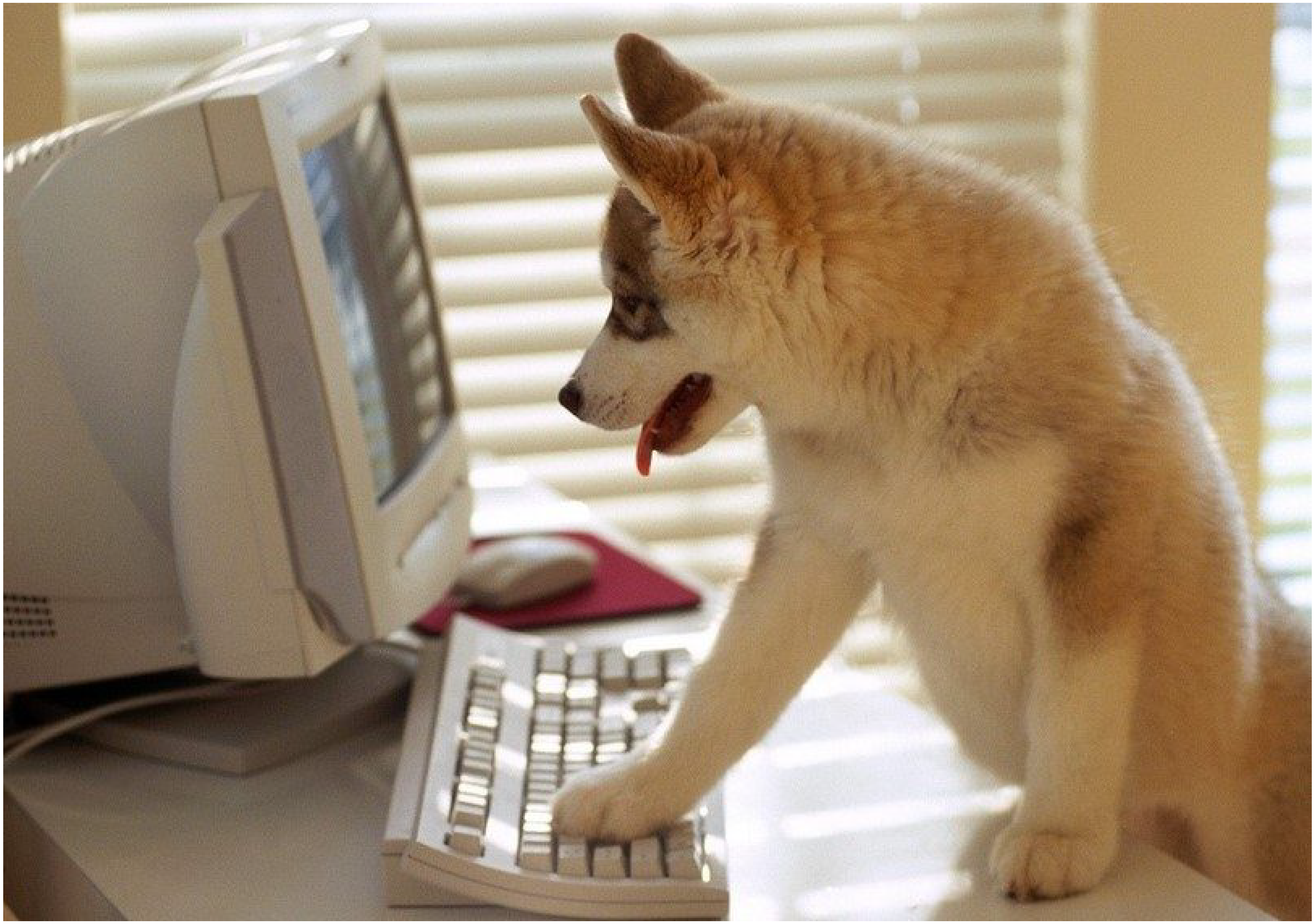


“On the Internet, nobody knows you’re a dog.”

THREATS TO DISTRIBUTED APPLICATIONS 1

"How do I know I am connecting to my bank?"

Maybe an attacker . . .

. . . sends you phishing email (pretending to be from your bank) and you click on a link in it

. . . gets access to your packet stream and changes the IP destination addresses

. . . hijacks the DNS entry of bigbank.com

. . . hijacks routing to the IP address of bigbank.com

THREATS TO DISTRIBUTED APPLICATIONS 2

Jane Q. Public

client's machine

Big Bank

server's machine

WHO COULD SPY ON THEM OR TAMPER WITH COMMUNICATION?

- the networks themselves
- people who . . .

  . . . tap a wire

  . . . penetrate an infrastructure machine

  . . . put a wireless receiver near a transmitter

  . . . connect to a wired broadcast medium such as a cable network

WHAT CAN ATTACKERS DO?

read data

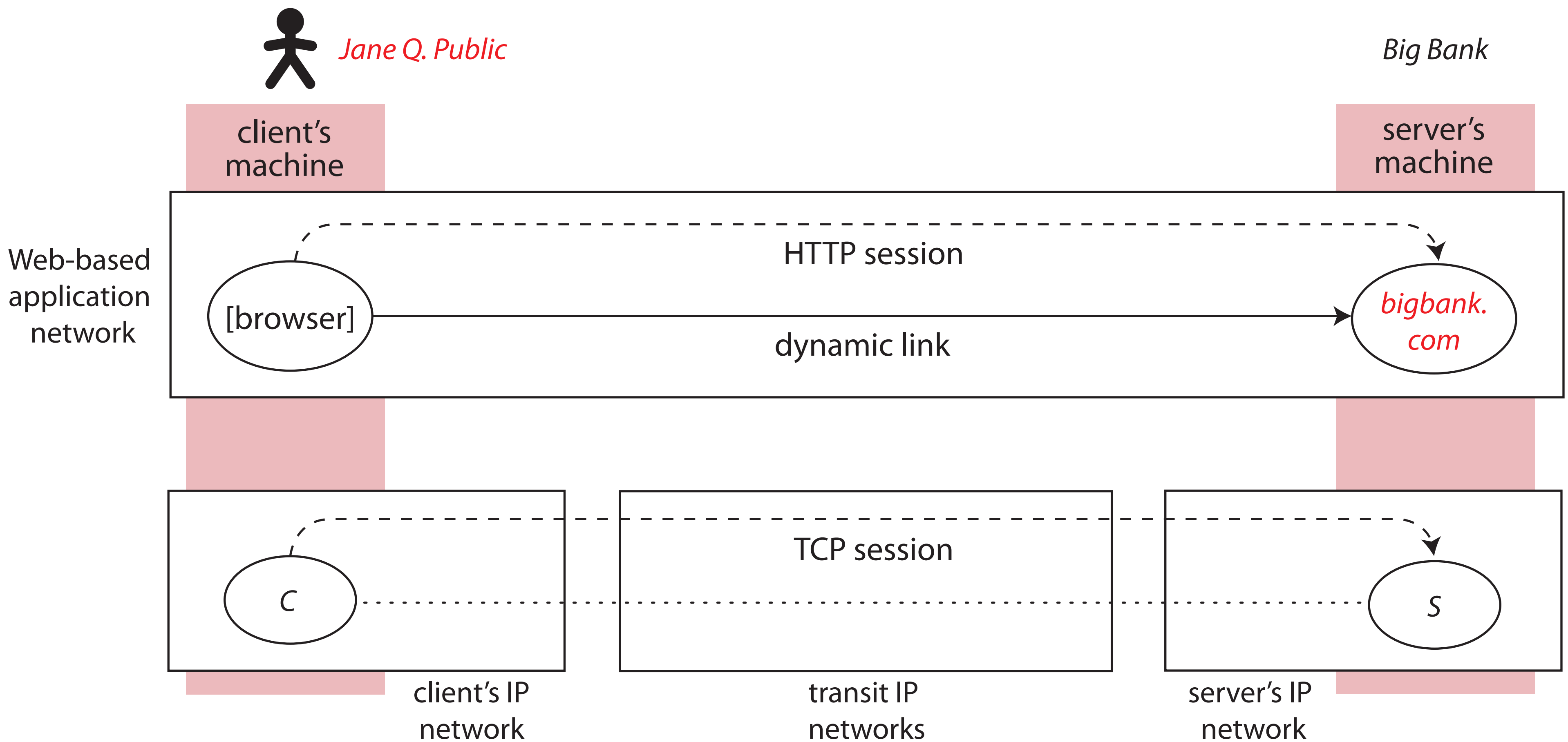
absorb packets

modify packets

inject packets into the stream

"WHOM AM I TALKING TO?"

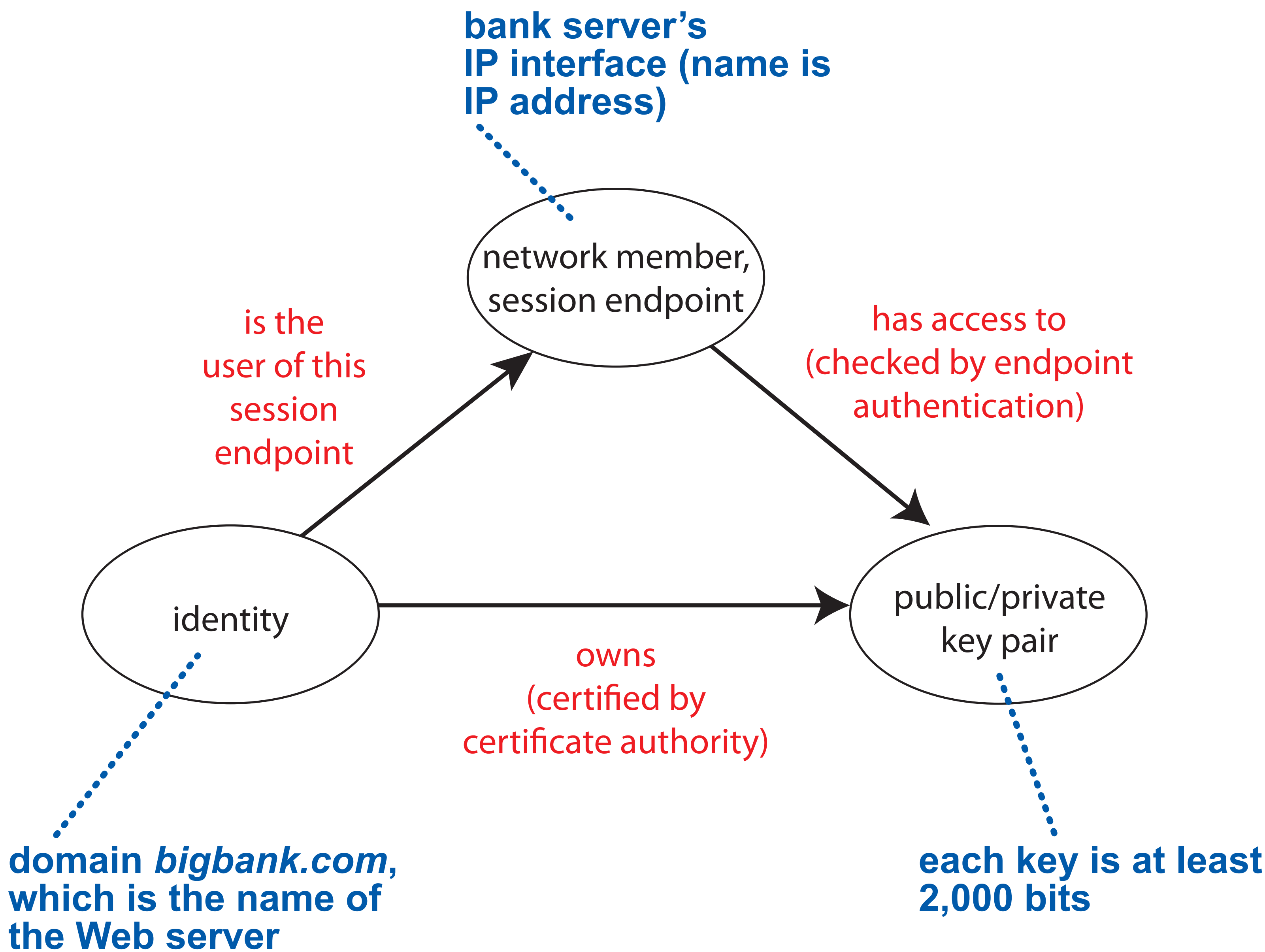
the answer (whatever it is) is an identity



most likely, the bank wants as identity the user name "Jane Q. Public," . . .

. . . and asks for a password for endpoint authentication

IDENTITIES USED IN SESSION PROTOCOLS



PUBLIC-KEY ENCRYPTION

$$K^{+}(K^{-}(\text{data})) = K^{-}(K^{+}(\text{data}))$$

BUT knowing one of the pair, it is very difficult to compute the other!

CHALLENGER B | **VERIFIABLE ENDPOINT A**

Are you A, who has public key K^{+}? Nonce n

$K^{-}(n)$

$K^{+}(K^{-}(\text{data})) = n$

OK!

THE TLS HANDSHAKE

accomplishes . . .

(1) agreement on cipher suite,

(2) endpoint authentication of server,

(3) key exchange

can also authenticate initiating endpoint of session, but this is not done by normal Web services (user has history with the site, not a certificate)

CLIENT B | **SERVER A**

cipher suites I support,
my nonce NB

$\longrightarrow$

important to do a good job of this

choice of cipher suite,
my nonce NA, my certificate

important to use an up-to-date one, cf. TLS 1.3

$\longleftarrow$

verify certificate, extract
public key K^+, generate
pre-master-secret

K^+(pre-master-secret)

$\longrightarrow$

using private key K , compute
K^-(K^+(pre-master-secret)) to get
pre-master-secret

from pre-master-secret, NB,
and NA, compute 2 encryption
keys and 2 authentication keys

from pre-master-secret, NB,
and NA, compute 2 encryption
keys and 2 authentication keys

lots of keys!

A HIGH-LEVEL IDENTITY IS MOBILE

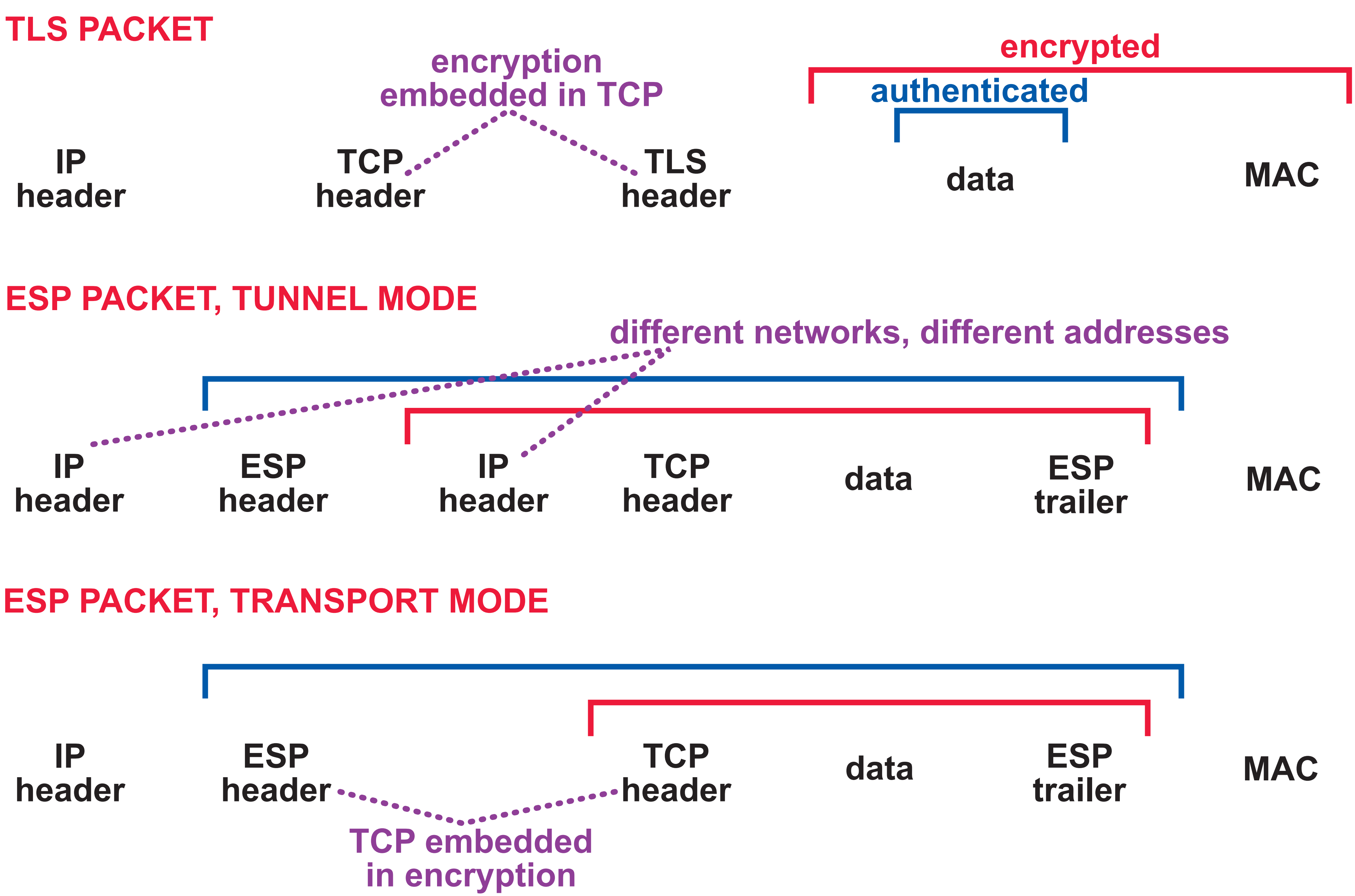
CLIENT CAN . . .

- log in from another computer
- disconnect identity from session by logging out
- move around while using a mobile device (even if the identity goes with the device)

SERVER CAN . . .

- lend keys and certificate to a trusted representative, e.g., a content-delivery network
- attach a digital signature to data, so its identity can travel anywhere with the data

DATA ENCRYPTION AND MESSAGE AUTHENTICATION 1



MESSAGE AUTHENTICATION 2

TLS PACKET

encrypted

authenticated

IP header	TCP header	TLS header	data	MAC

MAC = H (data + key)

authentication key, just
for this direction, and known
by both endpoints

cryptographic
hash function

recipient recomputes
same formula, checks
that it is the same as MAC

MESSAGE AUTHENTICATION 3: THE CATCH

MAC ensures that received packet came from sender without modification.

But attacker could still delete, re-order, or replay packets.

TLS PACKET

encrypted

authenticated

IP header | TCP header | TLS header | data | MAC

TCP sequence numbers won't help because attacker can change those to make sequence look right

so TLS packets have implicit sequence numbers (can make them implicit because of reliance on TCP)

MAC = H

(data + key + number)

ESP PACKET, TRANSPORT MODE

IP header | ESP header | TCP header | data | ESP trailer | MAC

can't rely on TCP, so ESP headers have explicit sequence numbers . . .

. . . which may not arrive in order

so receiver deletes packets with previously received numbers (replay attacks)

ENCRYPTION AND AUTHENTICATION 4: SCOPES

WHY ARE THE SCOPES OF ENCRYPTION AND AUTHENTICATION DIFFERENT?

TLS PACKET

authentication is extra-secure?

encrypted

authenticated

IP header	TCP header	TLS header	data	MAC

ESP PACKET, TUNNEL MODE

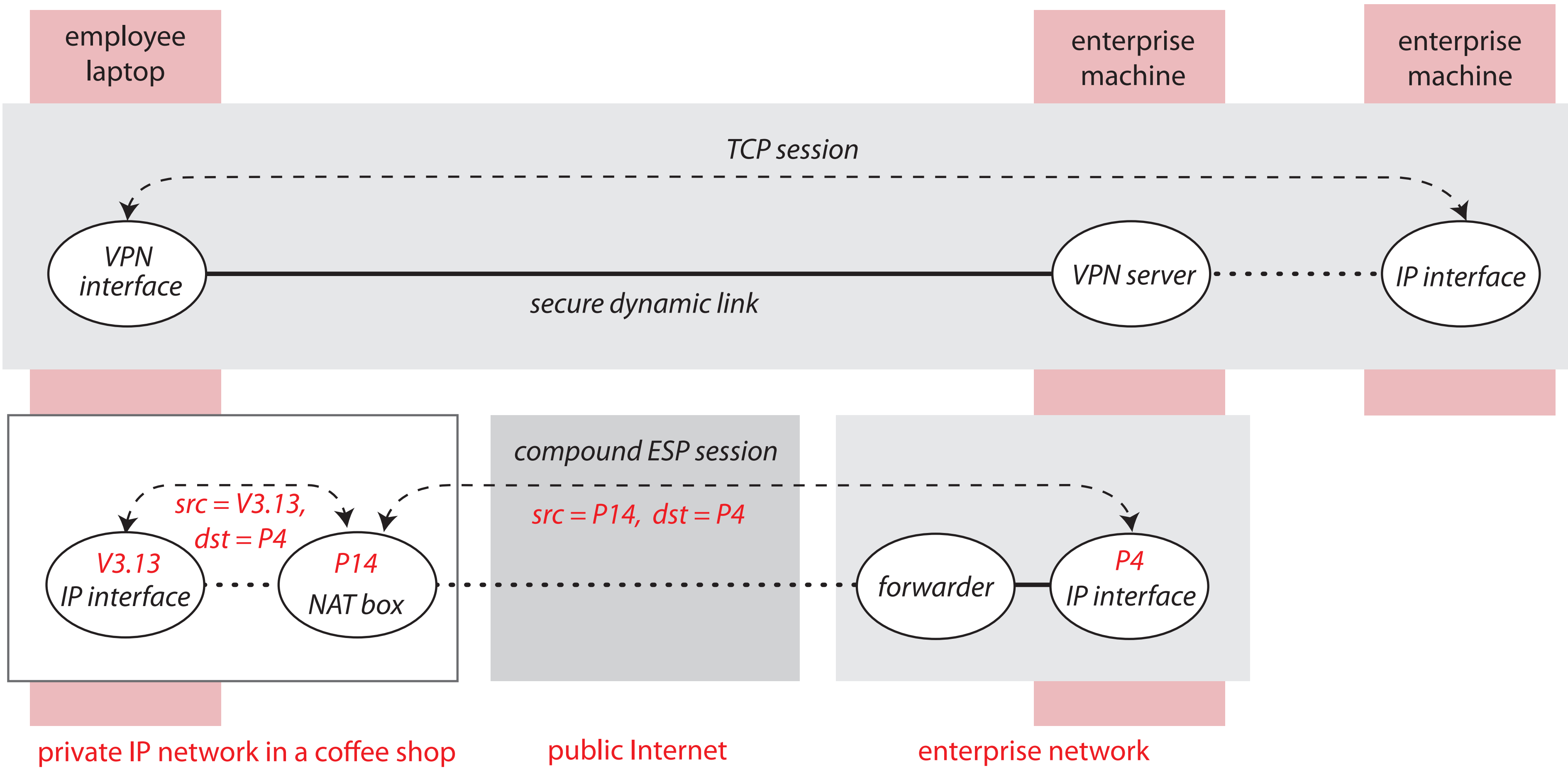
a middlebox can authenticate
even if it doesn't decrypt?

IP header	ESP header	IP header	TCP header	data	ESP trailer	MAC

why not authenticate
the IP header, too?

IPsec AH does this,
and packets can't
even pass through
a NAT

POOR COMPOSITION: SHOULD WORK, BUT DOES NOT



Why? A NAT cannot make a compound ESP session, because the session identifier is not standard.

Ugly hack: pretend UDP has persistent sessions, use with well-known port 4500, this signals endpoints that ESP is embedded inside UDP.

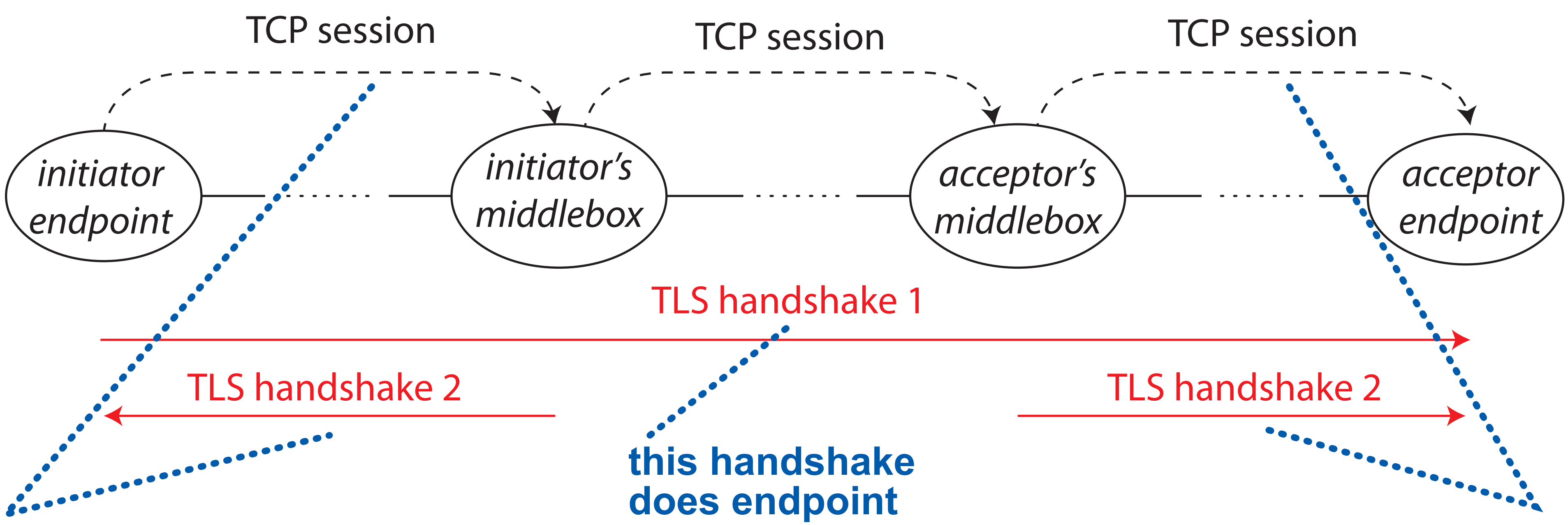
ENCRYPTION AND MIDDLEBOXES 1

IF THE MIDDLEBOXES HAVE THE SAME

INTERESTS AS ADJACENT ENDPOINTS,

THEY CAN BECOME PART OF THE TLS SCHEME

middleboxes could be doing security, performance optimization



endpoint authentication and key exchange for this hop

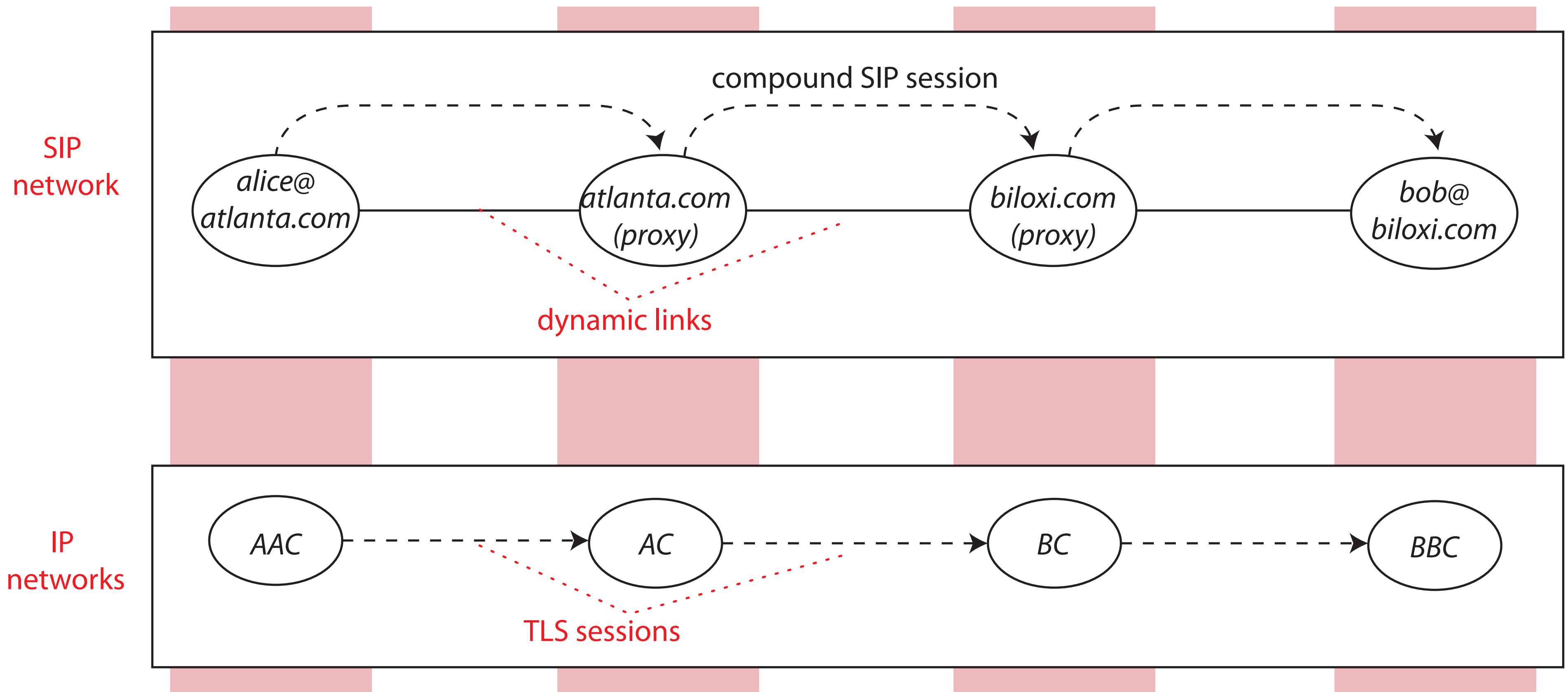
this handshake does endpoint authentication for the real endpoints, key exchange for the middle hop

endpoint authentication and key exchange for this hop

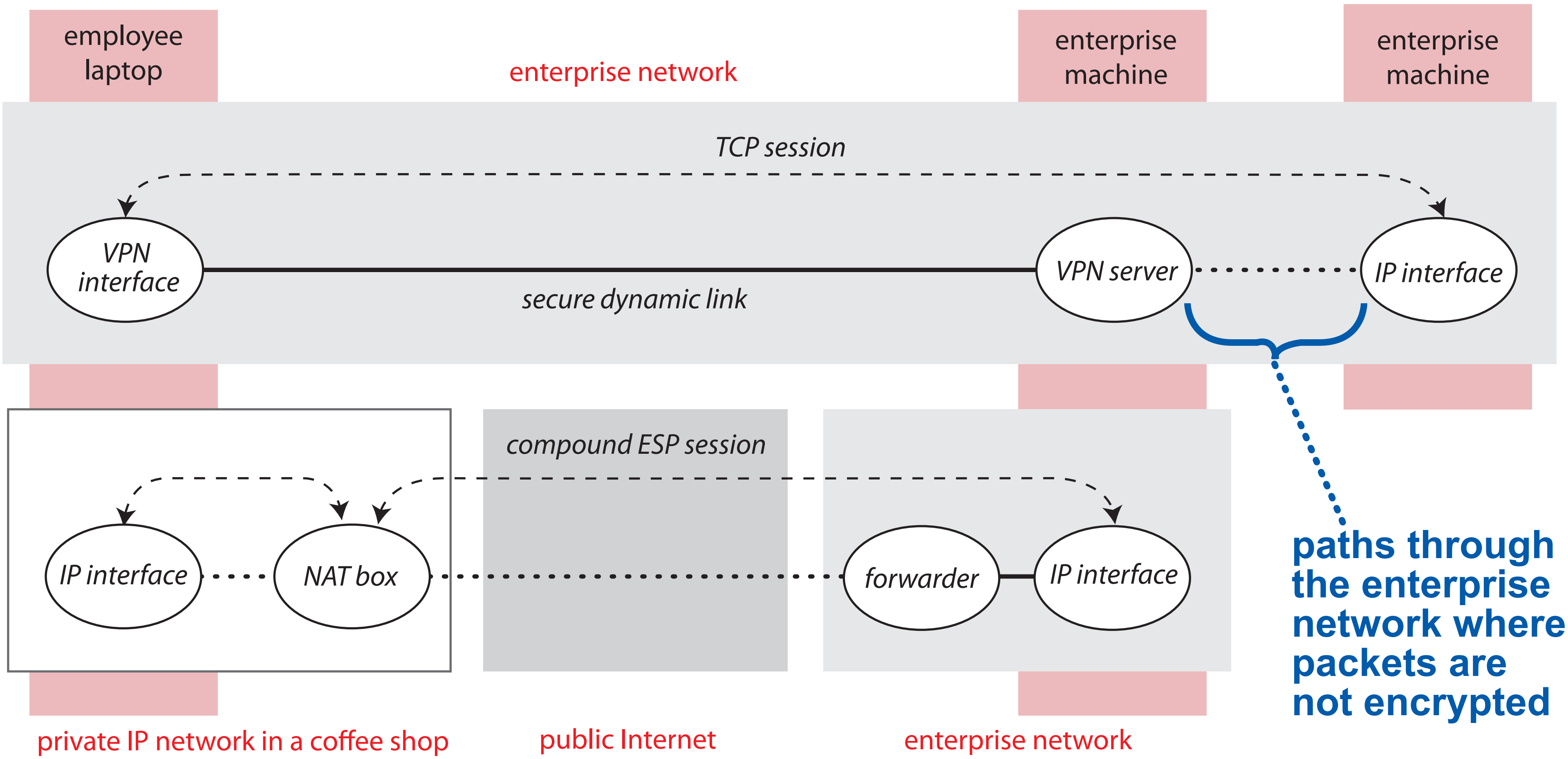
ENCRYPTION AND MIDDLEBOXES 2

TRUSTED MIDDLEBOXES ARE PART OF THE APPLICATION

—ABOVE THE ENCRYPTION!



ENCRYPTION AND MIDDLEBOXES 3



SIDESTEP THE ENCRYPTION!

ENCRYPTION AND MIDDLEBOXES 4

MIDDLEBOXES ARE BELOW THE ENCRYPTION!

encrypted

N1 header:	N2 headers:	N3 headers:	N4 headers:	N4 payload
Ethernet	IP, UDP, GTP	IP, UDP, ESP	IP, TCP	

at Level 4, everything in the packets is encrypted

?

at Level 3, data is encrypted, headers are not

IP + UDP + ESP

Level 2 is a cellular network, which has several middleboxes that may care about Level 3 headers

IP + UDP + GTP

SECURITY FOR CONTROL PROTOCOLS

BECAUSE THEY CHANGE THE STATE OF THE NETWORK!

CANNOT ALWAYS USE TLS OR ESP

- **in session-location mobility, an identity must update its own location, but may not have a certificate or past history with the server**
- **control protocols can be very high-volume (DNS, routers exchanging filtering information)**
- **protocols may be too old**

and attackers can afford a lot of tries, guessing how to get in, because there is little risk

HELP FOR CONTROL PROTOCOLS

at least, when updates are requested

DON'T ACCEPT UNSOLICITED REPLIES

e.g., ARP accepts unsolicited replies to requests . . .

. . . which are broadcast to every member of network . . .

. . . so any member of network can reply “I have requested IP address”

CHECK REPLIES FOR CREDIBILITY

in the U.S., the closest cnn.com server is not in Brazil

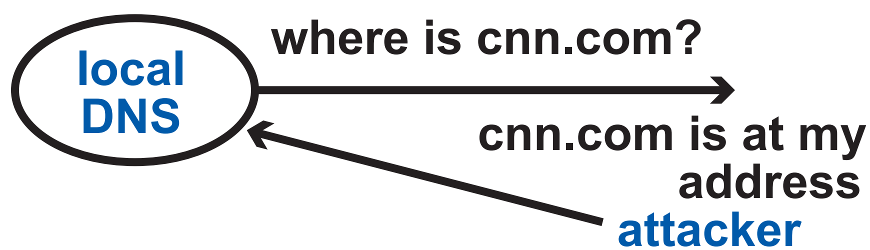
woe to those late-blooming U.S. services whose IP addresses are in Brazil

USE NONCES OR RANDOMIZATION

TO PREVENT “OFF-PATH” ATTACKS

attacker queries local server for cnn.com

so local server may need to query another server, if no cache or old cache



to prevent this, server can put a nonce (random number) in a new or unused field of the request . . .

. . . and expect the reply to carry the same information