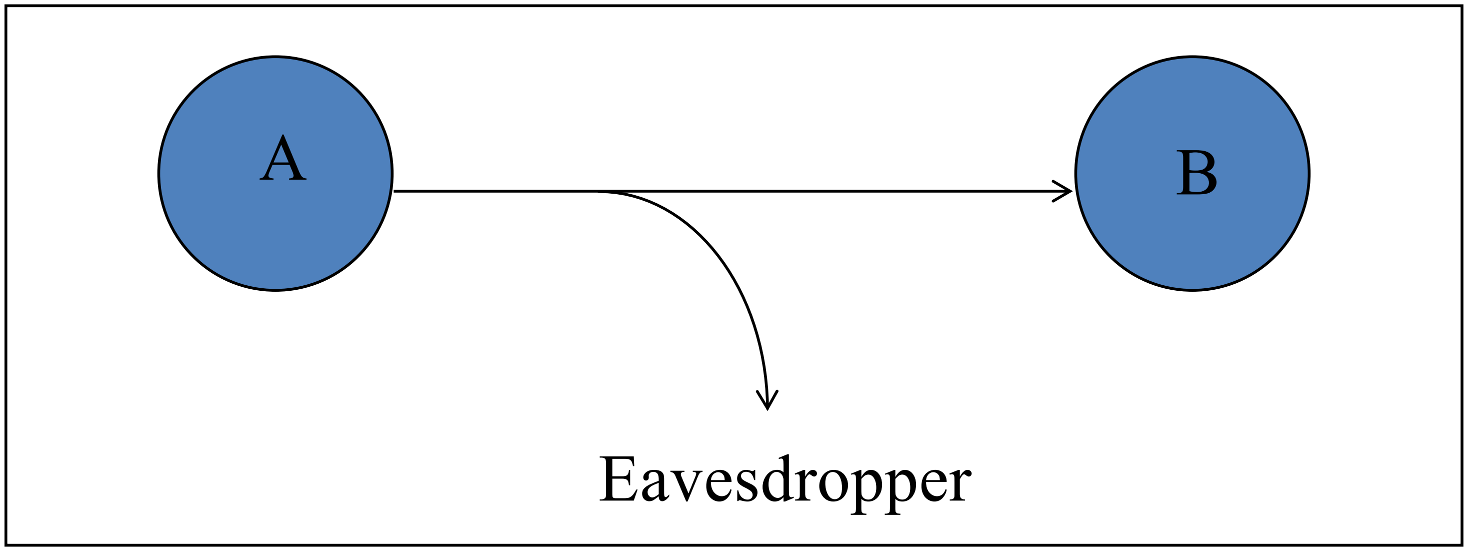
---

- **Access control:** Facilities to determine and enforce who is allowed access to what resources (host, software, network, …)

Source: Freedman (partial)

Eavesdropping - Message Interception (Attack on Confidentiality)

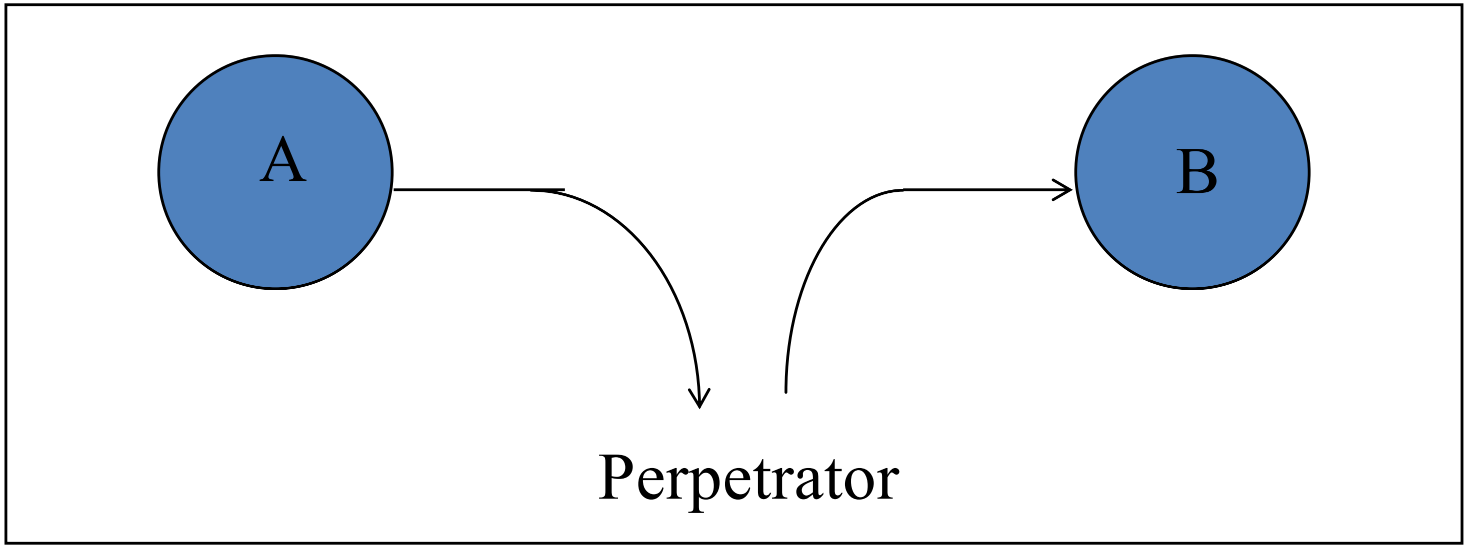
- Unauthorized access to information
- Packet sniffers and wiretappers (e.g. tcpdump)
- Illicit copying of files and programs



Source: Freedman

Integrity Attack - Tampering

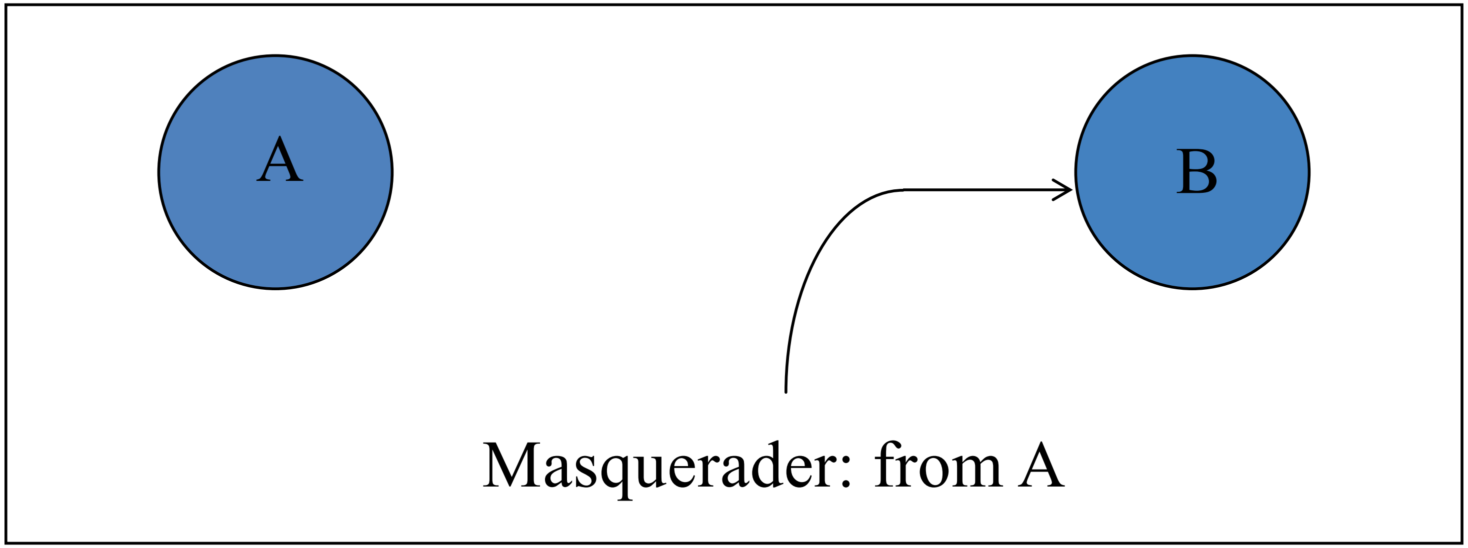
- Stop the flow of the message
- Delay and optionally modify the message
- Release the message again



Source: Freedman

Authenticity Attack - Fabrication

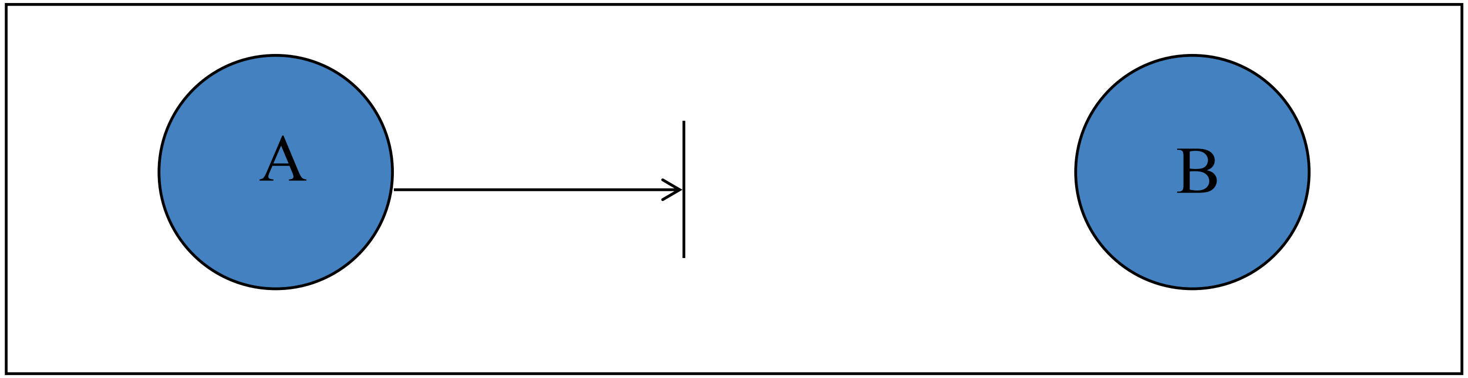
- Unauthorized assumption of other’s identity
- Generate and distribute objects under identity



Source: Freedman

Attack on Availability

- Destroy hardware (cutting fiber) or software
- Modify software in a subtle way
- Corrupt packets in transit



- Blatant *denial of service* (DoS):
  - Crashing the server
  - Overwhelm the server (use up its resource)
  - DDoS is a distributed DoS attack involving multiple machines from which the attack originates

Source: Freedman

Impact of Attacks

- Theft of confidential information
- Unauthorized use of
  - Network bandwidth
  - Computing resource
- Spread of false information
- Disruption of legitimate services

Source: Freedman

What technical means can we utilize to ensure data confidentiality?

Cryptography!

What is Cryptography?

- Comes from Greek word meaning "secret"
  - Primitives also can provide integrity, authentication

- Cryptographers invent secret codes to attempt to hide messages from unauthorized observers

$$\textbf{plaintext} \xrightarrow{\textbf{encryption}} \textbf{ciphertext} \xrightarrow{\textbf{decryption}} \textbf{plaintext}$$

- Modern encryption:
  - *Algorithm* public, *key* secret and provides security
  - May be symmetric (secret) or asymmetric (public)

Source: Freedman

What is Cryptography?

Modern cryptography relies on the assumption that:

- The algorithms are publicly known
- The key must be protected

Symmetric	**Assymetric**
Uses a single, secret key for both encryption and decryption	Uses a public key for encryption and a secret private key for decryption
Is fast due to small key sizes	Is slow due to larger keys (Relies on Integer factorization, Discrete Logarithms, or Elliptic curves)
Must be known by communicating parties in advance	Does not require sharing anything in advance

Source: Freedman, Kurose & Ross

Three types of functions serving different purposes

- **Cryptographic hash Functions**
  - **No keys at all**
  - **Used to ensure data integrity/authenticity but not confidentiality**
  - **Very quick**

- Secret-key functions (Symmetric key)
  - One key
  - Very secure, very difficult to distribute
  - Quick, compared to Asymmetric Keys

- Public-key functions (Asymmetric key)
  - Two keys
  - Very secure, easy to distribute
  - Very slow

Source: Freedman

Use of encryption and MAC/signatures

Confidentiality (Encryption)

Sender:

- Compute C = $Enc_K(M)$
- Send C

Receiver:

- Recover M = $Dec_K(C)$

Auth/Integrity (MAC / Signature)

Sender:

- Alice knows, s = shared secret
- Compute mac = Hash (M + s)
- Send <M, mac>

Receiver:

- Bob also knows s
- Compute mac' = Hash(M + s)
- Check mac' == mac

These are simplified forms of the actual algorithms

Secrets are often added to make these more secure

Source: Freedman, Kurose & Ross

Using Keys

- **Secret-key functions**

  secret *s* must be known by both ahead of time

client			receiver	
m	$M=s(m)$	M	$m=s(M)$	m

- **Public-key functions**

  reciever has a public key r_+ and private key r_-

client			reciever	
m	$M=r^+(m)$	M	$m=r^-(M)$	m

Source: Freedman

HTTP and security

- How do you secure HTTP?
- Ensure sites are the sites you really requested
- Ensure no one else can read or forge requests/responses

"Securing" HTTP

- Threat model
  - Eavesdropper listening on conversation (confidentiality)
  - Man-in-the-middle modifying content (integrity)
  - Adversary impersonating desired website (authentication, and confidentiality)

- Enter HTTP-S
  - HTTP sits on top of secure channel (SSL/TLS)
  - All (HTTP) bytes written to secure channel are encrypted and authenticated
  - Problem: What is actually authenticated to prevent impersonation? Which keys used for crypto protocols?

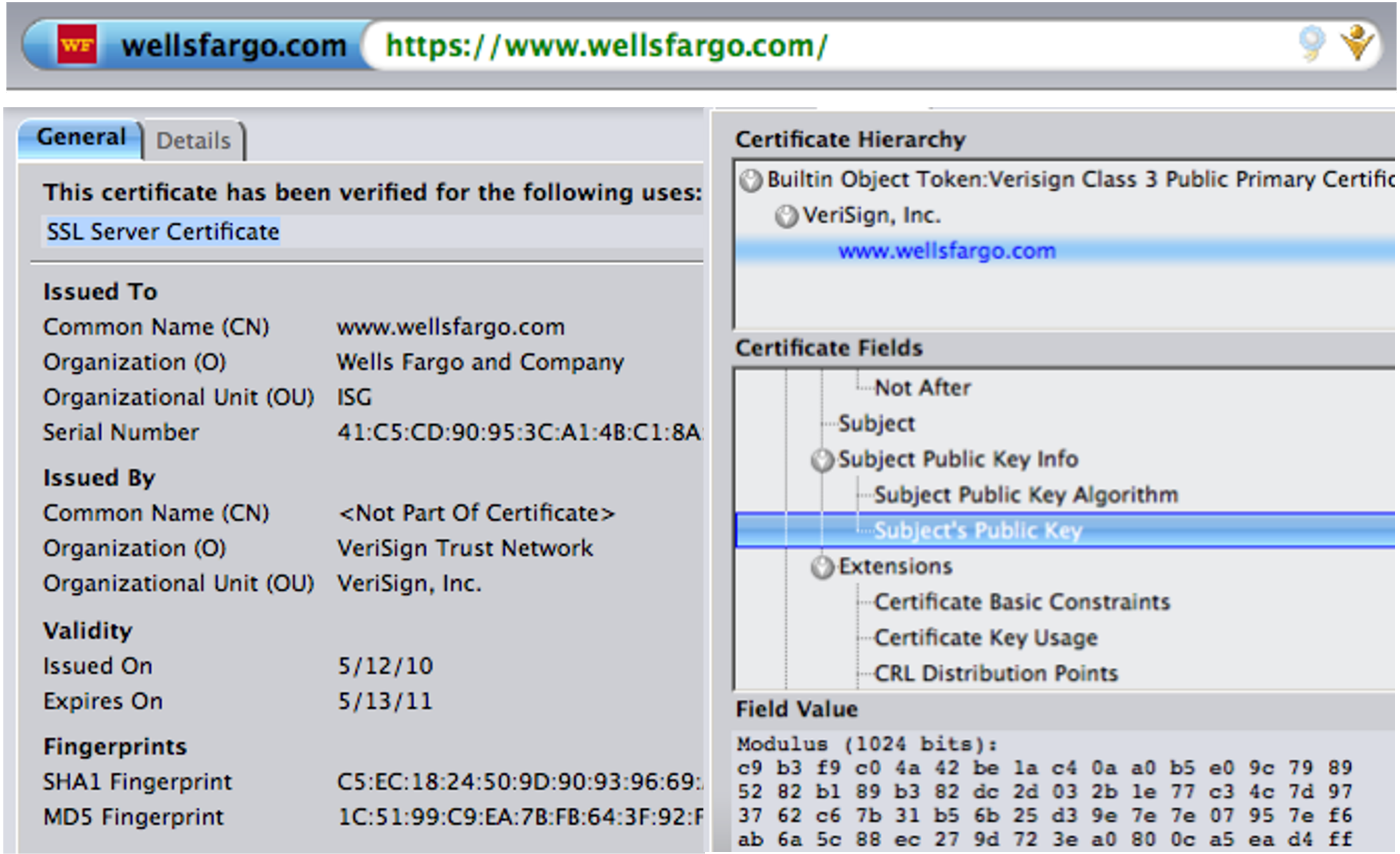
Source: Freedman

Learning a valid public key

- What is that lock?
- Securely binds domain name to public key (PK)
- Believable only if you trust the attesting body
- Bootstrapping problem: Who to trust, and how to tell if this message is actually from them?
- If PK is authenticated, then any message signed by that PK cannot be forged by non-authorized party

Source: Freedman

How to authenticate PK



Source: Freedman

SSL and TCP/IP

Application	Application
	SSL
TCP	TCP
IP	IP
normal application	*application with SSL*

- SSL provides application programming interface (API) to applications.
- C, Python and Java SSL libraries/classes readily available

Source: Kurose & Ross

Transport Layer Security (TLS) – Replaces SSL

RSA or Diffie-Hellman used for shared key negotiation

- Send new random value, list of supported ciphers
- Send pre-secret, encrypted under PK
- Create shared secret key from pre-secret and random
- Switch to new symmetric-key cipher using shared key

- Send new random value, digital certificate with PK
- Create shared secret key from pre-secret and random
- Switch to new symmetric-key cipher using shared key

Source: Freedman (partial)

Source: Freedman (partial)

Comments on HTTPS

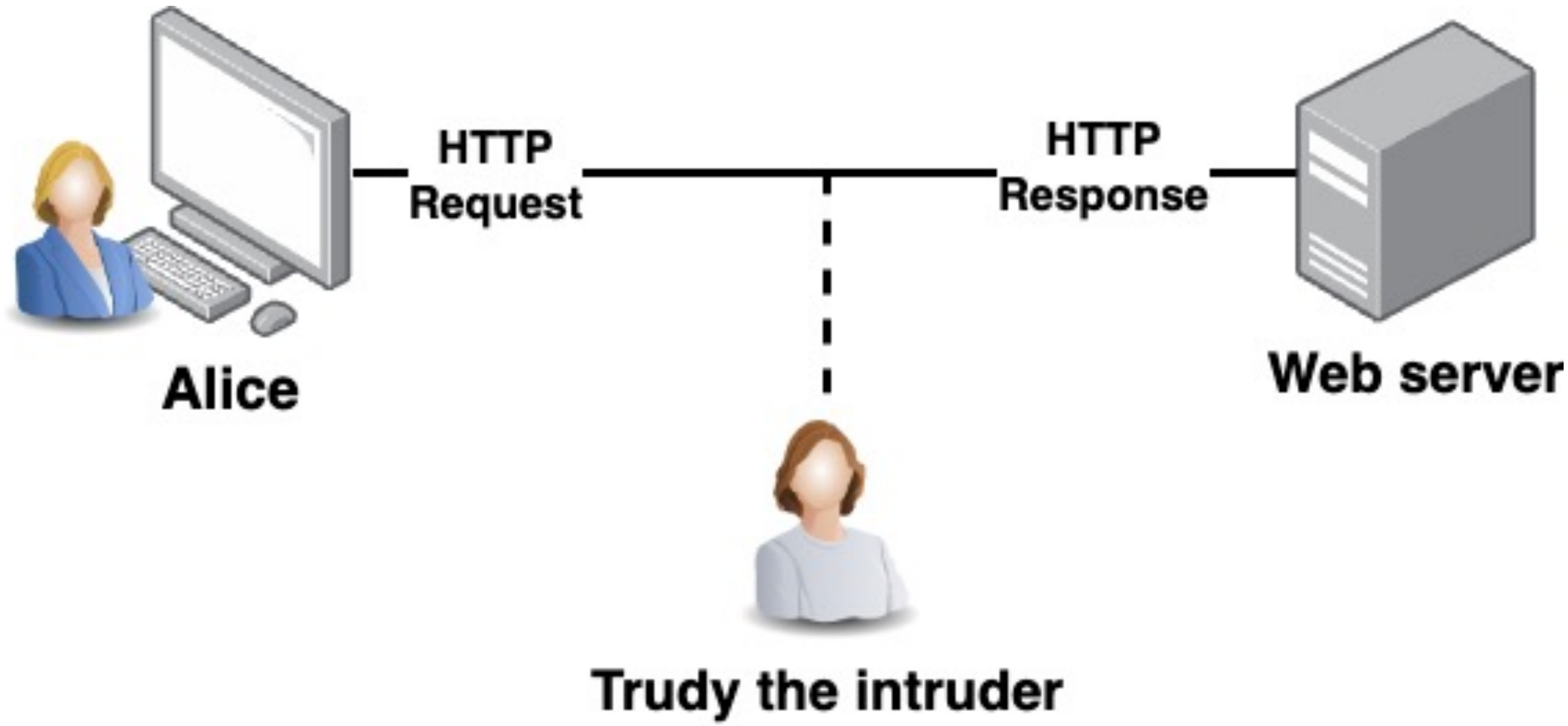
- Note that HTTPS authenticates server, not content
  - If CDN (Akamai) serves content over HTTPS for its customers, customer must trust Akamai not to change content

- Switch to symmetric-key crypto after public-key ops
  - Symmetric-key crypto much faster (100-1000x)
  - PK crypto can encrypt message only approx. as large as key (1024 bits – this is a simplification) – afterwards uses hybrid

- HTTPS on top of TCP, so reliable byte stream
  - Can leverage fact that transmission is reliable to ensure: each data segment received exactly once
  - Adversary can't successfully drop or replay packets

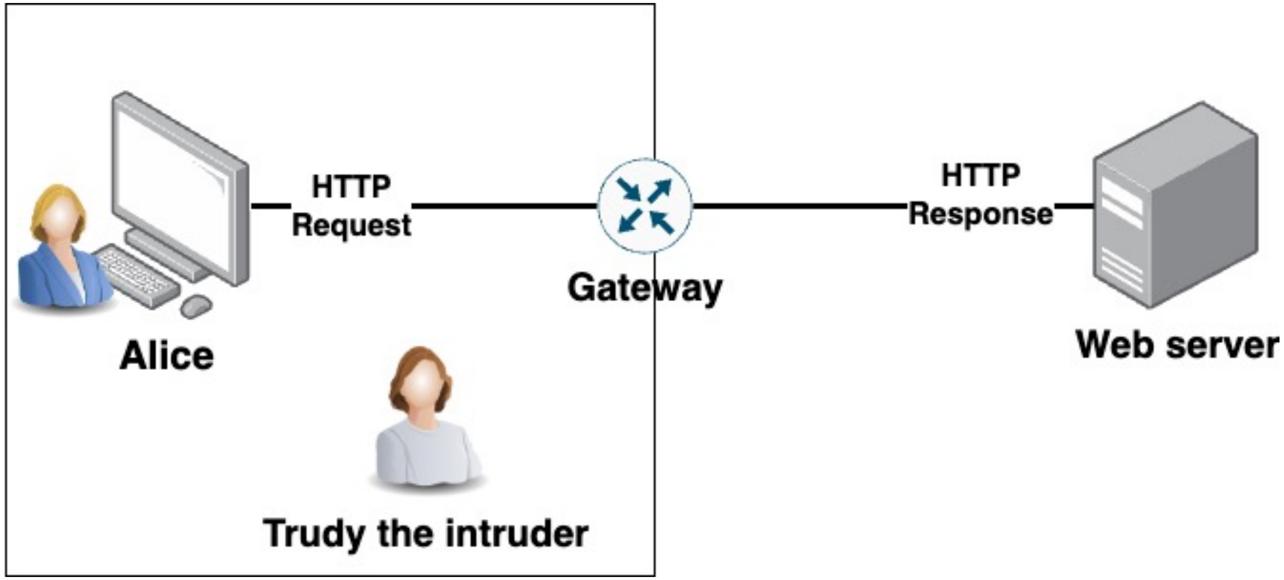
Man-in-the-middle attacks in the real world

- Alice communicates with a web server over HTTP
- Hackers objective: Intercept the communication between Alice and the web server



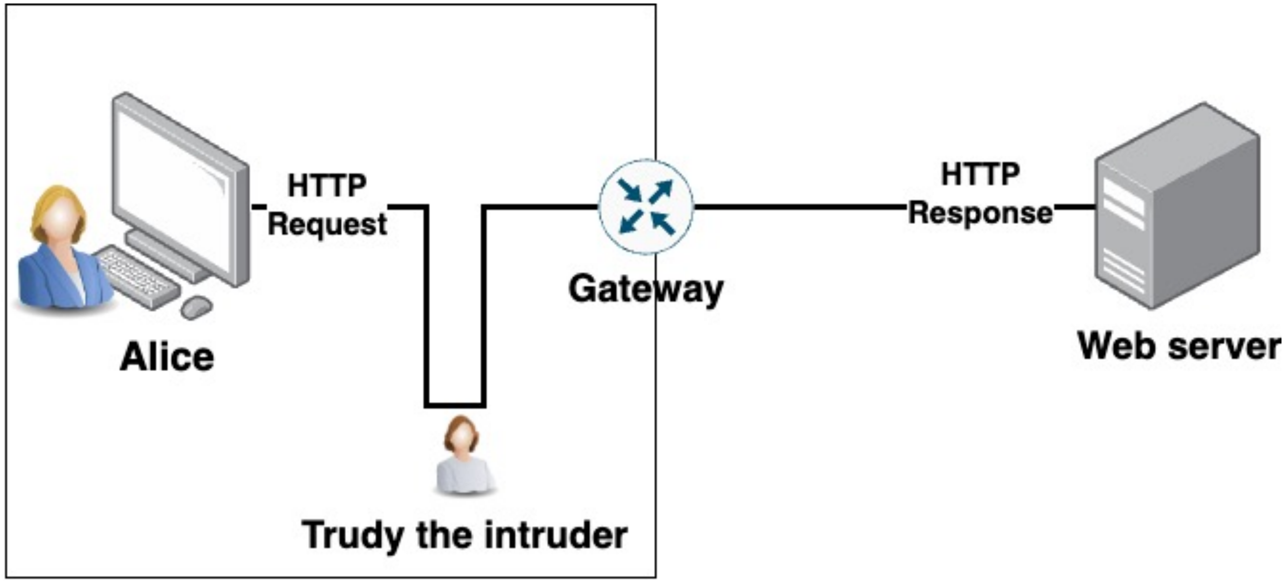
HTTP traffic interception recipe

- Route the traffic between Alice and the server through Trudys machine
  - Exploit a link layer weakness where we can fool Alice into thinking Trudy is the gateway
  - Intercept the TCP packets to/from Alice using Wireshark
  - Set proper filters to find interesting data, such as passwords, etc.



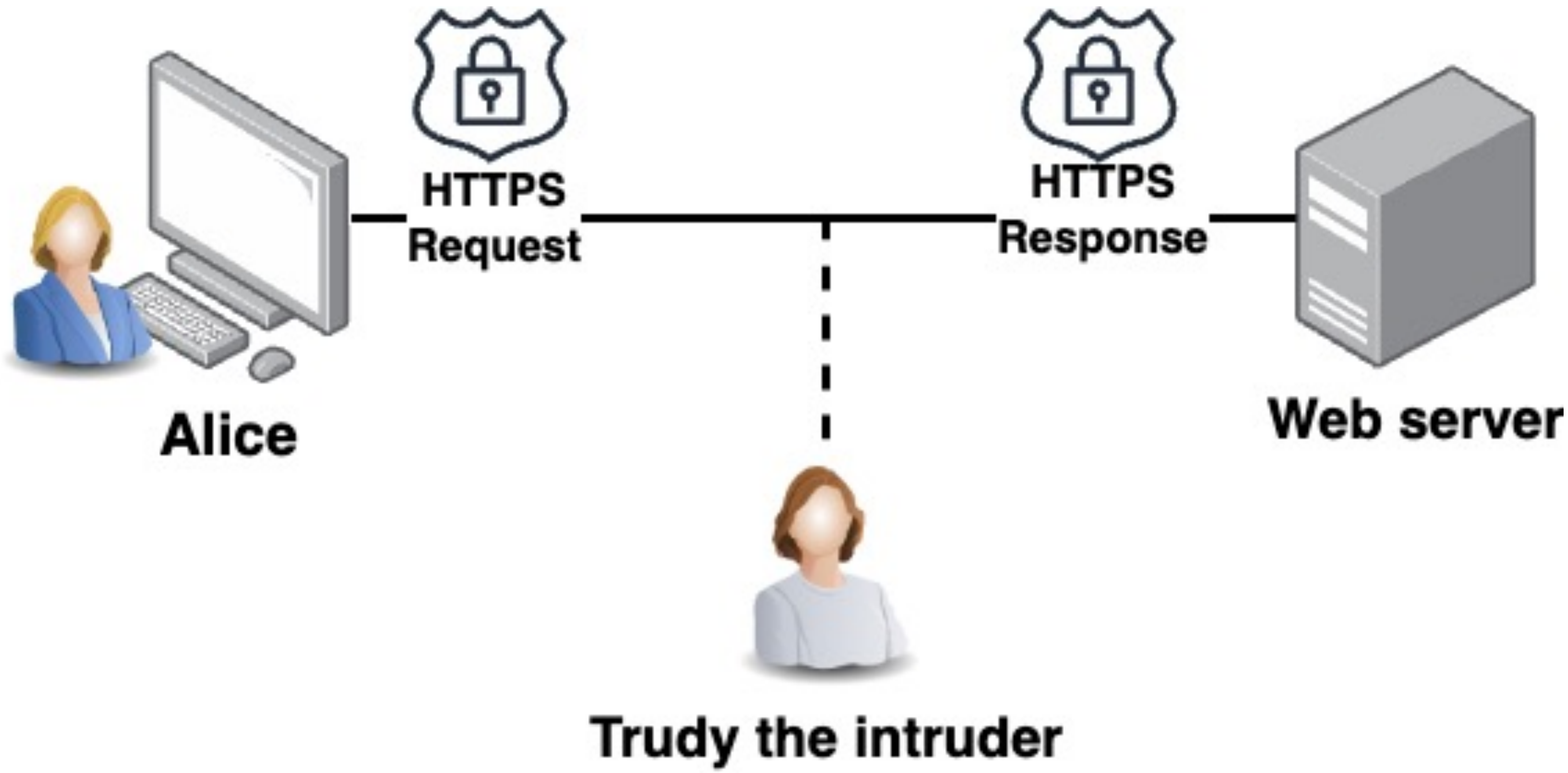
HTTP traffic interception recipe

- Route the traffic between Alice and the server through Trudys machine
  - Exploit a link layer weakness where we can fool Alice into thinking Trudy is the gateway (ARP spoofing)
  - Intercept the TCP packets to/from Alice using Wireshark
  - Set proper filters to find interesting data, such as passwords, etc.



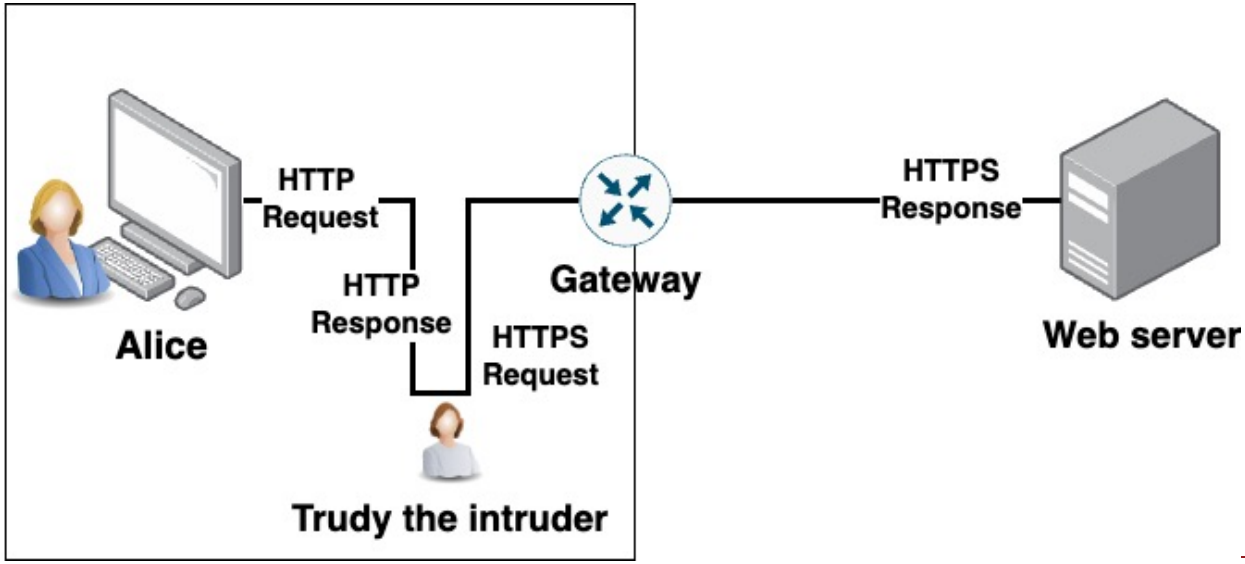
Man-in-the-middle attacks in the real world

- Alice now choses to use HTTPS
- Hackers objective: Intercept the communication between Alice and the web server
- It would be infeaseable to attempt brute-forcing the encryption

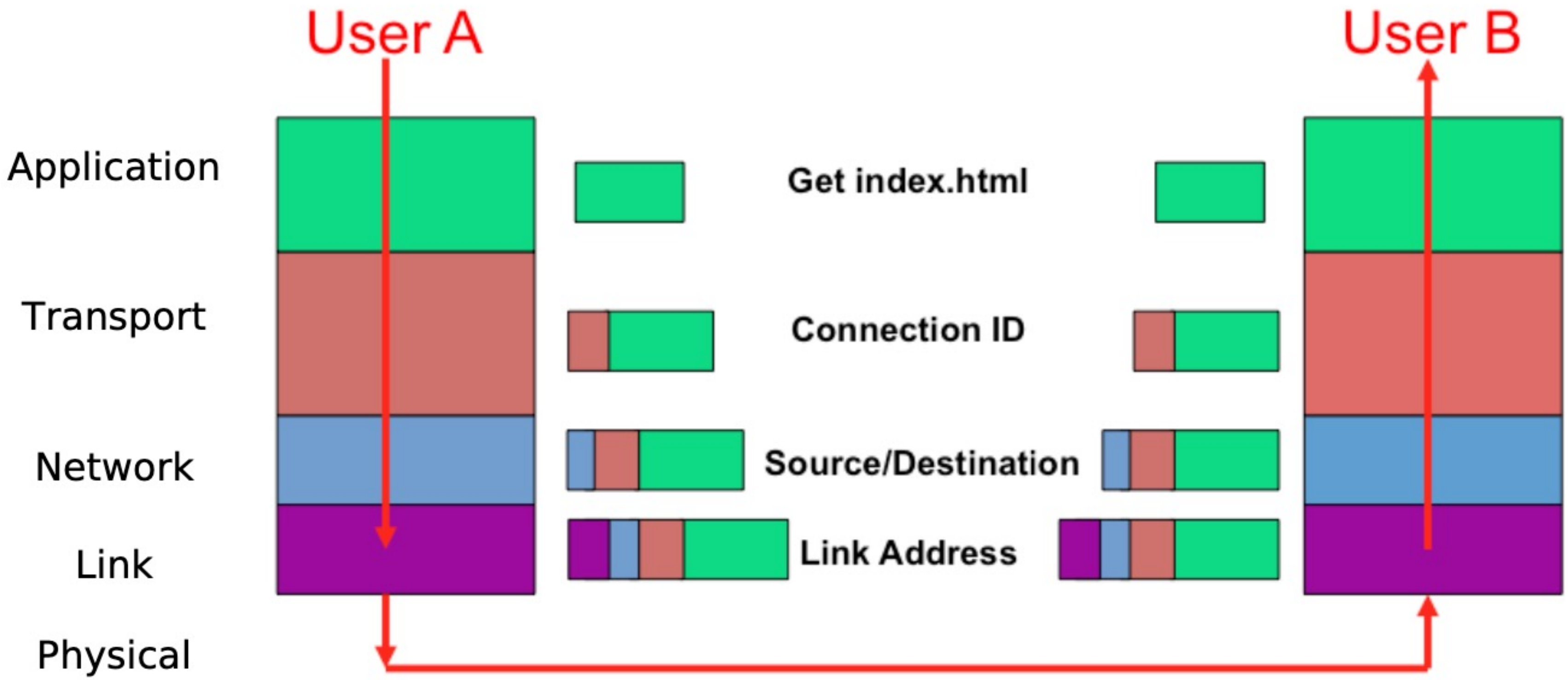


HTTPS traffic interception recipe

- Route the traffic between Alice and the server through Trudys machine
  - Exploit a link layer weakness where we can fool Alice into thinking Trudy is the gateway (ARP spoofing)
  - Trudy acts as a proxy, intercepting Alices HTTP request. Posing as the user, Alice establishes an HTTPS connection to the server.
  - Most users wont understand the error generated by the browser, and they will simply ignore it



Security at the network layer



Security at the core of the OSI model – the network layer

IP Security

- There are range of app-specific security mechanisms
  - eg. TLS/HTTPS, S/MIME, PGP, Kerberos,
- But security concerns that cut across protocol layers
- Implement by the network for all applications?

Enter IPSec!

Source: Freedman

IPSec

- General IP Security mechanism framework

- Allows one to provide
  - Access control, integrity, authentication, originality, and confidentiality

- Applicable to different settings
  - Narrow streams: Specific TCP connections
  - Wide streams: All packets between two gateways

Source: Freedman

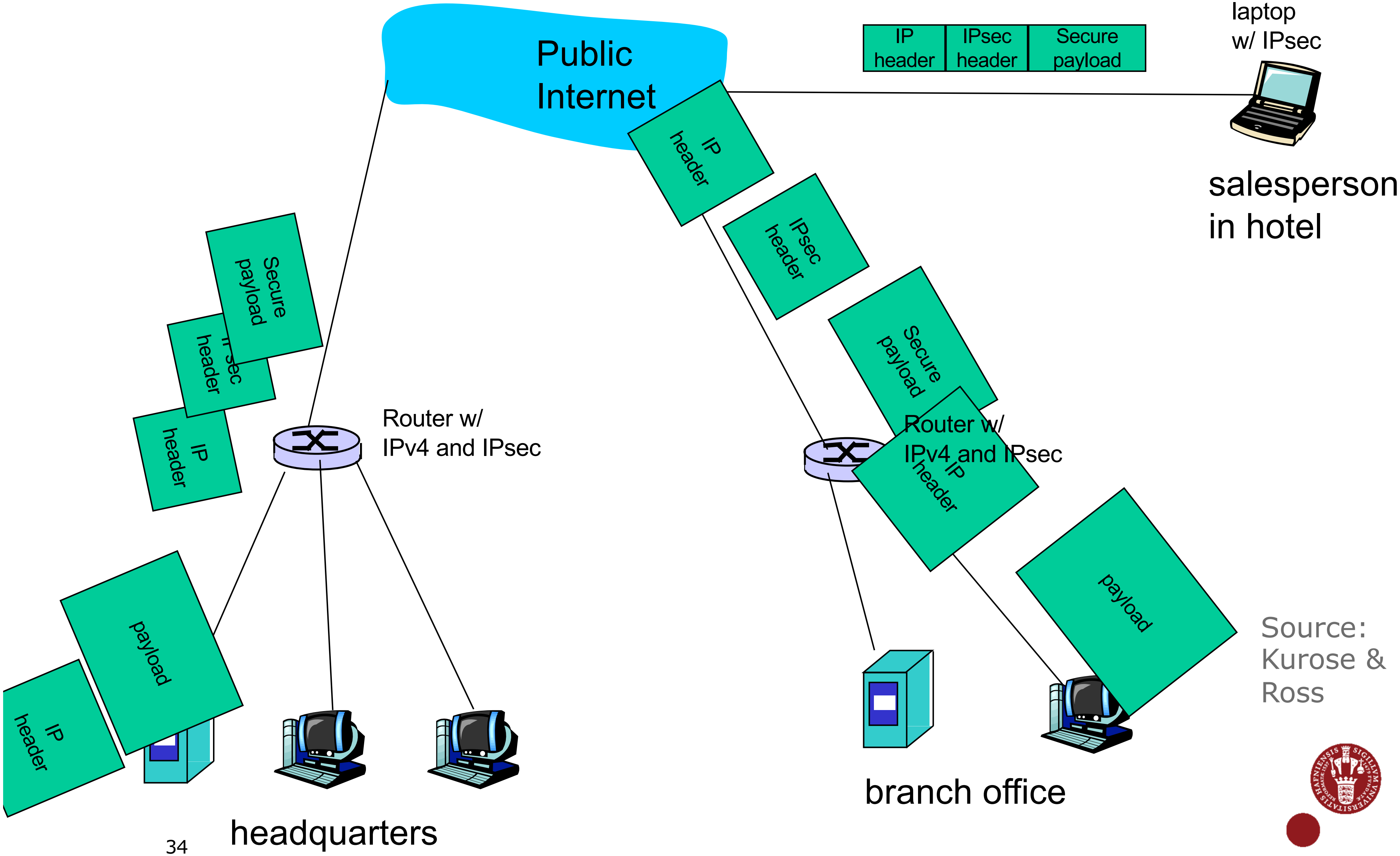
Virtual Private Networks (VPNs)

- institutions often want private networks for security.
  - costly: separate routers, links, DNS infrastructure.

- VPN: institution's inter-office traffic is sent over public Internet instead
  - encrypted before entering public Internet
  - logically separate from other traffic

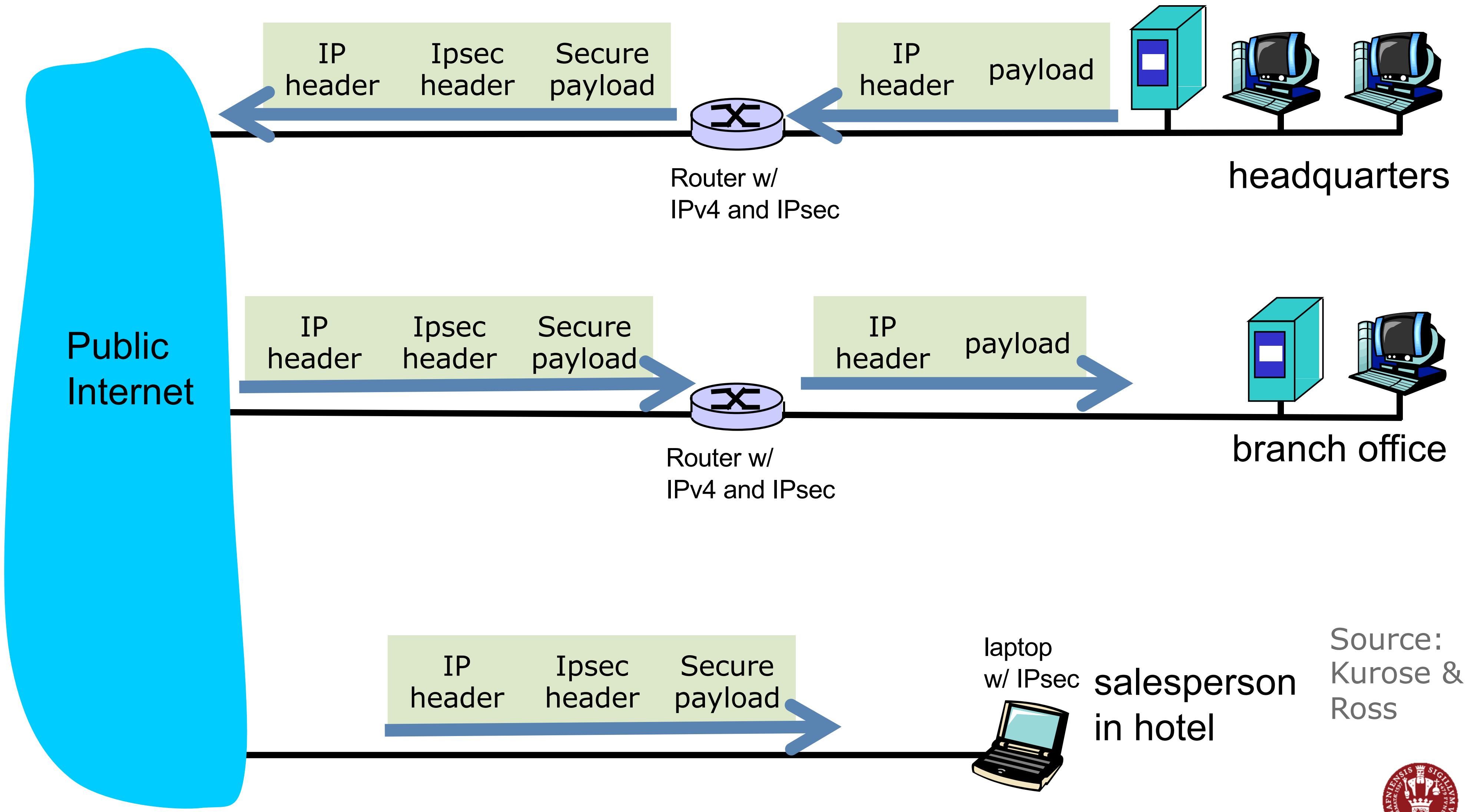
Source: Kurose & Ross

Virtual Private Network (VPN)



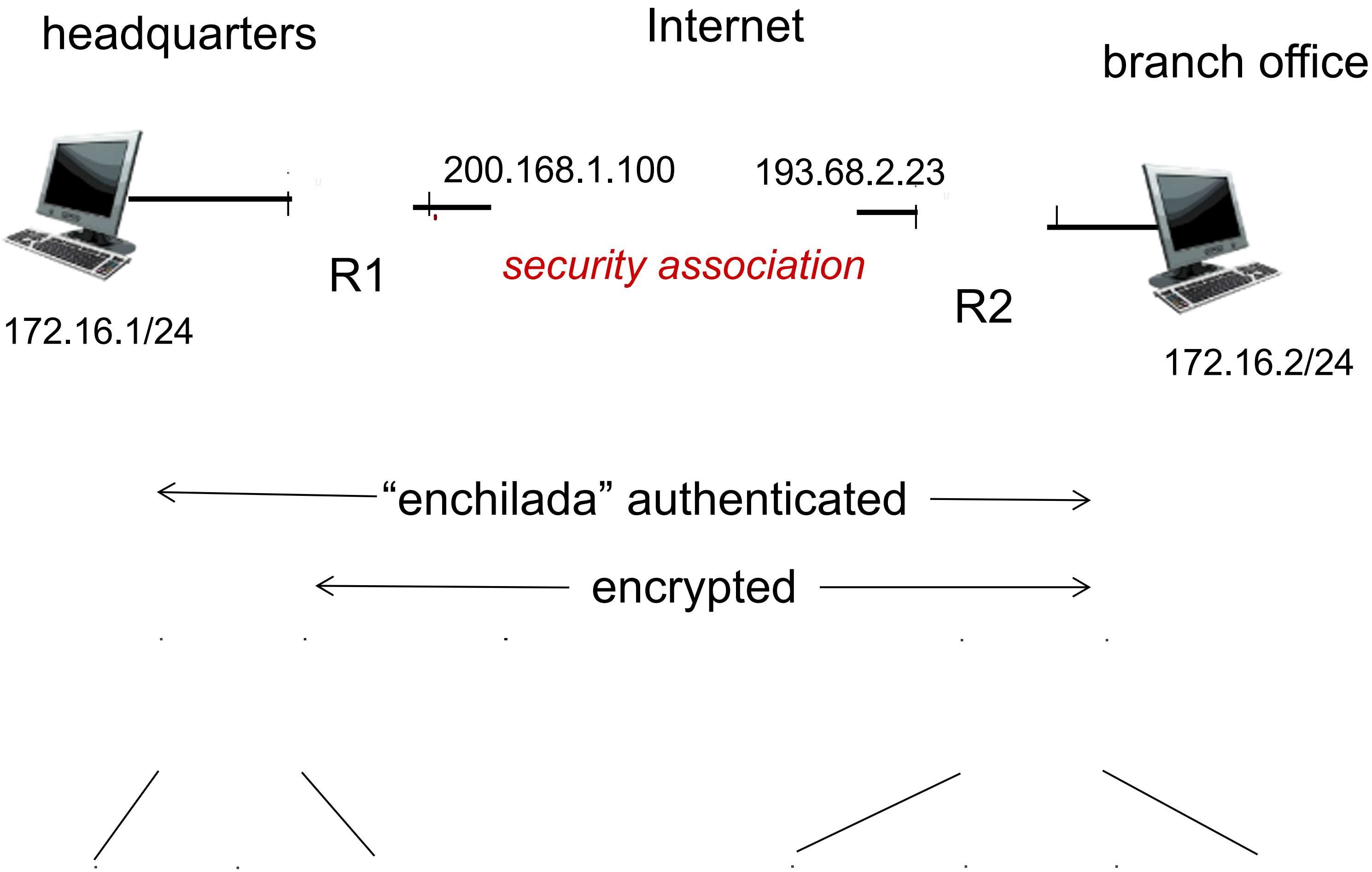
Source: Kurose & Ross

Virtual Private Network (VPN)



Source: Kurose & Ross

IPSec (What happens)



Source: Kurose & Ross

IP Security Architecture

- Specification quite complex (incl. RFC 2401, 2402, 2406, 2408)
  - Mandatory in IPv6, optional in IPv4
- Two security header extensions:
  - Authentication Header (AH)
    - Connectionless integrity, origin authentication
    - MAC over most header fields and packet body
    - Anti-replay protection
  - Encapsulating Security Payload (ESP)
    - These properties, plus confidentiality

Source: Freedman (partial)

Operational Security

- Ensure certain classes of attacks not possible by special rules and/or architectural decisions

- Rules often implemented in middleboxes

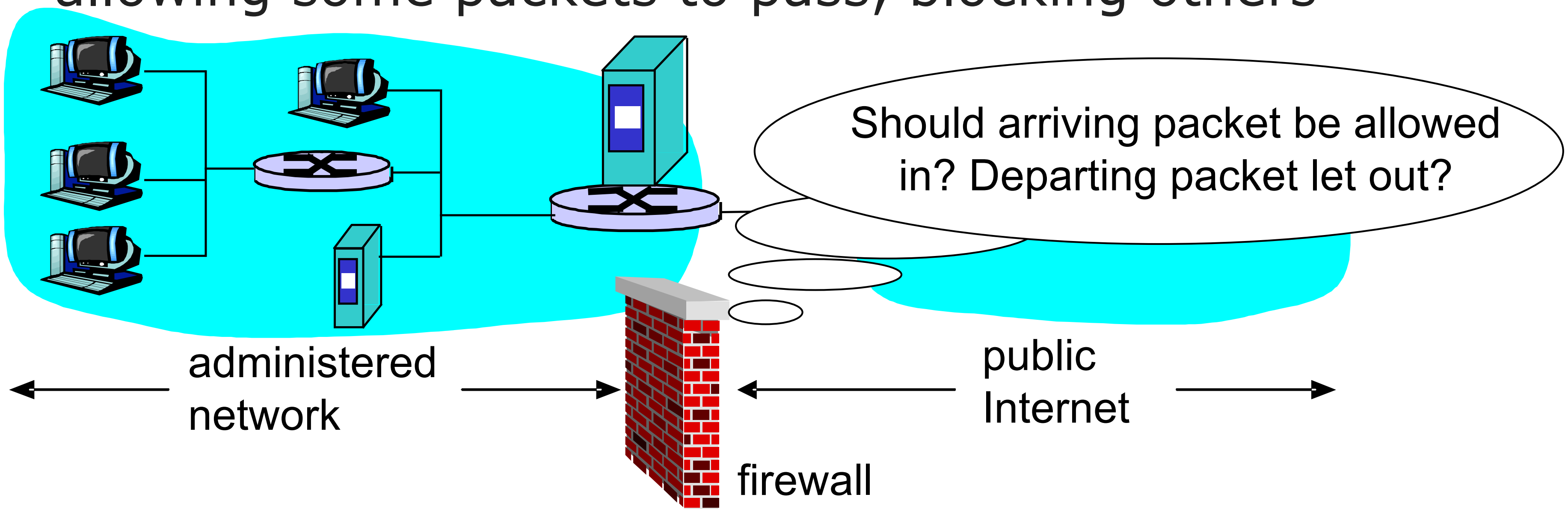
- Architectural decisions more holistic (e.g., how to prevent DoS attack?)

- Two instances today
  - Firewalls
  - DNS Security

Source: Freedman (partial)

Firewalls

•Isolates internal net from larger Internet, allowing some packets to pass, blocking others



•Firewall filters packet-by-packet, based on:

•Source/Dest IP address; Source/Dest TCP/UDP port numbers

•TCP SYN and ACK bits; ICMP message type

•Deep packet inspection on packet contents (DPI)

Packet Filtering Examples

- Block all packets with IP protocol field = 17 and with either source or dest port = 23.
  - All incoming and outgoing UDP flows blocked
  - All Telnet connections are blocked
- Block inbound TCP packets with SYN but no ACK
  - Prevents external clients from making TCP connections with internal clients
  - But allows internal clients to connect to outside
- Block all packets with TCP port of Quake

Source: Freedman

Source: Kurose & Ross (partial)

Configuring Firewall Rules

Policy	Firewall Setting
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios (UDP traffic) from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.
Prevent your network from being used for a smurf DoS attack (send ping's to targets broadcast address).	Drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

Clever Users Subvert Firewalls

- Example: filtering dorm access to a server
- Firewall rule based on IP addresses of dorms
- … and the server IP address and port number
- Problem: users may log in to another machine
- E.g., connect from the dorms to another host
- … and then onward to the blocked server

- Example: filtering P2P based on port #s
- Firewall rule based on TCP/UDP port numbers
- E.g., allow only port 80 (e.g., Web) traffic
- Problem: software using non-traditional ports
- E.g., write P2P client to use port 80 instead

Source: Freedman (partial)

Honourable mentions

- Security by obfuscation
- Use obscure/unique/outdated standards
- Not a strategy in and of itself
- Security by location
- Run on private networks
- Run on hard to reach hardware

But neither of these should be considered strategies in their own right

Source: Freedman (partial)

Nothing is secure forever

"You have 1 minute to design a maze that takes 2 minutes to solve" – some scriptwriter

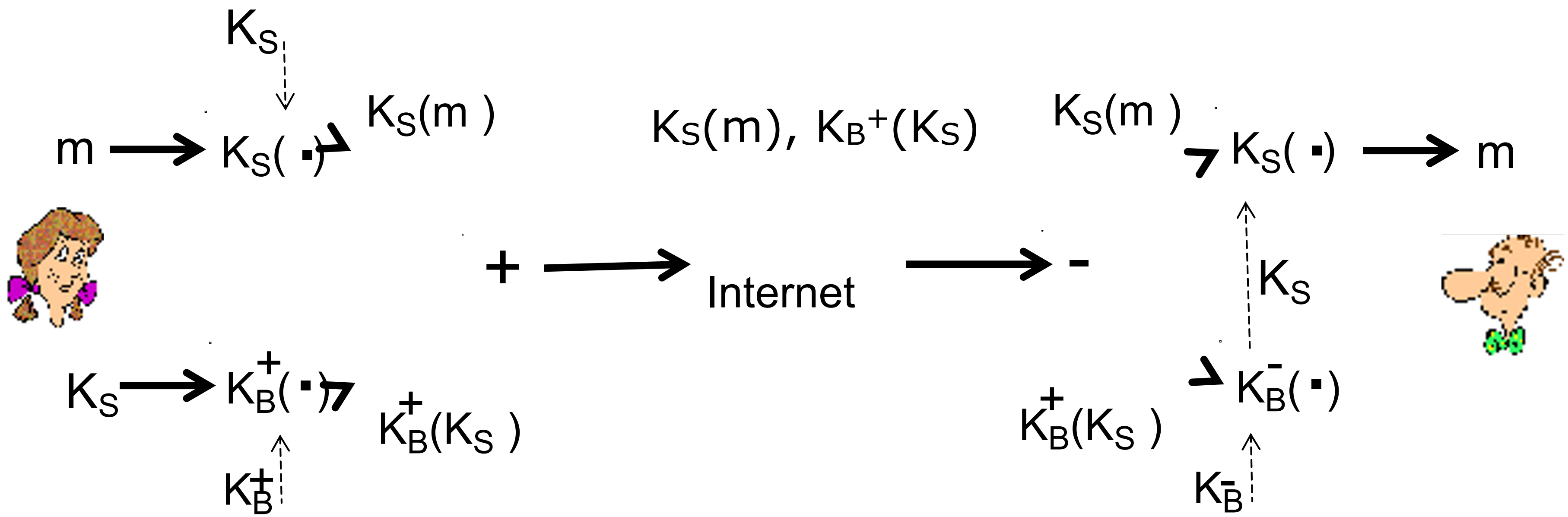
Source: Freedman (partial)

What should you be able to do after today?

- List security properties and related attacks
- Relate basic cryptographic schemes to their use in network protocols
- Explain the main mechanisms of HTTPS
- Explain the motivation and uses of IPSec
- Discuss operational security concerns and solutions, such as firewalls

An additional example concerning secure Email

Alice wants to send confidential e-mail, m, to Bob.

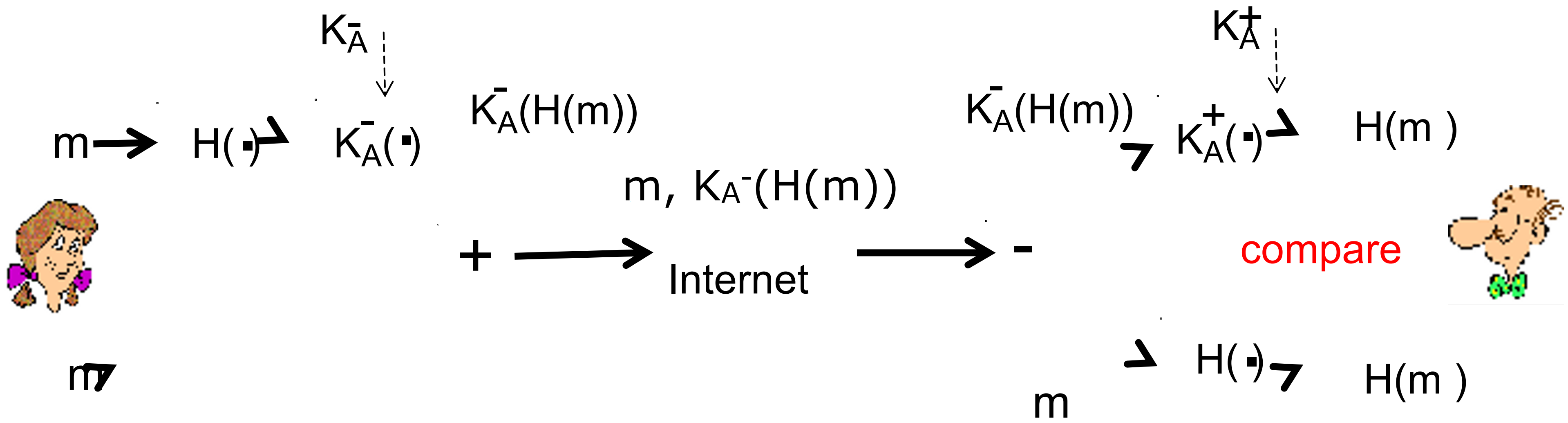


Use a random secret to encrypt and send that too.

Source: Kurose & Ross

Secure Email (continued)

Alice wants to provide sender authentication message integrity

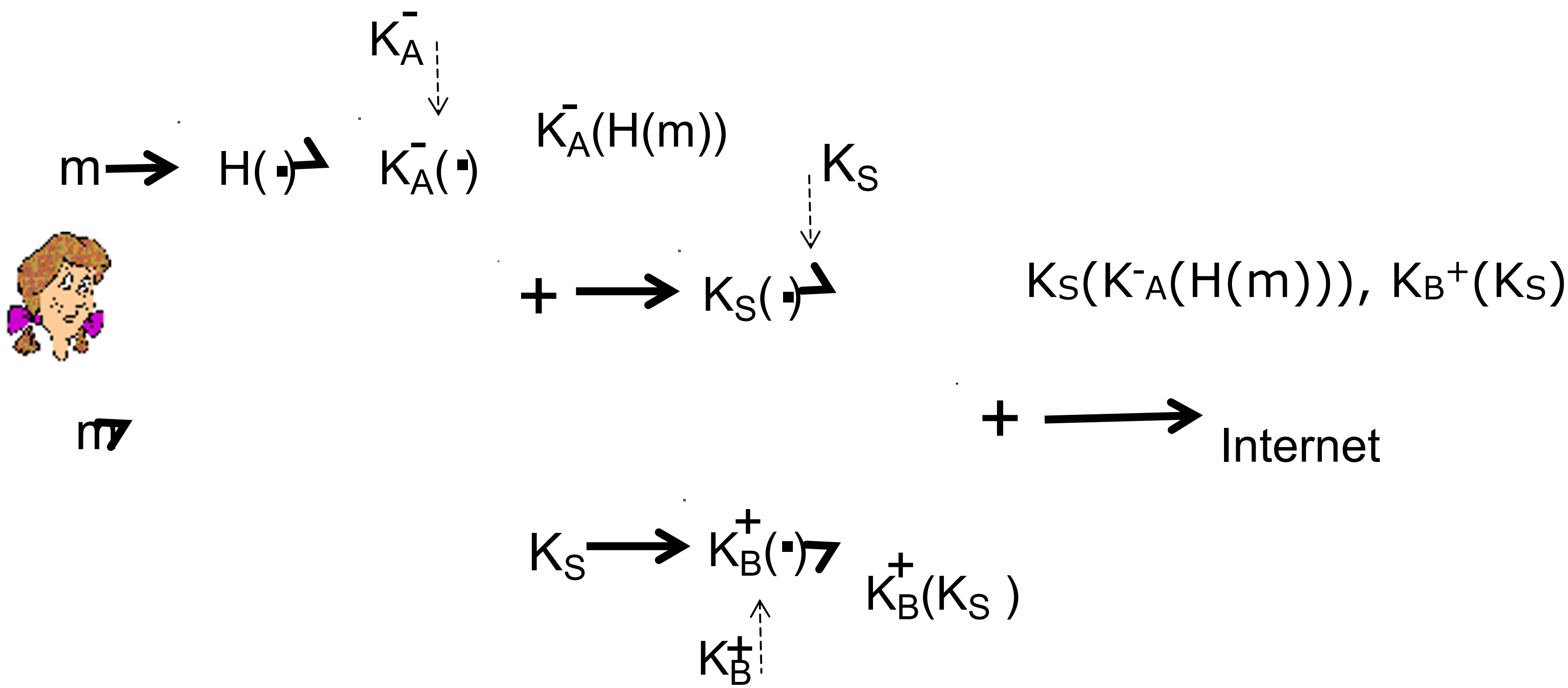


Use a private key to encrypt the hash of the message.

Source: Kurose & Ross

Secure Email (continued)

Alice wants to provide secrecy, sender authentication, message integrity.



Use both approaches in combination.

Source: Kurose & Ross