COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Mixnets and Onion Routing

Information Privacy with Applications

David Sidi (dsidi@email.arizona.edu)

Administrative Items

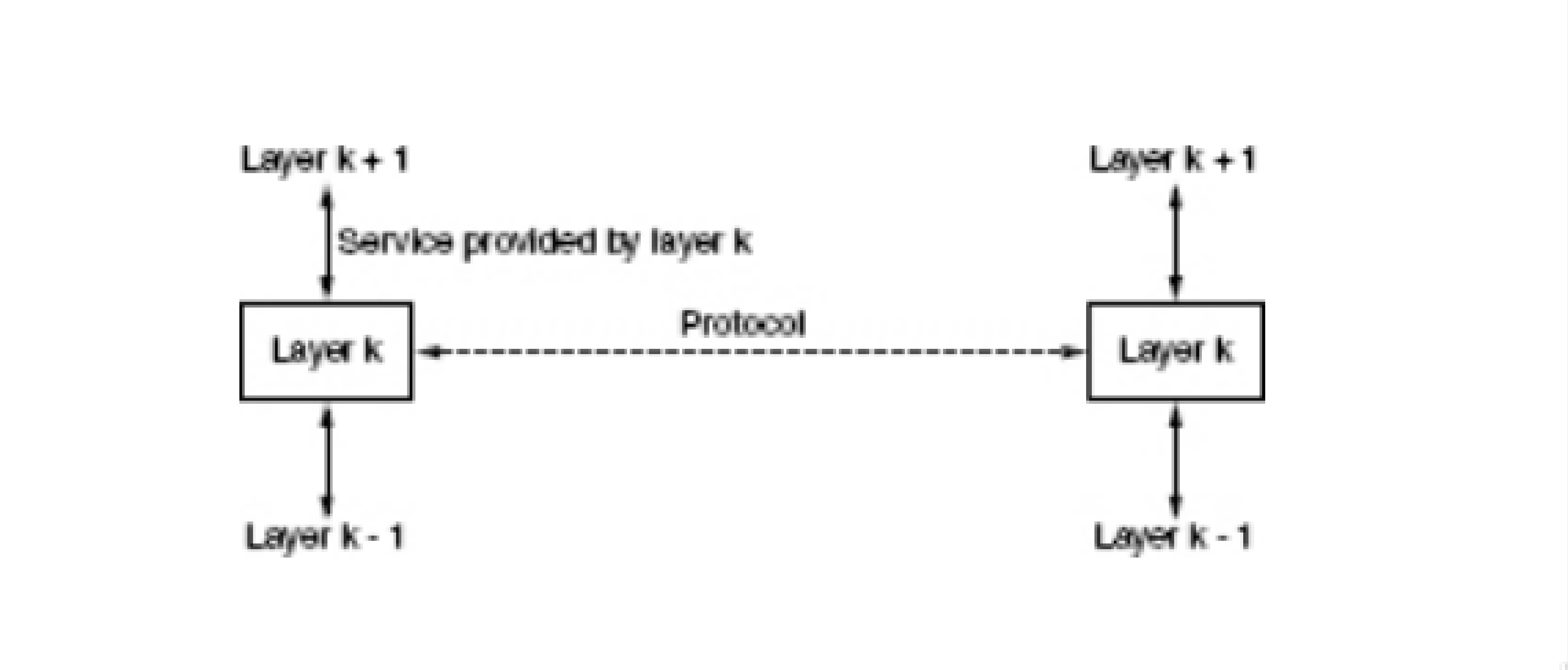
- Final project proposals: more examples
- Anonymity assignment: auditing by meddling in your connections, anonymous publication with tor

Student project example

As conversations were increasingly being had over the phone and internet and computational analysis became more efficient and powerful, someone had the idea that they could listen to what a user was typing on the other end of a conversation and recreate what their keystrokes, just by hearing the clicks of the keyboard. It's not perfect in all cases, but if there is a sufficient sample size an accuracy of more than 90% can be reached. The goal of this project is to prevent any useful data from being gathered through just listening.

Student project example 2

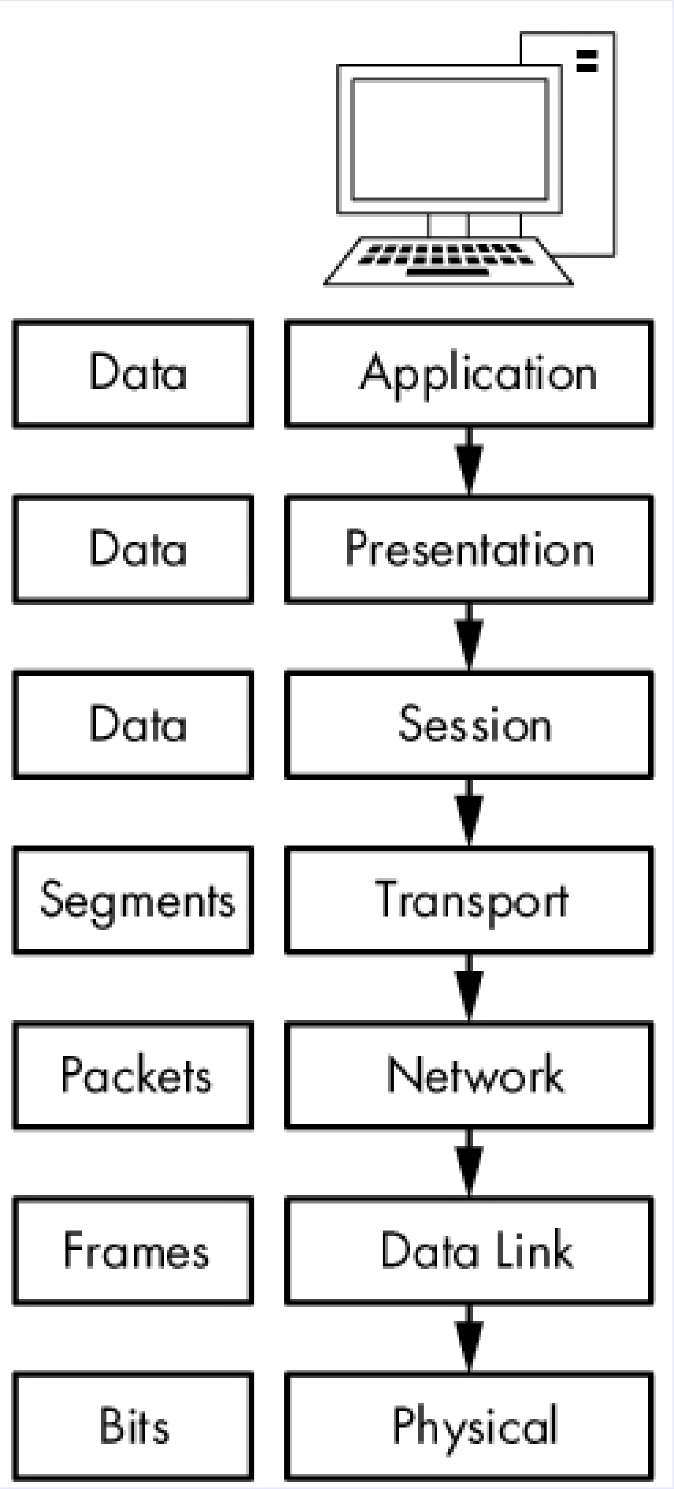
- Obfuscate ourselves on the web for protection
- Obfuscate traffic (headers and JavaScript values)
- Eventually add fake traffic too (fake requests)
- We are trying to address traffic surveillance, browser fingerprinting. We want to browse the web anonymously.
- Fingerprinting and tracking on the web is becoming rampant. We have to fake things because turning JavaScript off and other forms of disabling tracking is itself, a way of fingerprint.



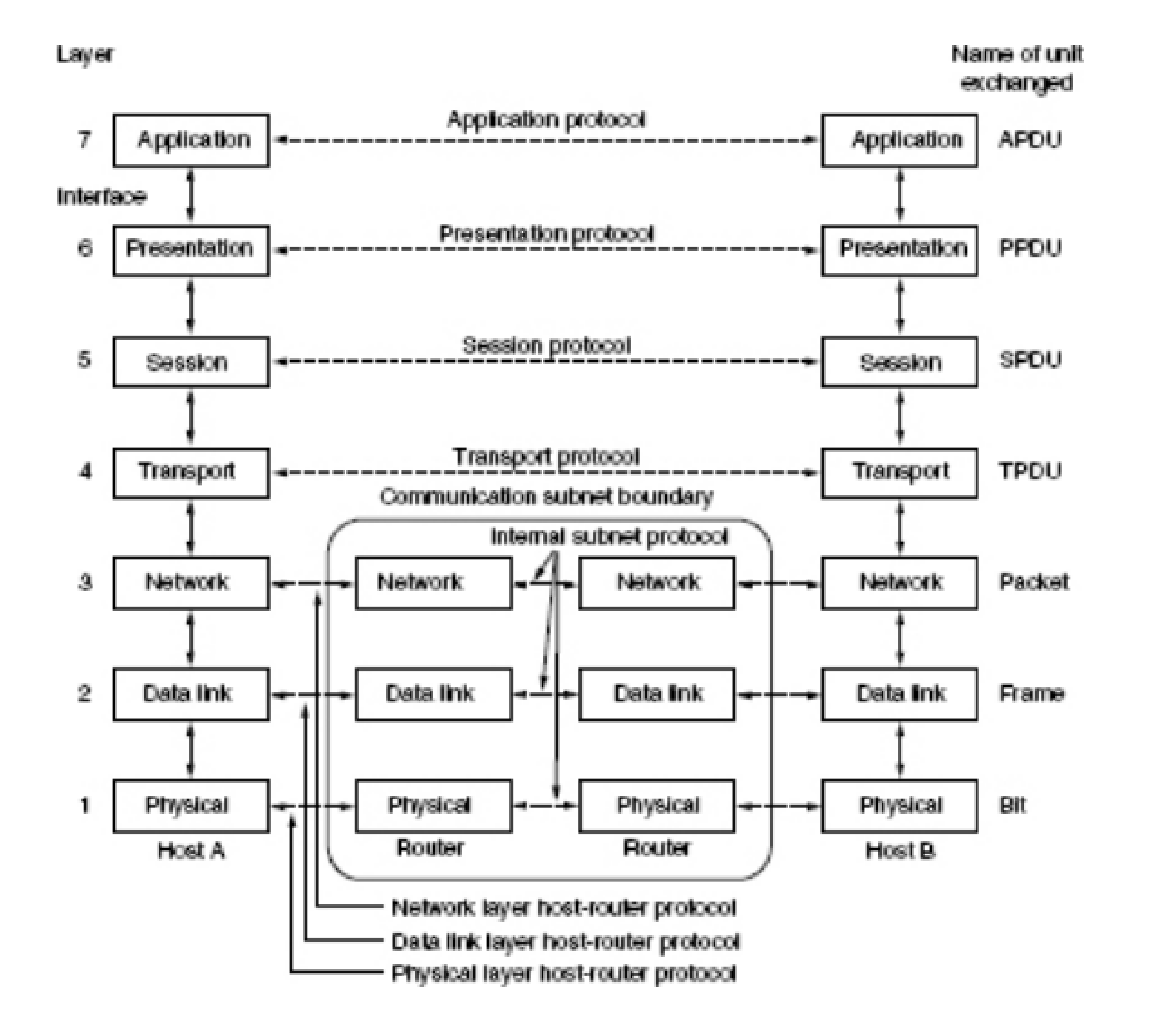
Tanenbaum, A. *Computer Networks*, Figure 1-19.

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

The Seven-Layer OSI Reference Model



“the way up is the way down.” -Heraclitus

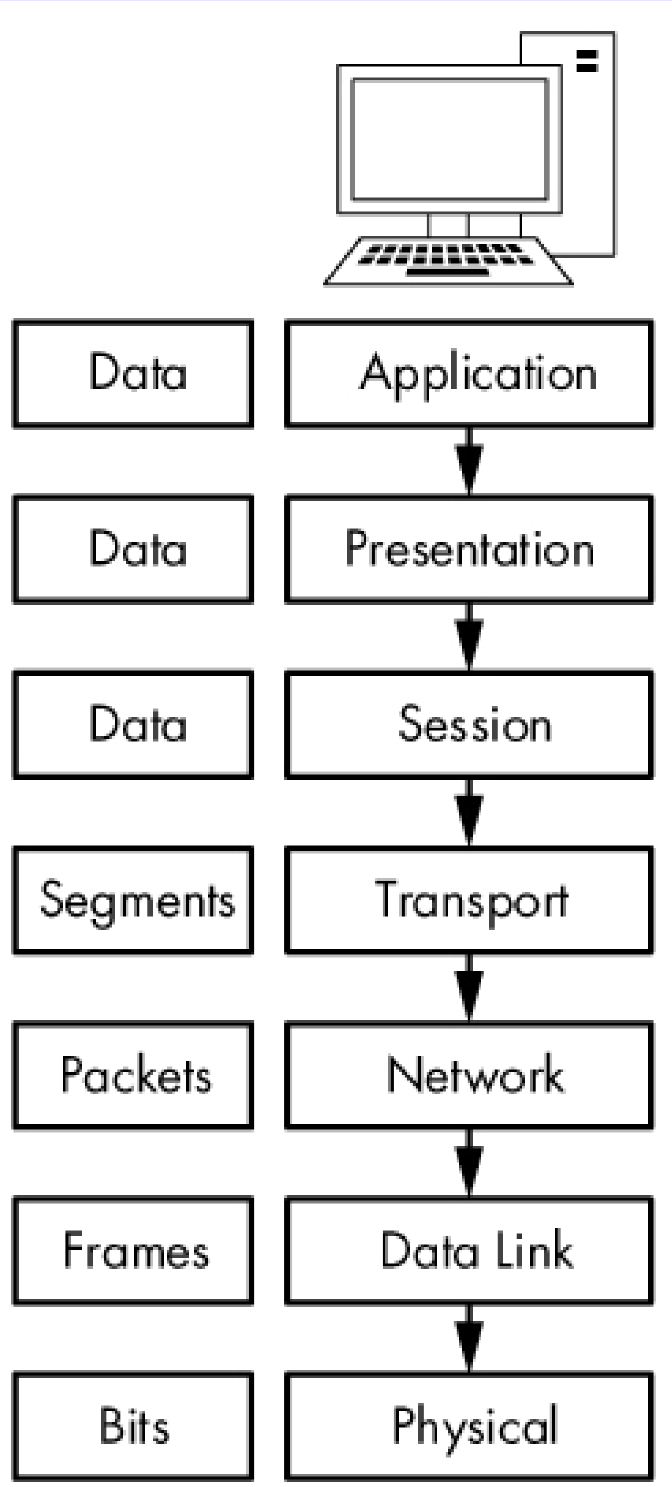


Tanenbaum, A. *Computer Networks*, Figure 1-19.

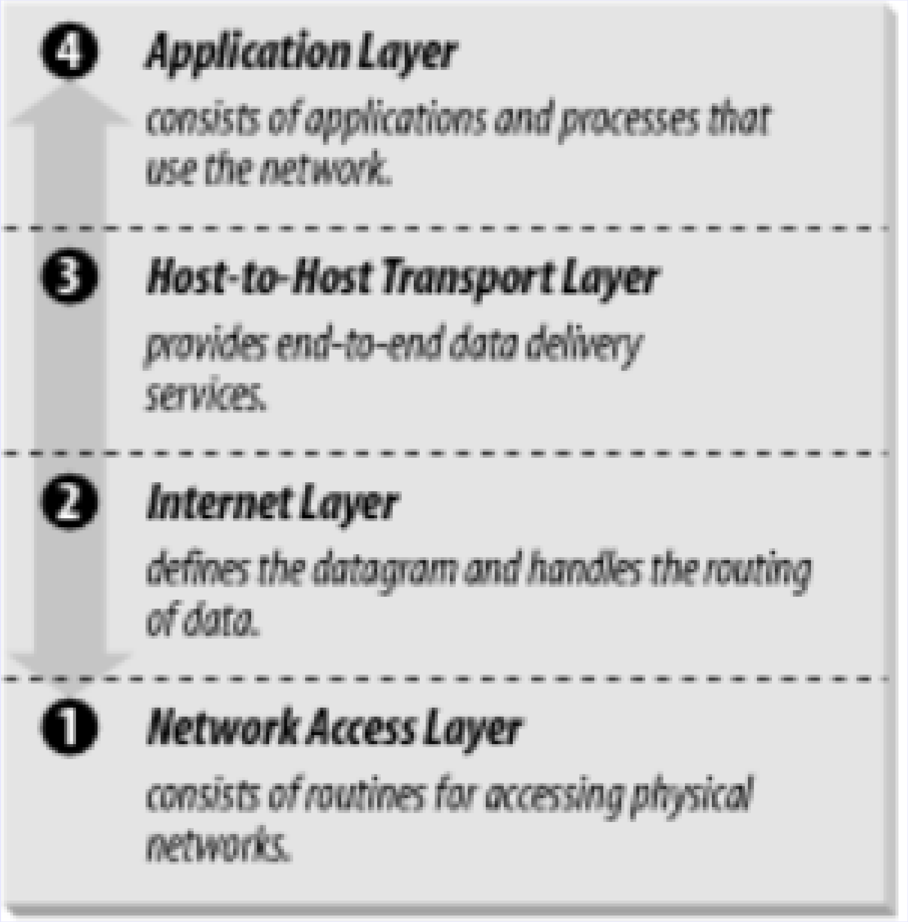
COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

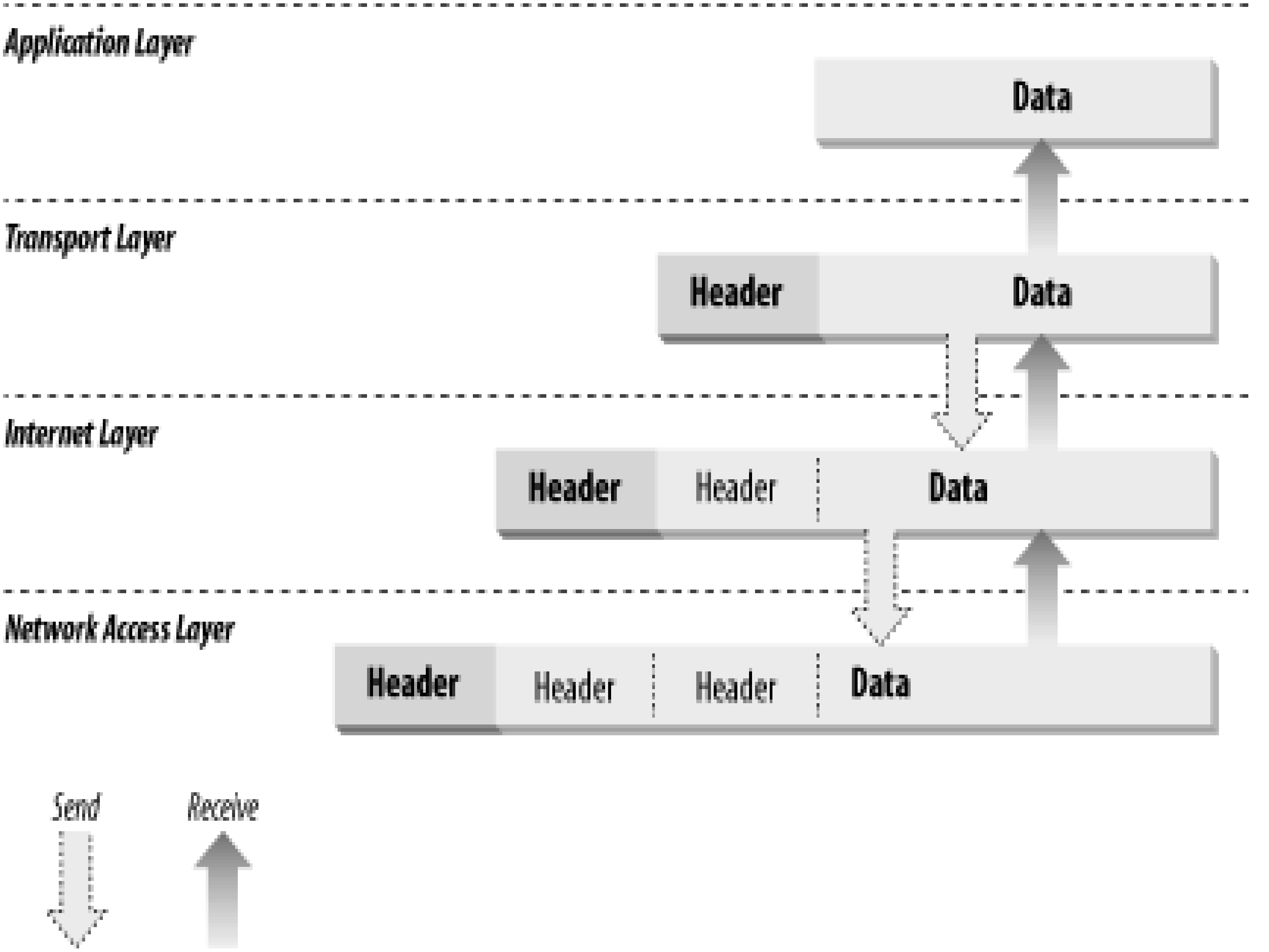
The Seven-Layer OSI Reference Model

“the way up is the way down.” -Heraclitus



TCP/IP Protocol Stack




Application Layer
Data
Transport Layer
Header
Data
Internet Layer
Header
Header
Data
Network Access Layer
Header
Header
Header
Data
Send
Receive

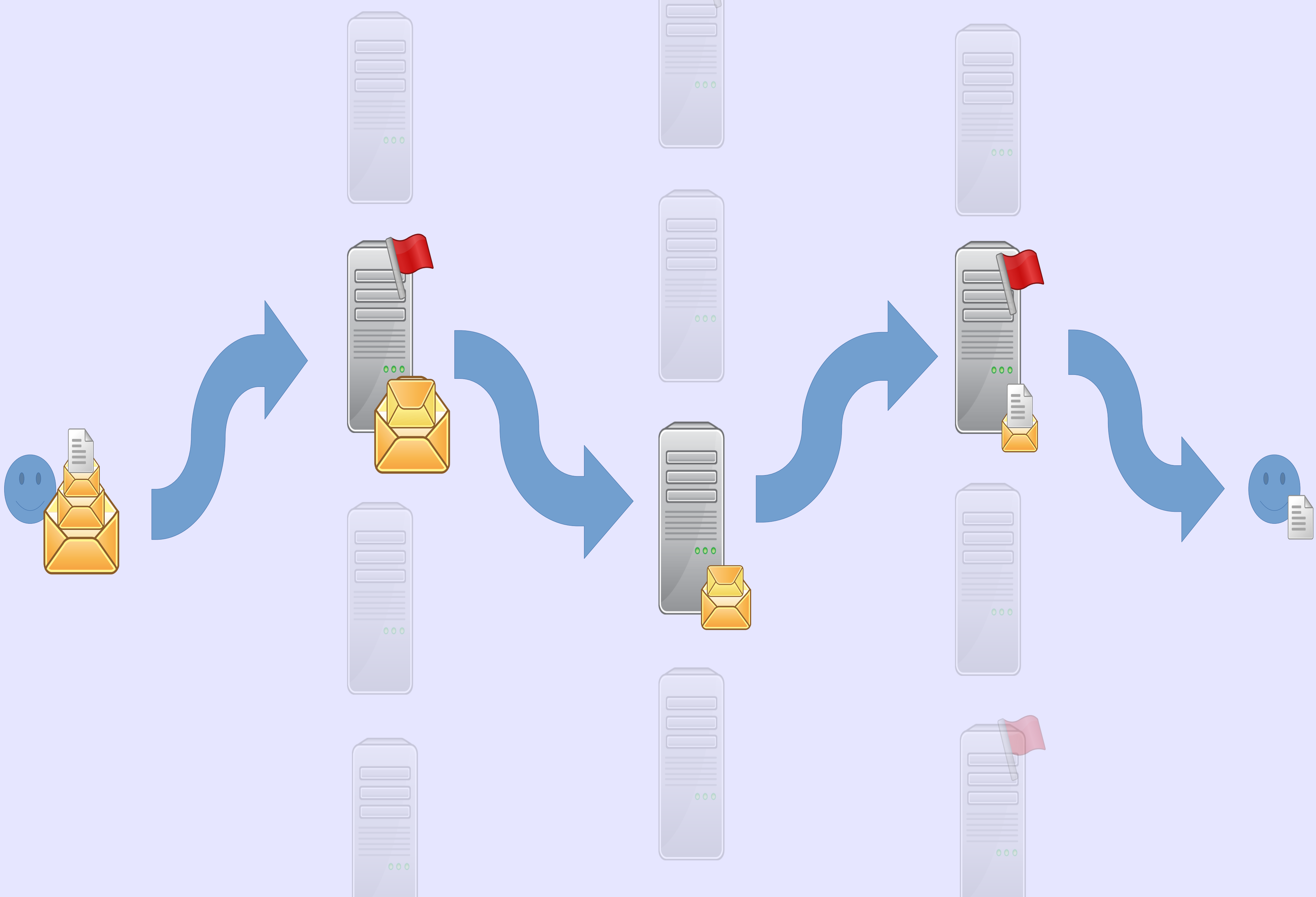
Chaum 1981:

process each item of mail before it is delivered. A participant prepares a message M for delivery to a participant at address A by sealing it with the addressee's public key K_a, appending the address A, and then sealing the result with the mix's public key K_1. The left-hand side of the following expression denotes this item which is input to the mix:

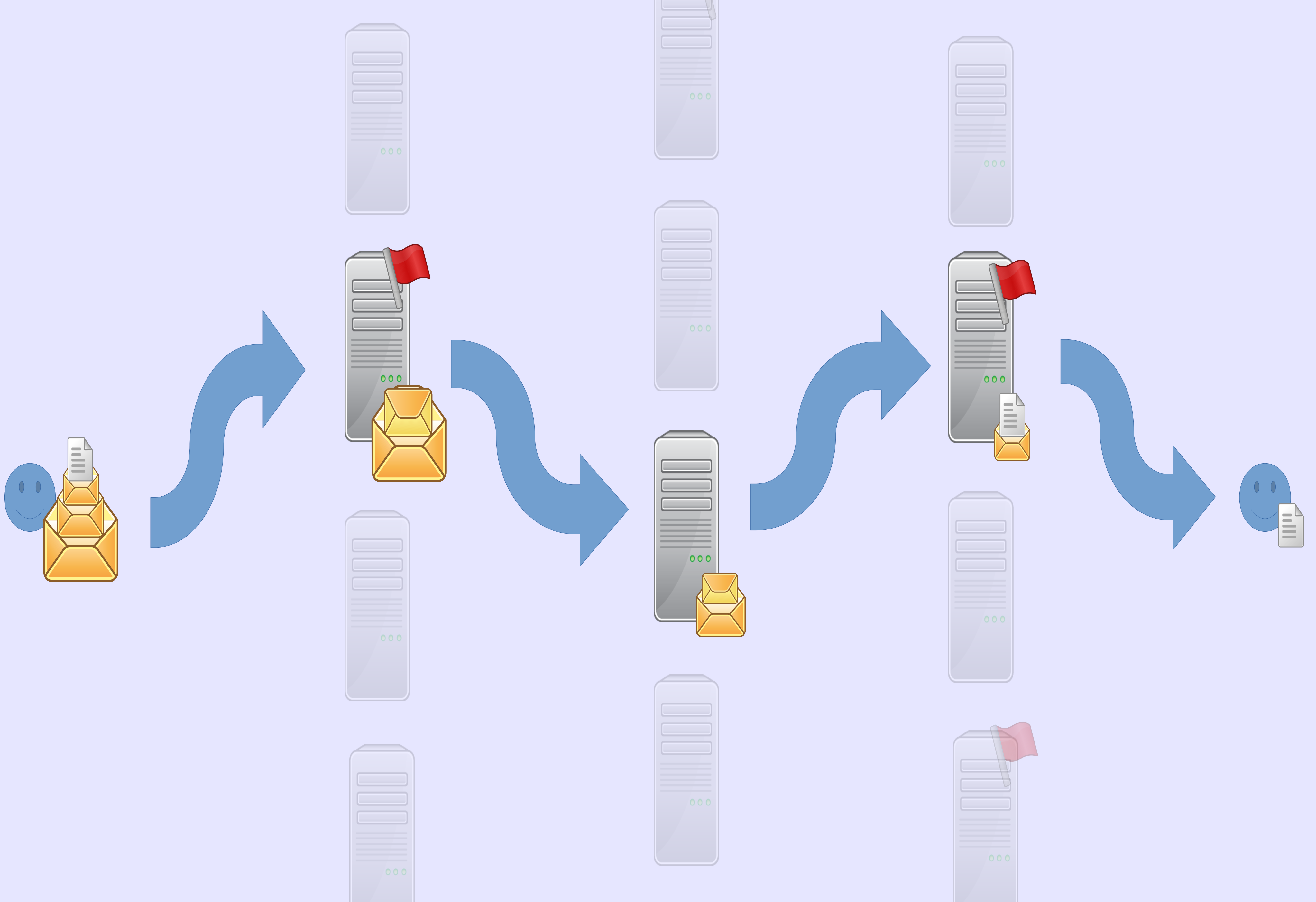
$$K_1(R_1, K_a(R_0, M), A) \rightarrow K_a(R_0, M), A.$$

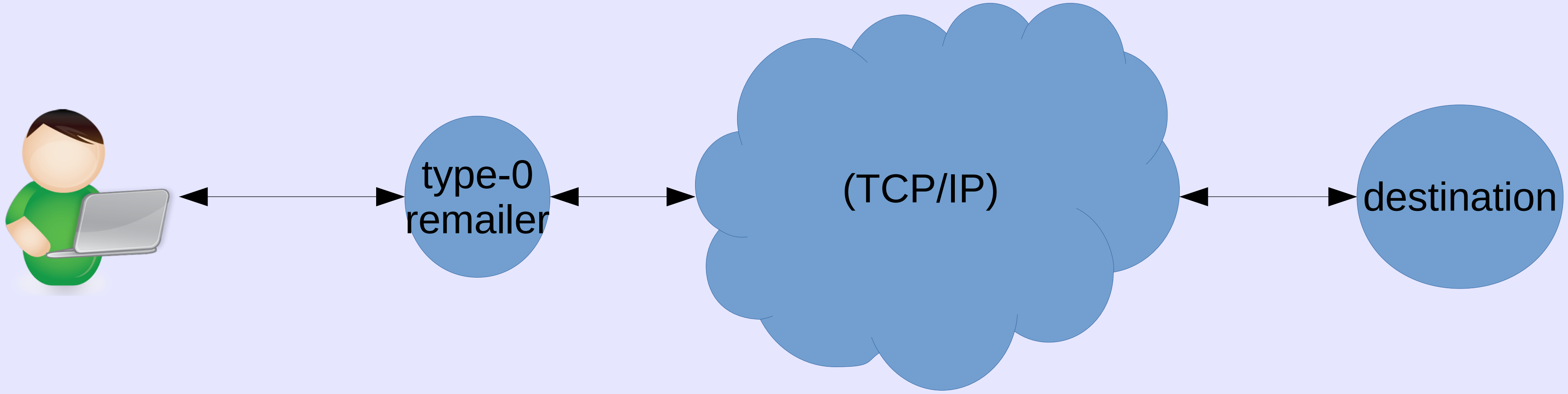
How does the message get to A?
One answer: overlay network

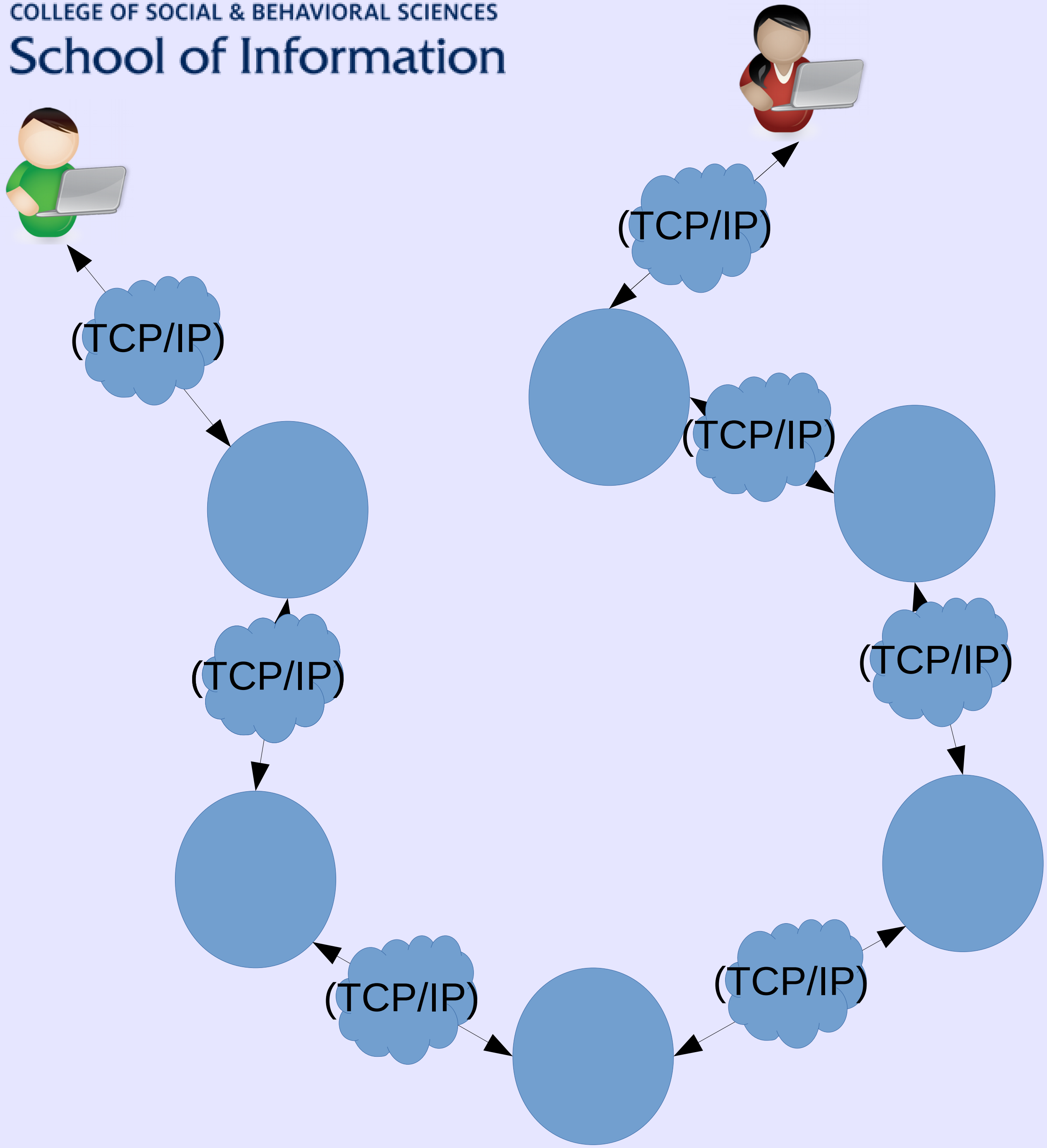
COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

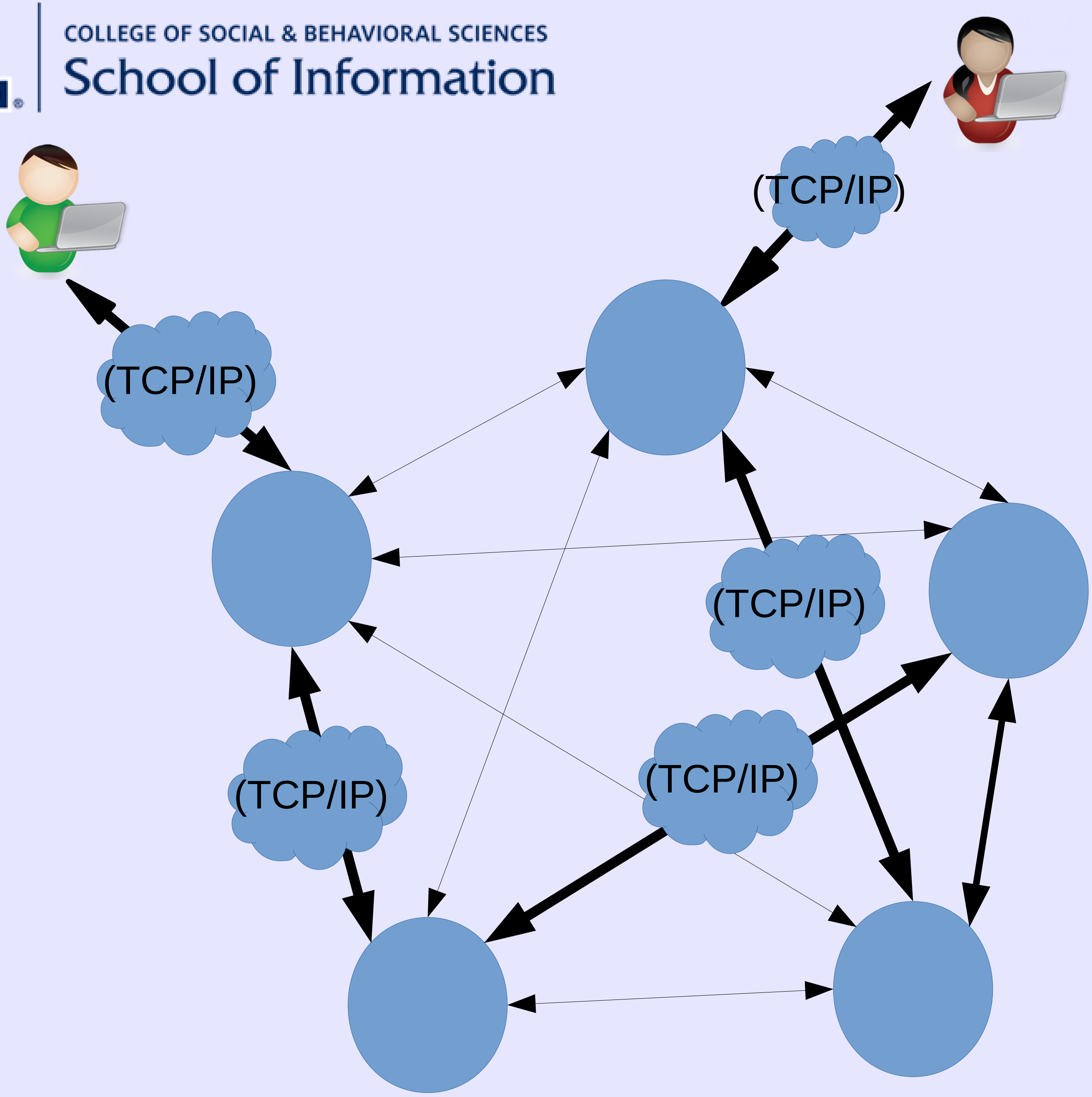


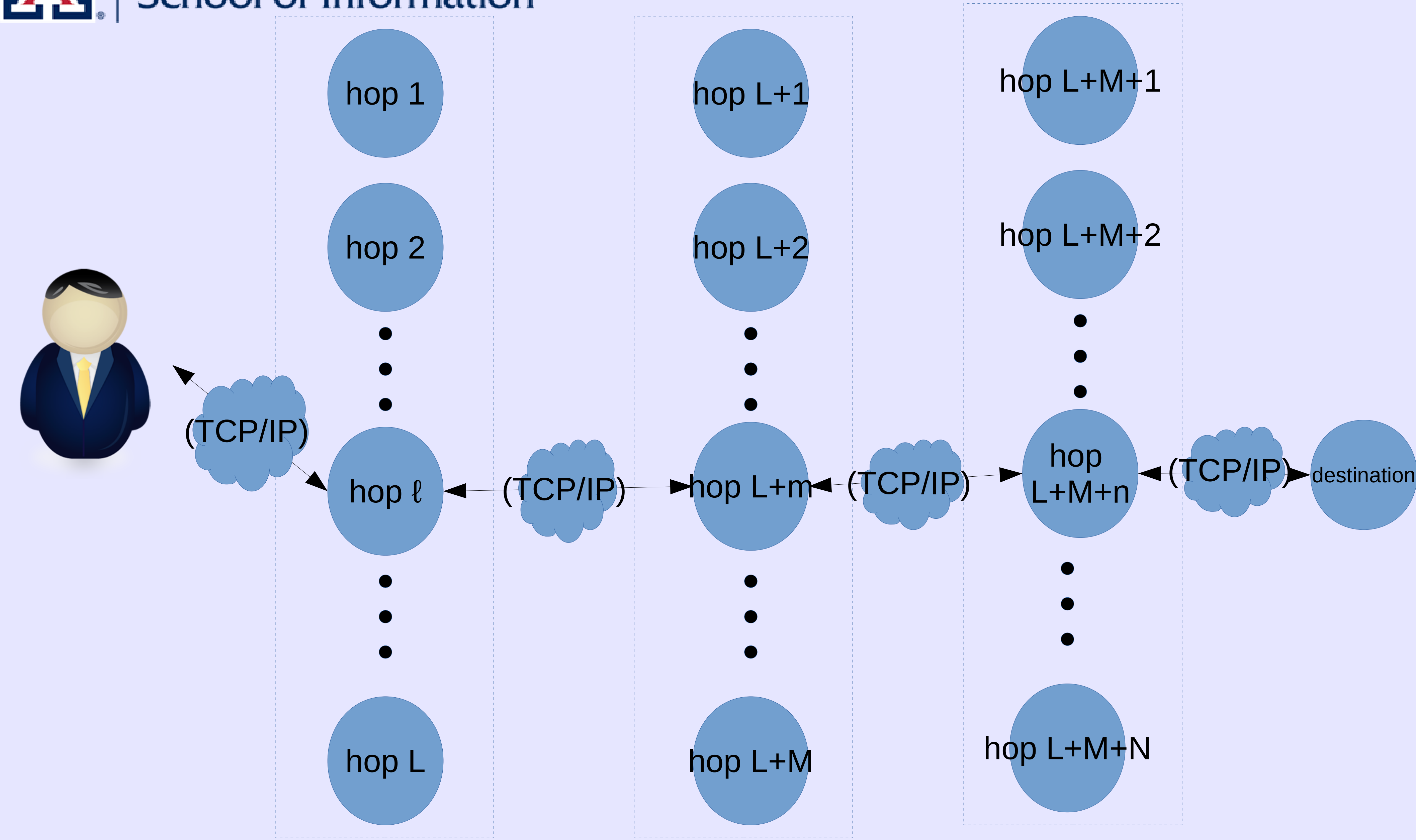
COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

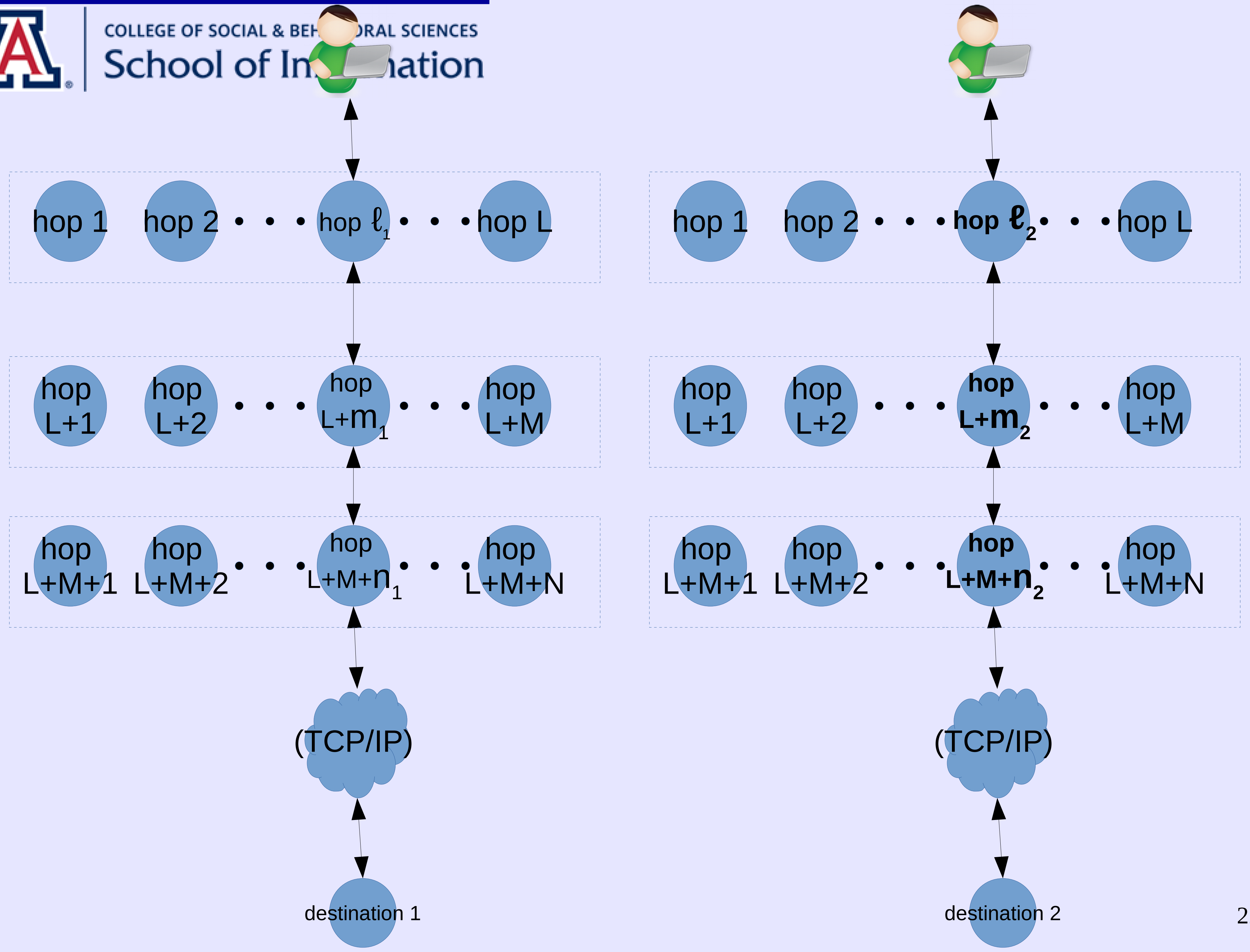



type-0
remailer
(TCP/IP)
destination


(TCP/IP)
(TCP/IP)
(TCP/IP)
(TCP/IP)
(TCP/IP)
(TCP/IP)
(TCP/IP)
(TCP/IP)


(TCP/IP)
(TCP/IP)
(TCP/IP)
(TCP/IP)
(TCP/IP)


hop 1
hop 2
hop ℓ
hop L
hop L+1
hop L+2
hop L+m
hop L+M
hop L+M+1
hop L+M+2
hop L+M+n
hop L+M+N
(TCP/IP)
(TCP/IP)
(TCP/IP)
(TCP/IP)
destination


hop 1
hop 2
hop ℓ_1
hop L
hop L+1
hop L+2
hop L+m_1
hop L+M
hop L+M+1
hop L+M+2
hop L+M+n_1
hop L+M+N
(TCP/IP)
destination 1
hop 1
hop 2
hop ℓ_2
hop L
hop L+1
hop L+2
hop L+m_2
hop L+M
hop L+M+1
hop L+M+2
hop L+M+n_2
hop L+M+N
(TCP/IP)
destination 2

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Topologies for Mixnets

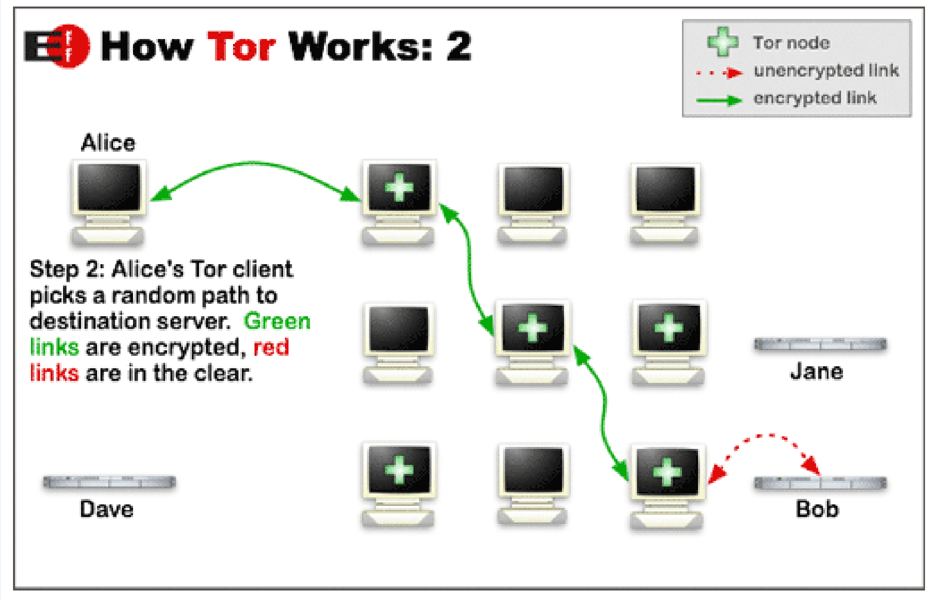
- Cascading: All nodes are always used, in the same order
- Scalability is a problem, requires setting up a fixed route with all nodes
- Only requires one honest node to preserve anonymity

Topologies for Mixnets

- User specified: user arbitrarily picks its route through the network
- Scalable, does not require initial configuration of a route
- Not anonymous if only one node is honest (nodes can figure out their positions)

Onion Routing

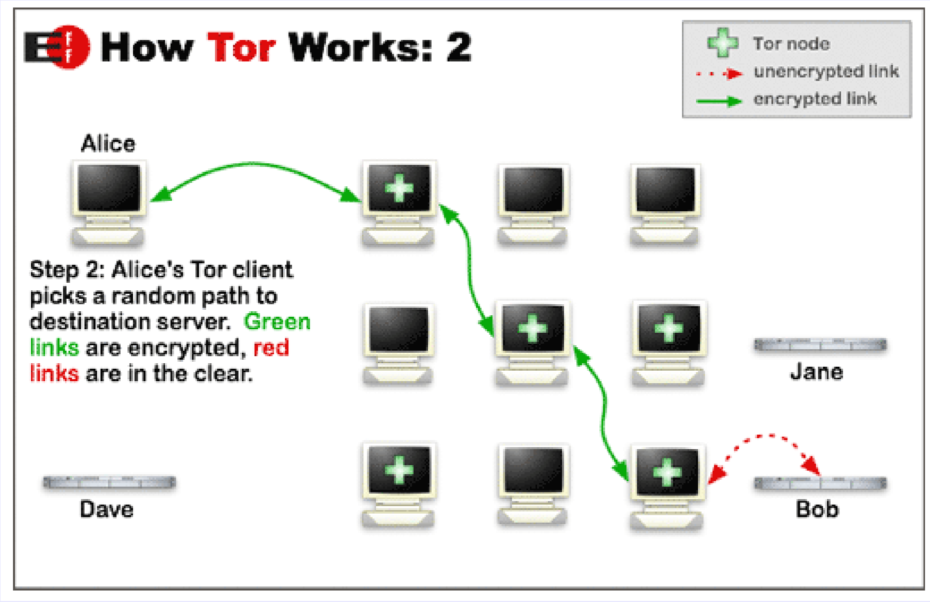
- First was from the US Naval Laboratory, 1996
  - pure peering at this stage, loafers!
- Freedom Network was an independent onion routing network from Zero Knowledge Systems
- Tor is a third-gen. onion routing network



COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Tor is an onion routing system

- Example: simplified, slightly out-of-date Tor (*link*)
- Distributed TCP overlay network
- Sets up a “virtual circuit” as a cascade of three onion relays (OR) from the initial client onion proxy (OP)
- guard (from “helper nodes”), relay, and exit nodes
  - each node only knows its immediate predecessor and successor



COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Tor is an onion routing system

- originally, onion routing systems sent an initial onion message that was “just layers” to set up the circuit; Tor does it in stages (“telescoping”)
- Next hop in the circuit is determined by unwrapping an “extend” relay cell with a symmetric key, which causes the OR to send its own “create” control cell

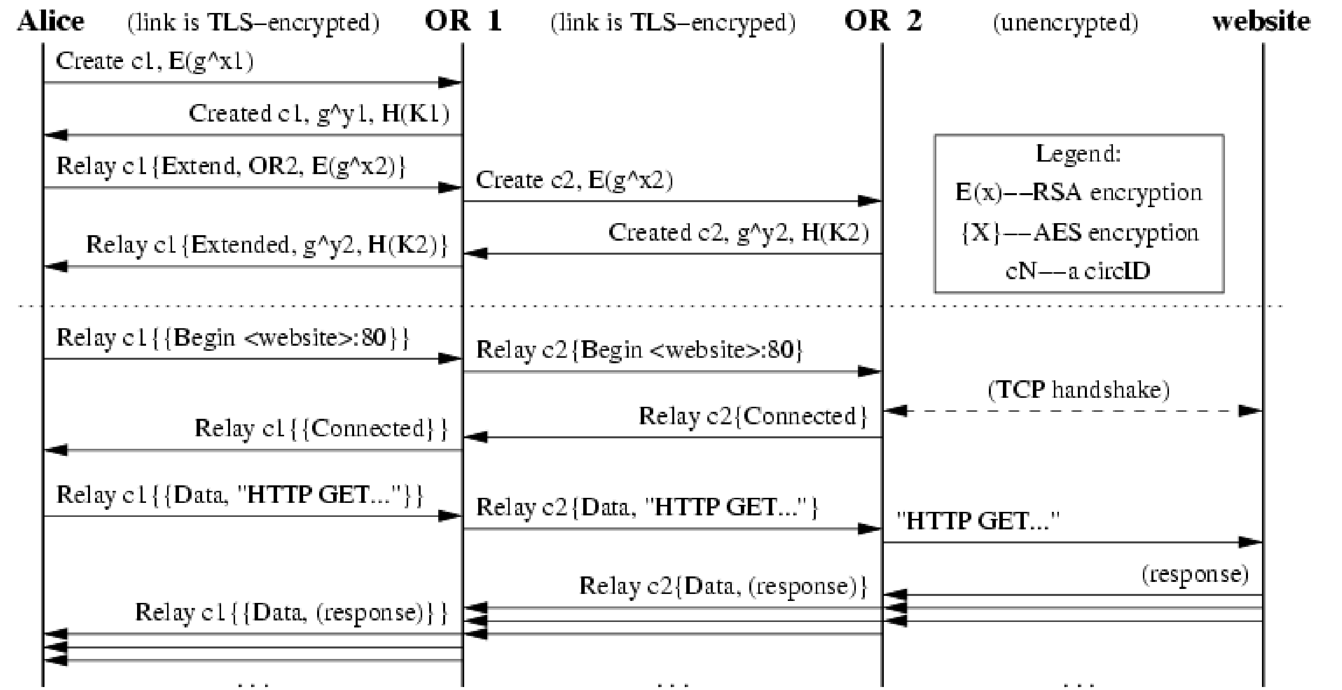


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

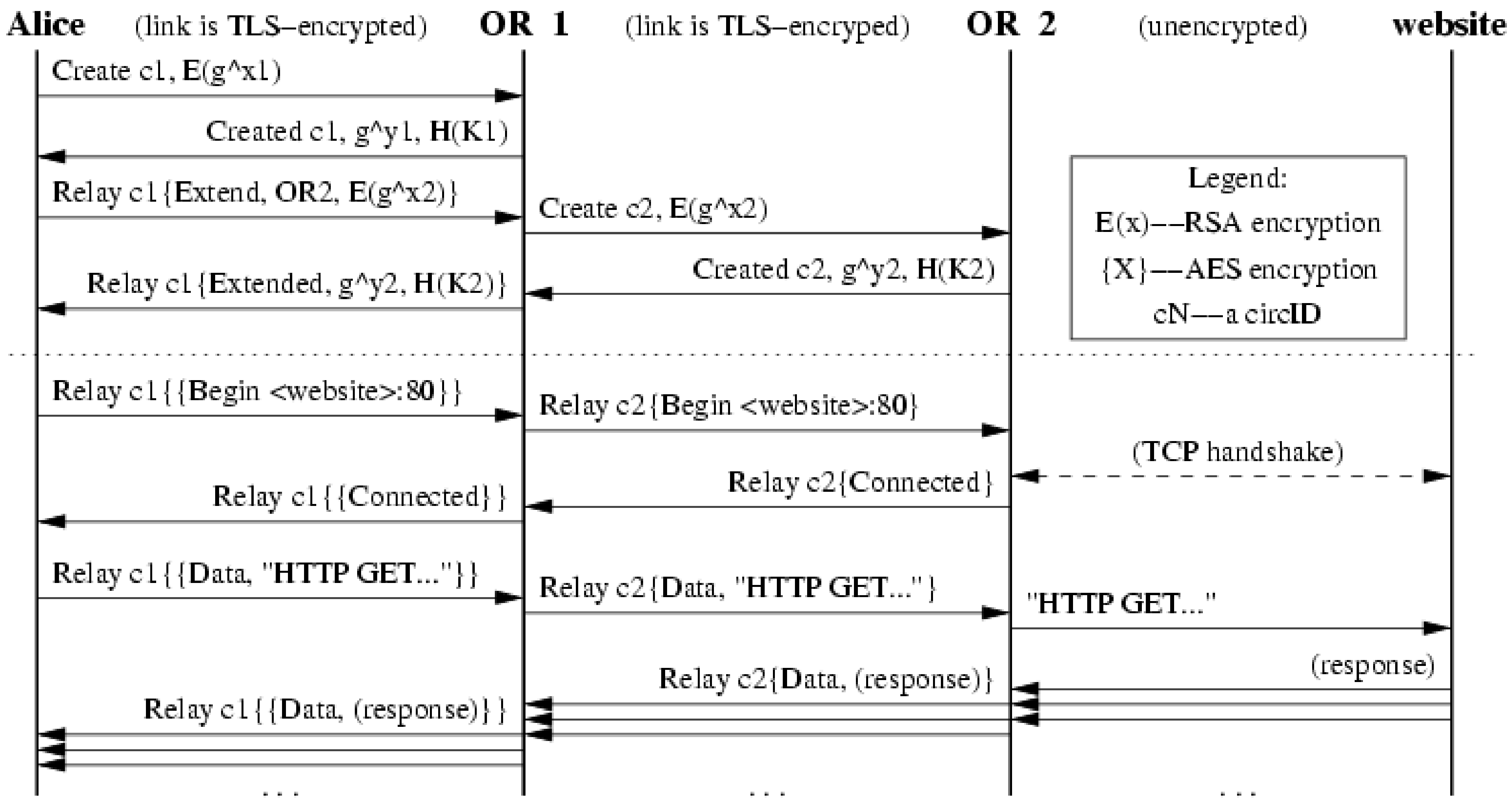


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

OP picks the route

- First picks the exit node E such that E's exit policy includes at least one pending stream that needs a circuit
- Choose N-1 distinct nodes (default is three), with some order
- Open a connection to the first (guard) node, negotiate session keys
- extend the circuit incrementally over the remaining N-1 nodes

Tor uses PKC to protect negotiation of a session key

- One hop at a time over an encrypted and authenticated channel
  - TLS: use *identity keys* to sign certs
- Use public-key cryptography (PKC) over this channel to set up an ephemeral session key
  - PKC is RSA (legacy) or Curve25519: use short-term *onion keys*
  - symmetric is AES, set up with DHE (legacy) or ECDHE
- Once ephemeral keys are set up OP layers them, and ORs unwrap them

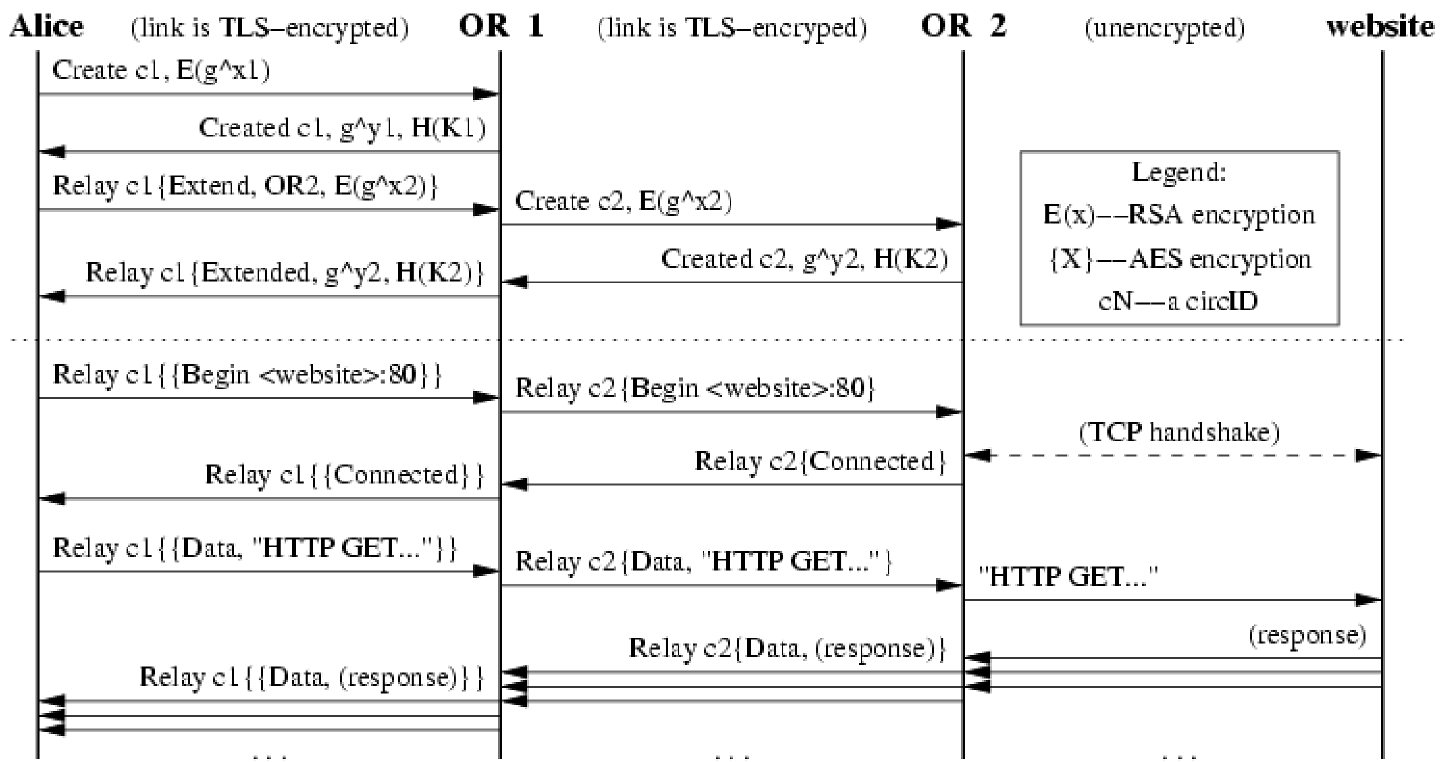


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

Discussion

- We have a public key for the guard. What reason did I give to not just use PKC to encrypt communications?

Session keys are negotiated using Diffie-Hellman Key Exchange

- First published in 1976; still around
- Alice and Bob want to share a secret key for use in a symmetric cipher. Every piece of information that they exchange is observed by their adversary Eve. How is it possible for Alice and Bob to agree on a key without making it available to Eve?

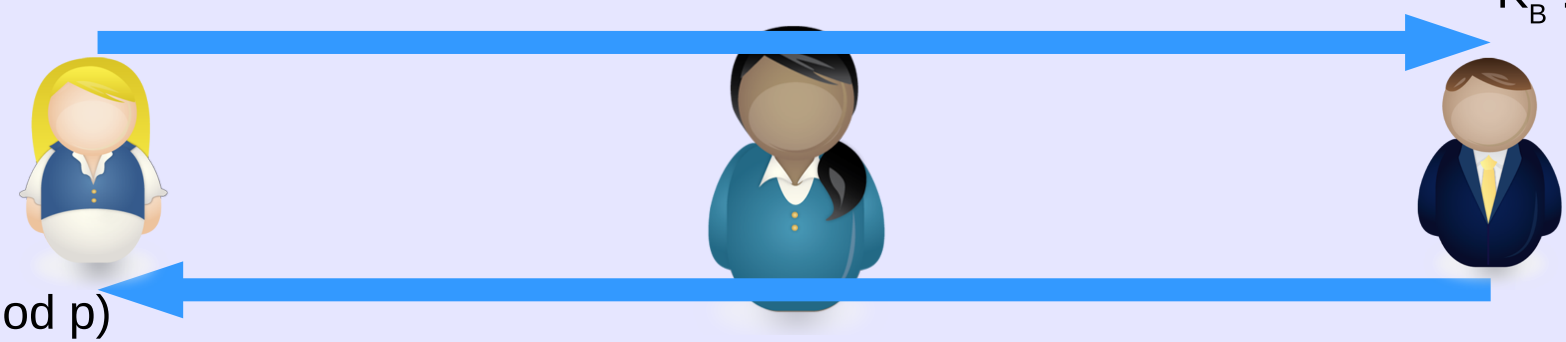
Diffie-Hellman

Publicly choose:
- a safe large prime *p* (e.g. Tor docs use rfc2409 section 6.2. But see Logjam)
- g, a primitive root mod p, with *2 <= g <= p-2*

Secretly generate:
- Alice and Bob randomly choose secret integers 1 <= *x, y* <= *p-2* respectively

$A := g^x \pmod p$

$K_B := A^y \pmod p$



$K_A := B^x \pmod p$

$B := g^y \pmod p$

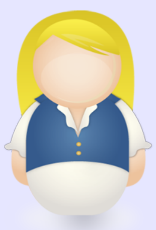
$K_A = (g^y)^x = (g^x)^y = K_B$ is the key

Diffie-Hellman

$x = \log_g A \pmod{p}$

$y = \log_g B \pmod{p}$

Diffie-Hellman



$$x = \log_g A \pmod p$$
$$y = \log_g B \pmod p$$

Discrete Log Problem is in NP

Diffie Hellman Problem is no harder than DL problem; there is no proof of the converse

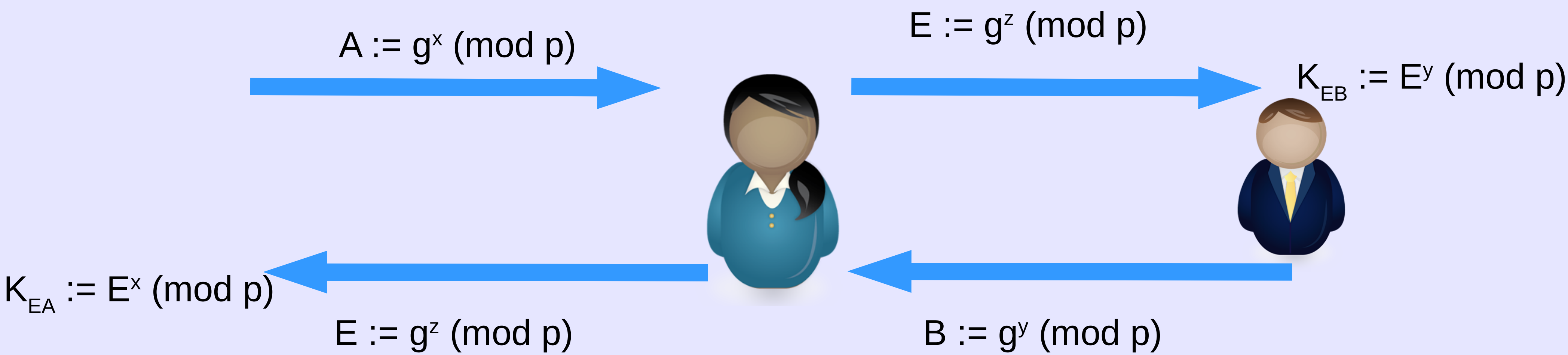
Question

- Ian Goldberg remarked that a good way to fight mass surveillance by a global passive adversary would be to “do a quick Diffie-Hellman” by default when setting up otherwise unprotected connections. He notes that this won’t help against an active attack. Can you guess what he means by an active attack?

MiTM Diffie-Hellman

Publicly choose:
a secure large prime *p*
g, a primitive root mod p, with $2 <= g <= p-2$

Secretly generate:
- Alice and Bob choose secret integers 0 <= $x, y <= p-2$ respectively
- Eve picks her own secret integer, z



COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

MiTM Diffie-Hellman

Publicly choose:
a secure large prime *p*
g, a primitive root mod p, with *2 <= g <*

K_{EA} := g^{xz} (mod p)
K_{EB} := g^{yz} (mod p)

Secretly generate:
- Alice and Bob choose secret integers 0 <= *x, y* <= *p-2* respectively
- Eve picks her own secret integer, *z*

A := g^x (mod p)

E := g^z (mod p)

K_{EB} := E^y (mod p)

K_{EA} := E^x (mod p)

E := g^z (mod p)

B := g^y (mod p)

Tor protocol, cont’d

- once session keys are agreed upon with DH, encrypt for the exit node, then encrypt the result for the relay node, and finally encrypt the result for the guard node
- send the layered result to the guard node
- guard node decrypts, gets the next hop and sends it on, then the middle key decrypts, ...
- on way back, each node uses the session key agreed on with the client OP, and passes to its neighbor in the circuit

Tor strengths and weaknesses

- Strengths
  - faster than mixnets
  - perfect forward secrecy
  - easy to run nodes, easy to use as a client: adds to security
  - bridges, pluggable transports for censorship circumvention
  - sandboxing
- Weaknesses
  - traffic analysis by a pervasive passive adversary
  - end-to-end timing attacks
  - content is revealed to exit node
  - blockable exit nodes

Who runs exit nodes?

- Universities (as of Oct 2017) (link)
  - MIT+, Michigan, CMU, UNC, Karlsruhrer IT, Stanford, Clarkson, U. Washington, Utah+, Caltech, RIT+, Bowdoin, Northeastern+, Princeton
- Bad people too! (Why might they do that?)
- Not Arizona :-(
  - yet :-)

Who runs hidden services?

- Propublica, Duckduckgo, Facebook, Scihub, Riseup, Protonmail, Debian, Whonix, The Intercept, Wikileaks, Securedrops for The Freedom of the Press Foundation, The Guardian, The Associated Press, NY TImes, USA Today, Washington Post, etc., TORCH (these are all onion links)
- A bunch of illegal stuff
- Hidden services are easy to set up (demo)
  - even inside firewalled networks