CS459/698
Privacy, Cryptography, Network and Data Security

Network Anonymity

Spring 2025, Monday/Wednesday 2:30pm-3:50pm

Recall a Little Bit About Privacy

Two “types” of information that could be privacy-sensitive:

- Data: refers to contents of messages, contents of a database...
- Metadata: any other information that is not data
  - When communication occurs
  - Who communicates
  - How often do they communicate
  - …
- Is metadata privacy important?

Recall a Little Bit About Privacy

Two “types” of information that could be privacy-sensitive:

- Data: refers to contents of messages, contents of a database...
- Metadata: any other information that is not data
  - When communication occurs
  - Who communicates
  - How often do they communicate
  - …

- Is metadata privacy important?
  - **Yes!!!**

The U.S. government "kill[s] people based on metadata"

Former head of the National Security Agency, Gen. Michael Hayden

Metadata Can Reveal a Lot

- Alice receives a call from a gynecologist then calls an abortion clinic.

- Bob visits the website of a local activist group then messages a large number of people. Later that day, some of those people are arrested at a protest.

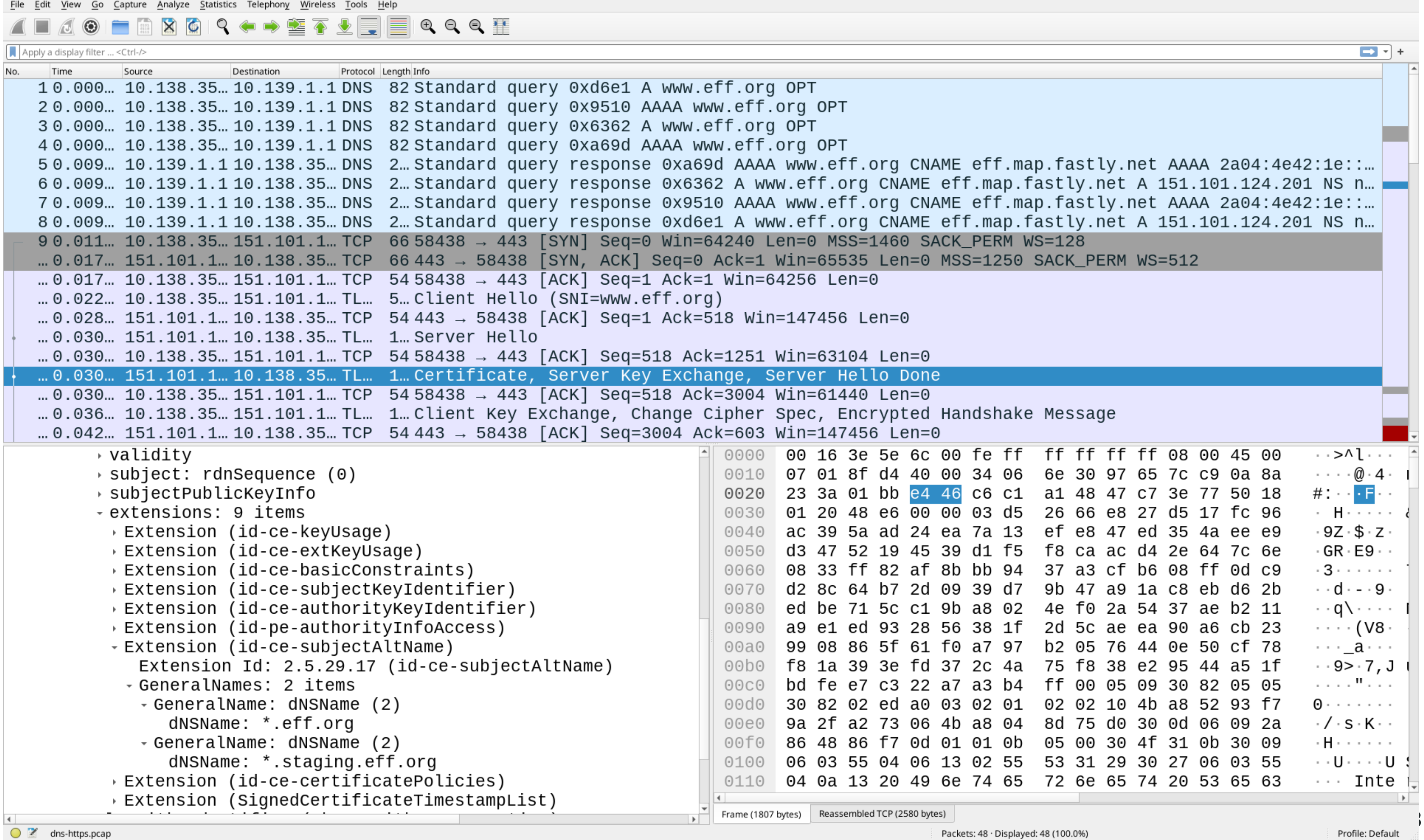
- Every day, Carol and Dave send dozens of messages to each other. One day, they stop sending messages altogether.

Anonymous Versus Confidential Communication

- Confidential communication encrypts **payload** (contents – HTTP/HTML, email, etc.)
- Parts of the communication that are not encrypted
  - Sometimes called meta-data
  - Network addresses (necessary for routing the message)
  - Email address, IP addresses (TCP ports)
  - Consider personal information
    - Your email provider likely knows “who” you are by your email address
    - Your ISP likely knows “who” you are by your IP address
  - Length (encryption does not hide the length – except minimally)
  - Timing

Metadata in Web Browsing

- Source leaked by source IP address
- Destination leaked by...
  - DNS queries
  - Destination IP address
  - TLS certificate (in some versions of TLS)
  - Server Name Indication

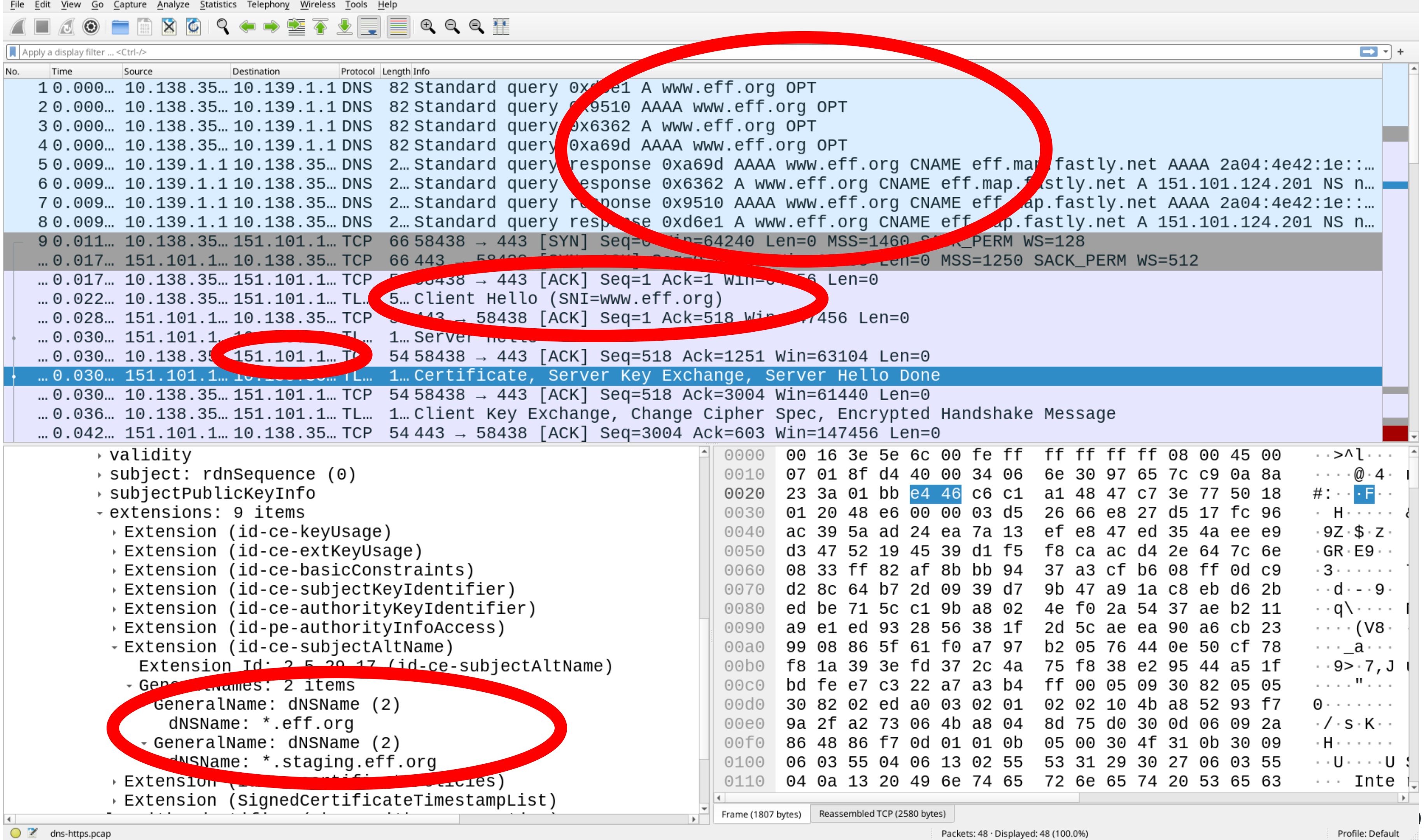

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

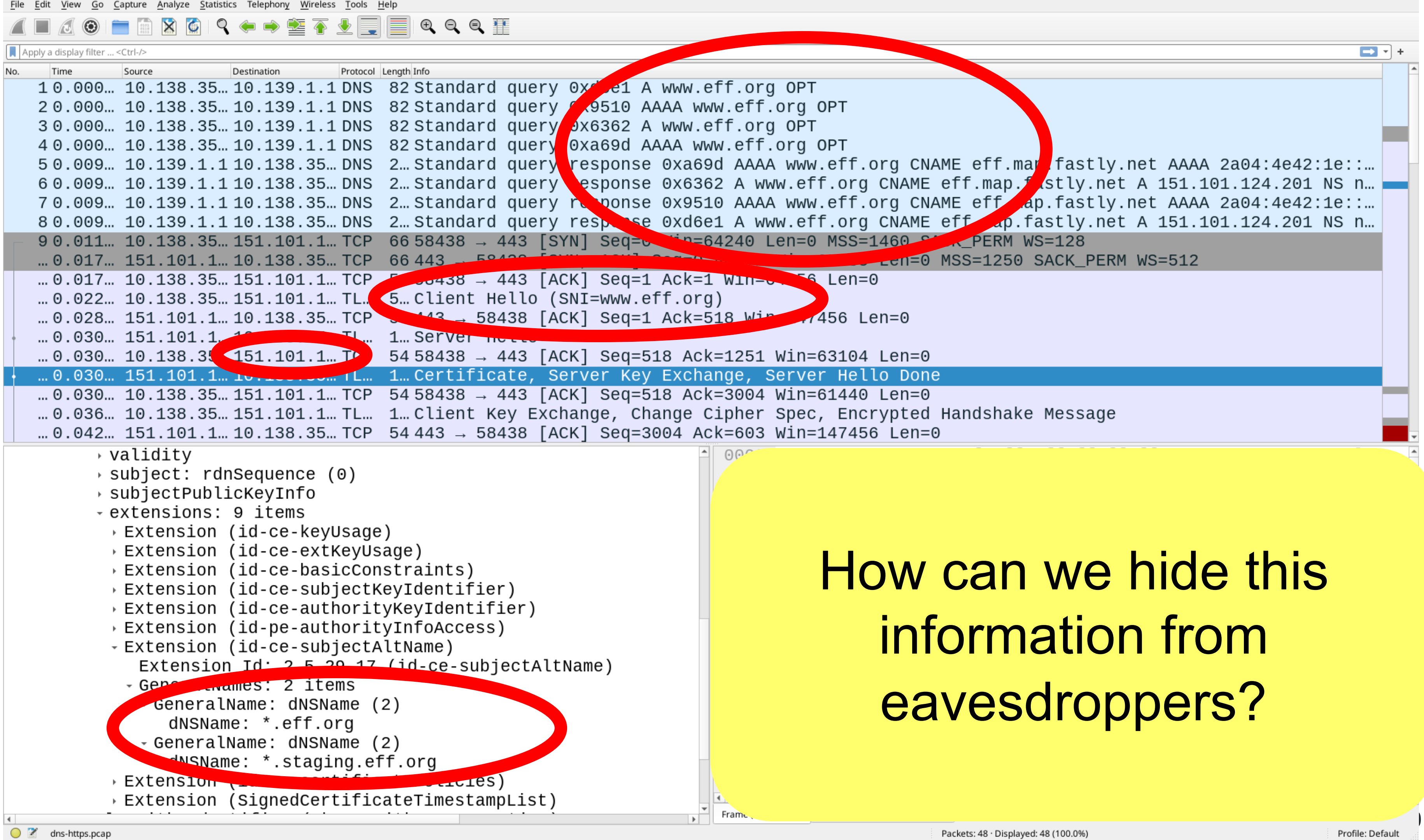
```
No.  Time      Source       Destination  Protocol Length Info
   1 0.000…  10.138.35… 10.139.1.1 DNS   82 Standard query 0xd6e1 A www.eff.org OPT
   2 0.000…  10.138.35… 10.139.1.1 DNS   82 Standard query 0x9510 AAAA www.eff.org OPT
   3 0.000…  10.138.35… 10.139.1.1 DNS   82 Standard query 0x6362 A www.eff.org OPT
   4 0.000…  10.138.35… 10.139.1.1 DNS   82 Standard query 0xa69d AAAA www.eff.org OPT
   5 0.009…  10.139.1.1 10.138.35… DNS   2… Standard query response 0xa69d AAAA www.eff.org CNAME eff.map.fastly.net AAAA 2a04:4e42:1e::…
   6 0.009…  10.139.1.1 10.138.35… DNS   2… Standard query response 0x6362 A www.eff.org CNAME eff.map.fastly.net A 151.101.124.201 NS n…
   7 0.009…  10.139.1.1 10.138.35… DNS   2… Standard query response 0x9510 AAAA www.eff.org CNAME eff.map.fastly.net AAAA 2a04:4e42:1e::…
   8 0.009…  10.139.1.1 10.138.35… DNS   2… Standard query response 0xd6e1 A www.eff.org CNAME eff.map.fastly.net A 151.101.124.201 NS n…
   9 0.011…  10.138.35… 151.101.1… TCP   66 58438 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
   … 0.017…  151.101.1… 10.138.35… TCP   66 443 → 58438 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1250 SACK_PERM WS=512
   … 0.017…  10.138.35… 151.101.1… TCP   54 58438 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
   … 0.022…  10.138.35… 151.101.1… TL…   5… Client Hello (SNI=www.eff.org)
   … 0.028…  151.101.1… 10.138.35… TCP   54 443 → 58438 [ACK] Seq=1 Ack=518 Win=147456 Len=0
   … 0.030…  151.101.1… 10.138.35… TL…   1… Server Hello
   … 0.030…  10.138.35… 151.101.1… TCP   54 58438 → 443 [ACK] Seq=518 Ack=1251 Win=63104 Len=0
   … 0.030…  151.101.1… 10.138.35… TL…   1… Certificate, Server Key Exchange, Server Hello Done
   … 0.030…  10.138.35… 151.101.1… TCP   54 58438 → 443 [ACK] Seq=518 Ack=3004 Win=61440 Len=0
   … 0.036…  10.138.35… 151.101.1… TL…   1… Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
   … 0.042…  151.101.1… 10.138.35… TCP   54 443 → 58438 [ACK] Seq=3004 Ack=603 Win=147456 Len=0
```

```
        › validity
        › subject: rdnSequence (0)
        › subjectPublicKeyInfo
        ⌄ extensions: 9 items
          › Extension (id-ce-keyUsage)
          › Extension (id-ce-extKeyUsage)
          › Extension (id-ce-basicConstraints)
          › Extension (id-ce-subjectKeyIdentifier)
          › Extension (id-ce-authorityKeyIdentifier)
          › Extension (id-pe-authorityInfoAccess)
          ⌄ Extension (id-ce-subjectAltName)
              Extension Id: 2.5.29.17 (id-ce-subjectAltName)
            ⌄ GeneralNames: 2 items
              ⌄ GeneralName: dNSName (2)
                  dNSName: *.eff.org
              ⌄ GeneralName: dNSName (2)
                  dNSName: *.staging.eff.org
          › Extension (id-ce-certificatePolicies)
          › Extension (SignedCertificateTimestampList)
```

```
0000  00 16 3e 5e 6c 00 fe ff  ff ff ff ff 08 00 45 00   ··>^l···
0010  07 01 8f d4 40 00 34 06  6e 30 97 65 7c c9 0a 8a   ····@·4·
0020  23 3a 01 bb e4 46 c6 c1  a1 48 47 c7 3e 77 50 18   #:···F··
0030  01 20 48 e6 00 00 03 d5  26 66 e8 27 d5 17 fc 96   · H·····
0040  ac 39 5a ad 24 ea 7a 13  ef e8 47 ed 35 4a ee e9   ·9Z·$·z·
0050  d3 47 52 19 45 39 d1 f5  f8 ca ac d4 2e 64 7c 6e   ·GR·E9··
0060  08 33 ff 82 af 8b bb 94  37 a3 cf b6 08 ff 0d c9   ·3······
0070  d2 8c 64 b7 2d 09 39 d7  9b 47 a9 1a c8 eb d6 2b   ··d·-·9·
0080  ed be 71 5c c1 9b a8 02  4e f0 2a 54 37 ae b2 11   ··q\····
0090  a9 e1 ed 93 28 56 38 1f  2d 5c ae ea 90 a6 cb 23   ····(V8·
00a0  99 08 86 5f 61 f0 a7 97  b2 05 76 44 0e 50 cf 78   ···_a···
00b0  f8 1a 39 3e fd 37 2c 4a  75 f8 38 e2 95 44 a5 1f   ··9>·7,J
00c0  bd fe e7 c3 22 a7 a3 b4  ff 00 05 09 30 82 05 05   ····"···
00d0  30 82 02 ed a0 03 02 01  02 02 10 4b a8 52 93 f7   0·······
00e0  9a 2f a2 73 06 4b a8 04  8d 75 d0 30 0d 06 09 2a   ·/·s·K··
00f0  86 48 86 f7 0d 01 01 0b  05 00 30 4f 31 0b 30 09   ·H······
0100  06 03 55 04 06 13 02 55  53 31 29 30 27 06 03 55   ··U····U
0110  04 0a 13 20 49 6e 74 65  72 6e 65 74 20 53 65 63   ··· Inte
```

Frame (1807 bytes)
Reassembled TCP (2580 bytes)
dns-https.pcap
Packets: 48 · Displayed: 48 (100.0%)
Profile: Default



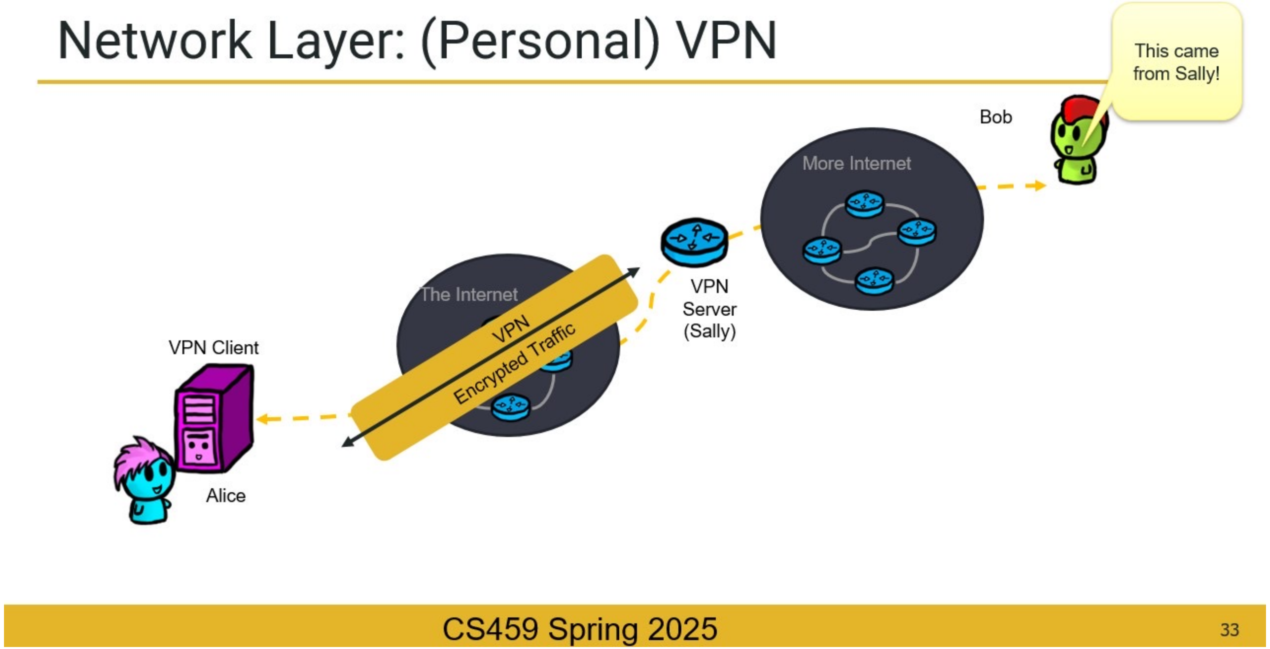

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No. Time     Source      Destination Protocol Length Info
  1 0.000… 10.138.35… 10.139.1.1 DNS 82 Standard query 0xd6e1 A www.eff.org OPT
  2 0.000… 10.138.35… 10.139.1.1 DNS 82 Standard query 0x9510 AAAA www.eff.org OPT
  3 0.000… 10.138.35… 10.139.1.1 DNS 82 Standard query 0x6362 A www.eff.org OPT
  4 0.000… 10.138.35… 10.139.1.1 DNS 82 Standard query 0xa69d AAAA www.eff.org OPT
  5 0.009… 10.139.1.1 10.138.35… DNS 2… Standard query response 0xa69d AAAA www.eff.org CNAME eff.map.fastly.net AAAA 2a04:4e42:1e::…
  6 0.009… 10.139.1.1 10.138.35… DNS 2… Standard query response 0x6362 A www.eff.org CNAME eff.map.fastly.net A 151.101.124.201 NS n…
  7 0.009… 10.139.1.1 10.138.35… DNS 2… Standard query response 0x9510 AAAA www.eff.org CNAME eff.map.fastly.net AAAA 2a04:4e42:1e::…
  8 0.009… 10.139.1.1 10.138.35… DNS 2… Standard query response 0xd6e1 A www.eff.org CNAME eff.map.fastly.net A 151.101.124.201 NS n…
  9 0.011… 10.138.35… 151.101.1… TCP 66 58438 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
  … 0.017… 151.101.1… 10.138.35… TCP 66 443 → 58438 [SYN, ACK] Seq=0 [illegible] Len=0 MSS=1250 SACK_PERM WS=512
  … 0.017… 10.138.35… 151.101.1… TCP 54 58438 → 443 [ACK] Seq=1 Ack=1 Win=6[illegible]56 Len=0
  … 0.022… 10.138.35… 151.101.1… TL… 5… Client Hello (SNI=www.eff.org)
  … 0.028… 151.101.1… 10.138.35… TCP 54 443 → 58438 [ACK] Seq=1 Ack=518 Win=147456 Len=0
  … 0.030… 151.101.1… 10.138.35… TL… 1… Server Hello
  … 0.030… 10.138.35… 151.101.1… TCP 54 58438 → 443 [ACK] Seq=518 Ack=1251 Win=63104 Len=0
  … 0.030… 151.101.1… 10.138.35… TL… 1… Certificate, Server Key Exchange, Server Hello Done
  … 0.030… 10.138.35… 151.101.1… TCP 54 58438 → 443 [ACK] Seq=518 Ack=3004 Win=61440 Len=0
  … 0.036… 10.138.35… 151.101.1… TL… 1… Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
  … 0.042… 151.101.1… 10.138.35… TCP 54 443 → 58438 [ACK] Seq=3004 Ack=603 Win=147456 Len=0
```

```
        › validity
        › subject: rdnSequence (0)
        › subjectPublicKeyInfo
        ⌄ extensions: 9 items
          › Extension (id-ce-keyUsage)
          › Extension (id-ce-extKeyUsage)
          › Extension (id-ce-basicConstraints)
          › Extension (id-ce-subjectKeyIdentifier)
          › Extension (id-ce-authorityKeyIdentifier)
          › Extension (id-pe-authorityInfoAccess)
          ⌄ Extension (id-ce-subjectAltName)
              Extension Id: 2.5.29.17 (id-ce-subjectAltName)
            ⌄ GeneralNames: 2 items
              ⌄ GeneralName: dNSName (2)
                  dNSName: *.eff.org
              ⌄ GeneralName: dNSName (2)
                  dNSName: *.staging.eff.org
          › Extension (id-ce-certificatePolicies)
          › Extension (SignedCertificateTimestampList)
```

How can we hide this information from eavesdroppers?

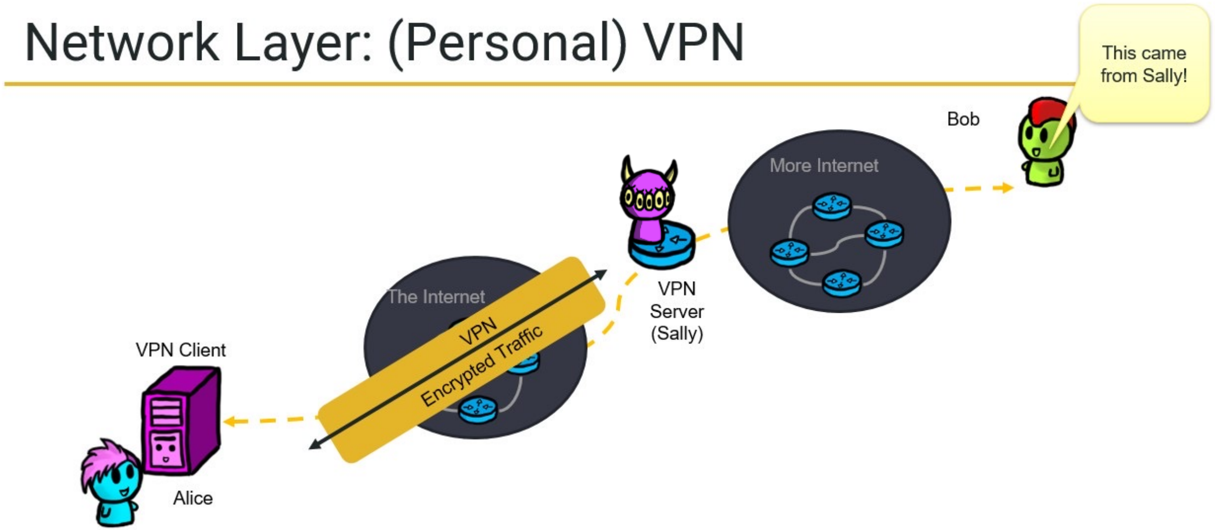
Recall Personal VPNs...



Network Layer: (Personal) VPN

CS459 Spring 2025 33

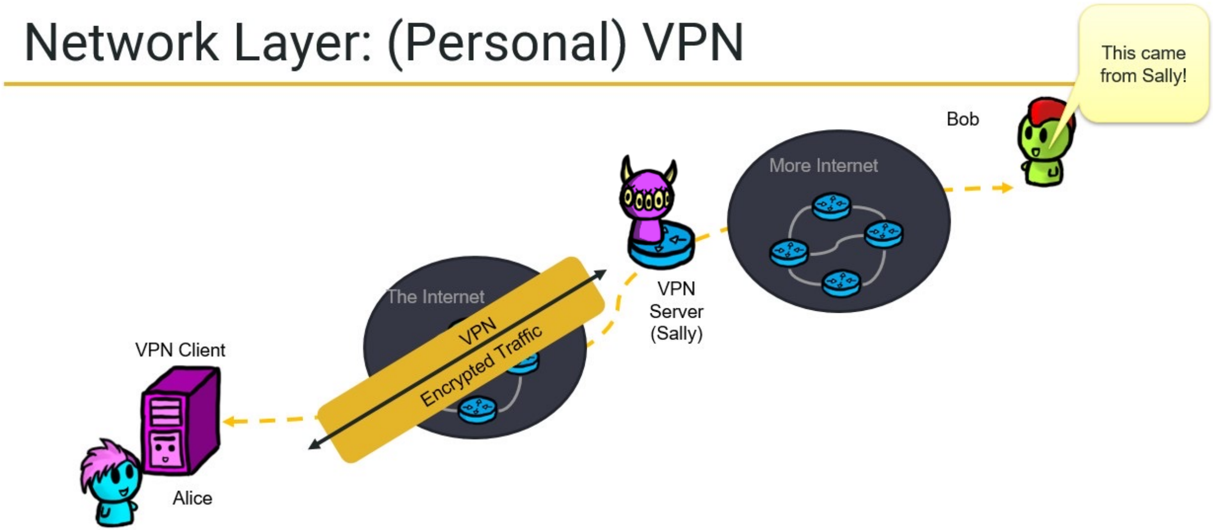
Privacy from the VPN Server?



Network Layer: (Personal) VPN

The VPN server knows both the sender and receiver.

Privacy from the VPN Server?

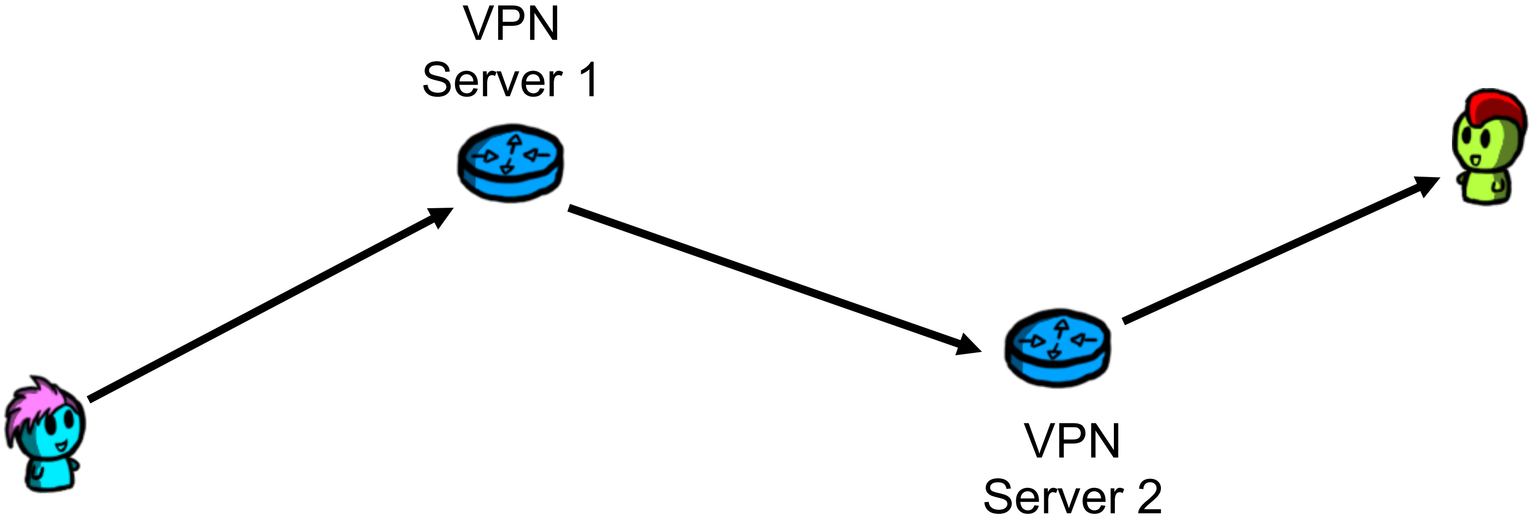


Network Layer: (Personal) VPN

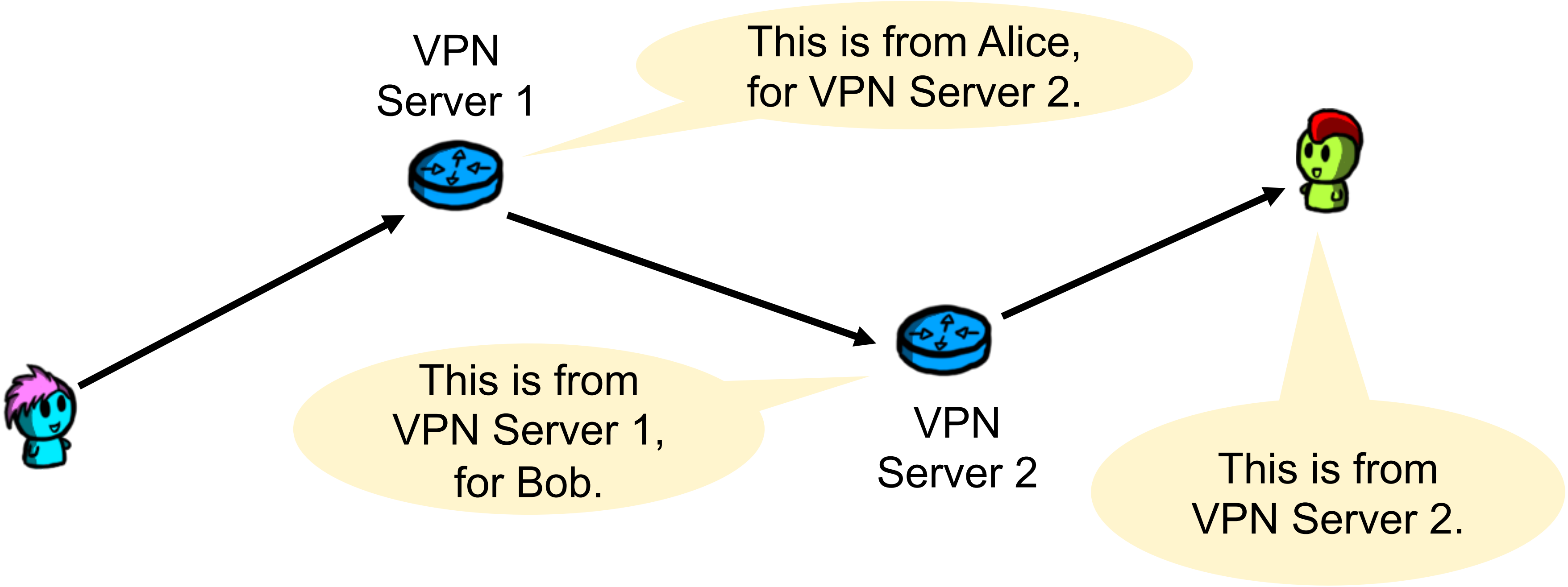
The VPN server knows both the sender and receiver.

What if we had multiple relays?

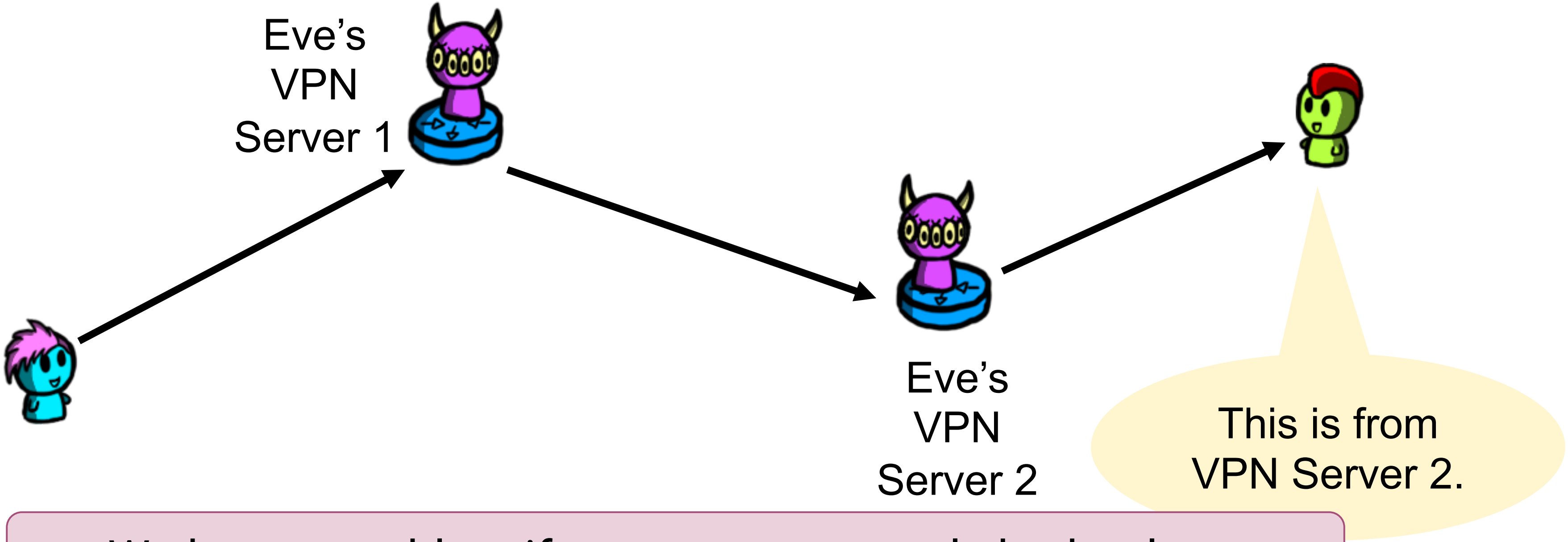
Multiple Relays



Multiple Relays



Multiple Relays



We have a problem if one person controls both relays.

Tor

Tor is?

Tor is a **low-latency** anonymous communication system

Tor is?

Tor is a **low-latency** anonymous communication system

Tor is?

Tor is a **low-latency** anonymous communication system

Tor has about **8,000 nodes** run by volunteers, scattered around the Internet; these are also called Onion Routers

Tor is?

Tor is a **low-latency** anonymous communication system

Tor has about **8,000 nodes** run by volunteers, scattered around the Internet; these are also called Onion Routers

Tor makes internet browsing unlinkably* anonymous. But Tor does not (and cannot) hide the existence of the transaction (website visit) altogether

Tor is?

Tor is a **low-latency** anonymous communication system

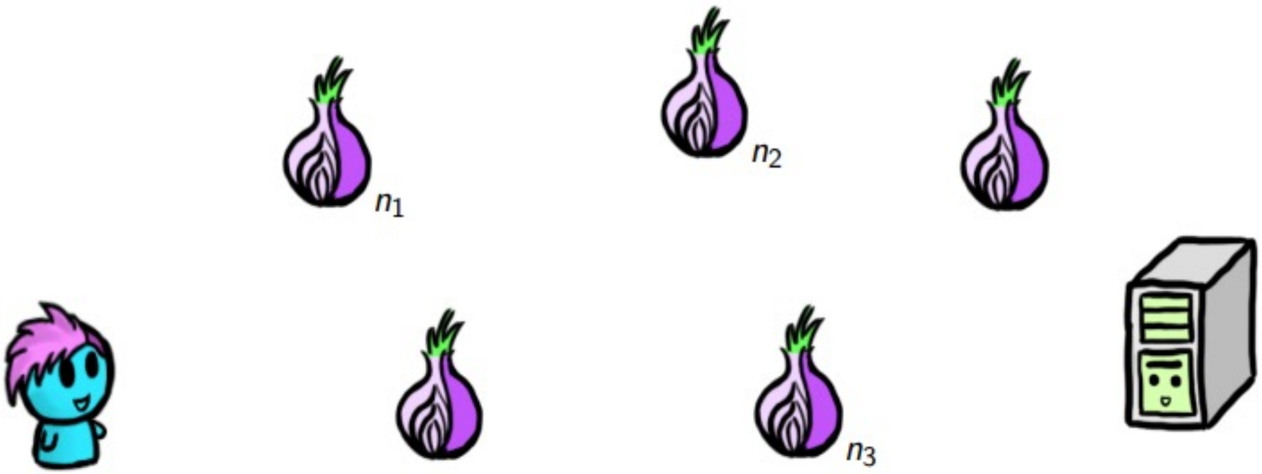
Tor has about **8,000 nodes** run by volunteers, scattered around the Internet; these are also called Onion Routers

Tor makes internet browsing unlinkably* anonymous. But Tor does not (and cannot) hide the existence of the transaction (website visit) altogether

Tor is not TOR!

Tor: Building a Circuit (I)

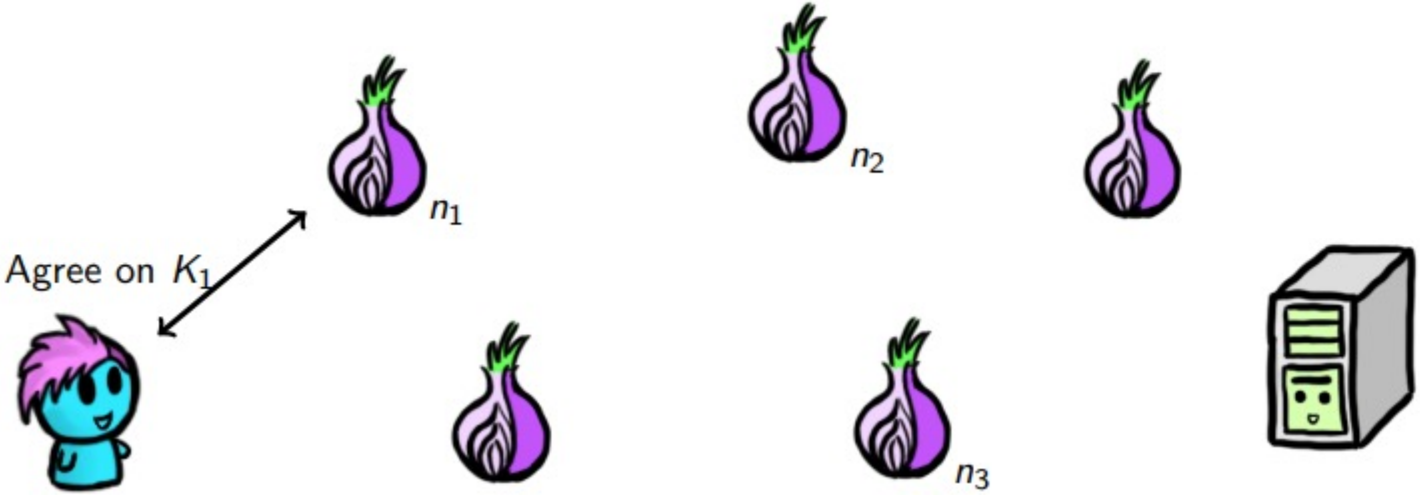
Goal: Alice wants to connect to a server without revealing her IP address



Alice has a global view of available Onion Routers (and their verification keys!)

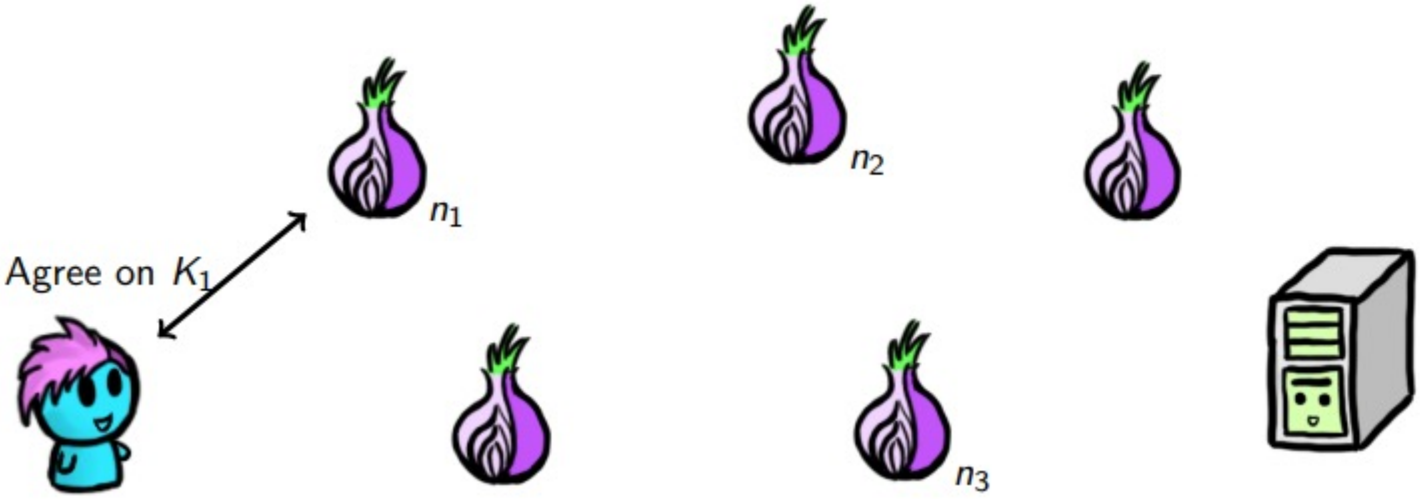
Tor: Building a Circuit (II)

Alice picks Tor node n_1 and uses PKC to establish an encrypted communication channel to it (much like TLS)



Tor: Building a Circuit (II)

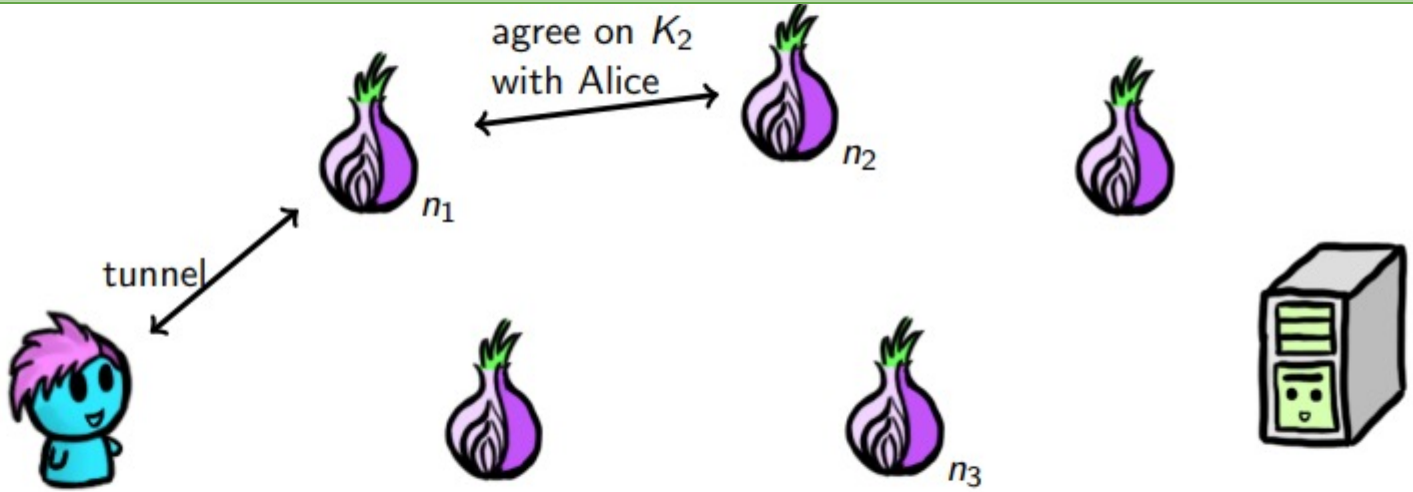
Alice picks Tor node n_1 and uses PKC to establish an encrypted communication channel to it (much like TLS)



The result is a secret key K_1 shared by Alice and n_1

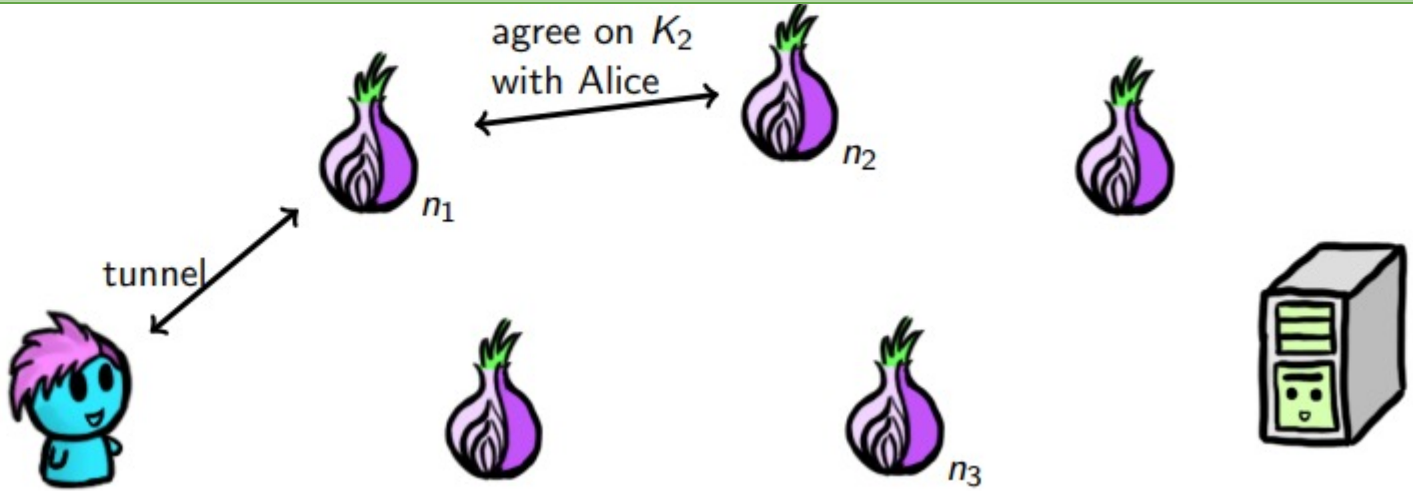
Tor: Building a Circuit (III)

Alice tells n_1 to contact a second node (n_2), and establishes a new encrypted comm.channel to n_2, tunneled within the previous one to n_1



Tor: Building a Circuit (III)

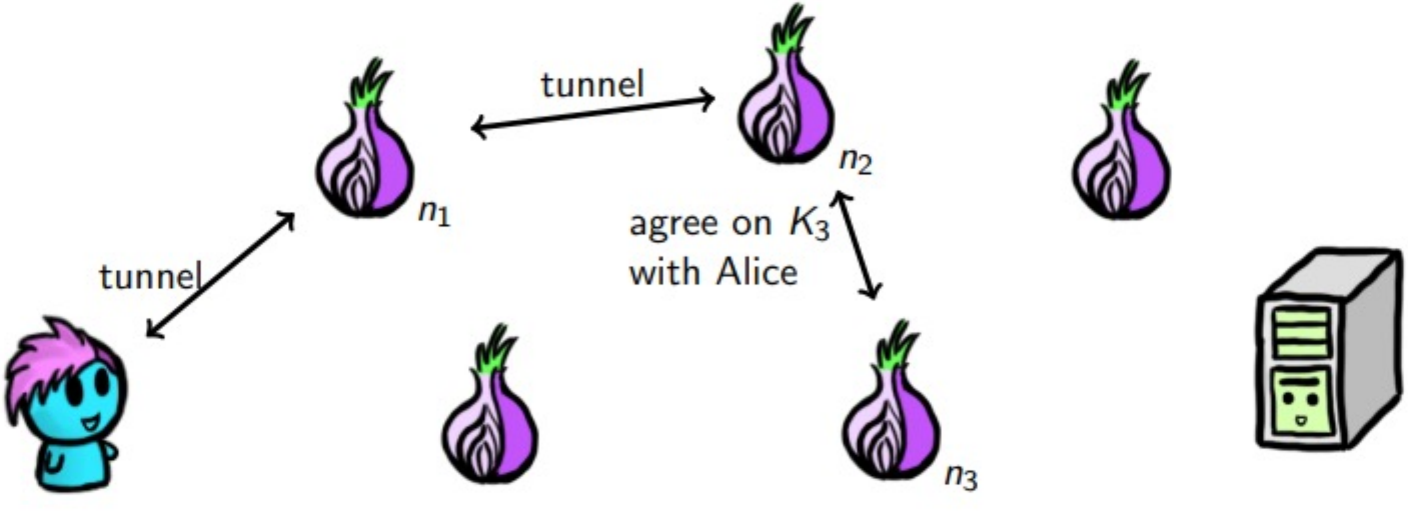
Alice tells n_1 to contact a second node (n_2), and establishes a new encrypted comm.channel to n_2, tunneled within the previous one to n_1



The result is a secret key K_2 shared between Alice and n_2, which is unknown to n_1

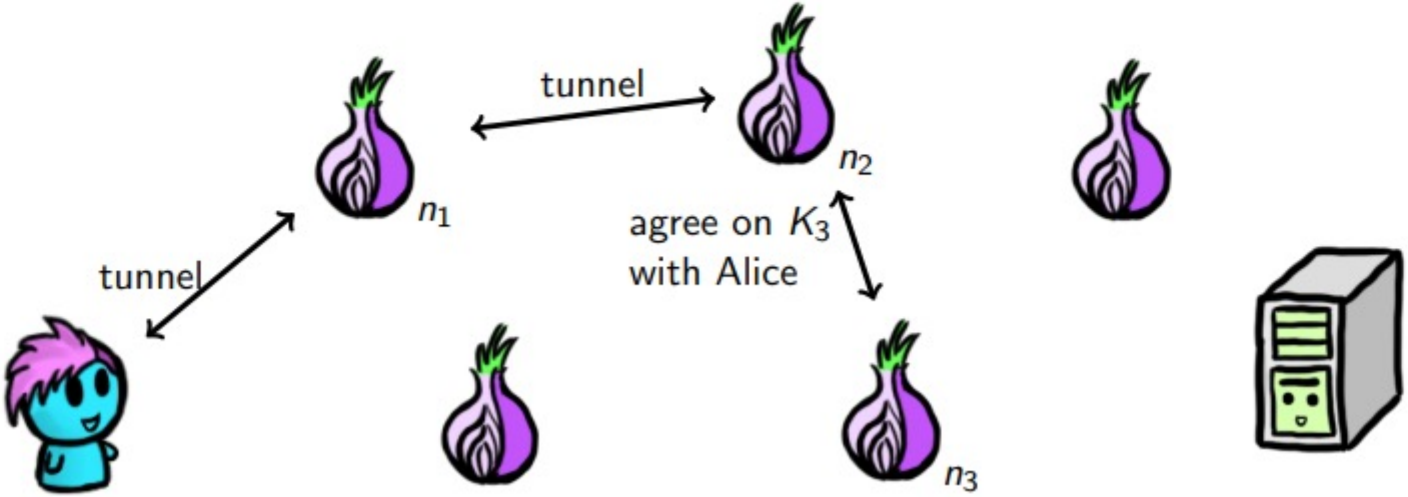
Tor: Building a Circuit (IV)

Alice tells n_2 to contact a third node (n_3), establishes a new encrypted communication channel to n_3, tunneled within the previous one to n_2



Tor: Building a Circuit (IV)

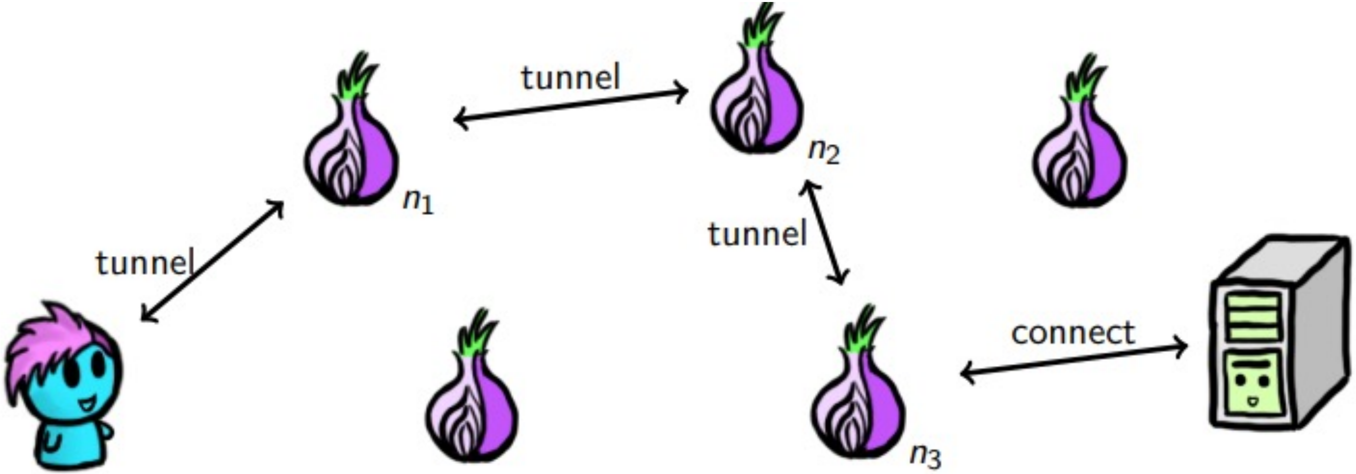
Alice tells n_2 to contact a third node (n_3), establishes a new encrypted communication channel to n_3, tunneled within the previous one to n_2



The result is a secret key K_3 shared between Alice and n_3, which is unknown to n_1 and n_2

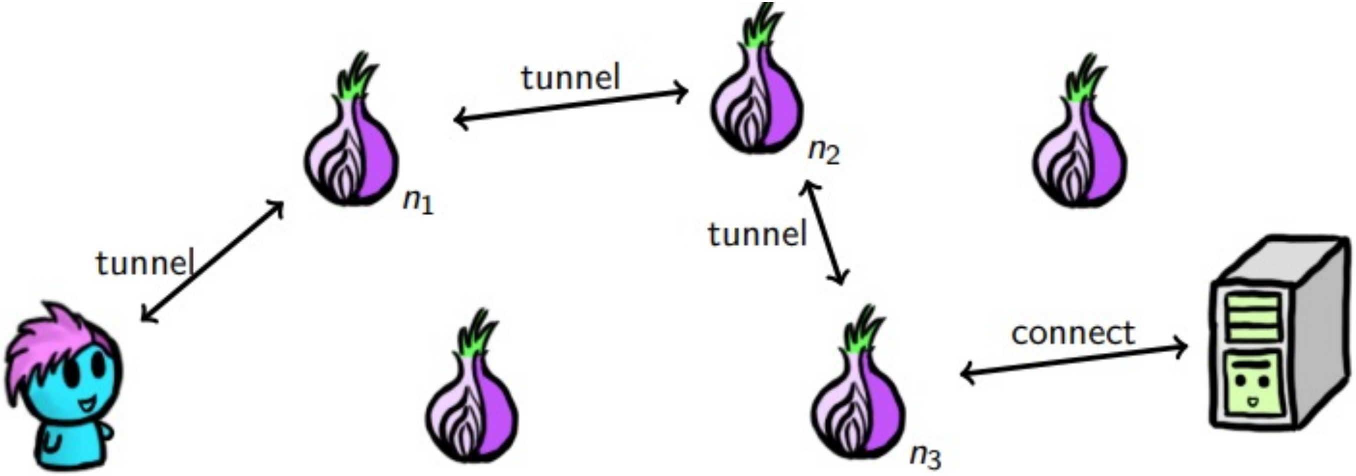
Tor: Building a Circuit (V)

... And so on, for as many steps as she likes (usually 3) ...



Tor: Building a Circuit (V)

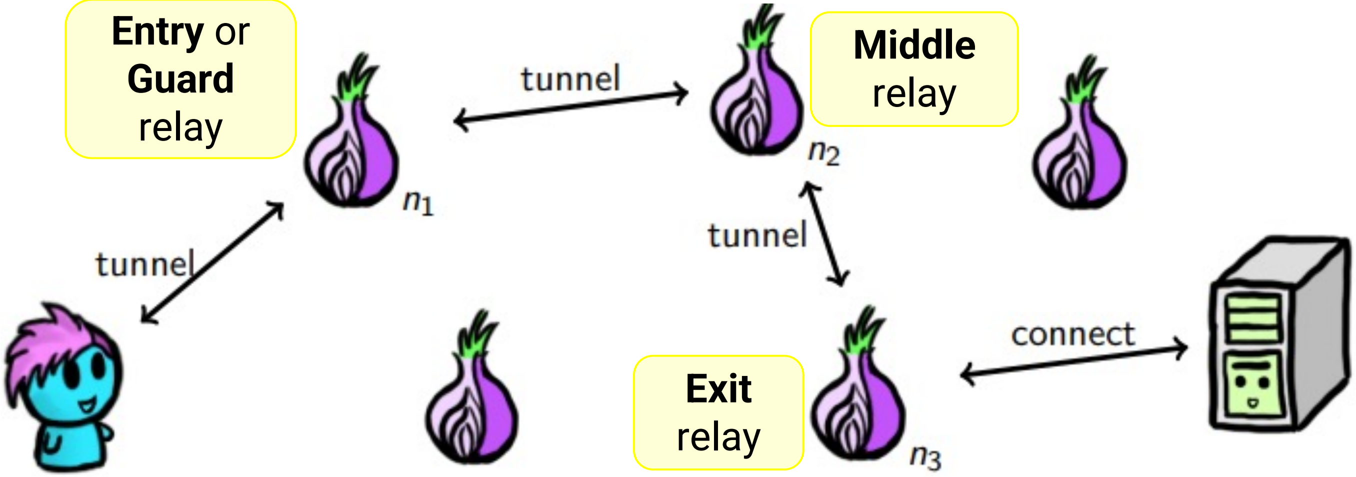
... And so on, for as many steps as she likes (usually 3) ...



Alice tells the last node (within the layers of tunnels) to connect to the website

Tor: Building a Circuit (V)

... And so on, for as many steps as she likes (usually 3) ...



Entry or **Guard** relay

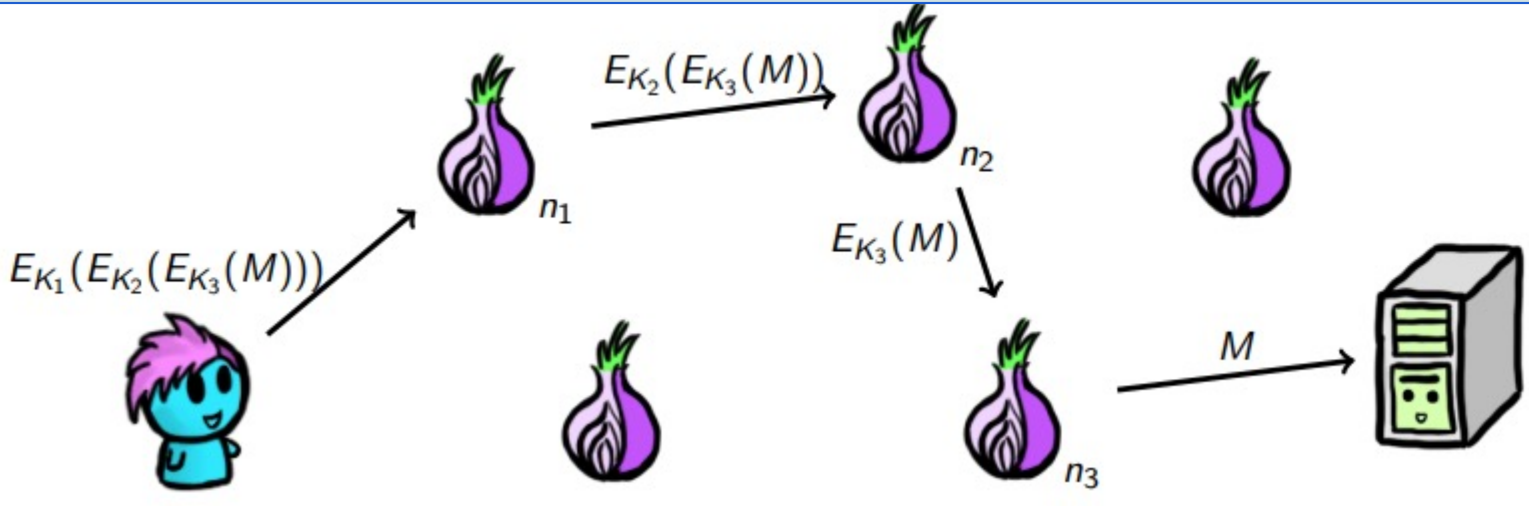
Middle relay

Exit relay

Alice tells the last node (within the layers of tunnels) to connect to the website

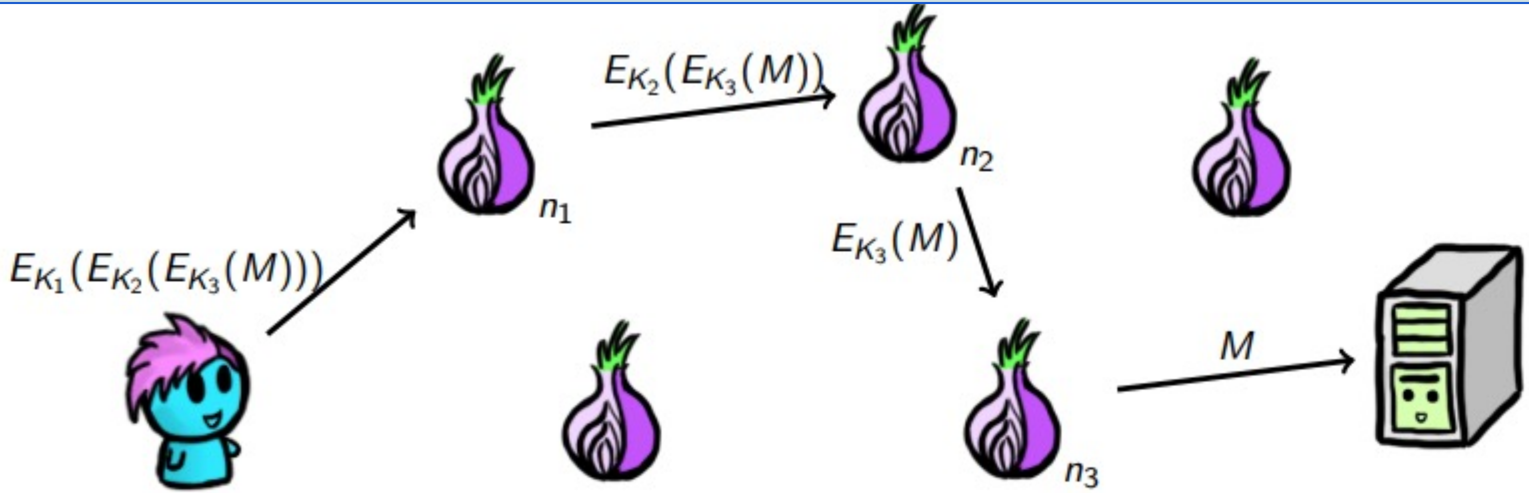
Sending Messages with Tor

Alice encrypts her message “like an onion”; each node peels a layer off and forwards it to the next step



Sending Messages with Tor

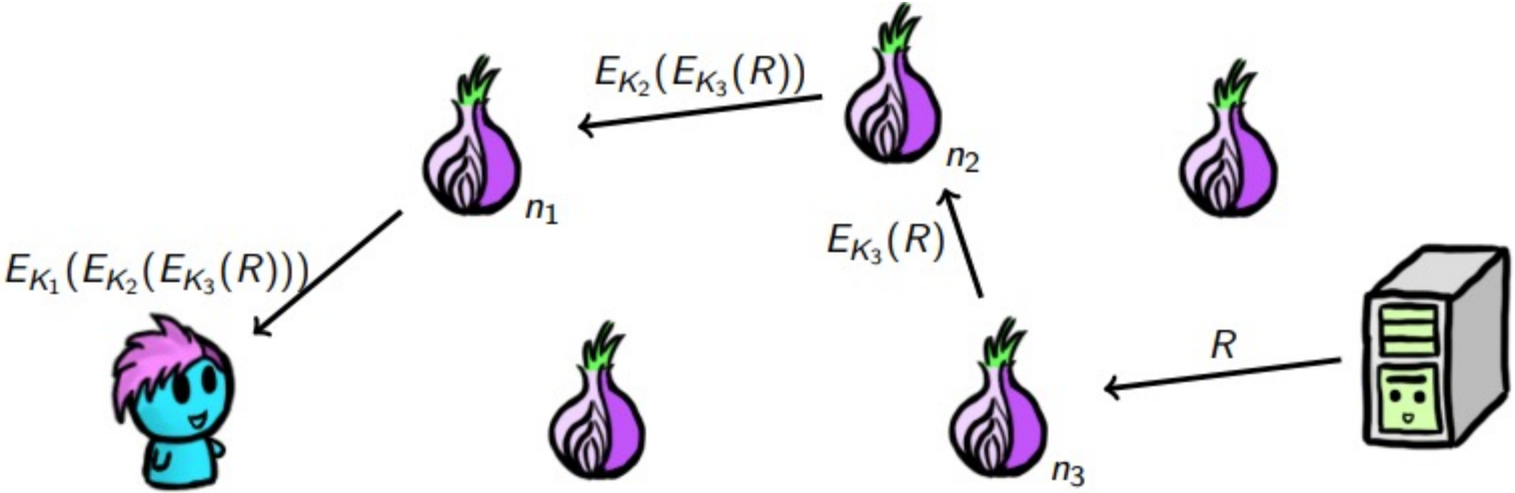
Alice encrypts her message “like an onion”; each node peels a layer off and forwards it to the next step



If connecting to a web server, M may be encrypted (e.g., TLS)

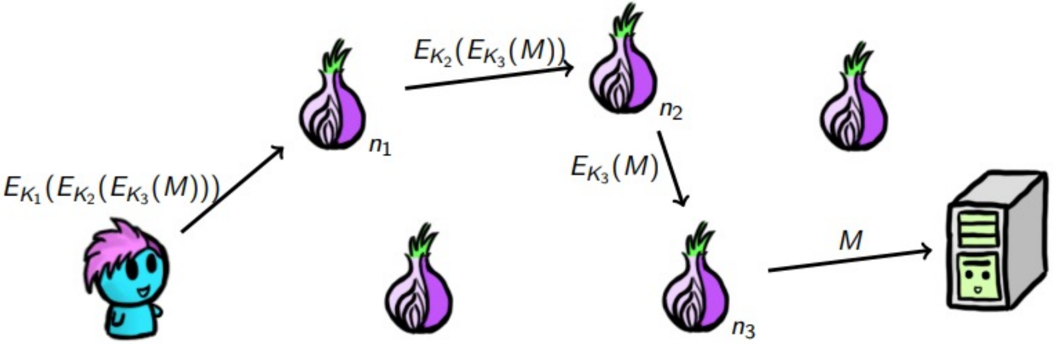
Replies in Tor

The server replies with R, sending it back to n_3. The nodes encrypt the message back and Alice decrypts all the layers.



Who knows what?

	Alice’s identity	Destination	Content
n1	Yes	No	No
n2	No	No	No
n3	No	Yes	Maybe
Destination	No	Yes (self)	Yes



Answer this…

Q: Why must Alice choose all nodes, instead of letting each node pick the next one?

Answer this…

Q: Why must Alice choose all nodes, instead of letting each node pick the next one?

A: A malicious node would pick another malicious node. The user must have the ability to choose the nodes

Answer this…

Q: Why must Alice choose all nodes, instead of letting each node pick the next one?

A: A malicious node would pick another malicious node. The user must have the ability to choose the nodes

Q: What happens if Eve can inspect all network links? (a global passive adversary)

Answer this…

Q: Why must Alice choose all nodes, instead of letting each node pick the next one?

A: A malicious node would pick another malicious node. The user must have the ability to choose the nodes

Q: What happens if Eve can inspect all network links? (a global passive adversary)

A: Tor does not protect against a global passive adversary. The adversary could de-anonymize Alice.

Answer some more…

Q: What happens when Eve can inspect the incoming and outgoing traffic of a single node?

Answer some more…

Q: What happens when Eve can inspect the incoming and outgoing traffic of a single node?

A: Alice is probably fine... but we'll see attacks in this setting in the next lecture

Answer some more...

Q: What happens when Eve can inspect the incoming and outgoing traffic of a single node?

A: Alice is probably fine... but we'll see attacks in this setting in the next lecture

Q: What happens when Eve can inspect the incoming and outgoing traffic of the first and last nodes?

Answer some more…

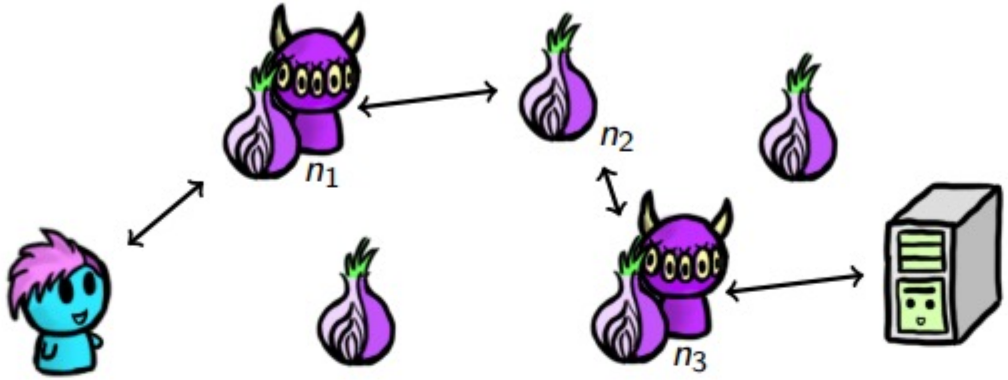
Q: What happens when Eve can inspect the incoming and outgoing traffic of a single node?

A: Alice is probably fine… but we'll see attacks in this setting in the next lecture

Q: What happens when Eve can inspect the incoming and outgoing traffic of the first and last nodes?

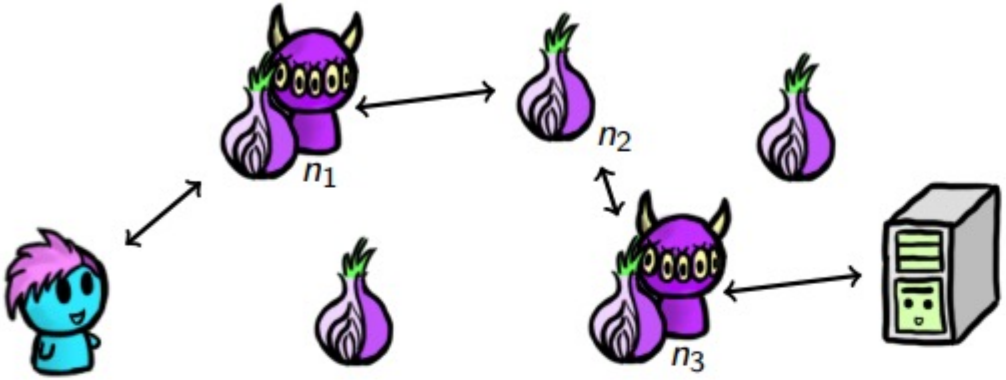
A: Traffic correlation attacks can easily de-anonymize Alice

Last One...For Now



Q: : Why do we usually pick 3 nodes?

Last One...For Now



Q: : Why do we usually pick 3 nodes?

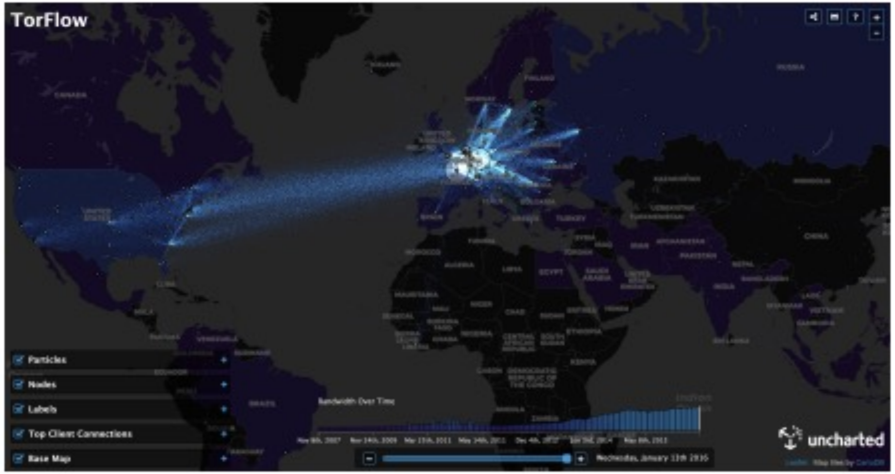
A: It’s a sweet spot between privacy and latency. More nodes usually do not provide more anonymity.

Path Selection

- We want nodes run by different people
  - Avoid multiple nodes in same MyFamily (run by same entity)
  - What about dishonest operators? (sock puppet/Sybil attack)
- Path selection algorithms can help
  - With anonymity: by picking nodes that are in different countries/ISPs
  - With performance: latency is affected by this
- Don’t forget that countries can collaborate as well

Path Selection

- We want to avoid a global passive adversary: choose nodes in different ISPs/countries
- How concentrated is the geographical distribution of Tor relays?

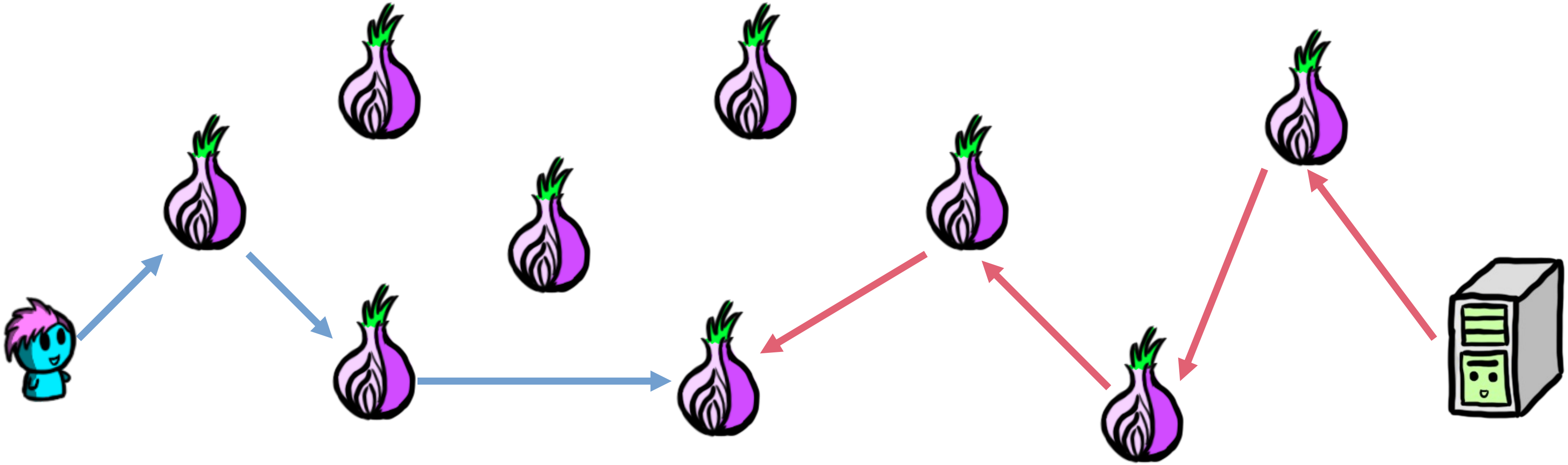


Onion Services

- What if the server wants anonymity too?
  - Onion services! (Also called “hidden services”)

Onion Services

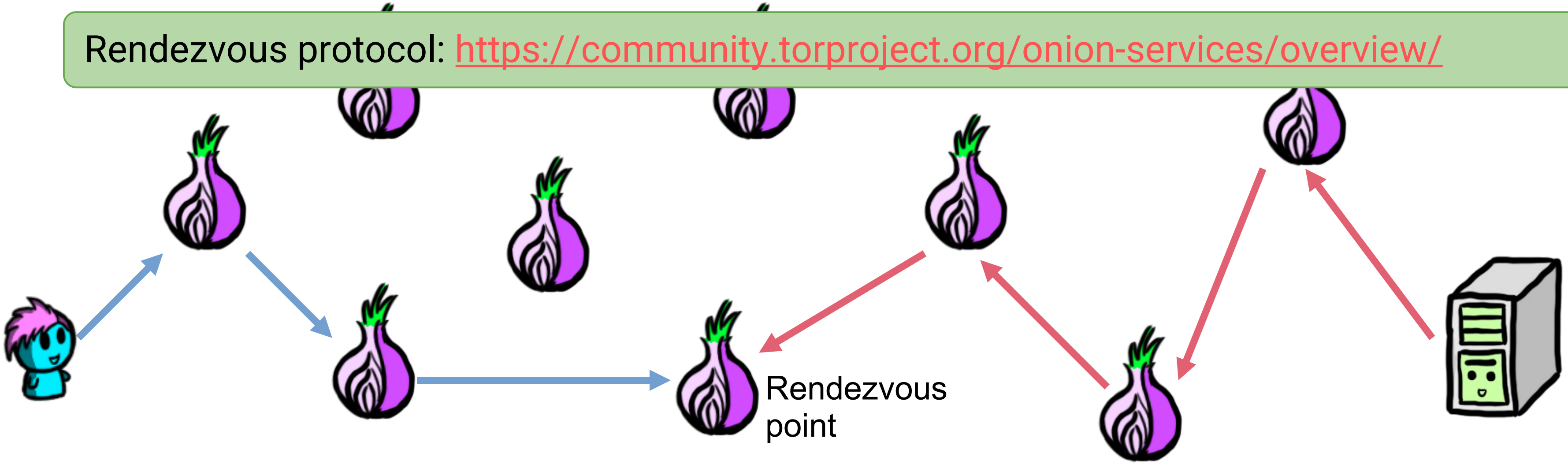
- What if the server wants anonymity too?
  - Onion services! (Also called "hidden services")



Onion Services

- What if the server wants anonymity too?
  - Onion services! (Also called “hidden services”)

Rendezvous protocol: https://community.torproject.org/onion-services/overview/



Onion Addresses

- Long addresses:
  - uwcryspionvholmkfxoqt2xns5mvnct34ytacugxtqpqrnka2oqm6kqd.onion
- Address contains ECC public key for authentication
  - Built-in security
  - No need to rely on HTTPS
- How does this compare to CA system?

Limitations of Tor

- Does not defend against global adversary
- Only protects IP address from destination
  - Users can be identified through browser fingerprinting
    - (Tor Browser tries to defend against this)

A Simple Linkage Attack Based on Length

- You record your sibling’s wedding, encrypt the recording and upload it to an anonymous storage server
- The file is 15,837,448,756 bytes large
- Two weeks later you download it again
- Eve is observing the network traffic to and from the anonymous storage server

Q: Can Eve determine that both access were by the same person?

A Simple Linkage Attack Based on Length

- You record your sibling’s wedding, encrypt the recording and upload it to an anonymous storage server
- The file is 15,837,448,756 bytes large
- Two weeks later you download it again
- Eve is observing the network traffic to and from the anonymous storage server

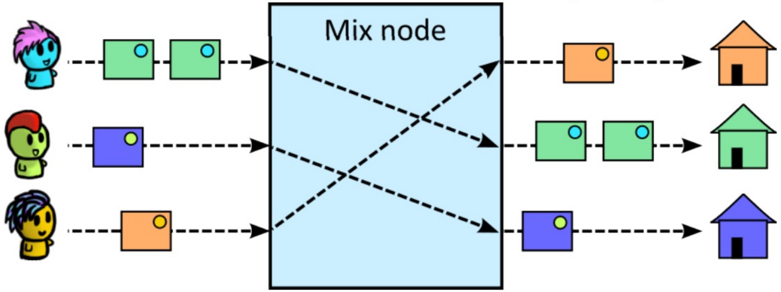
Q: Can Eve determine that both access were by the same person?

A: Well enough

Mixes

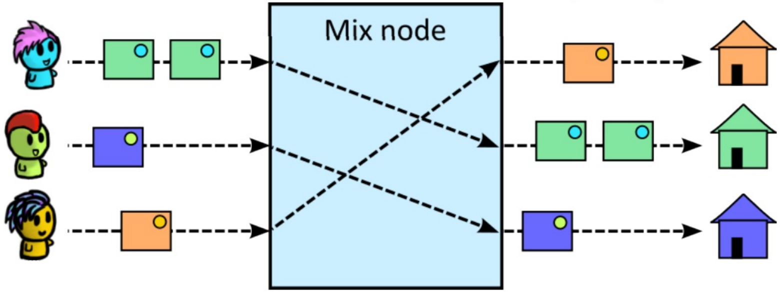
Mixes: Basic Operations

How do we provide anonymity?

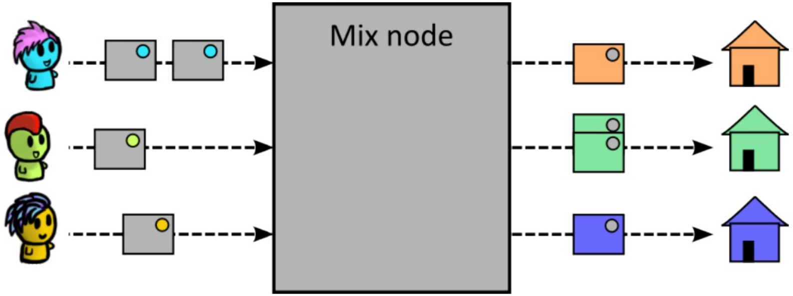


Mixes: Basic Operations

How do we provide anonymity?

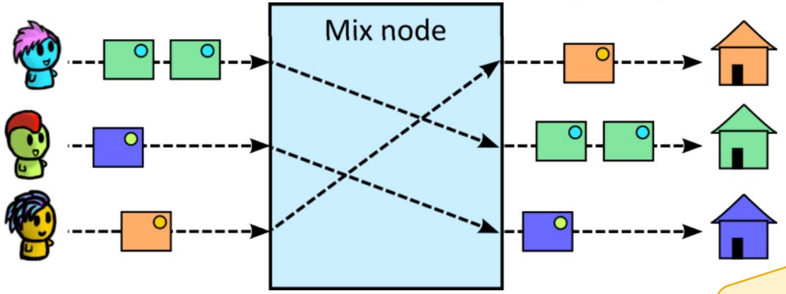


Delay messages!

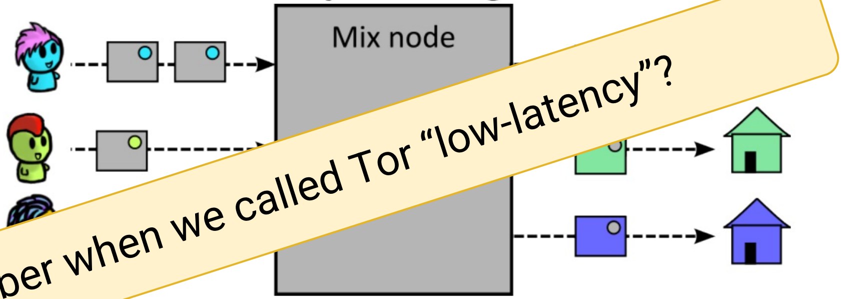


Mixes: Basic Operations

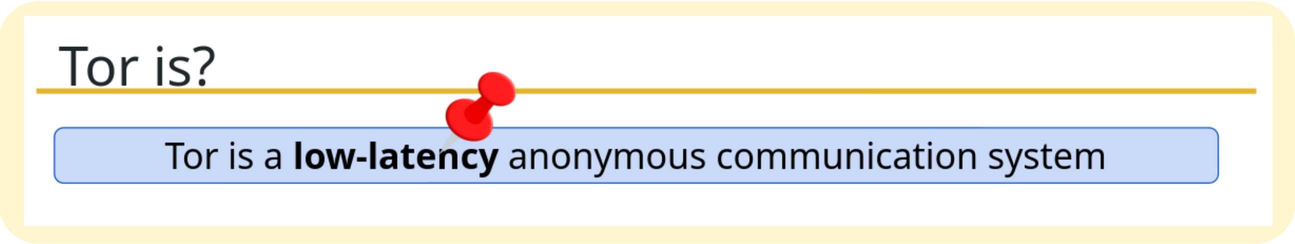
How do we provide anonymity?



Delay messages!



Remember when we called Tor “low-latency”?

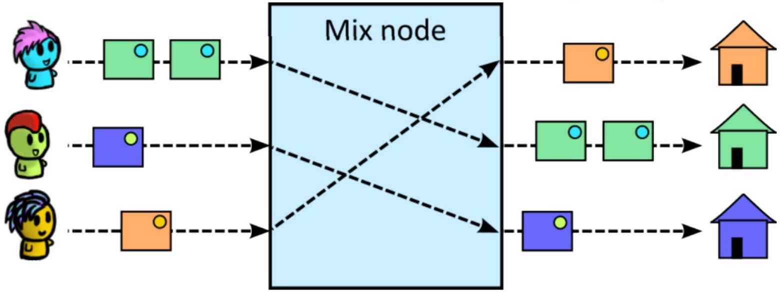


Tor is?

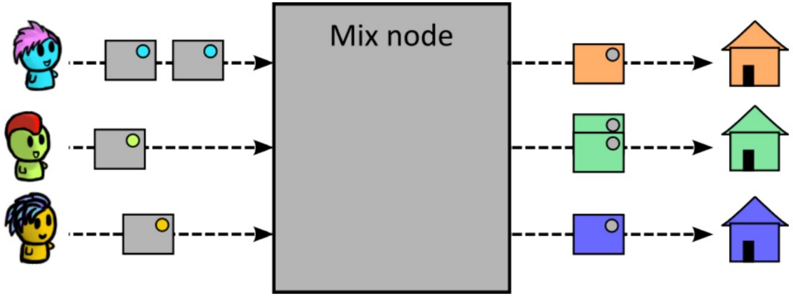
Tor is a **low-latency** anonymous communication system

Mixes: Basic Operations

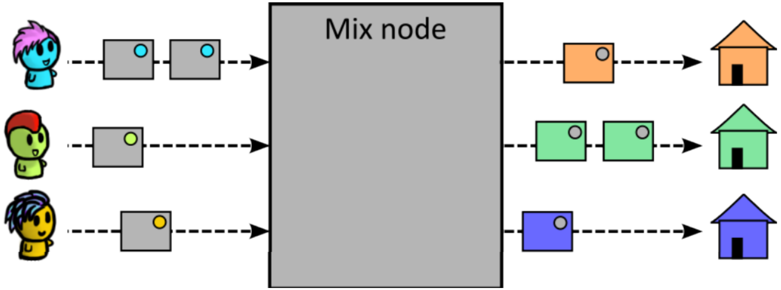
How do we provide anonymity?



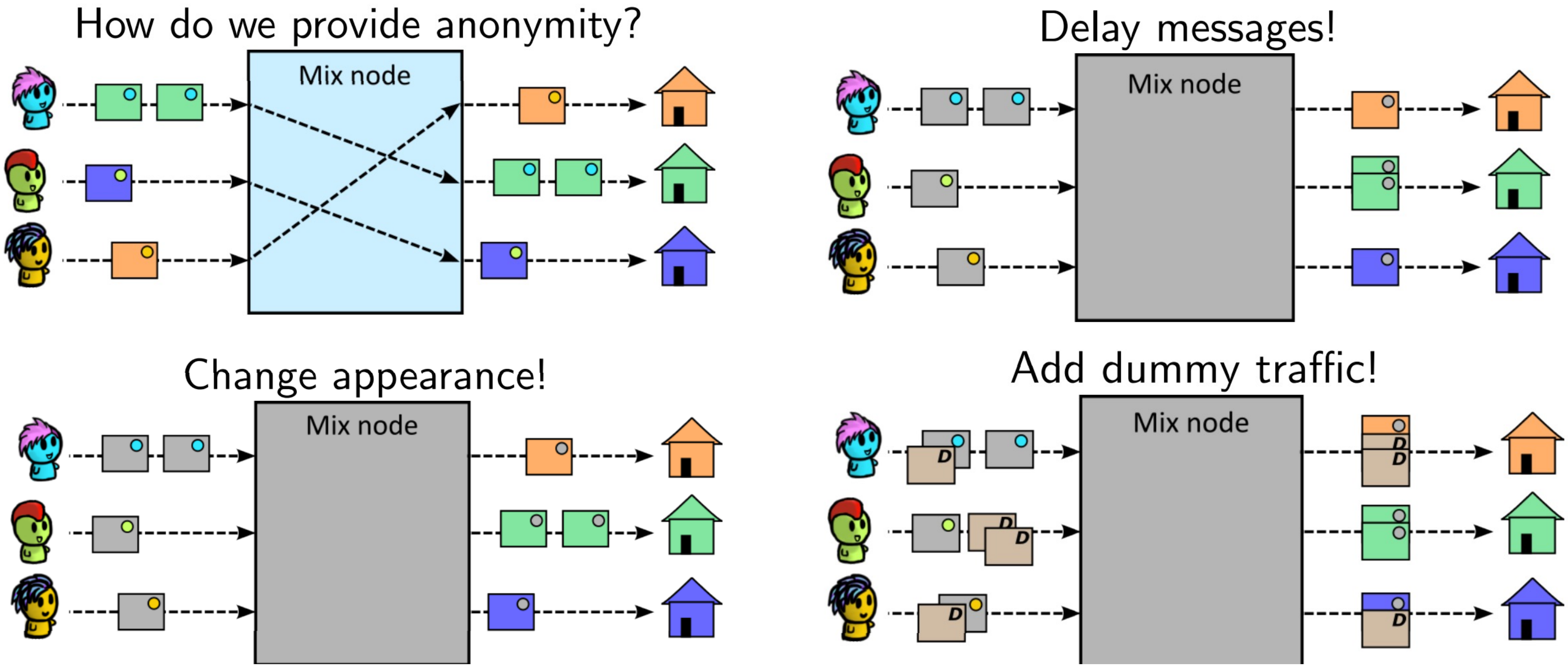
Delay messages!



Change appearance!



Mixes: Basic Operations



How do we provide anonymity?

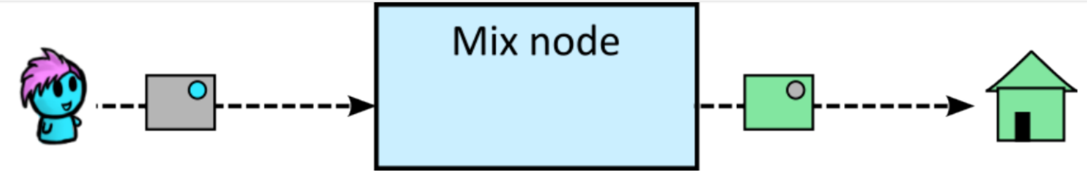
Delay messages!

Change appearance!

Add dummy traffic!

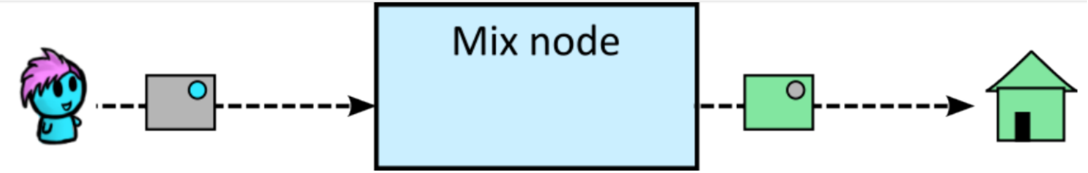
Operation 1: Changing Appearance

Q: How can we achieve this? (clue: we have some crypto tools!)

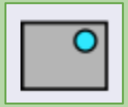


Operation 1: Changing Appearance

Q: How can we achieve this? (clue: we have some crypto tools!)



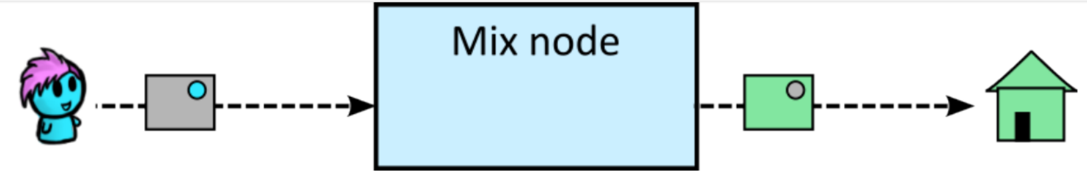
A: We can encrypt the output message with the Mix's key

 $= E_{K\_mix}($ $)$

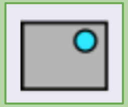
$= E_{K\_Bob}(m)$

Operation 1: Changing Appearance

Q: How can we achieve this? (clue: we have some crypto tools!)



A: We can encrypt the output message with the Mix's key

 $= E_{K\_mix}($ $)$

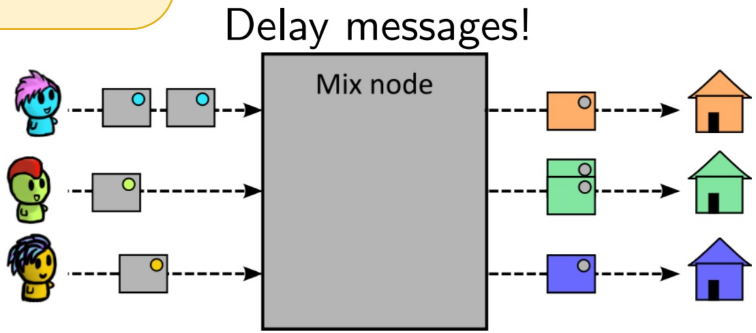
$= E_{K\_Bob}(m)$

This "layered encryption" concept is the same as in onion routing!

Operation 2: Delaying Messages

Q: How do we do this?

- Do we add a random delay to each message?
- Do we add a deterministic delay to each message?
- Do we add a constant delay to each message?

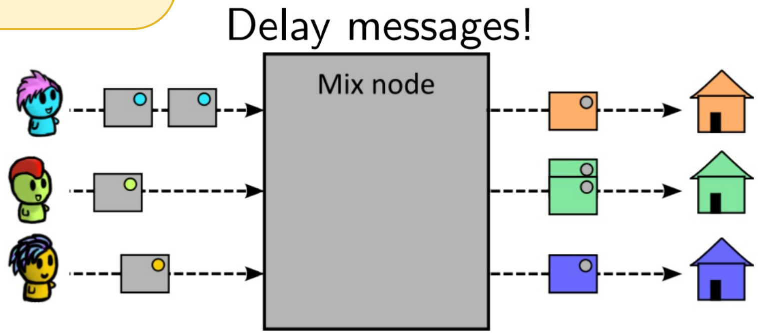


Operation 2: Delaying Messages

Q: How do we do this?

- Do we add a random delay to each message?
- Do we add a deterministic delay to each message?
- Do we add a constant delay to each message?

A: Yes. Yes. No.

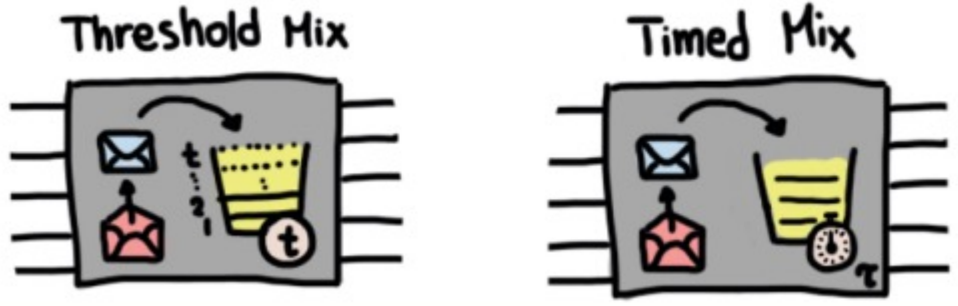


Deterministic delay: it's not constant, it depends on the arrival time and/or other messages. We will see some examples next!

Threshold and Timed Mixes

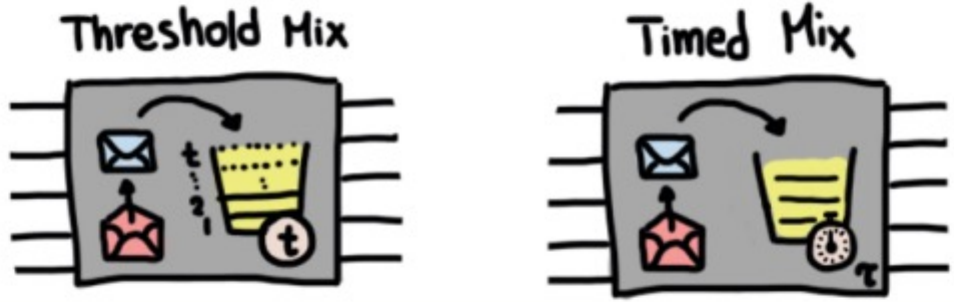
- Some popular mixes types are threshold and timed mixes.
- These mixes gather messages until a **flushing condition** triggers.
- When this condition happens, this marks the end of a **round**
  - Threshold mix: it gathers t messages, then it flushes them.
  - Timed mix: it gathers messages until a timer set to τ seconds expires, then it flushes them.

Threshold and Timed Mixes



Q: Which of the two is better?

Threshold and Timed Mixes

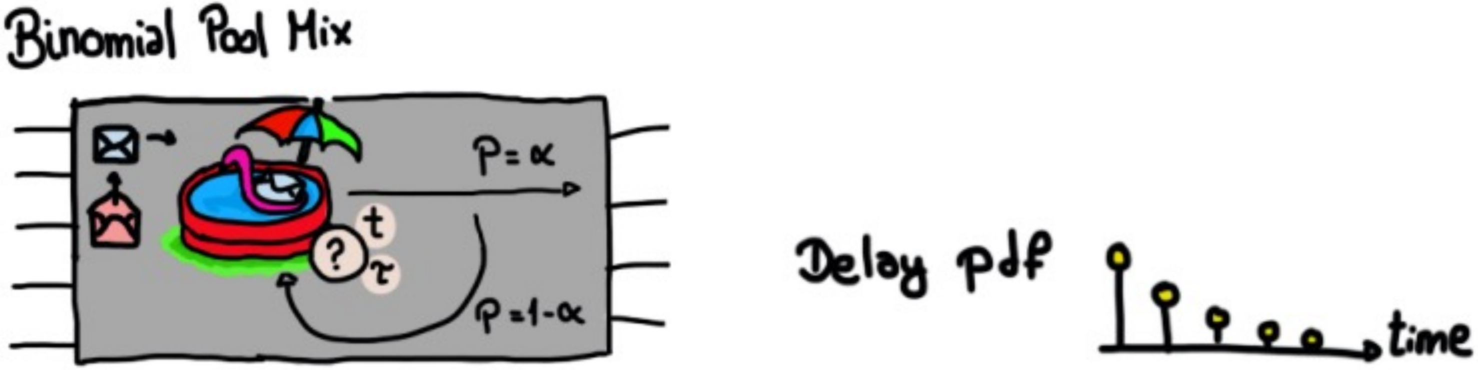


Q: Which of the two is better?

A: It depends... the threshold mix ensures a certain mixing size, the timed mix ensures a maximum message delay.

Pool Mixes

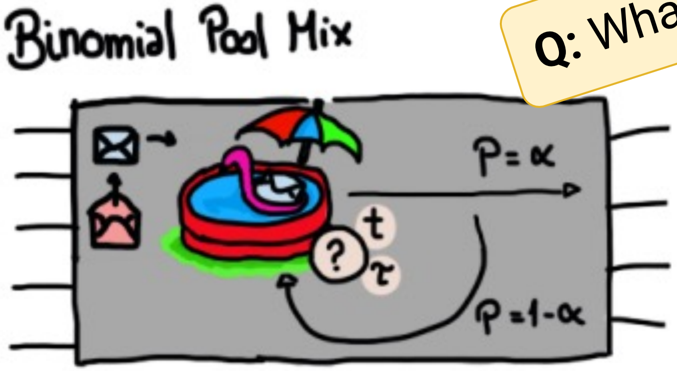
- When a (threshold/timed) mix keeps some messages inside after a round ends, it is called a **pool mix.**
- The **binomial** pool mix keeps each message inside with probability α

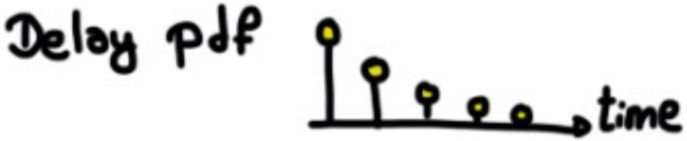


Pool Mixes

- When a (threshold/timed) mix keeps some messages inside after a round ends, it is called a **pool mi**
- The **binomial** pool mix keeps each m side with probability α

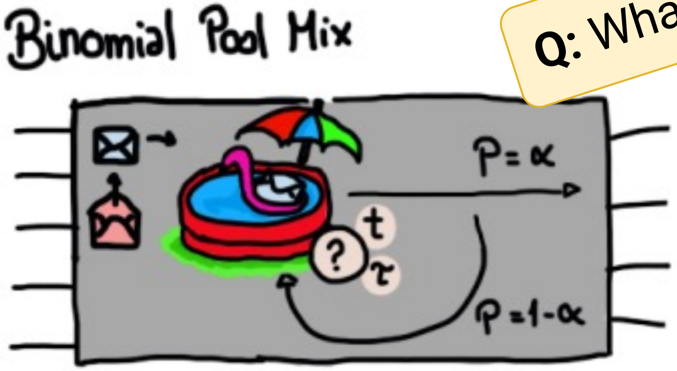
Q: What are the pros and cons of this?





Pool Mixes

- When a (threshold/timed) mix keeps some messages inside after a round ends, it is called a **pool mi**
- The **binomial** pool mix keeps each m side with probability α



Delay pdf
time

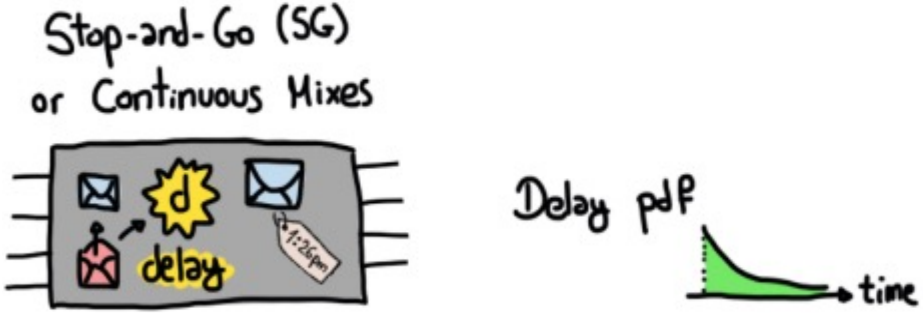
Q: What are the pros and cons of this?

A: Pros, more anonymity

A: Cons, more delay

Continuous-time or Stop-and-Go (SG) Mixes

- Some mixes do not work on “batches” or “rounds”, and instead delay each message independently: these are called continuous-time mixes or Stop-and-Go (SG) mixes.
- Mixes that delay messages following an exponential distribution are very popular (Loopix, Nym).
- The user can choose the delay and include it in the message



Mixnets

Mixnets

Sending messages through a **single mix is not great**

Q: Why?

Mixnets

Sending messages through a **single mix is not great**

Q: Why?

A: There's a single point of failure, and the mix knows the message correspondence.

Mixnets

Sending messages through a **single mix is not great**

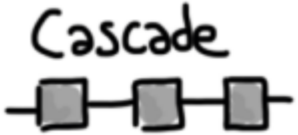
Q: Why?

A: There's a single point of failure, and the mix knows the message correspondence.

- We can chain mixes to create a mixnet.
- Mixnets have different topologies, depending on which nodes a message can travel between.

Mixnet Topologies

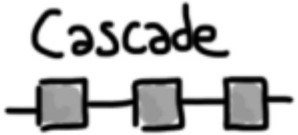
Let’s discuss pros and cons of each topology!



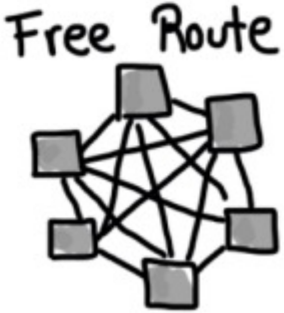
One after another

Mixnet Topologies

Let’s discuss pros and cons of each topology!



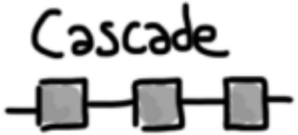
One after another



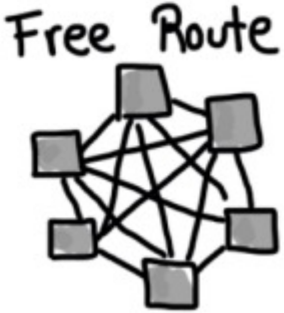
All of them are connected

Mixnet Topologies

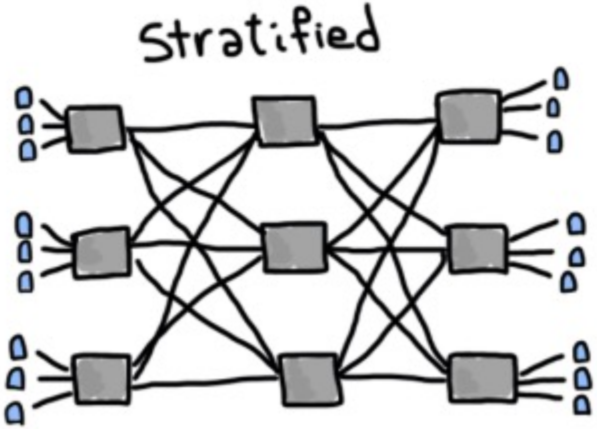
Let’s discuss pros and cons of each topology!



One after another



All of them are connected

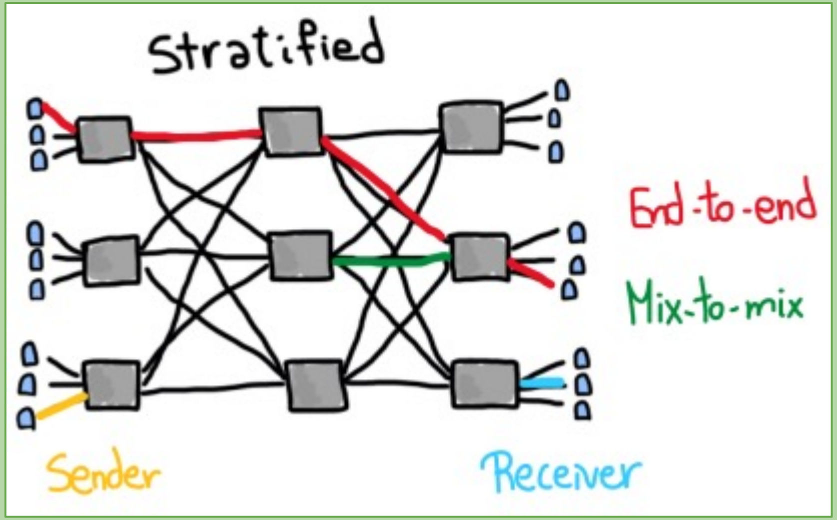


Each layer is fully connected to the next layer

Operation 3: Dummy Messages

Q: Where do we add dummy traffic?

A: Anywhere, everywhere!



Checkpoint 1

Q: What are the three basic operations of a mix node to provide anonymity? Why is each operation important?

Checkpoint 1

Q: What are the three basic operations of a mix node to provide anonymity? Why is each operation important?

A: Change appearance, delay messages, add dummy traffic

Checkpoint 1

Q: What are the three basic operations of a mix node to provide anonymity? Why is each operation important?

A: Change appearance, delay messages, add dummy traffic

Q: Threshold mixes: pros and cons of increasing the threshold t?

Checkpoint 1

Q: What are the three basic operations of a mix node to provide anonymity? Why is each operation important?

A: Change appearance, delay messages, add dummy traffic

Q: Threshold mixes: pros and cons of increasing the threshold t?

A: Increasing t improves anonymity but increases delay

Checkpoint 2

Q: Timed mixes: pros and cons of increasing the time τ?

Checkpoint 2

Q: Timed mixes: pros and cons of increasing the time τ?

A: Increasing τ improves anonymity but increases delay

Checkpoint 2

Q: Timed mixes: pros and cons of increasing the time τ?

A: Increasing τ improves anonymity but increases delay

Q: Binomial pool mix: pros and cons of increasing the probability of forwarding a message α?

Checkpoint 2

Q: Timed mixes: pros and cons of increasing the time τ?

A: Increasing τ improves anonymity but increases delay

Q: Binomial pool mix: pros and cons of increasing the probability of forwarding a message α?

A: Increasing α decreases anonymity and delay

Checkpoint 3

Q: Dummy traffic: pros and cons of increasing the amount of dummy messages?

Checkpoint 3

Q: Dummy traffic: pros and cons of increasing the amount of dummy messages?

A: More dummies require more bandwidth, but increase anonymity

Checkpoint 3

Q: Dummy traffic: pros and cons of increasing the amount of dummy messages?

A: More dummies require more bandwidth, but increase anonymity

Q: What happens if the number of senders increases?

Checkpoint 3

Q: Dummy traffic: pros and cons of increasing the amount of dummy messages?

A: More dummies require more bandwidth, but increase anonymity

Q: What happens if the number of senders increases?

A: Depends on the actual mix/setting, but usually **anonymity loves company**. More people using the system usually improves its anonymity level.

Anonymity Trade-Offs Summary

Anonymity has a cost. We can increase anonymity by:

- Adding more message delay
  - It has to be added “cleverly” (e.g., a constant delay does not work)
- Adding more dummy traffic
  - It has to be added “cleverly” (e.g., simulating real sending behavior)
- When the number of users increases
  - Effectiveness depends on the type of mix, the mix topology, etc.

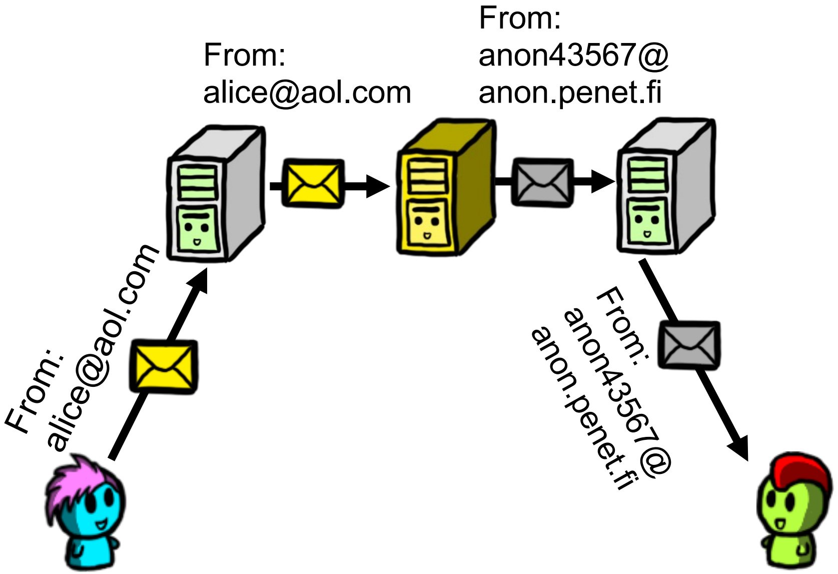
Remailers, A Brief History

See Prof. Goldberg’s papers on PETs for the Internet:
https://cypherpunks.ca/~iang/pubs/privacy-compcon97.pdf
https://cypherpunks.ca/~iang/pubs/pet2.pdf
https://cypherpunks.ca/~iang/pubs/pet3.pdf

Remailers: Very Simple Type 0, (1993–1996)

The best known being anon.penet.fi.

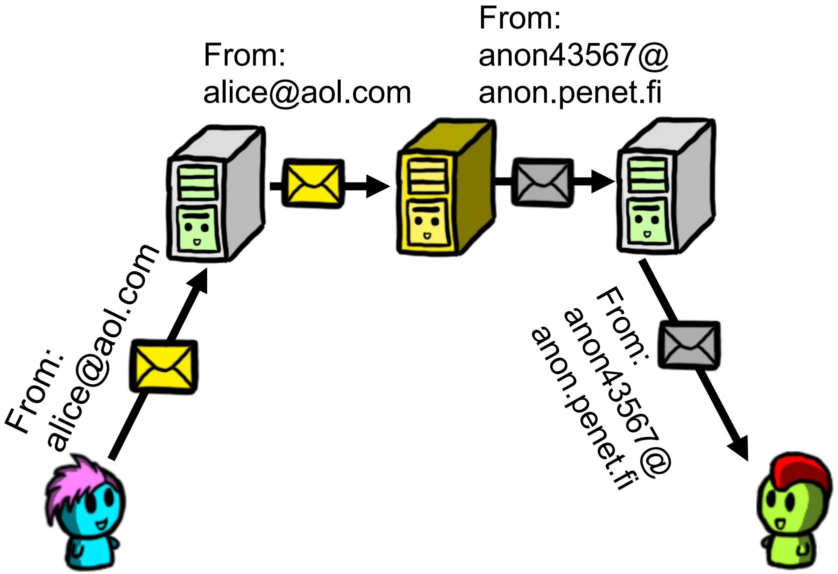
- Send email to anon.penet.fi
- It is forwarded to your intended recipient
- “From” address is changed to anon43567@anon.penet.fi
  - (Original address stored in a table for replies)



Remailers: Very Simple Type 0, (1993–1996)

The best known being anon.penet.fi.

- Send email to anon.penet.fi
- It is forwarded to your intended recipient
- “From” address is changed to anon43567@anon.penet.fi
  - ○ (Original address stored in a table for replies)
- Replies to the anon address get mapped back to your real address and delivered to you
- ≈ 10,000 emails per day (≈ 700,000 users)



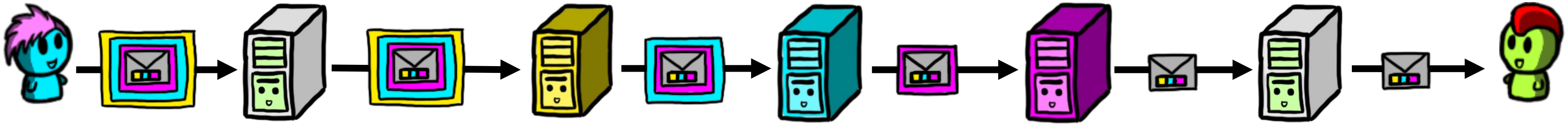
Anon.penet.f, works as long as…

- No one’s watching the Internet connections to or from anon.penet.fi
- The operator of anon.penet.fi, the machine (hardware), and the software all remain trustworthy and uncompromised
- The mapping of anon addresses to real addresses is kept secret

Unfortunately, a lawsuit forced Julf (the operator) to turn over parts of the list, and he shut down the whole thing, since he could no longer legally protect it

Cypherpunk (Type 1) Remailers

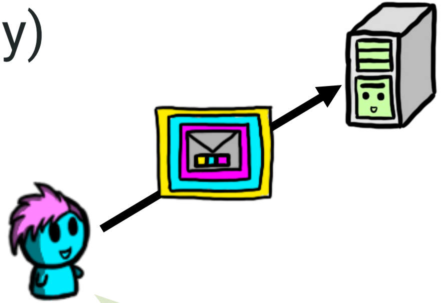
- Removed the central point of trust
- Messages are now sent through a “chain” of several remailers, with dozens to choose from
- Each step in the chain is encrypted to avoid observers following the messages through the chain
- Remailers also delay and reorder messages
- Support for pseudonymity is dropped: no replies!



Nymservers / Pseudonymous remailers

How to do replies? (i.e., recovering pseudonymity)

- Alice registers an address with nym.alias.net
- Alice uploads a “reply block”
  - Contains multiple type I remailer addresses
  - Layered encryption
- Alice tells Bob to reply to her alias



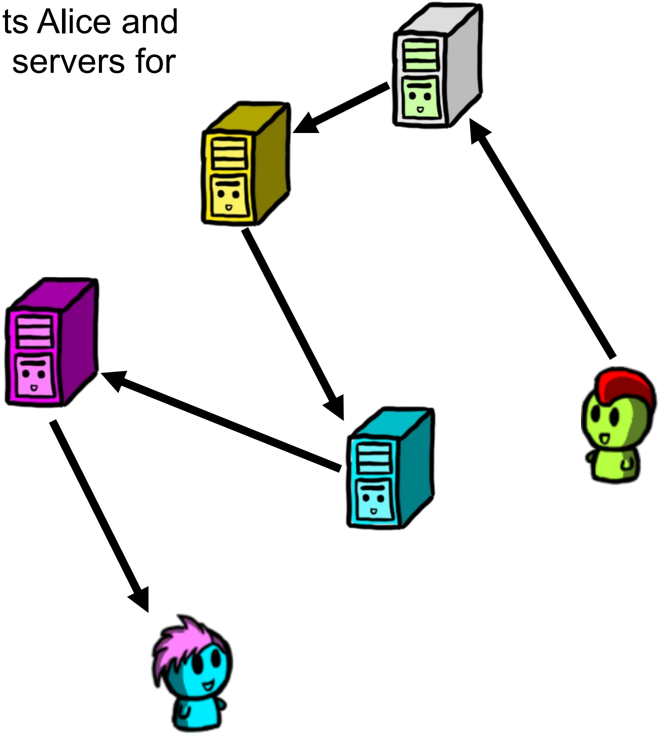
Send messages to janedoe@nym.alias.net through this series of remailers.

Nymservers / Pseudonymous remailers

How to do replies? (i.e., recovering pseudonymity)

- Alice registers an address with nym.alias.net
- Alice uploads a “reply block”
  - Contains multiple type I remailer addresses
  - Layered encryption
- Alice tells Bob to reply to her alias
- When Bob replies, the nymserver sends the message through type I remailers

(Visual omits Alice and Bob’s main servers for simplicity)



Type II remailers

Mixmaster (type II) remailers appeared in the late 1990s

- Constant-length messages to avoid an observer watching “that big file” travel through the network
- Protections against replay attacks
- Improved message reordering

Requires a special email client to construct the message fragments

Type III remailers

Mixminion (type III) remailer appears in the 2000s

- Native (and much improved) support for pseudonymity
  - No longer reliant on type I reply blocks
  - Instead, relies on mix networks
- Improved protection against replay and key compromise attacks

But it’s not very well deployed or mature, i.e., “you shouldn’t trust Mixminion with your anonymity yet”