# CS489/698 Privacy, Cryptography, Network and Data Security

## Network Anonymity

Fall 2024, Tuesday/Thursday 02:30pm-03:50pm

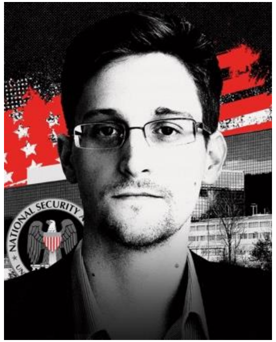
# Recall a Little Bit About Privacy

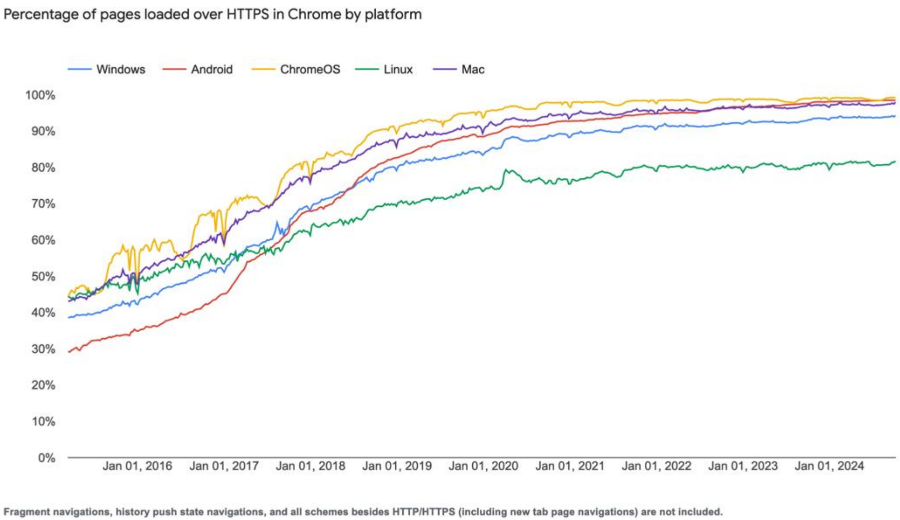
Two “types” of information that could be privacy-sensitive:

- Data: refers to contents of messages, contents of a database...
  - Data streams headers (e.g., source and destination IP)
  - Body (may or may not be encrypted)
- Meta-data: any other information that is not data
  - When communication occurs
  - Who communicates
  - How often do they communicate
  - …

# Encryption on the Internet

- When the Internet was designed, the main concern was reliability
  - The data should get transmitted successfully
  - SSL used only for the financial transactions
- Big change in the mid-2010s
  - Encryption at scale
  - HTTPS become ubiquitous



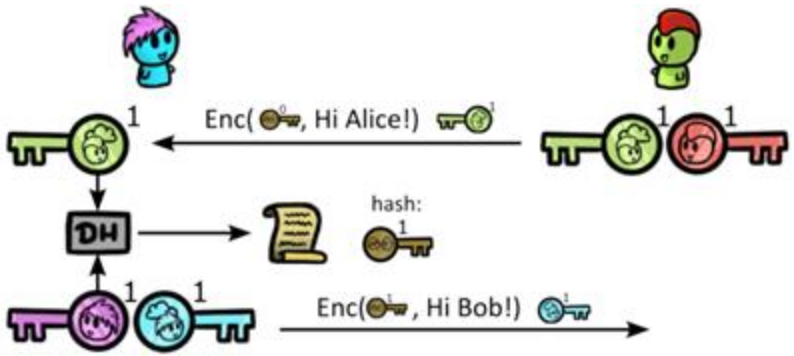


# Encryption on the Internet

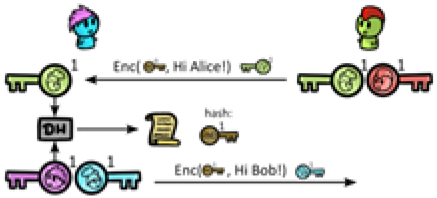
- You certainly use encryption on the Internet
  - SSL used only for the financial transactions
- You may also be familiar with “end-to-end” encrypted messaging apps
  - e.g., iMessage, WhatsApp, Signal
  - Even the servers facilitating the communication cannot read the contents of the messages

# Encryption on the Internet

- You certainly use encryption on the Internet
  - SSL used only for the financial transactions
- You may also be familiar with "end-to-end" encrypted messaging apps
  - e.g., iMessage, WhatsApp, Signal
  - Even the servers facilitating the communication cannot read the contents of the messages

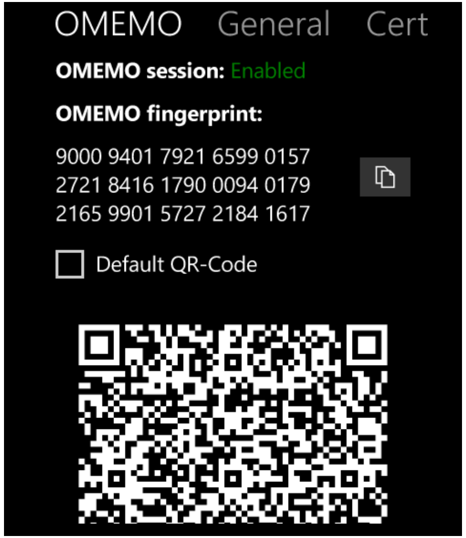


# How do we Trust these Keys?

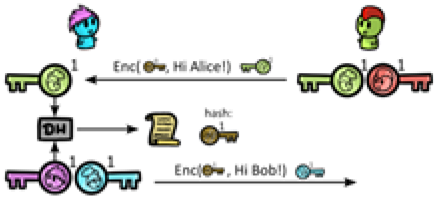



- You Manual Verification – OMEMO: QR Codes

XMPP

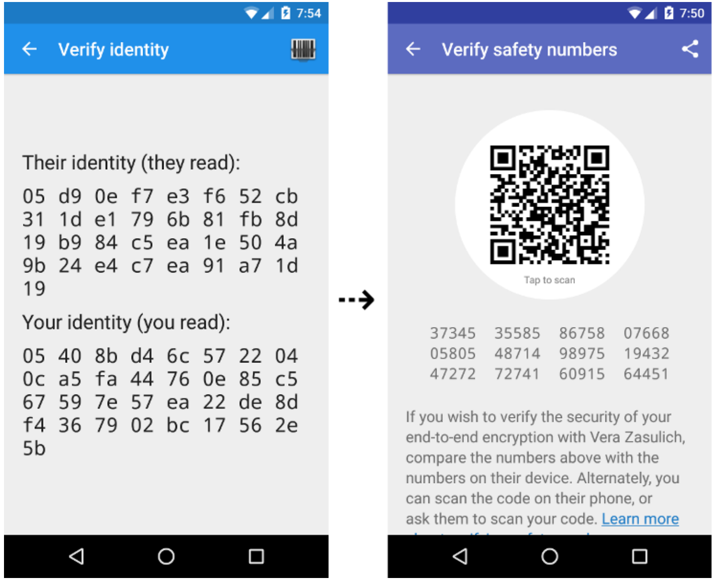


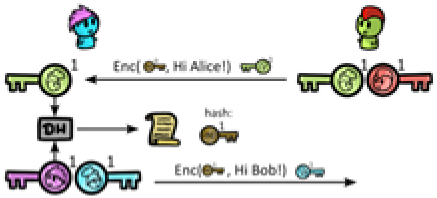
# How do we Trust these Keys?




- You Manual Verification – Signal: Safety number

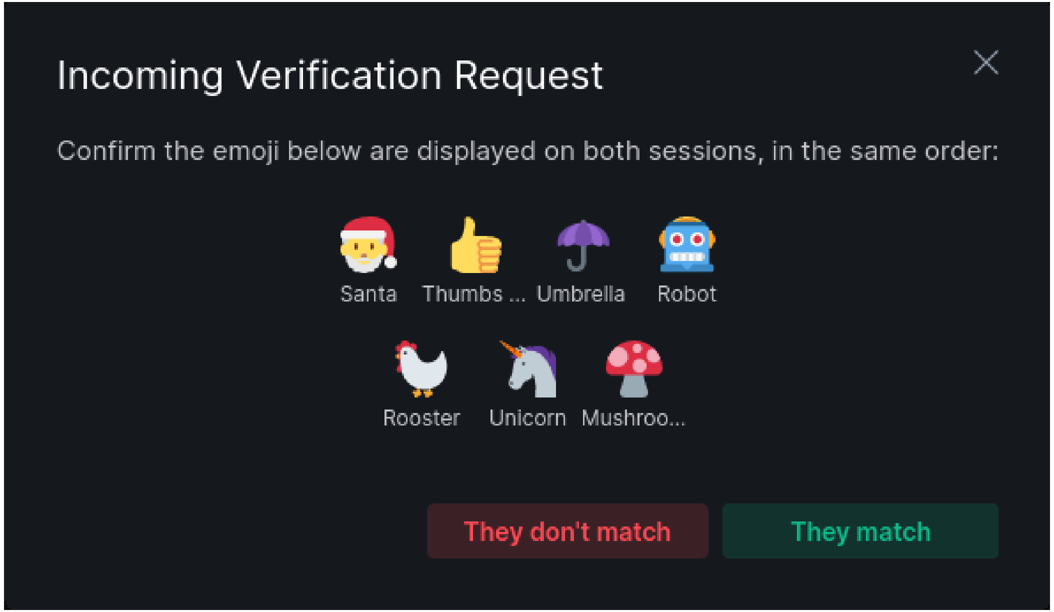
Signal






# How do we Trust these Keys?

- You Manual Verification – Matrix: Emoji



# Metadata

- Protecting data with encryption is necessary for security and privacy, but not sufficient
- Just knowing who your friends are gives a lot of information about you
  - This can be a big problem in some countries

"We kill people based on metadata"
— Gen. Michael Hayden, former director of NSA and CIA

# 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

# Protecting metadata from whom

- That is, what is the threat model? Who might be adversarial?

- ASes the network traffic passes through
  - Note that this could include governments
  - In the extreme, a global adversary

- The servers facilitating the communications
  - Even if they're compelled by law or force

- Sometimes even the person you're talking to
  - More limited in what we can protect

# 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

# PETs to Control Data Leakage and Meta-data

- Anonymity in communication (privacy as masks): how to hide who communicates with whom.
  - Tor
  - Remailers (Mixes)

**Today:** the meta-data of communication

E.g., Who communicates with whom, how often, from where…

# Tor

# Onion routing

- This new technology was onion routing (Syverson, Goldschlag, and Reed, 1997), and its successor, Tor (Dingledine, Mathewson, Syverson, 2004)

# Why Tor?

Tor is a successful privacy enhancing technology that works at the transport layer with **≈2 million** daily users

# Why Tor?

Tor is a successful privacy enhancing technology that works at the transport layer with **≈2 million** daily users

**Q:** Why do we need Tor when we have TLS?

# Why Tor?

Tor is a successful privacy enhancing technology that works at the transport layer with **≈2 million** daily users

**Q:** Why do we need Tor when we have TLS?

**A:** TLS **protects data**, but…We also **want to protect metadata** about the communication: e.g., IP addresses, browser fingerprints.

# Tor is?

Tor is a low-latency anonymous communication system

# Tor is?

Tor is a low-latency anonymous communication system

Tor has about **7 000 nodes** scattered around the Internet; these are also called Onion Routers

# Tor is?

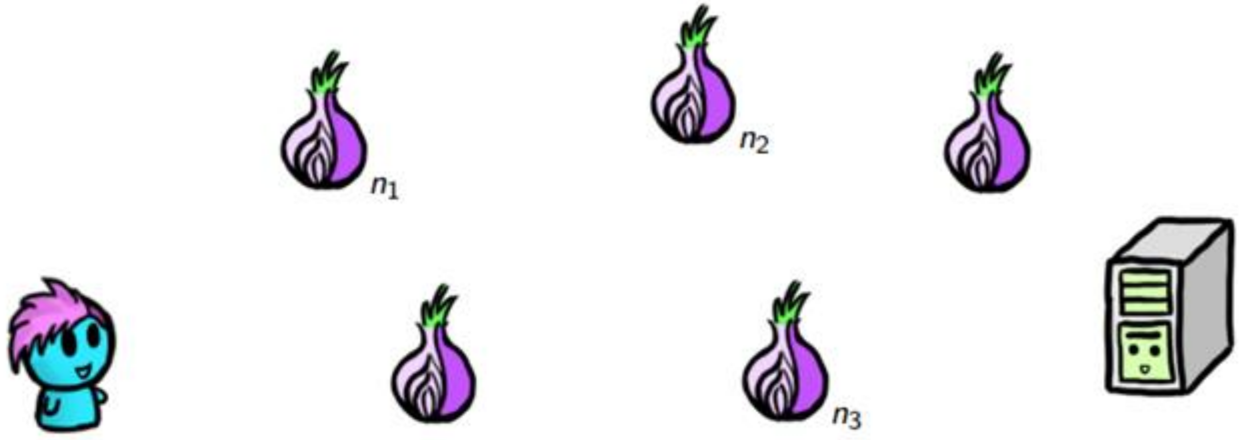
Tor is a low-latency anonymous communication system

Tor has about **7 000 nodes** 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: Building a Circuit (I)

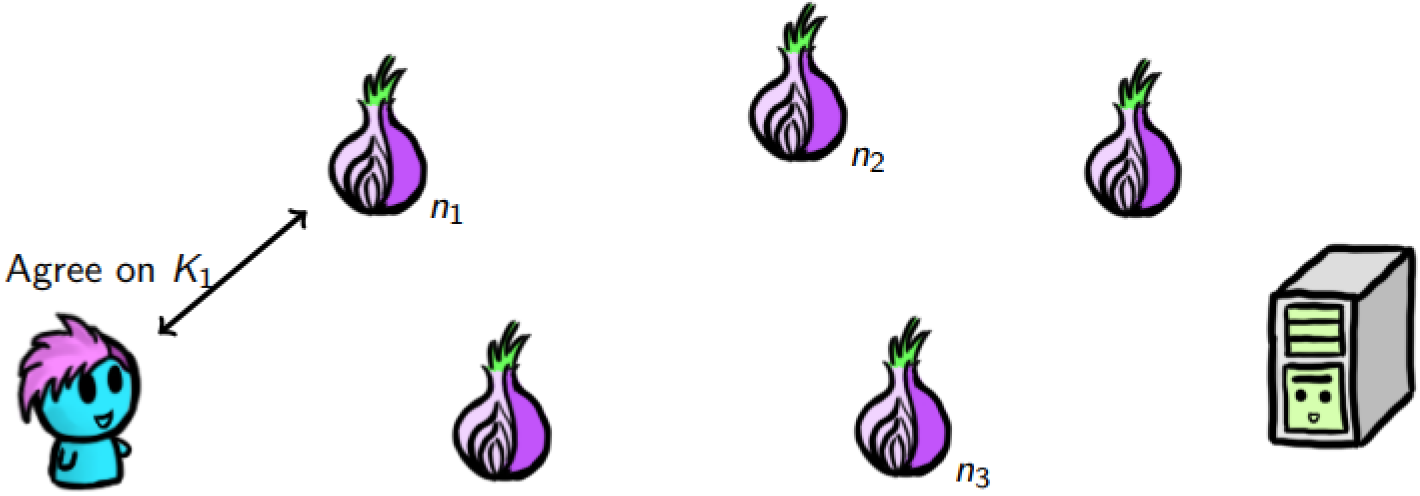
**Goal:** Alice wants to connect to a server without revealing her IP address




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

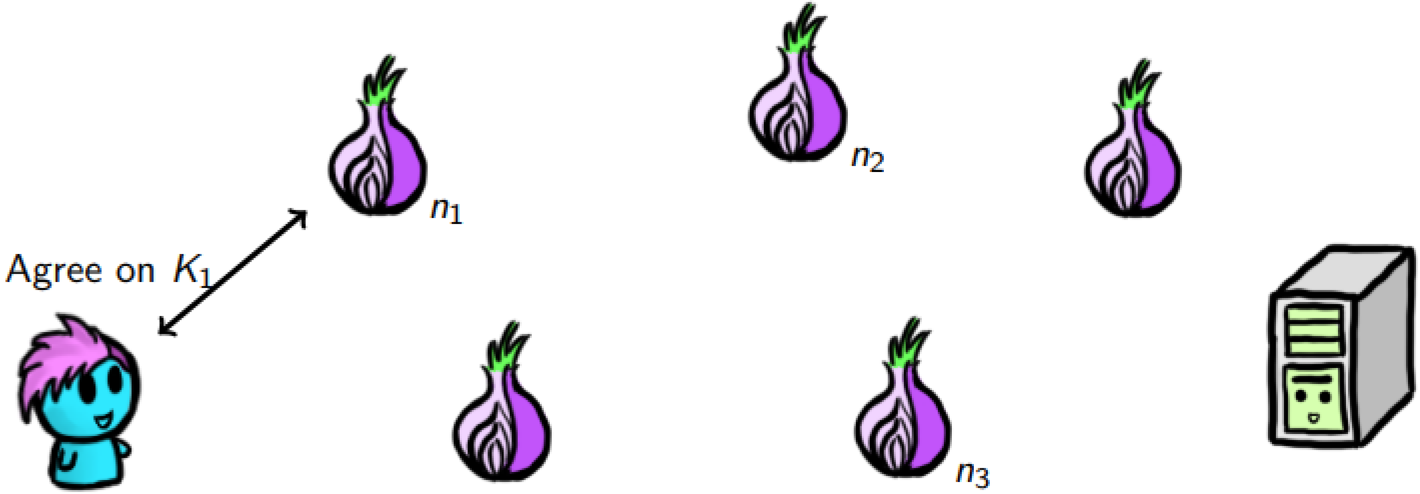
# Tor: Building a Circuit (II)

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



# Tor: Building a Circuit (II)

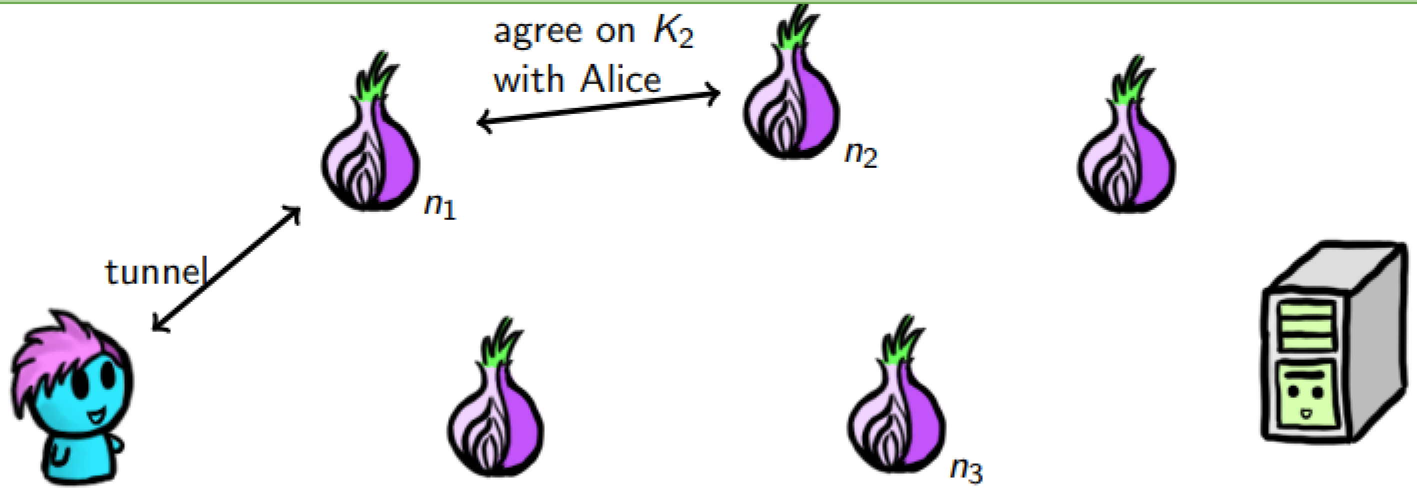
Alice picks Tor nodes ($n_1$) and uses PKC to establish an encrypted communication channel to it (much like TLS)




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

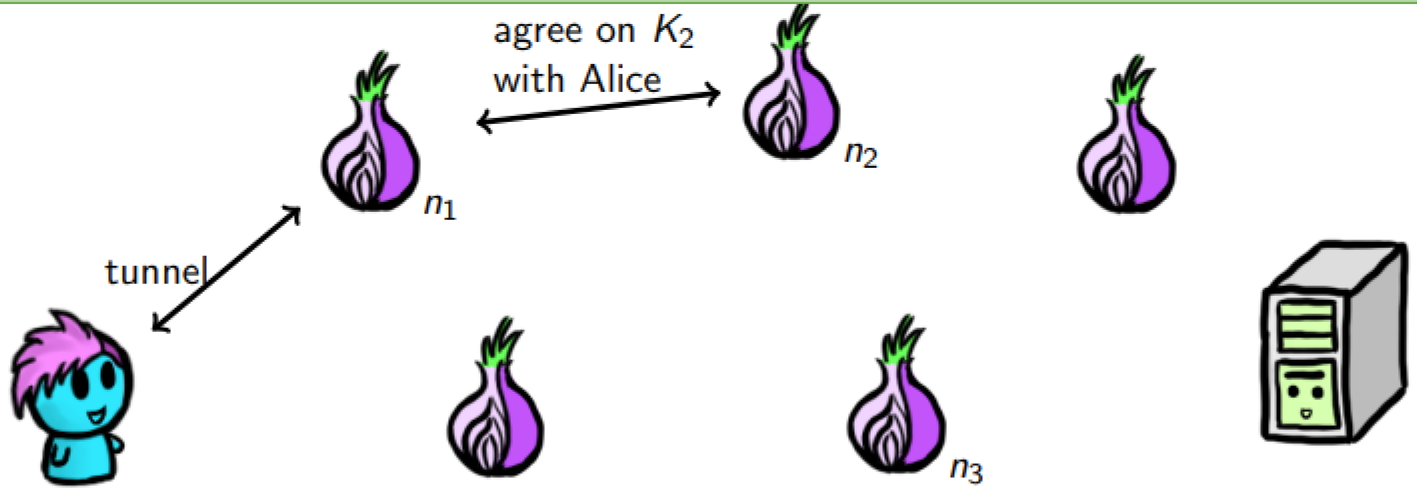
# Tor: Building a Circuit (III)

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)

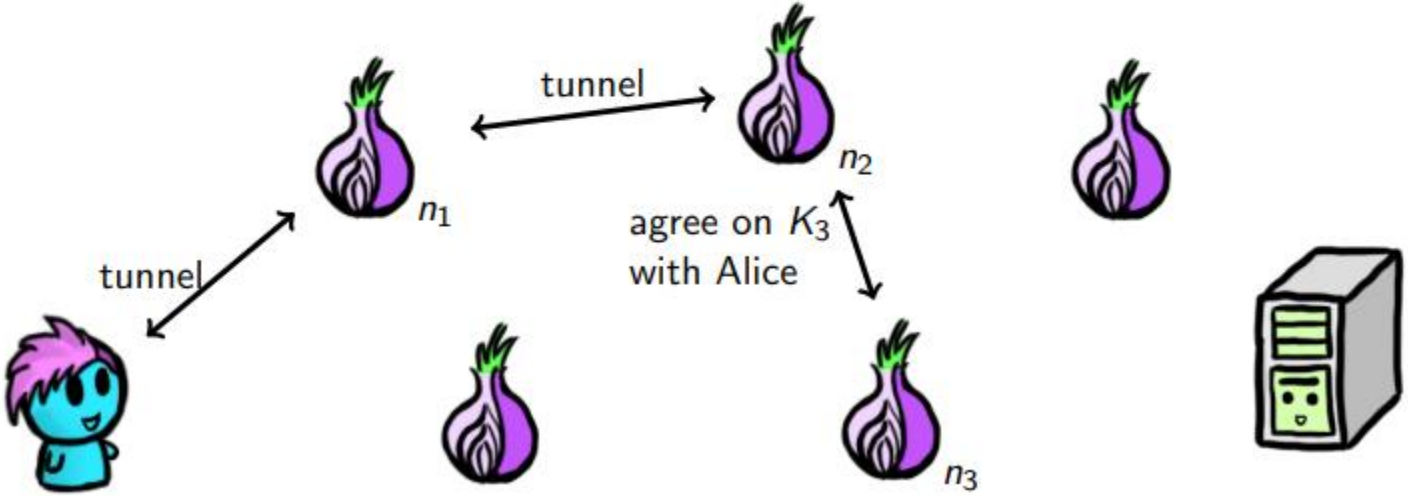
Alice tells $n_1$ to contact a second node ($n_2$), and establishes a new encrypted comm.channel to $n_2$, tunneled within the previous one to $n_1$




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

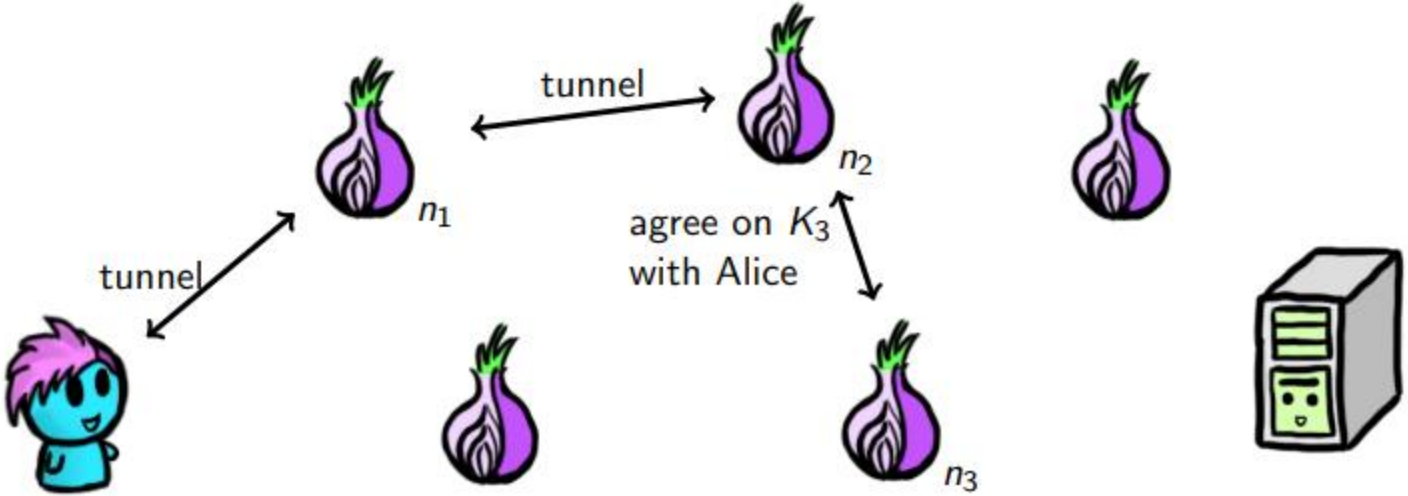
# Tor: Building a Circuit (IV)

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)

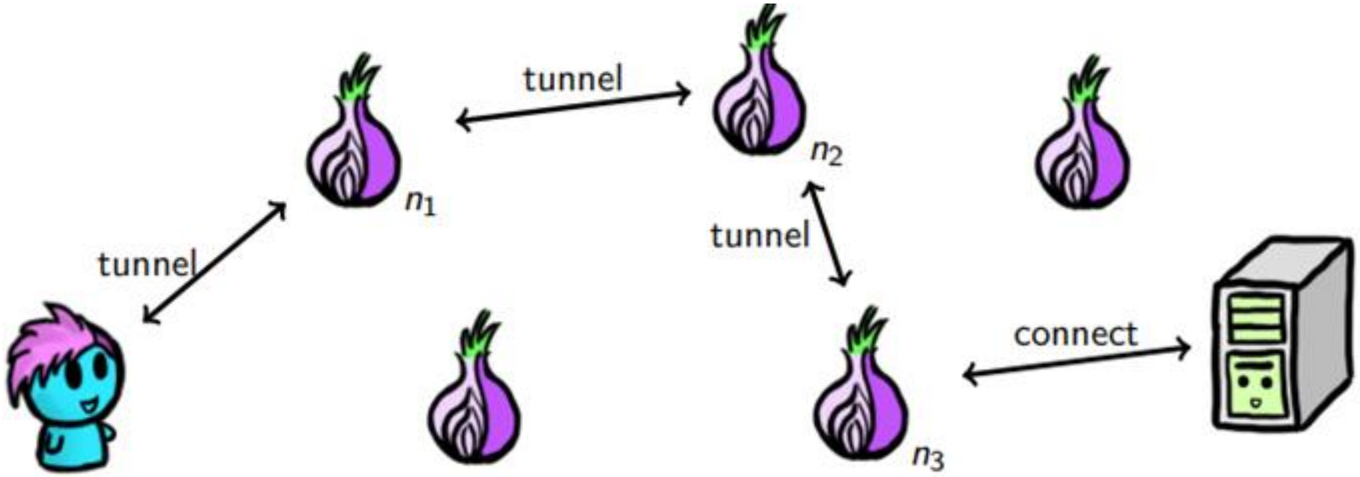
Alice tells $n_2$ to contact a third node ($n_3$), establishes a new encrypted communication channel to $n_3$, tunneled within the previous one to $n_2$




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

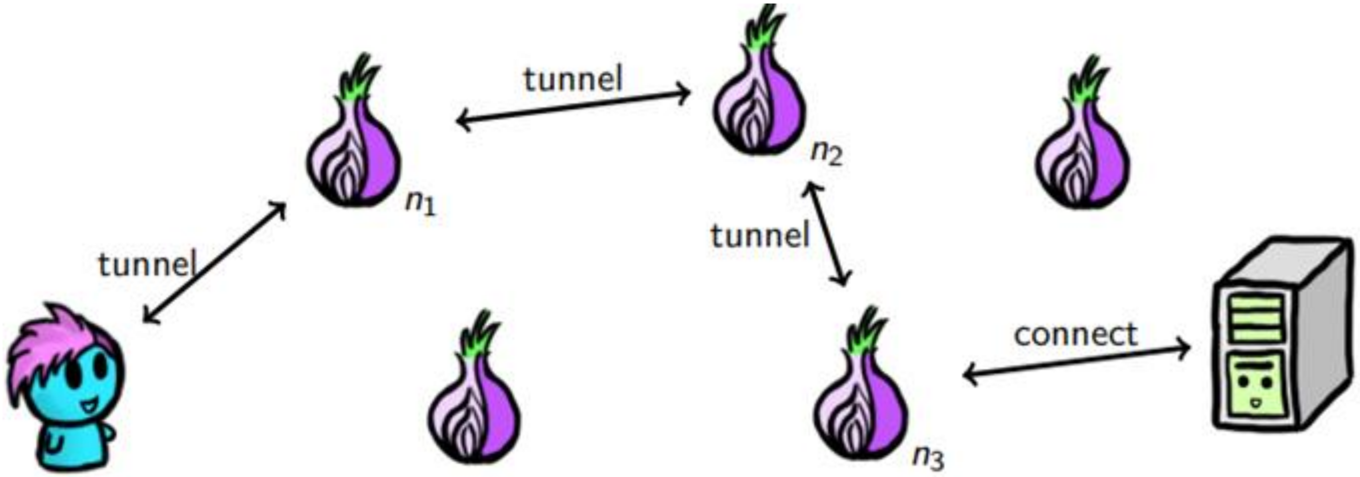
# Tor: Building a Circuit (V)

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



# Tor: Building a Circuit (V)

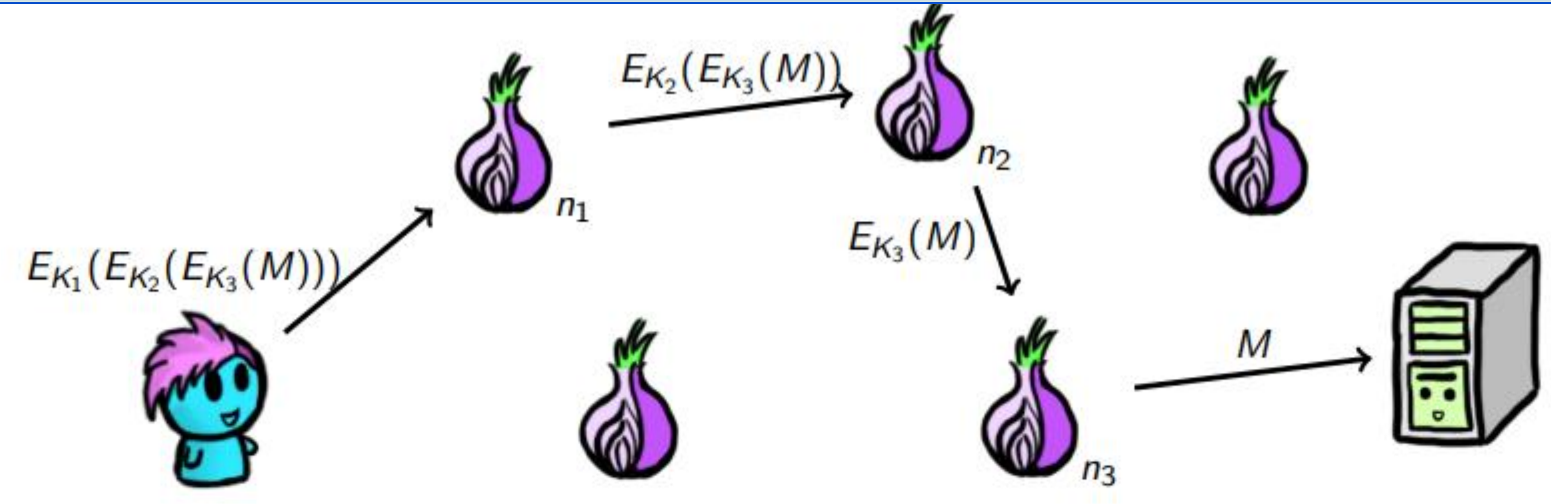
... And so on, for as many steps as she likes (usually 3) ...




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

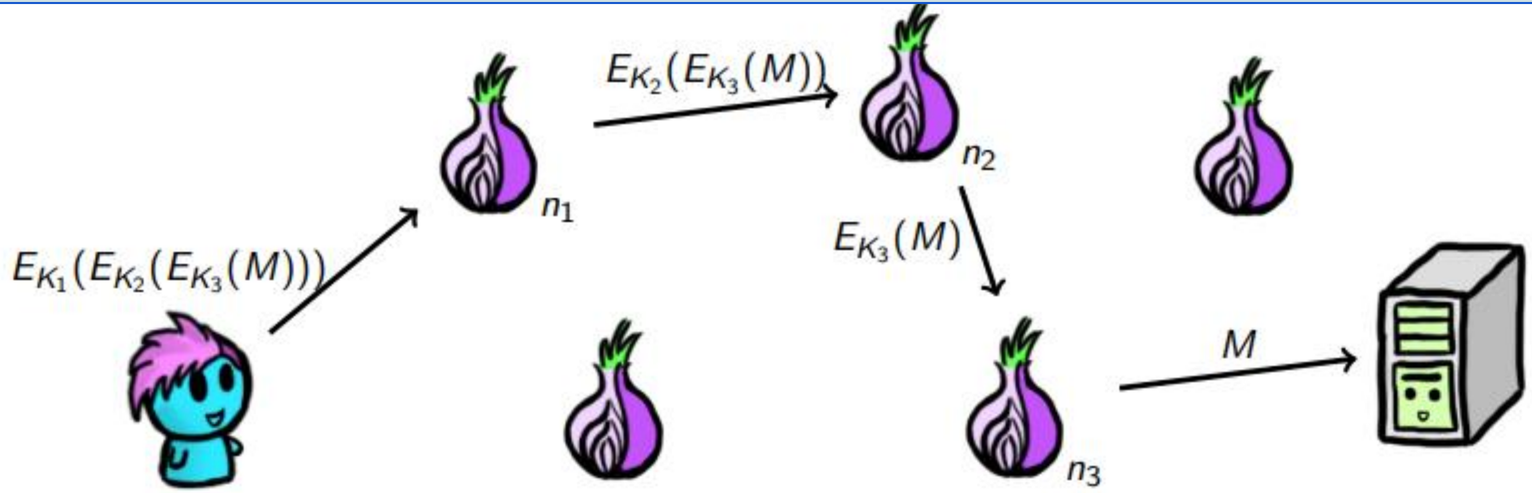
# Sending Messages with Tor

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



# Sending Messages with Tor

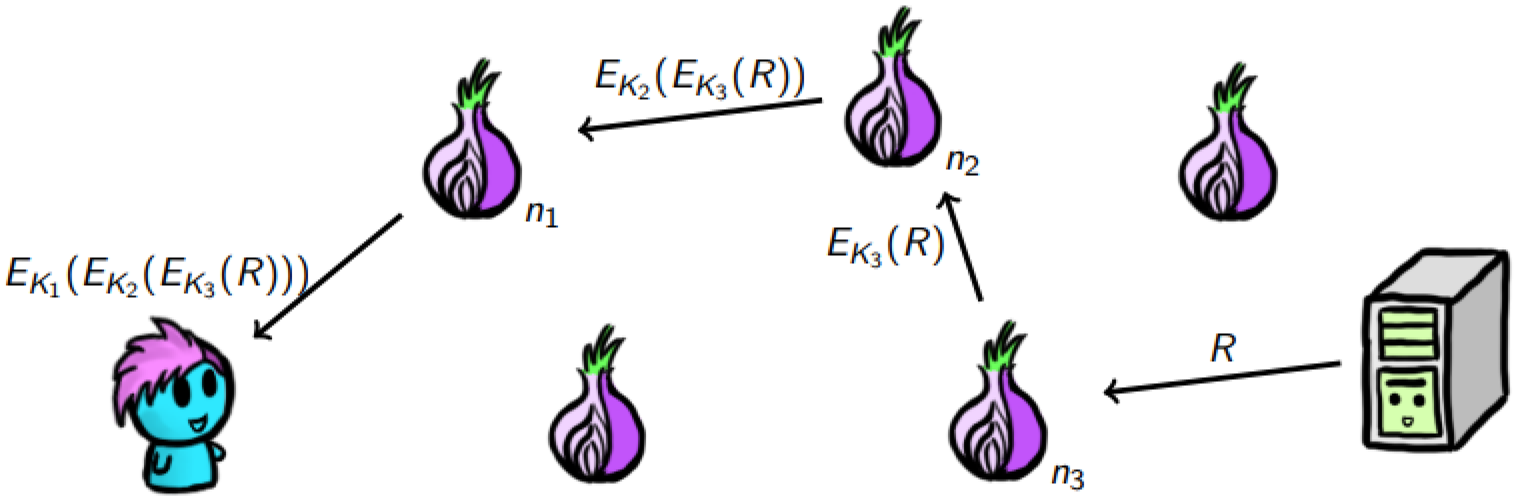
Alice encrypts her message “like an onion”; each node peels a layer off and forwards it to the next step




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

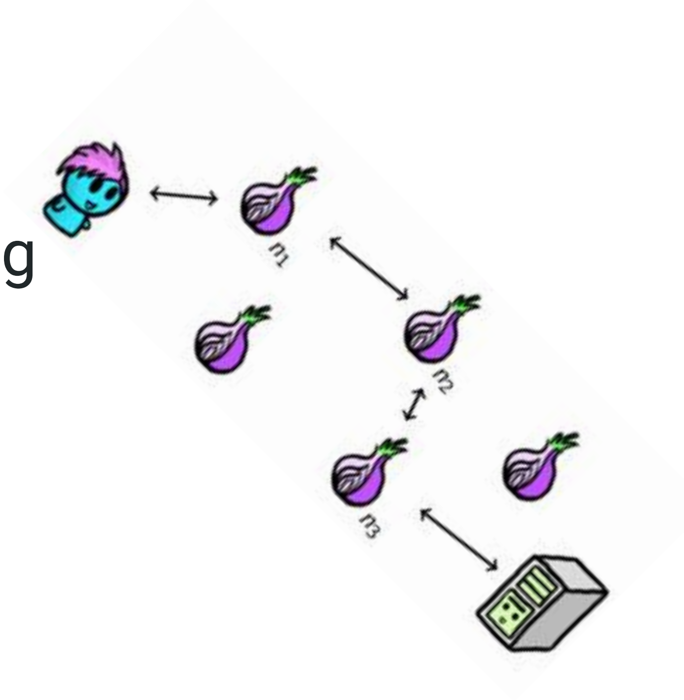
# Replies in Tor

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



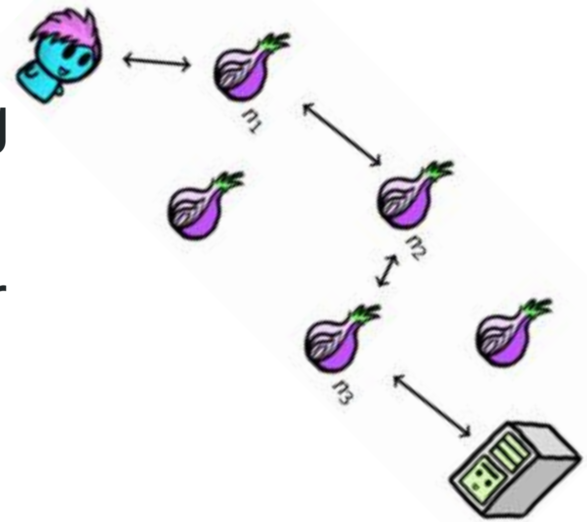
# Who knows what?

- Node $n_1$ knows that Alice is using Tor, and that her next node is $n_2$, but does not know which website Alice is visiting



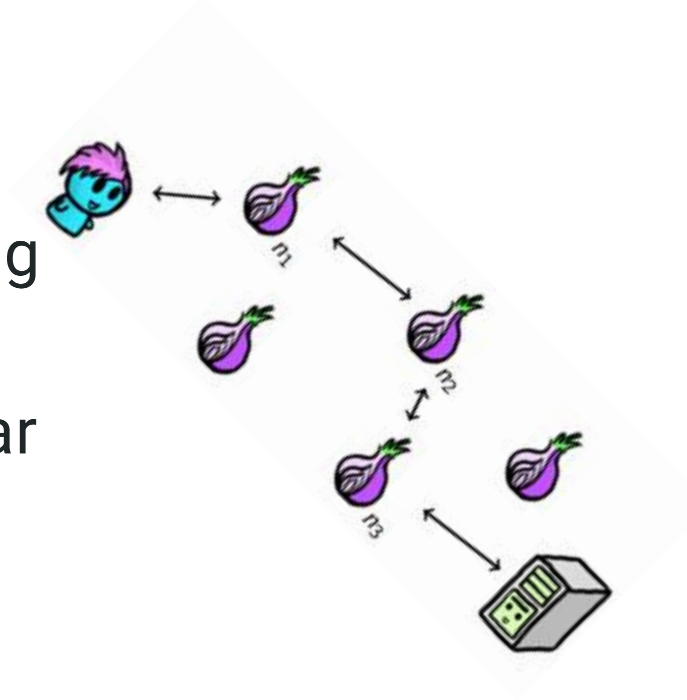
# Who knows what?

- Node $n_1$ knows that Alice is using Tor, and that her next node is $n_2$, but does not know which website Alice is visiting
- Node $n_3$ knows some Tor user (with previous node $n_2$) is visiting a particular website, but doesn't know who



# Who knows what?

- Node $n_1$ knows that Alice is using Tor, and that her next node is $n_2$, but does not know which website Alice is visiting
- Node $n_3$ knows some Tor user (with previous node $n_2$) is visiting a particular website, but doesn't know who
- The website itself only knows that it got a connection from Tor node $n_3$



# 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 good

# Answer some more…

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

**A:** Alice is probably good

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

# Answer some more…

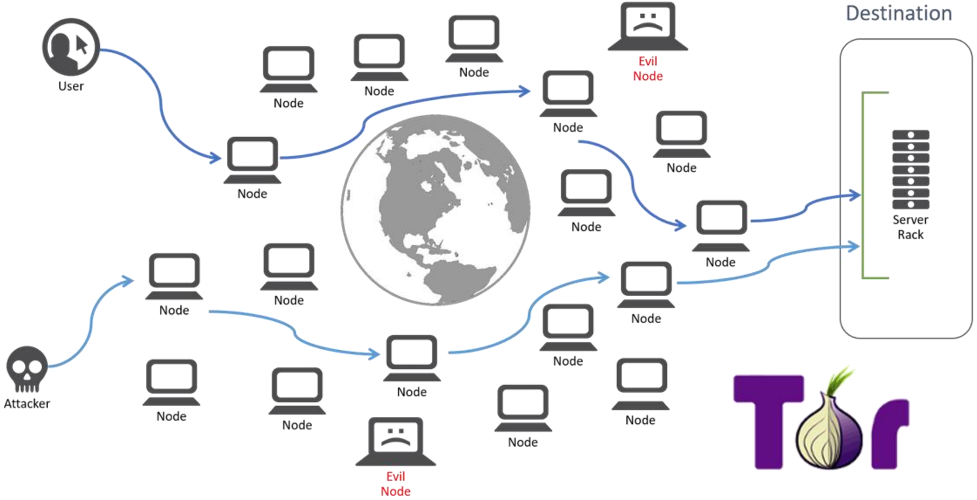
**Q:** What happens when Eve can inspect the incoming and outgoing traffic of a single node?

**A:** Alice is probably good

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

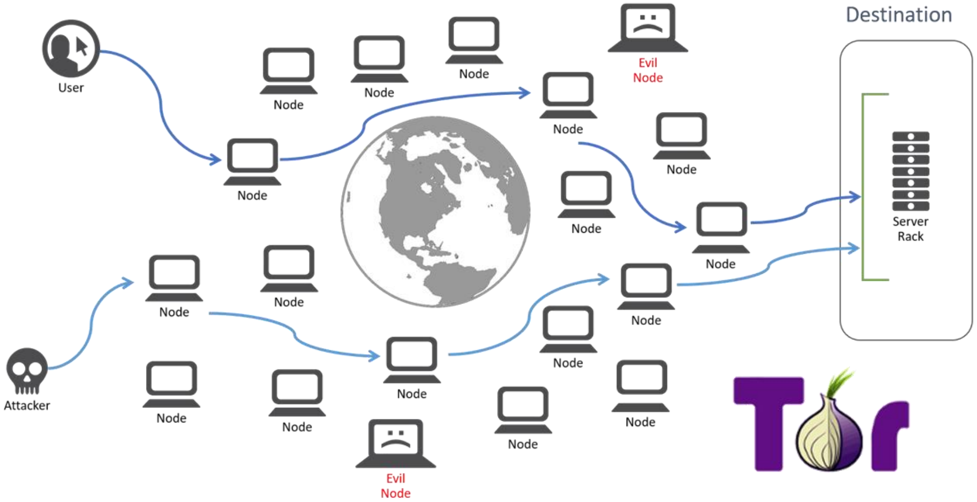
**A:** Traffic correlation attacks can easily de-anonymize Alice

# 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

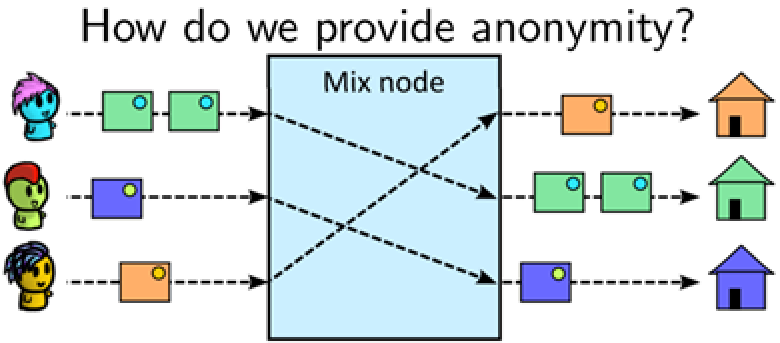
- We want to avoid a global passive adversary: choose nodes in different ISPs/countries
- How concentrated is the geographical distribution of Tor relays?

# Path Selection

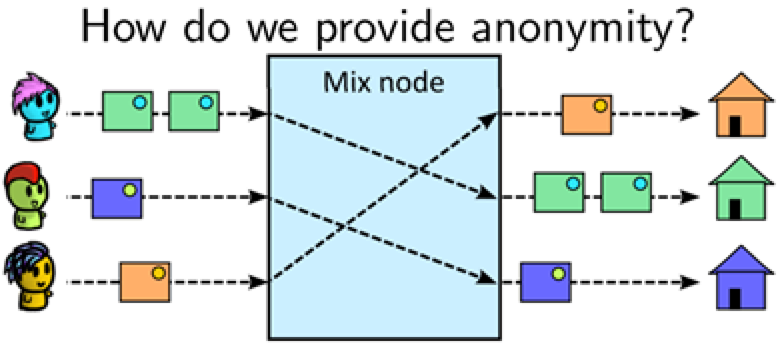
- Path selection algorithms can help
  - With anonymity: by picking nodes that are in different countries/ISPs
  - With performance: latency is affected by this
- Don’t forget that countries can collaborate as well
- We cannot use defenses that work in mixes (due to delay); those are called high-latency anonymous communication systems for a reason!
  - Mix-nodes focus on mixing and anonymizing traffic, making it more difficult to trace the source of the data.

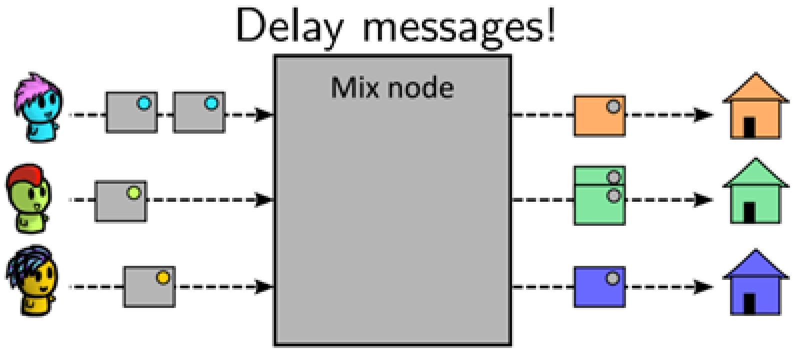
# Mixes

# Mixes: Basic Operations

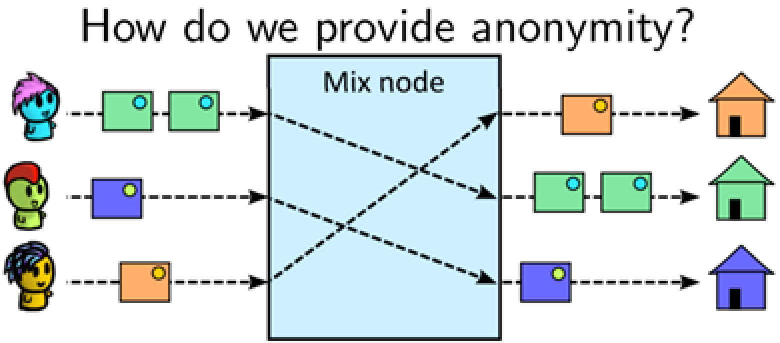


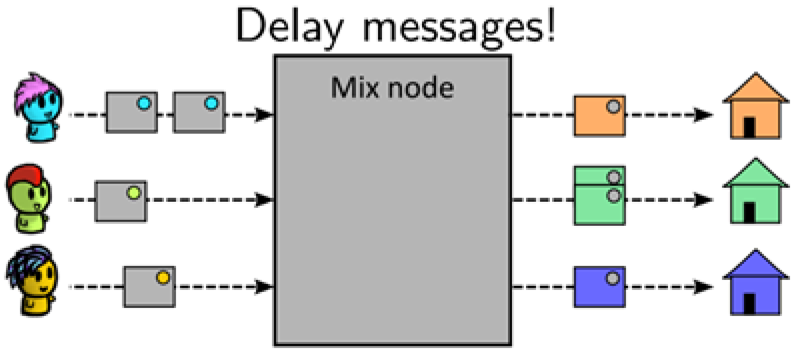
# Mixes: Basic Operations

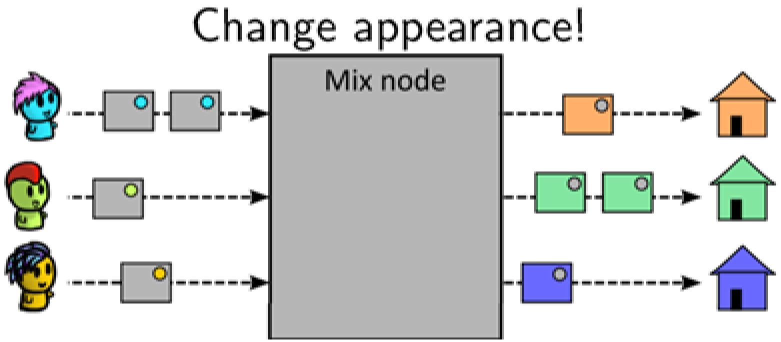


# Mixes: Basic Operations

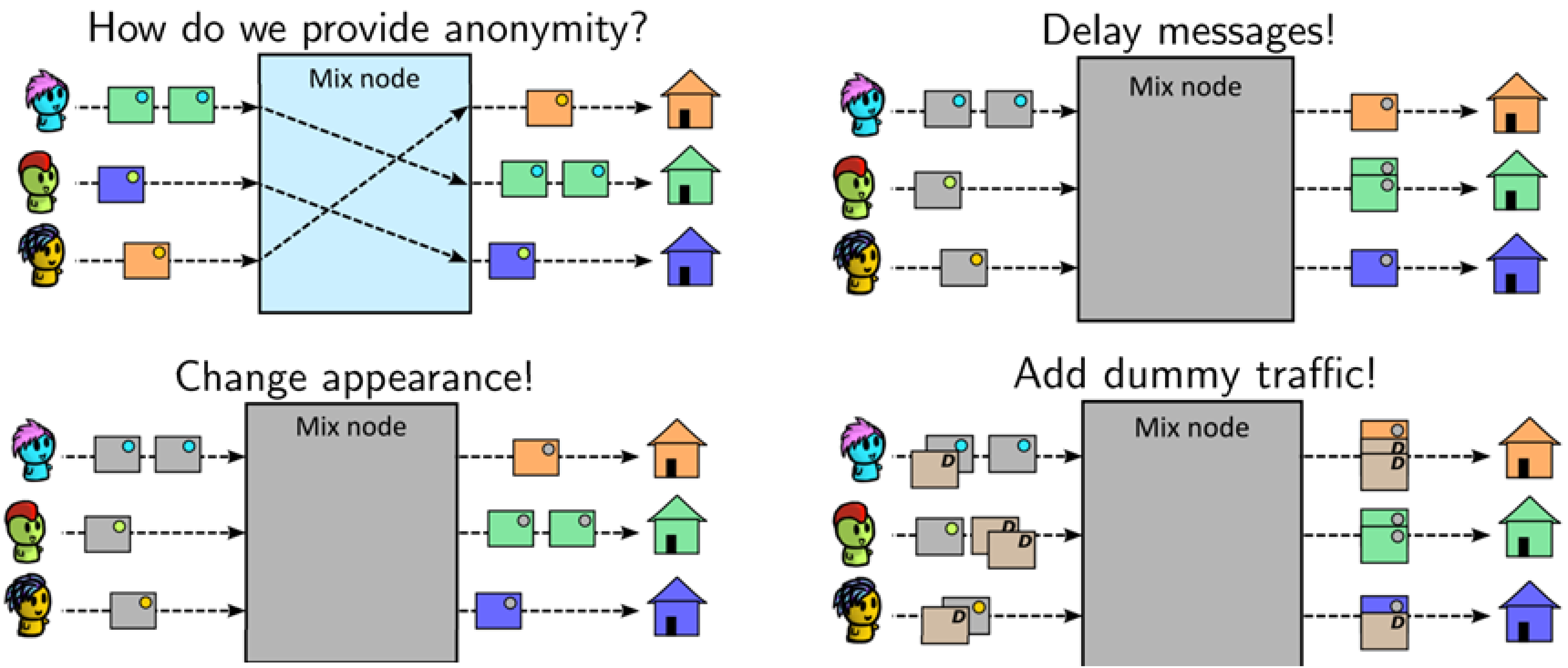


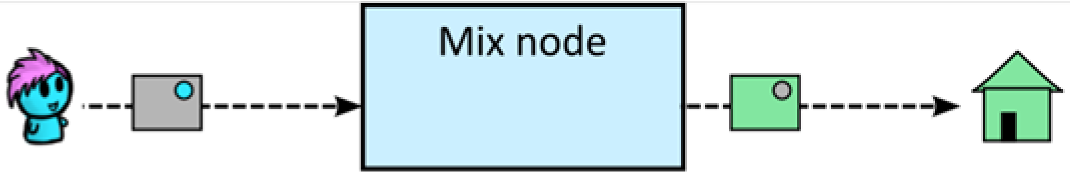

(e.g., changing headers or adding dummy data)

# Mixes: Basic Operations



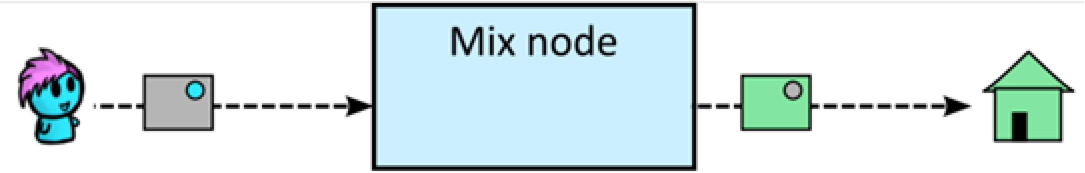
# Operation 1: Changing Appearance

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

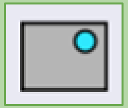
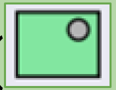


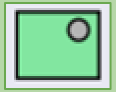
# Operation 1: Changing Appearance

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



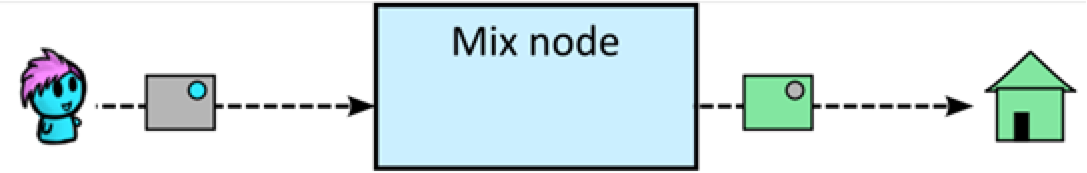

**A:** We can encrypt the output message with the Mix's key

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

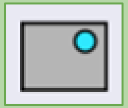
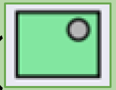
 $= E_{K\_Bob}(m)$

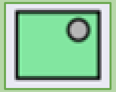
# Operation 1: Changing Appearance

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




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

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

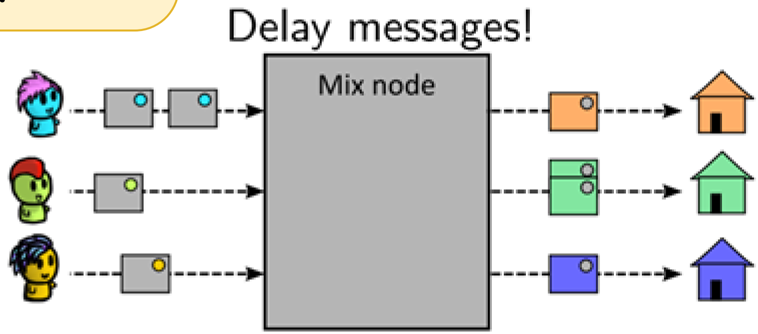
 $= E_{K\_Bob}(m)$

This "layered encryption" concept is called onion routing, and we saw it earlier in Tor.

# 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?

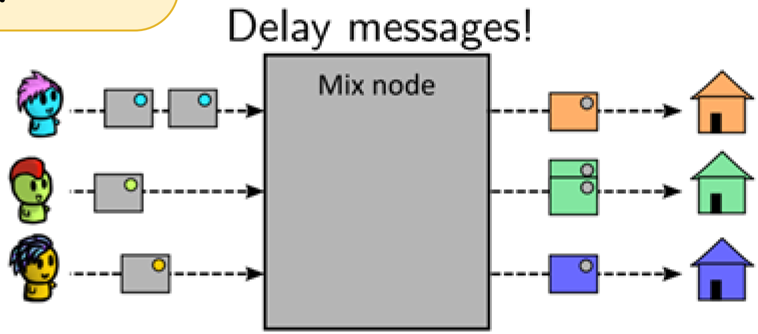


# Operation 2: Delaying Messages

**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.

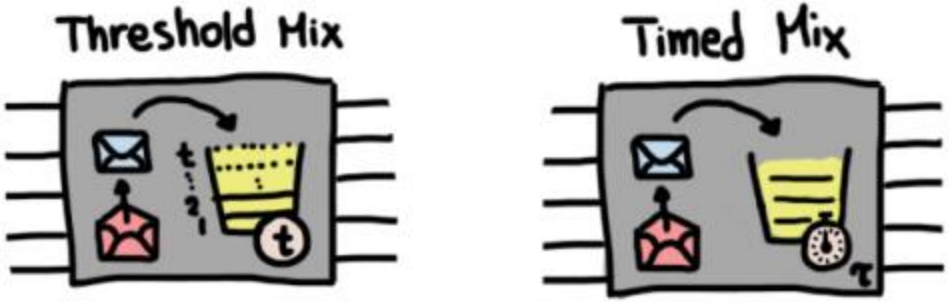



Deterministic delay: it's not constant, it depends on the arrival time and/or other messages. We will see some examples next!

# Threshold and Timed Mixes

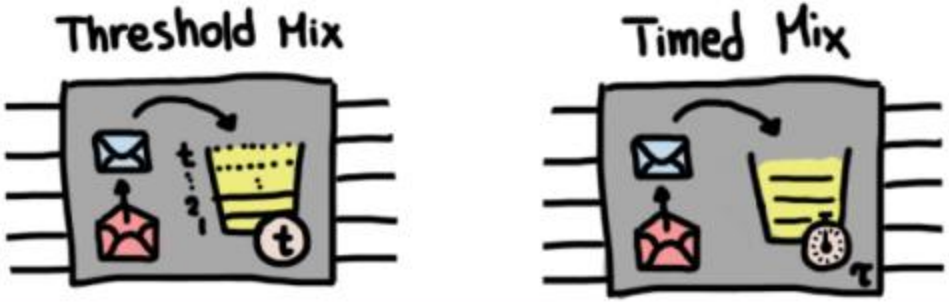
- Some popular mixes types are threshold and timed mixes.
- These mixes gather messages until a **flushing condition** triggers.
- When this condition happens, this marks the end of a **round**
  - Threshold mix: it gathers $t$ messages, then it flushes them.
  - Timed mix: it gathers messages until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

# Threshold and Timed Mixes




**Q:** Which of the two is better?

# Threshold and Timed Mixes

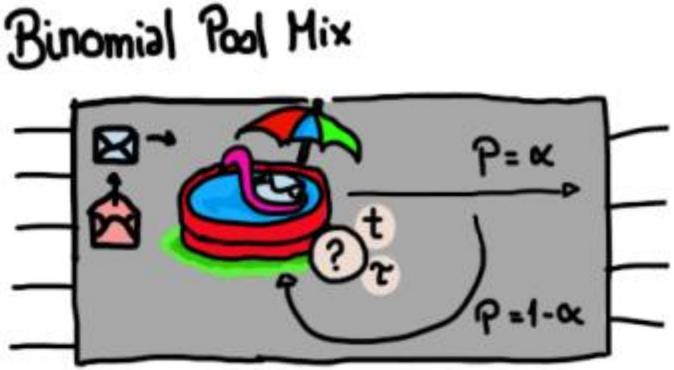


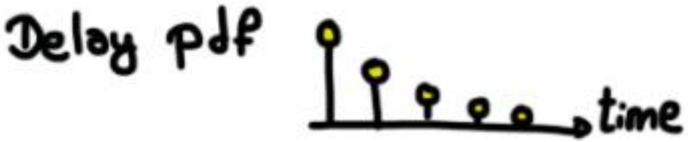

**Q:** Which of the two is better?

**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

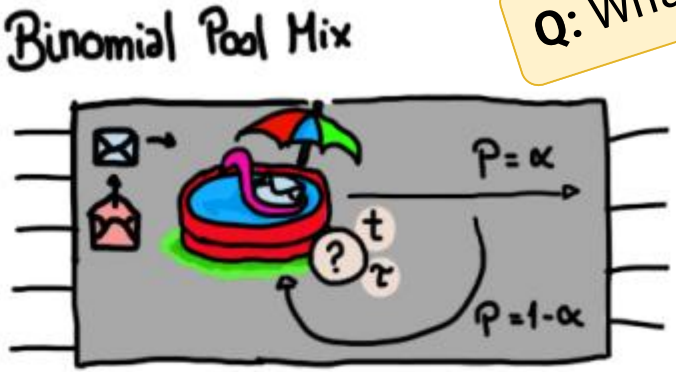


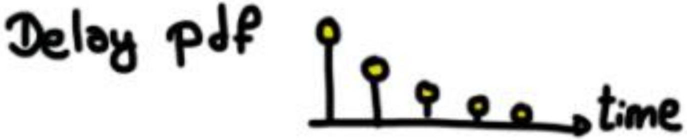



# Pool Mixes

- 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

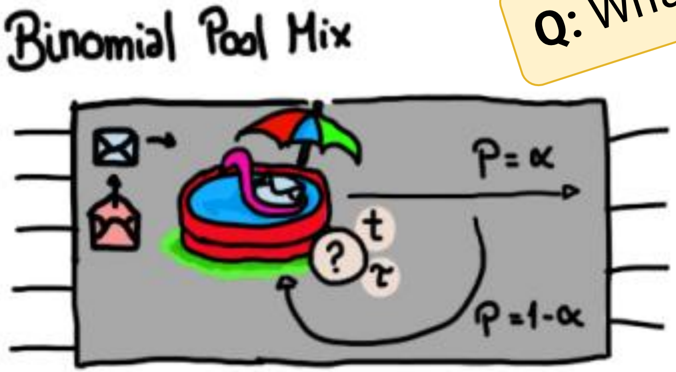
**Q:** What are the pros and cons of this?



# 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





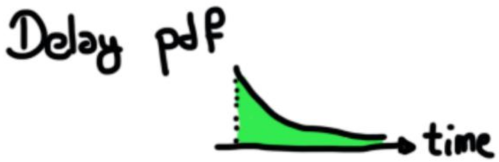
**Q:** What are the pros and cons of this?

**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

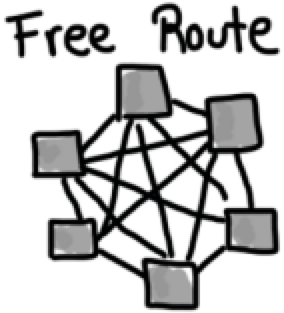
Let's discuss pros and cons of each topology!


**One after another**

- Pros: Increased Anonymity, Robustness, Reduced Correlation Risks, Scalability, Layered Security
- Cons: Latency, Complexity, Resource Intensive, Single Point of Failure

# Mixnet Topologies

Let's discuss pros and cons of each topology!

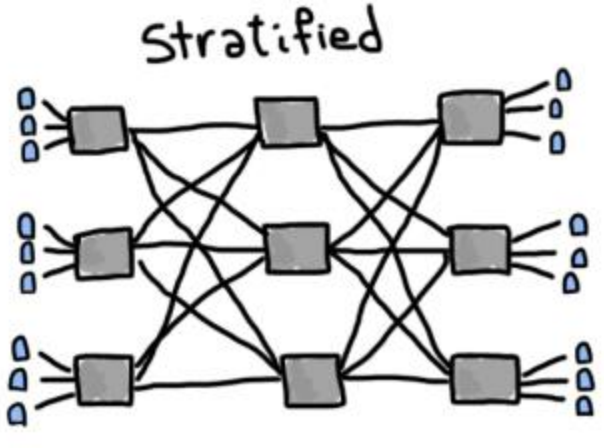



**All of them are connected**

- Pros: Increased Anonymity, Resilience to Node Compromise, Reduced Latency, Load Balancing
- Cons: Complexity, Resource Intensive, Vulnerability to Certain Attacks (e.g., If an adversary controls multiple nodes), Difficulty in Implementing Security

# Mixnet Topologies

Let's discuss pros and cons of each topology!




**Each layer is fully connected to the next layer**

- Pros: Enhanced Anonymity, Granular Control, Resilience, Traffic Management
- Cons: Complexity, Dependency on Layer Integrity(e.g., if one layer is compromised, it could jeopardize the entire system)

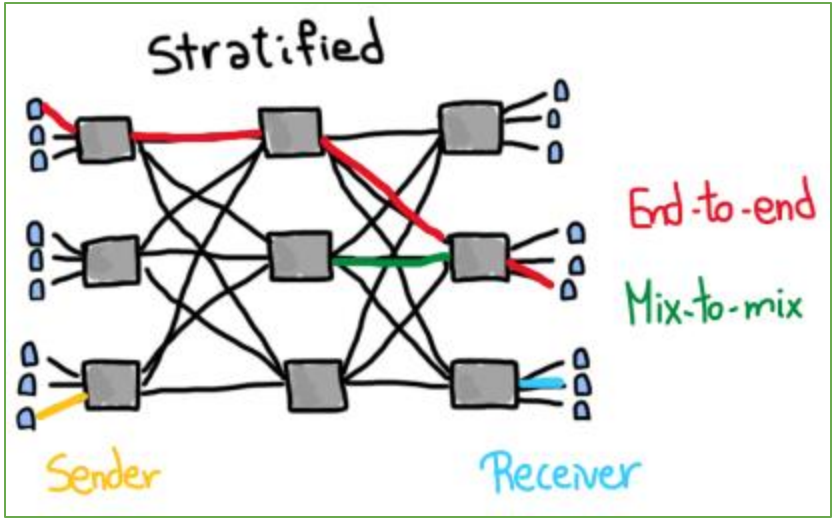
# Operation 3: Dummy Messages

**Q:** Where do we add dummy traffic?

# Operation 3: Dummy Messages

**Q:** Where do we add dummy traffic?

**A:** Anywhere, everywhere!



# Recap 1

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

# Recap 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

# Recap 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?

# Recap 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

# Recap 2

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

# Recap 2

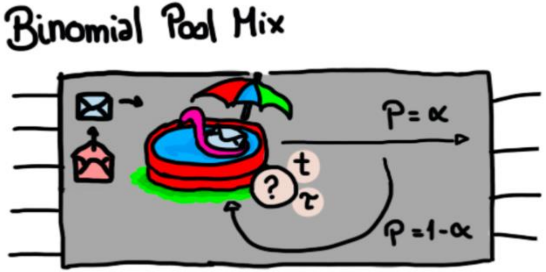
**Q:** Timed mixes: pros and cons of increasing the time τ?


**A:** Increasing τ improves anonymity but increases delay

# Recap 2

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


**A:** Increasing τ improves anonymity but increases delay



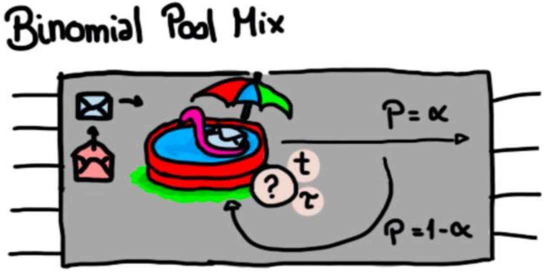

**Q:** Binomial pool mix: pros and cons of increasing the probability of forwarding a message α?

# Recap 2

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


**A:** Increasing τ improves anonymity but increases delay

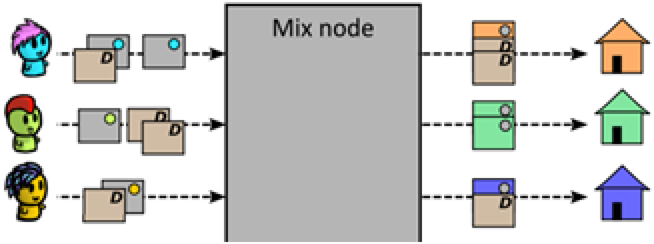
**Q:** Binomial pool mix: pros and cons of increasing the probability of forwarding a message α?




**A:** Increasing α decreases anonymity and delay

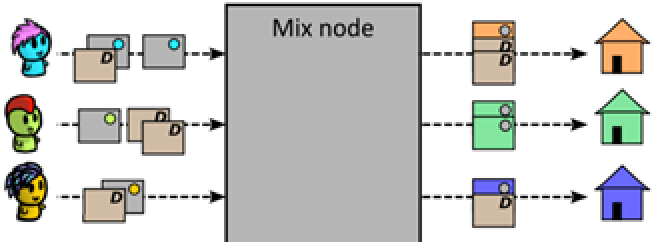
# Recap 2

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



# Recap 2

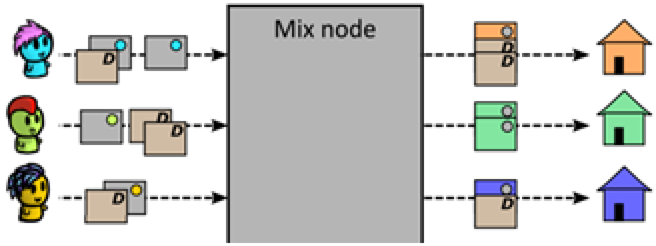
**Q:** Dummy traffic: pros and cons of increasing the amount of dummy messages?




**A:** More dummies require more bandwidth, but increase anonymity

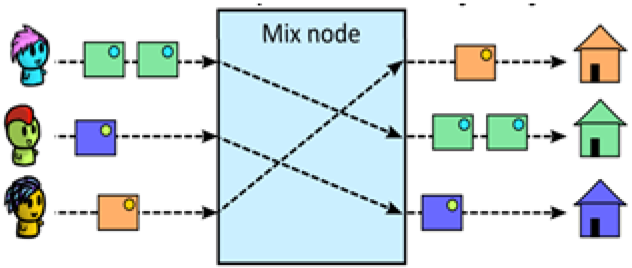
# Recap 2

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



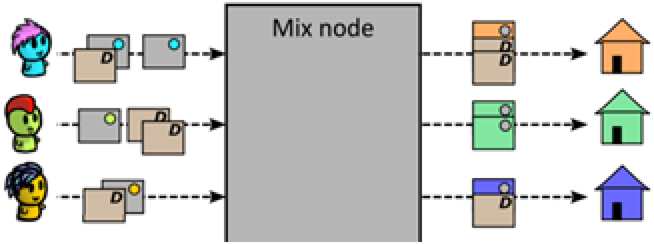

**A:** More dummies require more bandwidth, but increase anonymity

**Q:** What happens if the number of senders increases?



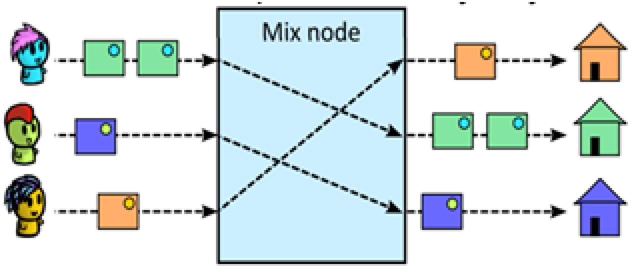
# Recap 2

**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

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

# 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
  - (but your original address is stored in a table)
- 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.fi, works as long as…

- No one's ‘ … o or from anon.pen…
- The opera… ardware), and the s… uncompr…
- The mapp… esses is kept secret

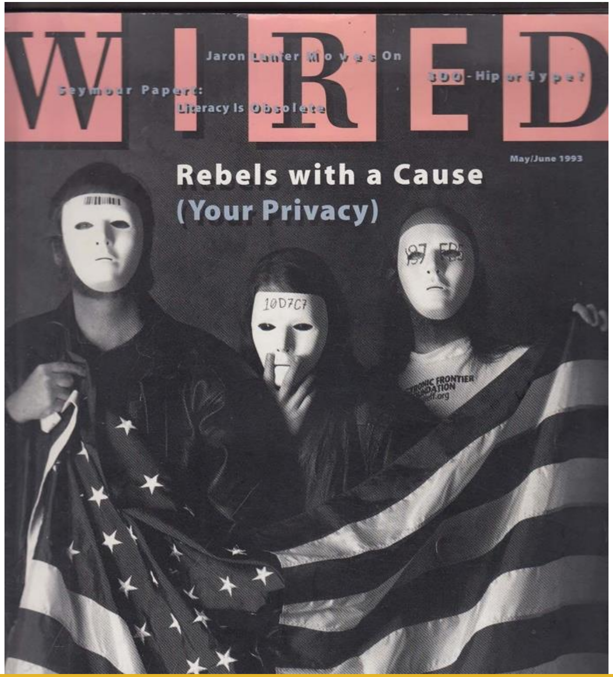

Unfortunately, a … rts of the list, and he shut down th… rotect it

# Type 1 Remailers

## Cypherpunk (type 1) 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!



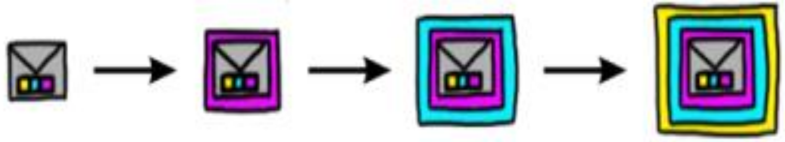

*An assumed name that conceals one's true identity

# Nym servers / Pseudonymous remailers

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

- “nym servers” mapped pseudonyms to “reply blocks” that contained a nested encrypted chain of type I remailers.
- Alice picks a list of nym servers

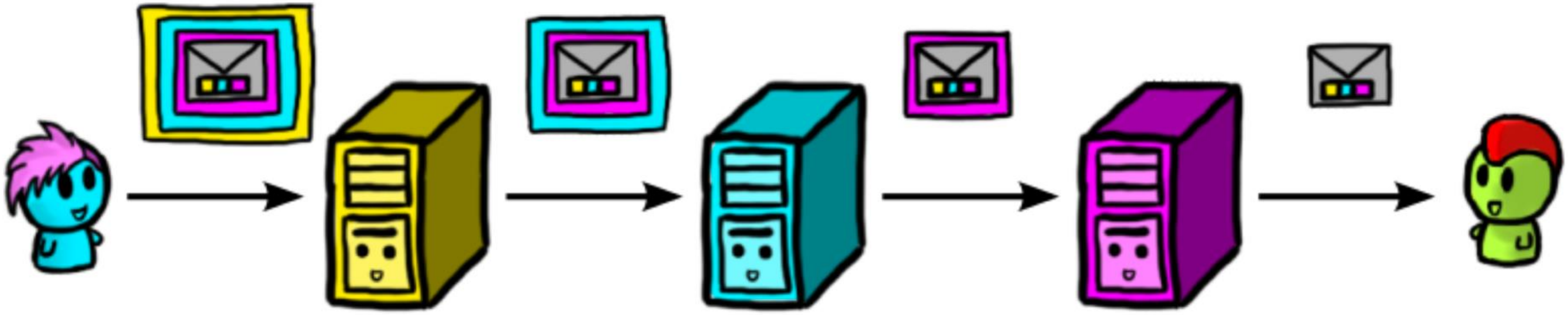
- Then, Alice builds her message using layered encryption
- The message contains a chain of reply blocks(+Alice pseudonymous address)



# Nym servers / Pseudonymous remailers

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

- The remailer forwards the message, but the sender's identity remains hidden. The reply block is included in the forwarded message.



- Bob replies by attaching his response to the end of the reply blocks
  - The reply is anonymized before being sent to the original sender

# 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"

# The Nym Network [Claudia Diaz, Harry Halpin, and Aggelos Kiayias (2021)]