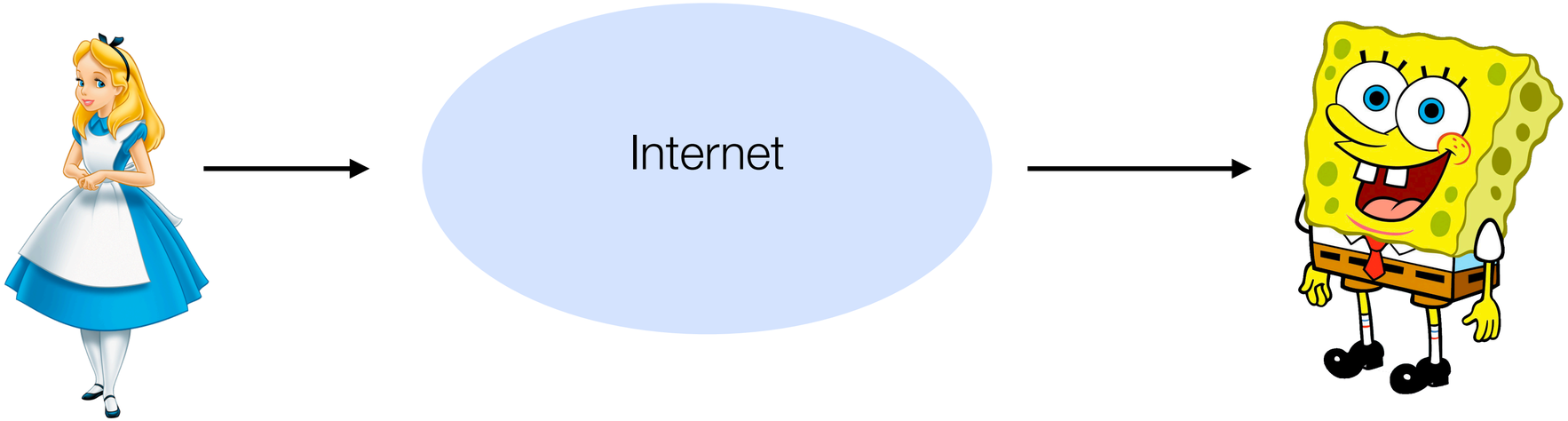
CS 350S: Privacy-Preserving Systems

Anonymous Messaging I

End-to-end encrypted communication

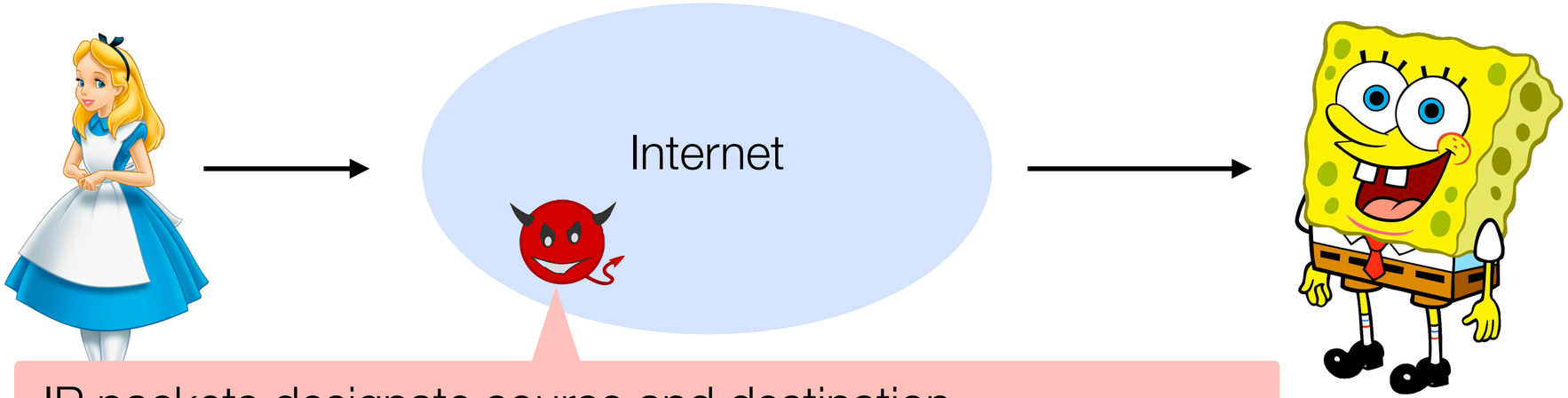
TLS / HTTPS provides confidentiality for communication on the internet



End-to-end encrypted communication

TLS / HTTPS provides confidentiality for communication on the internet

… but an adversary monitoring traffic can learn who is talking to who



Anonymity

State of being not identifiable within a set of subjects

Hides the link between an action and an identity

- Encrypted message and the sender + recipient
- Reddit post and a user’s identity
- Vote and the voter’s identity
- Financial transaction and the sender + recipient

Confidentiality vs. Anonymity

Confidentiality: defined with respect to one challenger and an adversary

Anonymity: defined across many users

- Anonymity set: set of subjects that Bob is not identifiable in (want to be large!)

Applications of anonymity

- Hiding transactions or web browsing from advertisers or government surveillance
- Untraceable electronic mail (whistleblowers, political dissidents, socially sensitive communications, confidential business negotiations)
- Digital cash (electronic currency with properties of paper money)
- Anonymous electronic voting
- Censorship-resistant publishing

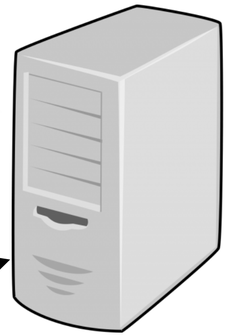
Outline

1. **Mixnets**
2. DC nets
3. Tor
4. Student presentation

Chaum's mix [Chaum81]

Limitation? Mix is central point of attack!

$\mathrm{Enc}_{\mathrm{pk}_{\mathrm{Mix}}}(\mathrm{Enc}_{\mathrm{pk}_{\mathrm{Bob}}}(m_1), \mathrm{Bob})$



$\mathrm{Enc}_{\mathrm{pk}_{\mathrm{Bob}}}(m_1)$

$\mathrm{sk}_{\mathrm{Bob}}$

$\mathrm{Enc}_{\mathrm{pk}_{\mathrm{Mix}}}(\mathrm{Enc}_{\mathrm{pk}_{\mathrm{Patrick}}}(m_2), \mathrm{Patrick})$

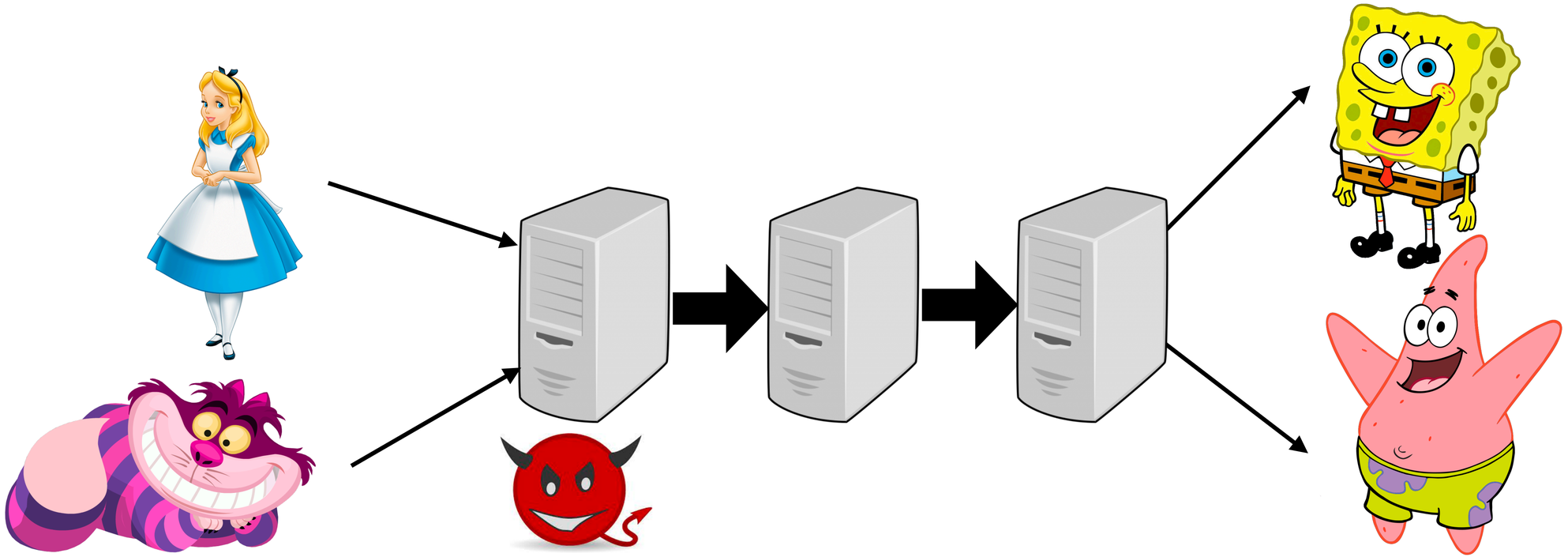
$\mathrm{sk}_{\mathrm{Mix}}$

$\mathrm{Enc}_{\mathrm{pk}_{\mathrm{Patrick}}}(m_2)$

$\mathrm{sk}_{\mathrm{Patrick}}$

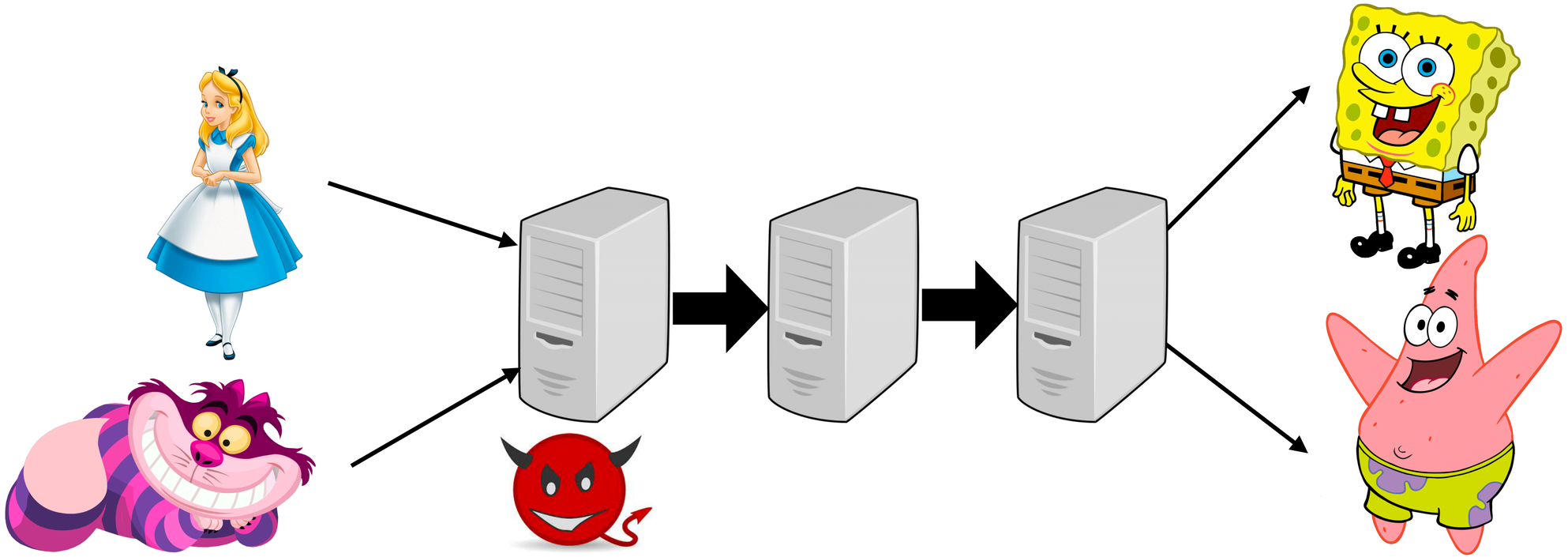
Mix cascade [Chaum81]

- Messages sent through sequence of mixes
  - Can form arbitrary network of mixes (“mixnet”)
- An attacker might control some mixes — even a single good mix guarantees anonymity



Mix cascade [Chaum81]

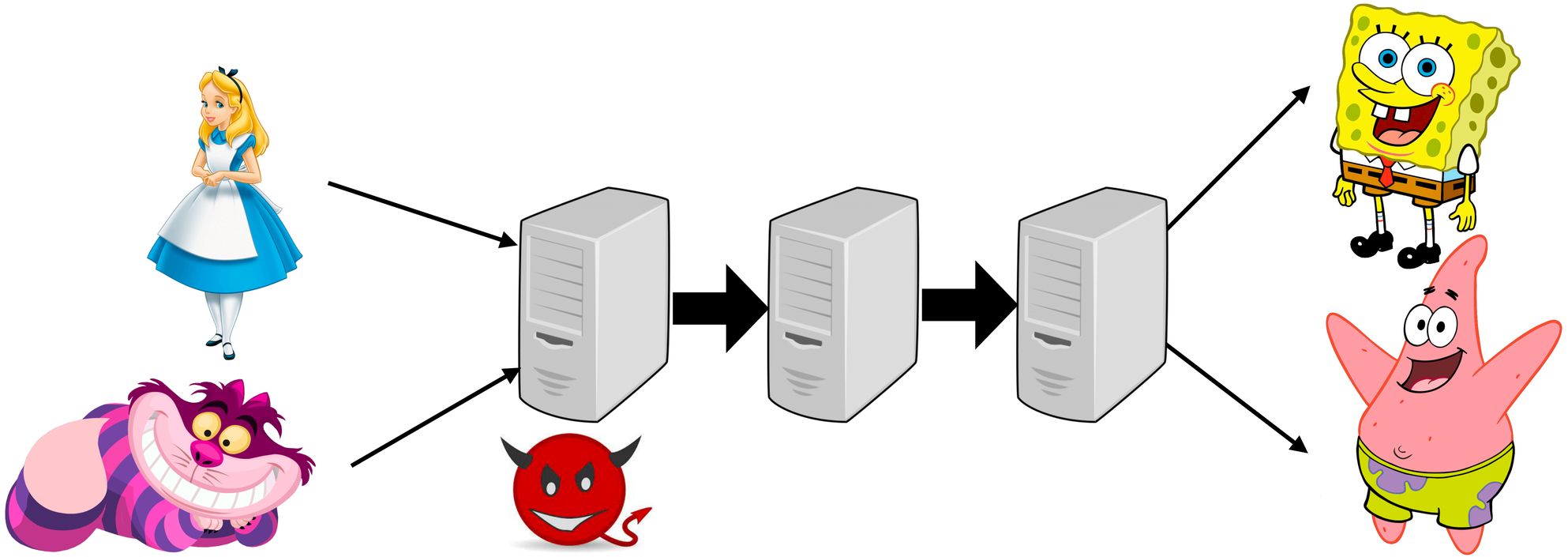
Limitations?



Mix cascade [Chaum81]

Limitations?

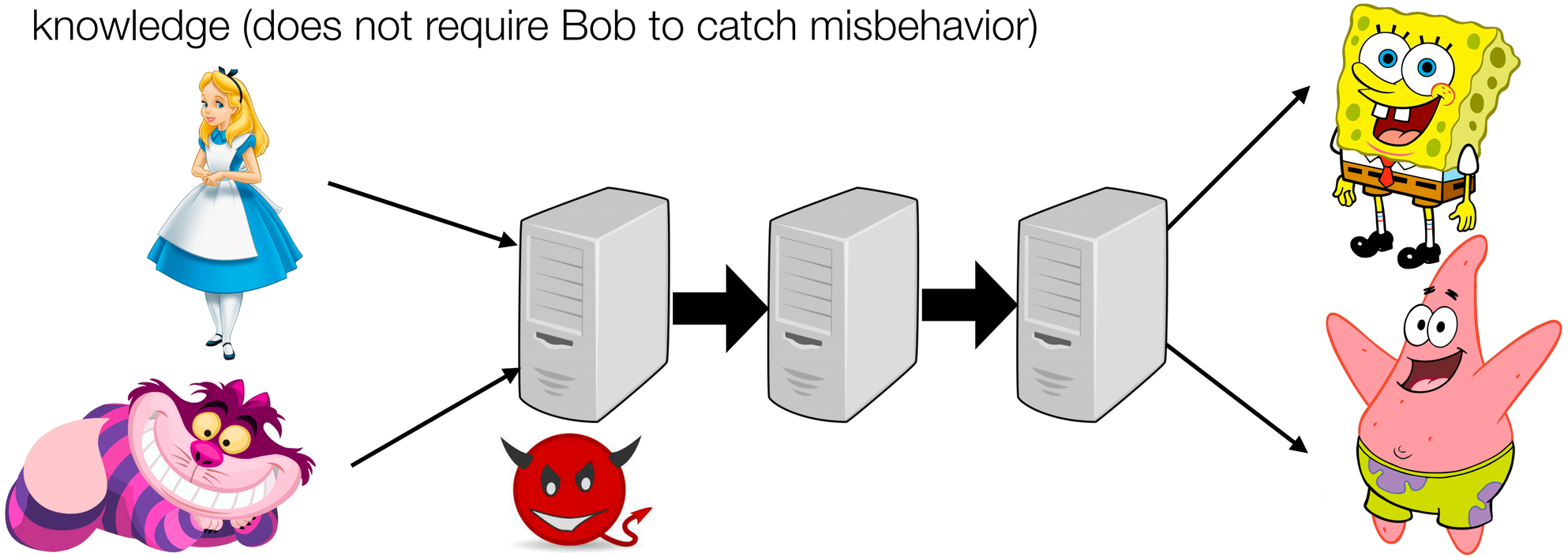
- High latency — need to wait for many users to submit messages for large anonymity set
- Only semi honest attackers — malicious mix can drop all messages except Alice's and break privacy



Mix cascade [Chaum81]

Defending against a malicious mix that drops all but Alice's message:

- One approach:
  - Each mix signs (1) the input messages and (2) the output batch
  - Bob can prove that a mix that received his message did not include it in its output batch
- Another approach: Each mix proved that it correctly executed its operations in zero-knowledge (does not require Bob to catch misbehavior)



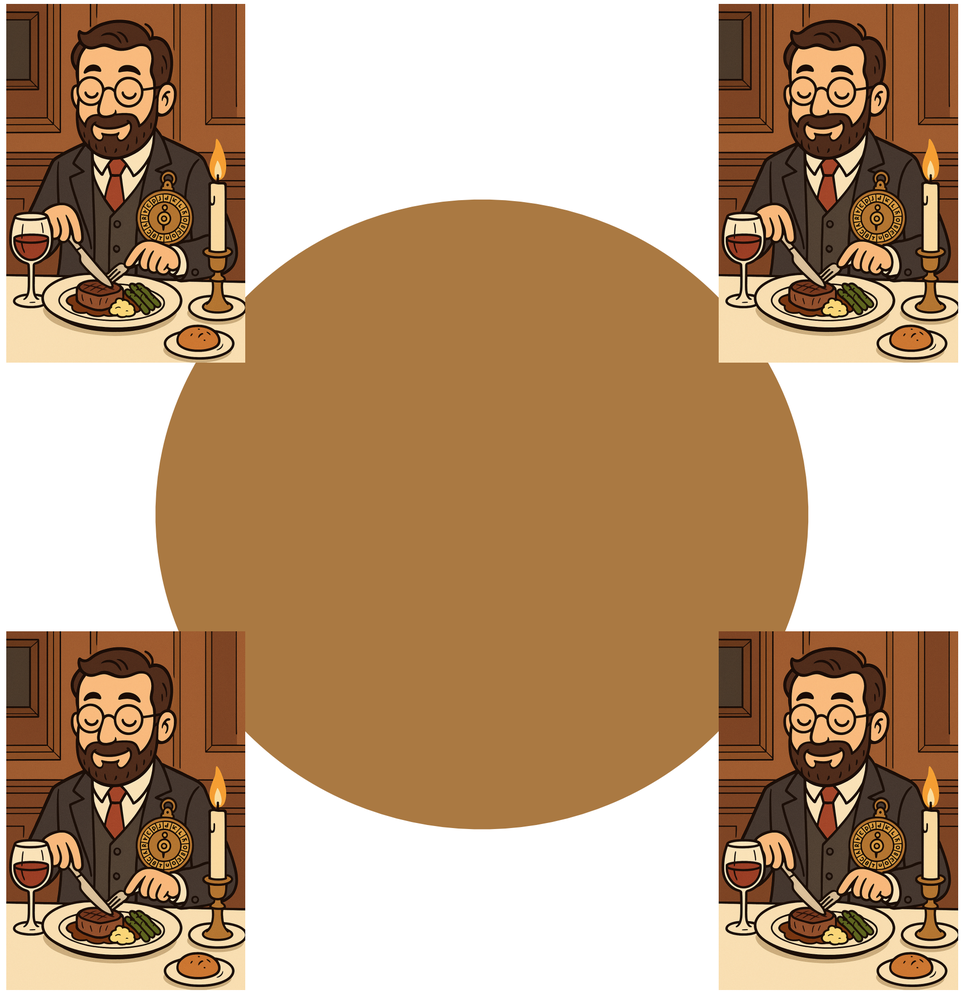
Outline

1. Mixnets
2. **DC nets**
3. Tor
4. Student presentation

Dining-cryptographers (DC) nets [Chaum88]

Cryptographers are having dinner when the waiter tells them that someone has anonymously paid the bill

The cryptographers want to know: is the NSA paying, or one of the cryptographers?
… but without learning which cryptographer is paying



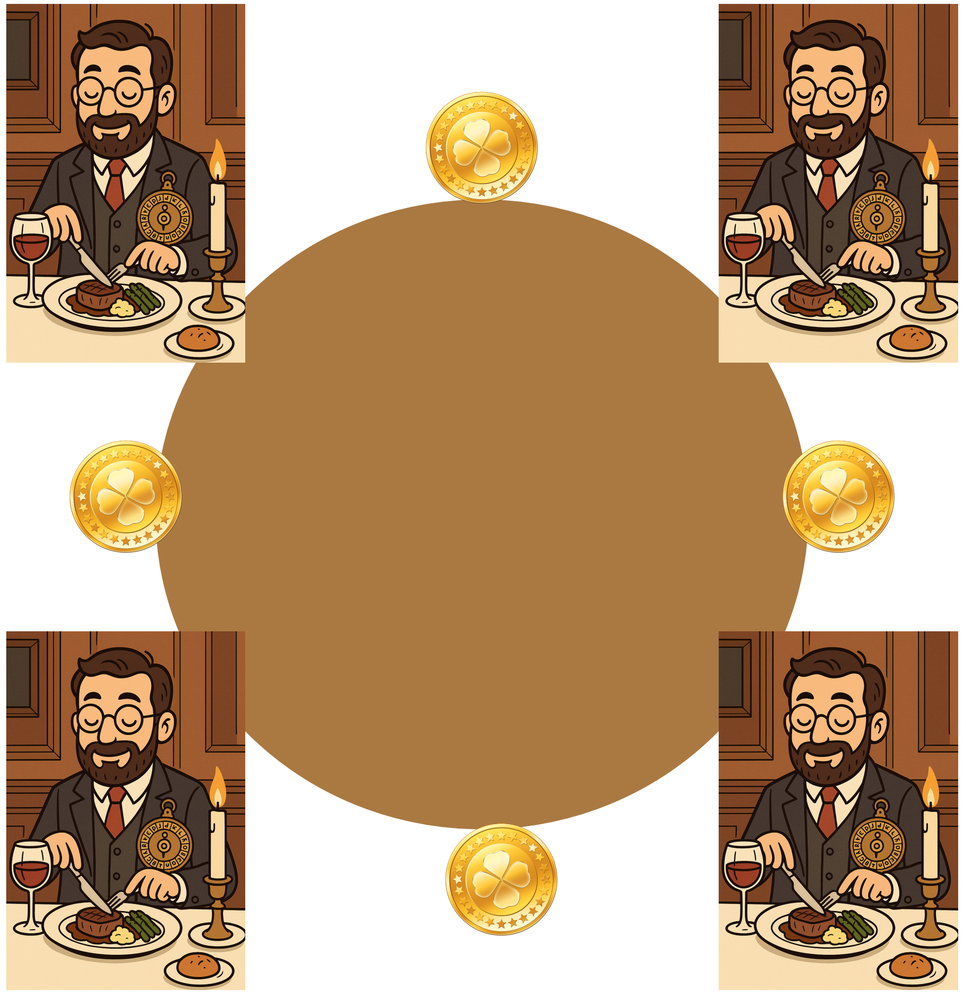
[Image from ChatGPT]

Dining-cryptographers (DC) nets [Chaum88]

The cryptographers want to know: is the NSA paying, or one of the cryptographers?
… but without learning which cryptographer is paying

Protocol:

1. Each pair of adjacent cryptographers flips a coin that only they can see
2. Each cryptographer says whether or not the two coins are the same or different
   If the cryptographer paid for dinner, he/she states the *opposite* (i.e., if the same, says different)
3. Odd # differences: cryptographer paid
   Even # differences: NSA paid



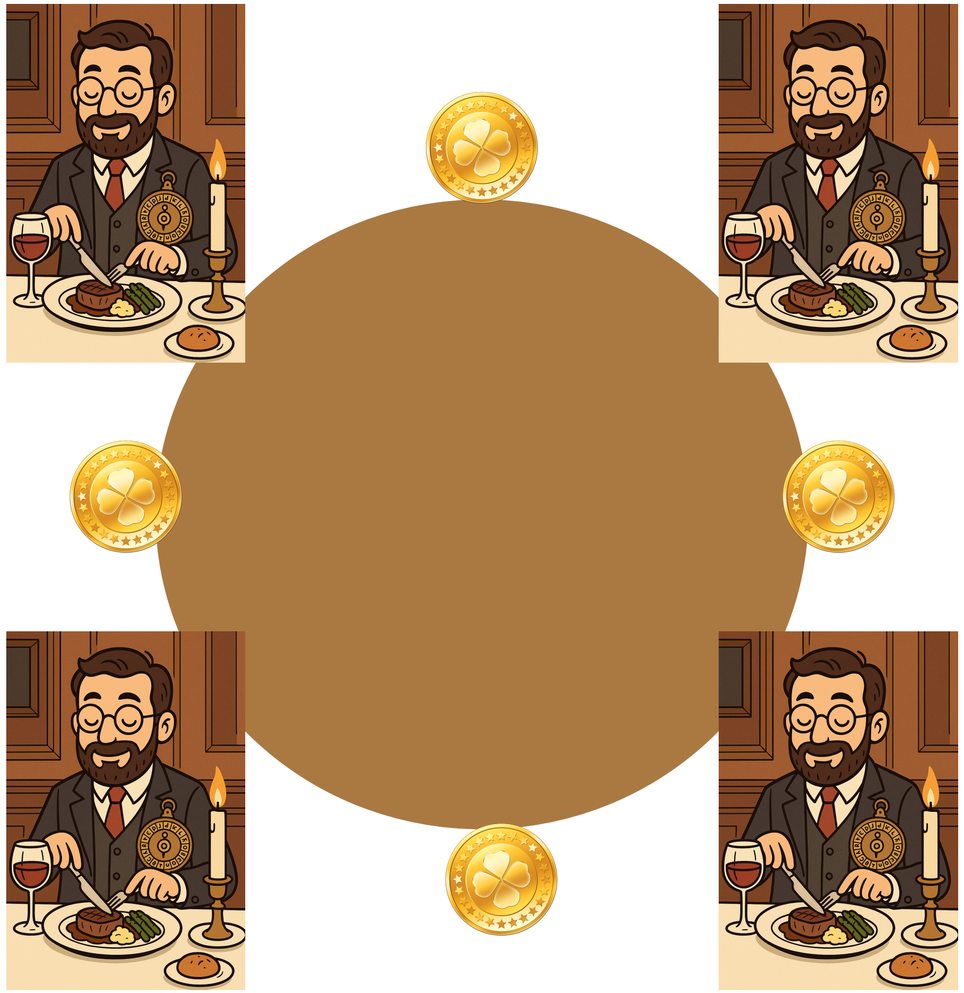
Dining-cryptographers (DC) nets [Chaum88]

Protocol:

1. Each pair of adjacent cryptographers flips a coin that only they can see
2. Each cryptographer says whether or not the two coins are the same or different
   If the cryptographer paid for dinner, he/she states the *opposite* (i.e., if the same, says different)
3. Odd # differences: cryptographer paid
   Even # differences: NSA paid

Why **correct**?

- Each actual observed difference comes in pairs
- If a cryptographer paid, there is one extra difference



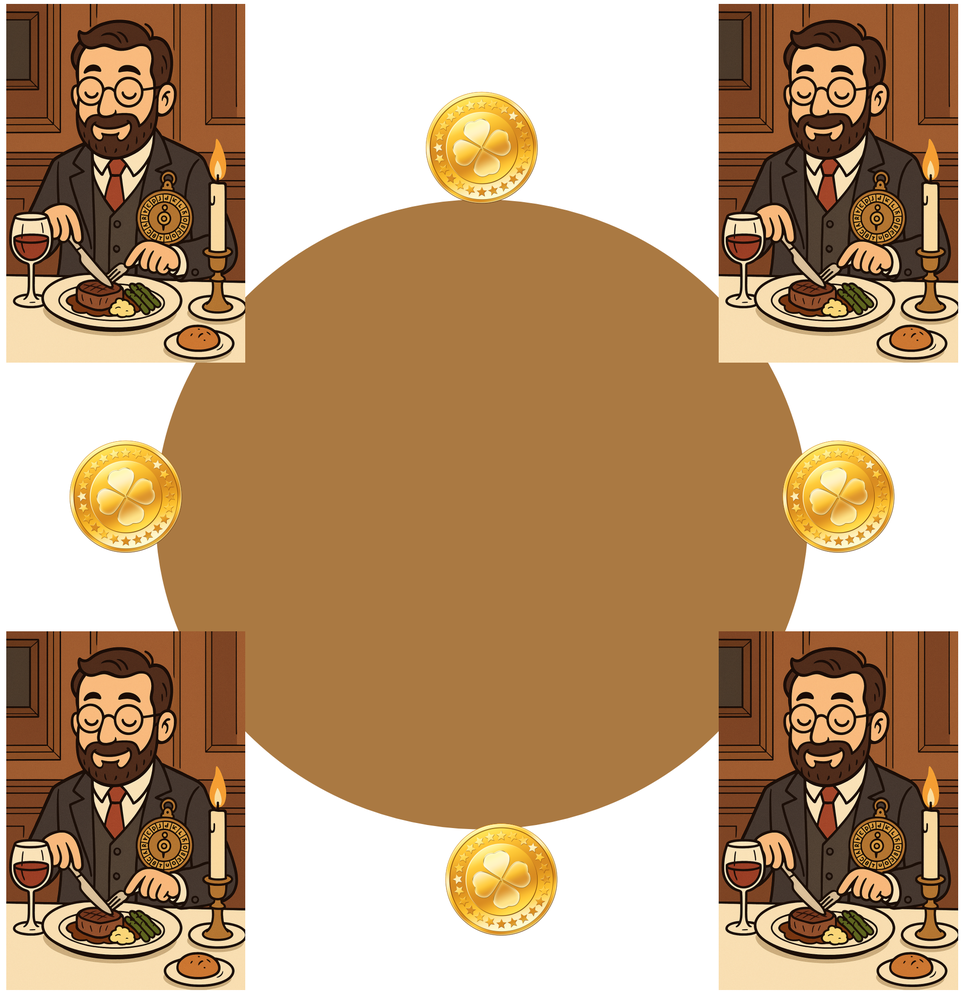
Dining-cryptographers (DC) nets [Chaum88]

Protocol:

1. Each pair of adjacent cryptographers flips a coin that only they can see
2. Each cryptographer says whether or not the two coins are the same or different
   If the cryptographer paid for dinner, he/she states the *opposite* (i.e., if the same, says different)
3. Odd # differences: cryptographer paid
   Even # differences: NSA paid

Why **private**?

- A cryptographer's statement is equally likely, regardless of whether he/she paid
- Information-theoretic privacy



Generalizing DC-nets

- One user wants to send a message $m \in \{0,1\}$
- Each user i samples a random bit r_i and sends it to “adjacent” neighbor $i-1$
- Each user announces: $r_i \oplus r_{i+1}$
- Sender announces: $r_i \oplus r_{i+1} \oplus m$
- XOR together all announcements to recover the message bit

Generalize to messages with more bits by running the protocol for each bit

Why is this correct (i.e., output the message bit)?

- Each random bit is XOR’d together twice (canceling out), while message bit only XOR’d once

Generalizing DC-nets

- One user wants to send a message $m \in \{0,1\}$
- Each user i samples a random bit r_i and sends it to “adjacent” neighbor $i - 1$
- Each user announces: $r_i \oplus r_{i+1}$
- Sender announces: $r_i \oplus r_{i+1} \oplus m$
- XOR together all announcements to recover the message bit

Generalize to messages with more bits by running the protocol for each bit

Limitations?

- With n parties, requires $O(n)$ communication for one sender to send 1 bit
- Doesn’t work if one party drops out

Outline

1. Mixnets
2. DC nets
3. **Tor**
4. Student presentation

Summary

Mixnets

- Need high latency for good security: need to wait longer for large batches of requests (i.e., large anonymity set)

DC-nets

- Information-theoretic security, but high communication overheads ($O(n)$ communication with n parties for 1 sender to send a single bit)

Tor

- Low latency, but security guarantees are more heuristic and less precise

Tor security goals

Adversary can

- Observe/modify some fraction of network traffic
- Operate/compromise some fraction of onion routers and directory servers

Goal: Preventing adversary from observing Alice's network behavior

Non-goals

- Defending against global passive adversary
- Preventing traffic confirmation attacks (suspects Alice talking to Bob)

Tor system goals

Deployability

- Cheap for nodes to run (“reasonable” bandwidth), no heavy liability burden on operators, not difficult/expensive to implement
- Non-anonymous parties (e.g., websites) should not have to run new software (exception: rendezvous points)

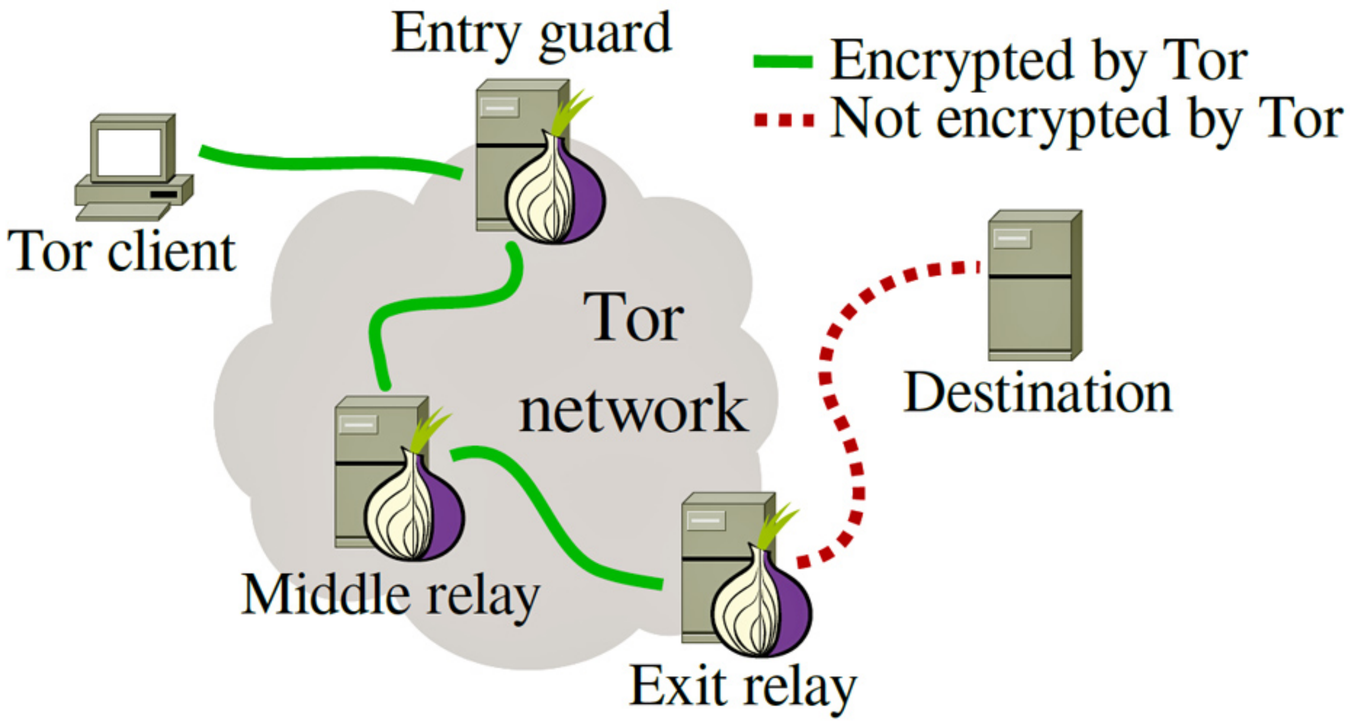
Usability

- Making a system more usable can increase the anonymity set —> better security!
- Should not require modifying applications, no prohibitive delays

Flexibility

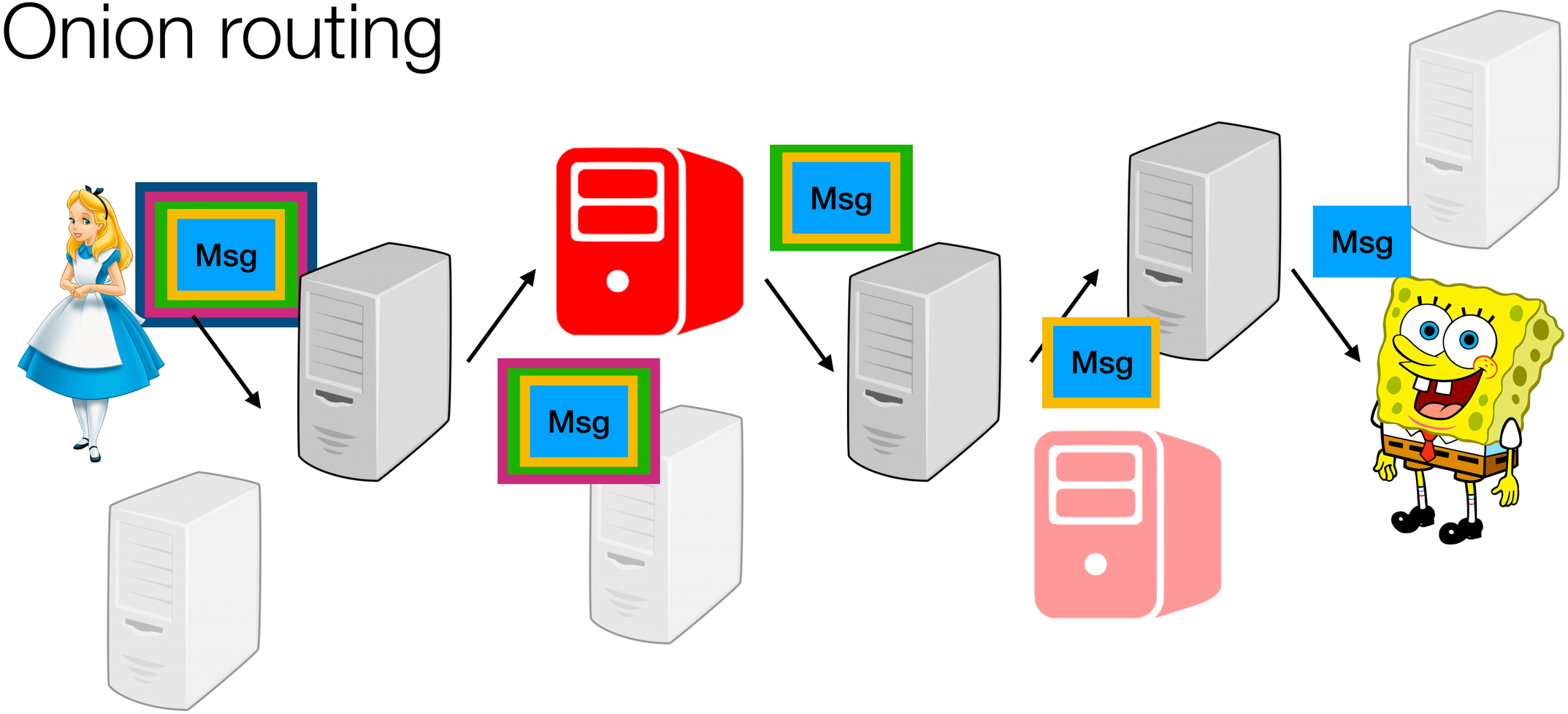
Simple design

Tor architecture



https://arstechnica.com/information-technology/2014/01/scientists-detect-spoiled-onions-trying-to-sabotage-tor-privacy-network/

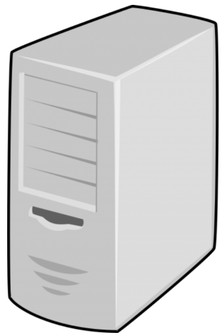
Onion routing



Onion routing: “peel off” a layer of decryption at each layer

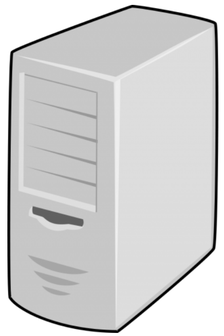
Constructing a circuit

Key exchange



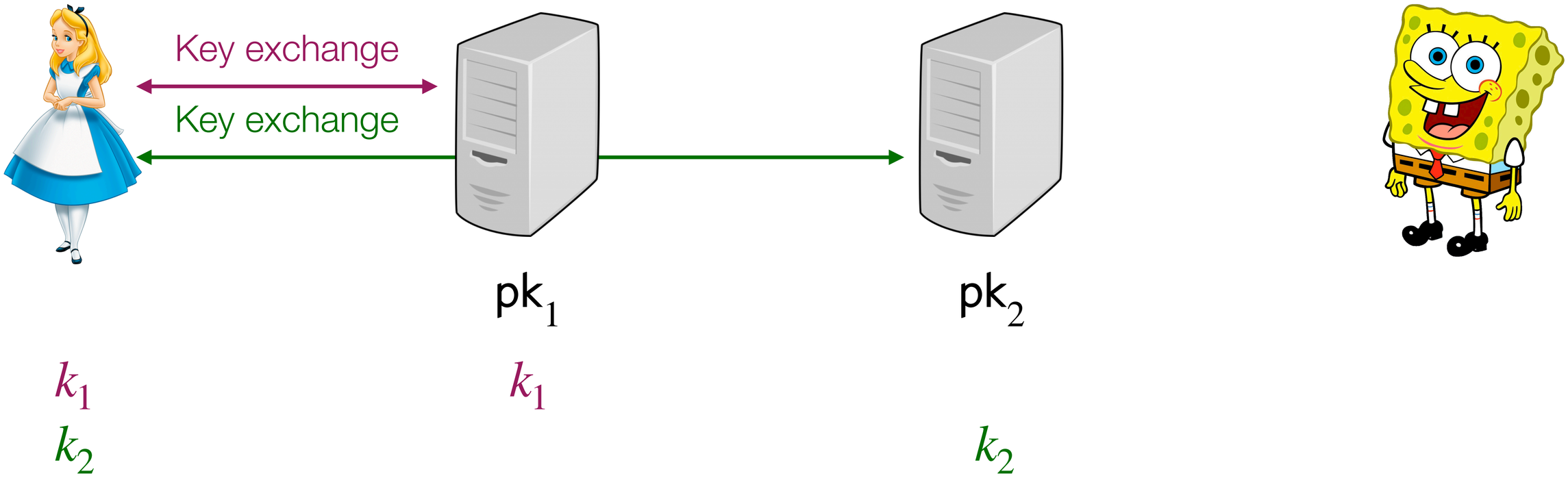
pk_1

k_1

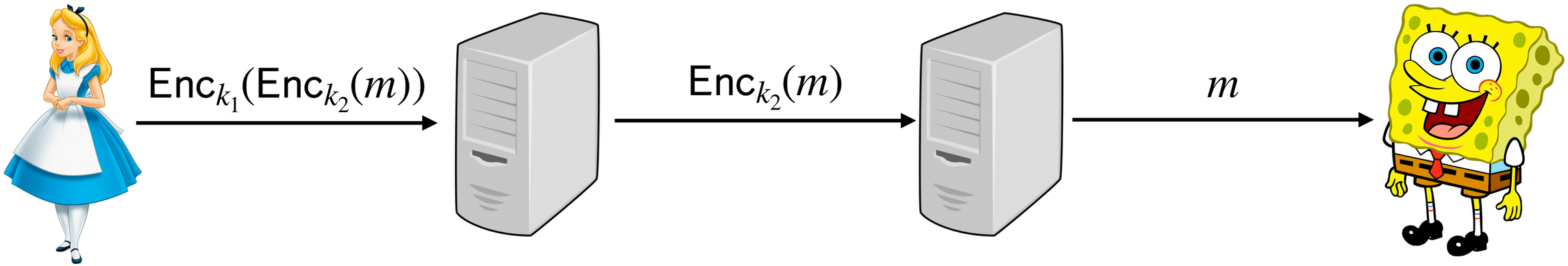


pk_2

Constructing a circuit



Sending message to Bob



k_1
k_2

k_1

k_2

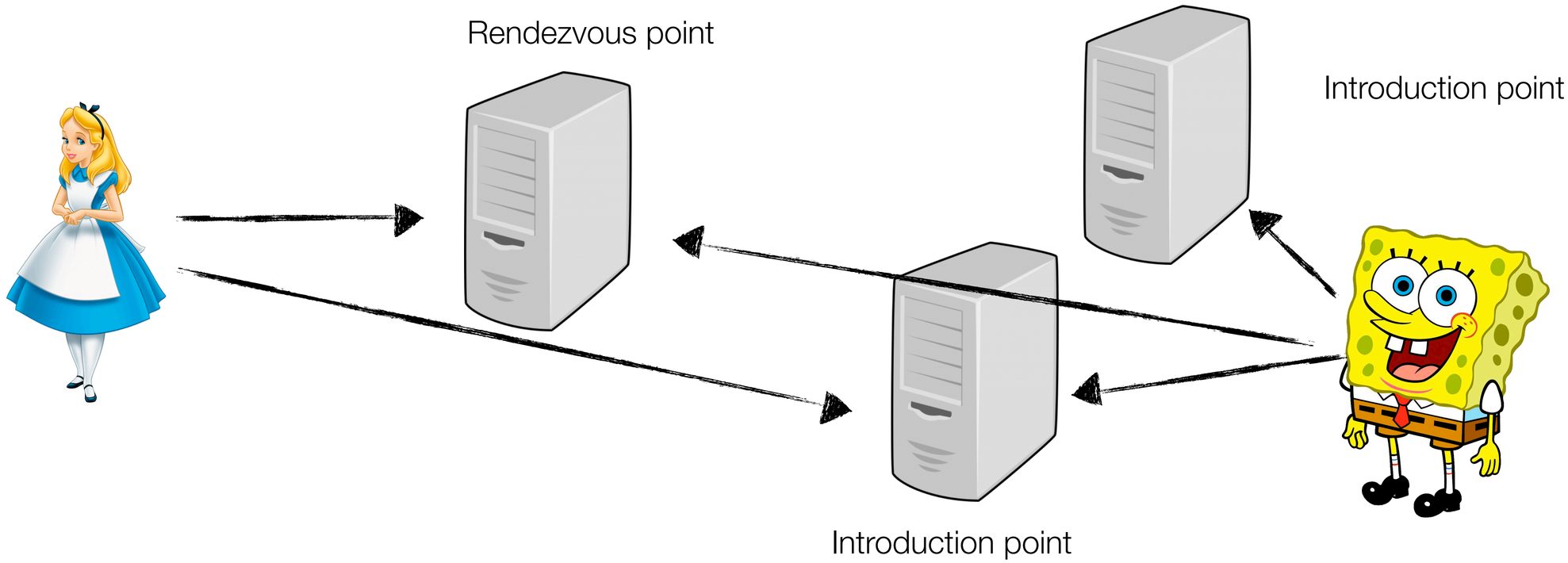
Alice should ensure that m is encrypted to Bob using HTTPS

Directory servers

- Maintain the list of Tor relays that clients can connect to
- Clients can connect to a directory server to fetch the list
- List needs to be agreed upon and signed by a majority of directory servers
- Important that an attacker cannot submit a large number of attacker-controlled nodes (“Sybil attack”)

Hidden services

- Bob wants to offer a TCP service without revealing his IP address
- Bob can advertise several introduction points
- Alice chooses a rendezvous point, tells Bob where the rendezvous point is by connecting to Bob’s introduction point, then waits for him to connect to rendezvous point



Limitations of Tor

- Susceptible to fingerprinting attacks (traffic pattern can reveal information about which site a user is connecting to)
- Attacker use timing and traffic volume to correlate endpoints
- Censors can block access to Tor relays (Tor bridges are proxies that will connect clients to Tor relays and try to remain hidden from censors)
- Using Tor in the first place may seem “suspicious” (particularly in certain countries)
- Exit nodes could send illegal or abusive traffic
- Need to trust directory servers to correctly admit, for the most part, honest Tor relays
- Clients need to be able to contact the directory servers in order to fetch the list of Tor relays
- Security guarantees are heuristic
- An adaptive adversary can compromise nodes along a path
- …

The anonymous Internet

Daily Tor users per 100,000 Internet users

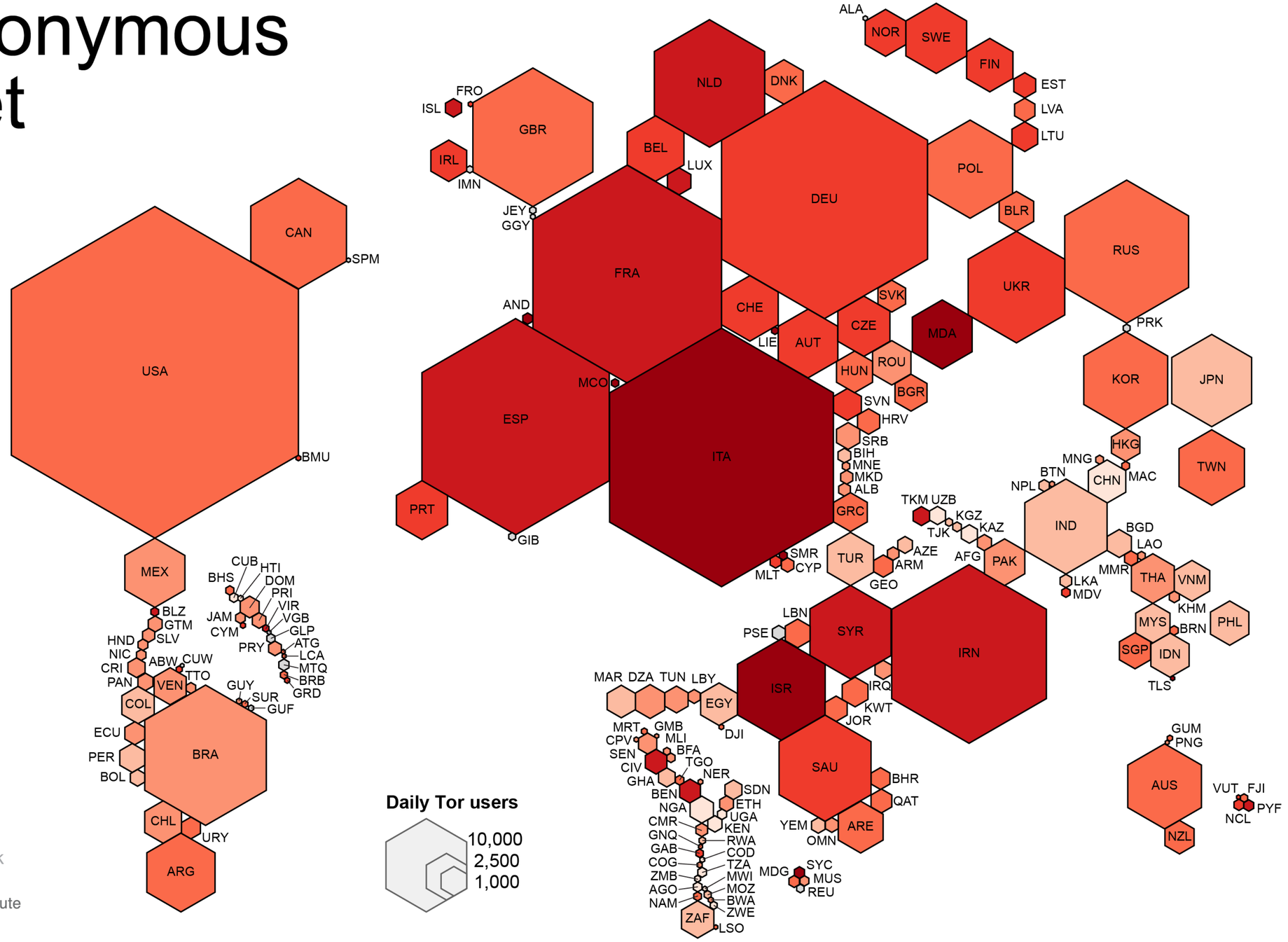
- > 200
- 100 - 200
- 50 - 100
- 25 - 50
- 10 - 25
- 5 - 10
- < 5
- no information

Average number of Tor users per day calculated between August 2012 and July 2013

data sources:
Tor Metrics Portal
metrics.torproject.org
World Bank
data.worldbank.org

by Mark Graham (@geoplace) and Stefano De Sabbata (@maps4thought)
Internet Geographies at the Oxford Internet Institute
2014 • geography.oii.ox.ac.uk

Oxford Internet Institute
University of Oxford



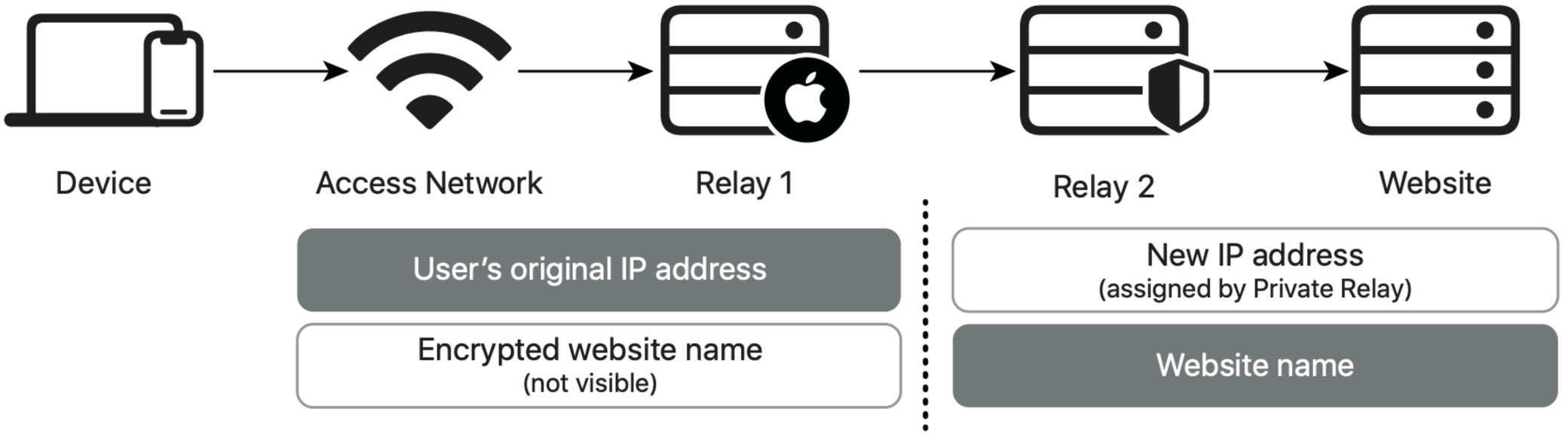
Daily Tor users

10,000
2,500
1,000

Apple private relay

- Ingress relay knows source IP address, but not destination
- Egress relay knows destination, but not source IP address
- Hides link between source and destination
- Also encrypts DNS requests

Run by a third-party CDN (e.g., Cloudflare)



https://www.apple.com/icloud/docs/iCloud_Private_Relay_Overview_Dec2021.pdf

Outline

1. Mixnets
2. DC nets
3. Tor
4. **Student presentation**

References

Chaum, David. "The dining cryptographers problem: Unconditional sender and recipient untraceability." Journal of cryptology 1, no. 1 (1988): 65-75.

Chaum, David L. "Untraceable electronic mail, return addresses, and digital pseudonyms." Communications of the ACM 24, no. 2 (1981): 84-90.

Dingledine, Roger, Nick Mathewson, and Paul Syverson. "Tor: The second-generation onion router." (2004).

https://6893.csail.mit.edu/lec18.pdf

Vitaly Shmatikov CS 259

https://courses.cs.washington.edu/courses/cse484/22au/slides/cse484-lecture21-au22.pdf

https://classes.cs.uchicago.edu/archive/2018/fall/23200-1/22.pdf

https://www.apple.com/icloud/docs/iCloud_Private_Relay_Overview_Dec2021.pdf