Protocols for anonymity

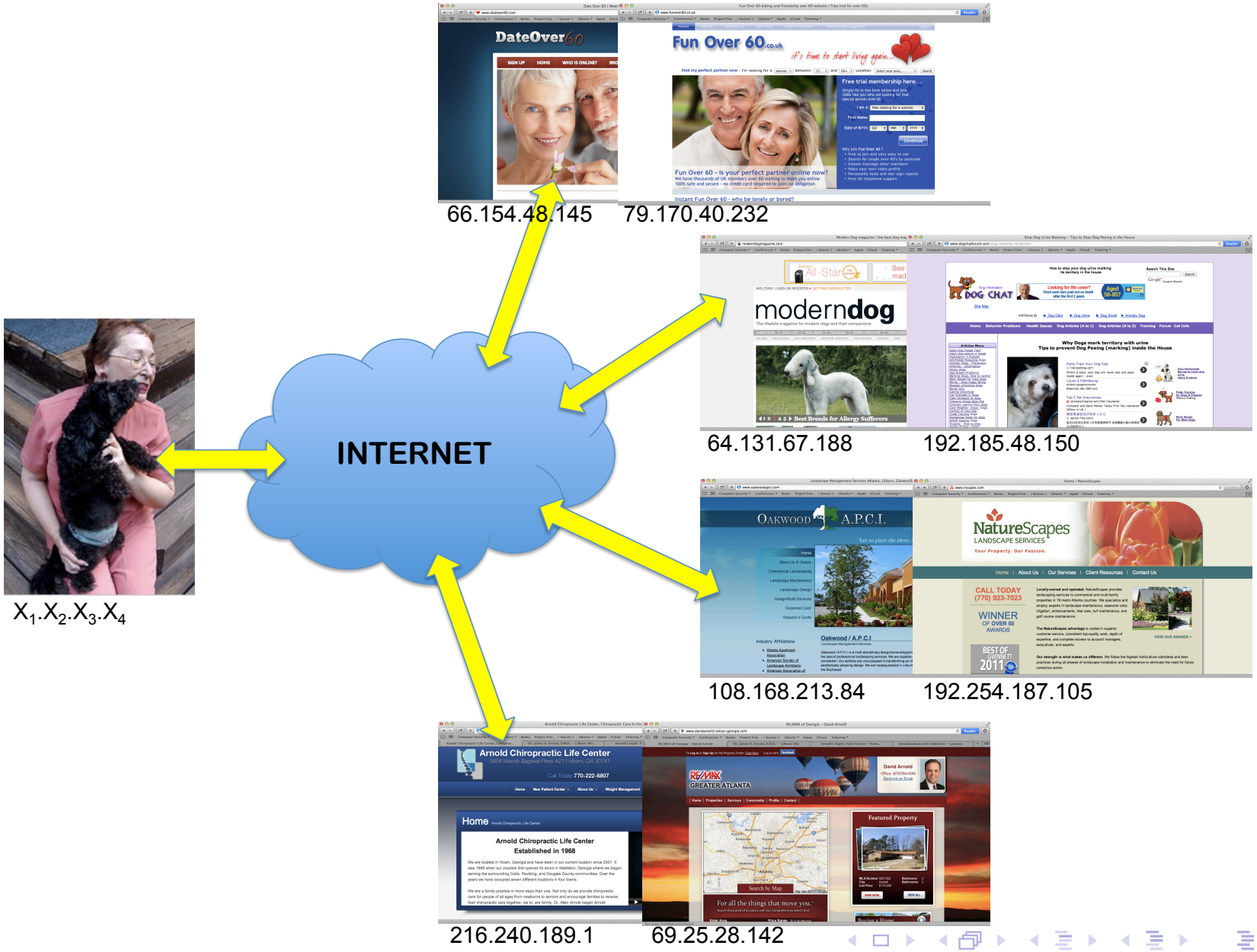
Myrto Arapinis
School of Informatics
University of Edinburgh

October 31, 2016

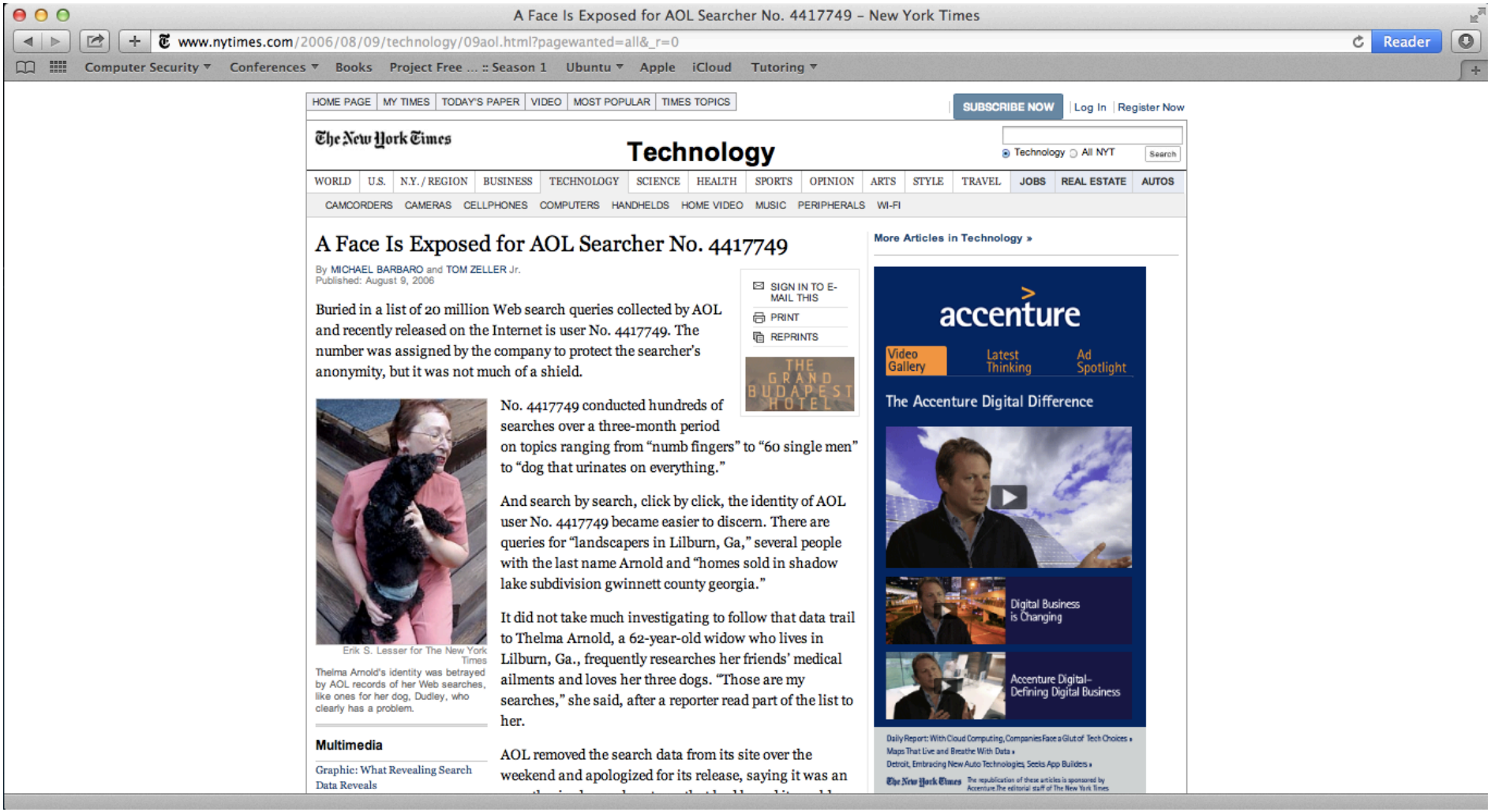
Context

- The Internet is a public network:
  - network routers see all traffic that passes through them
- Routing information is public:
  - IP packet headers contain source and destination of packets
- Encryption does not hide identities:
  - encryption hides payload, but not routing information

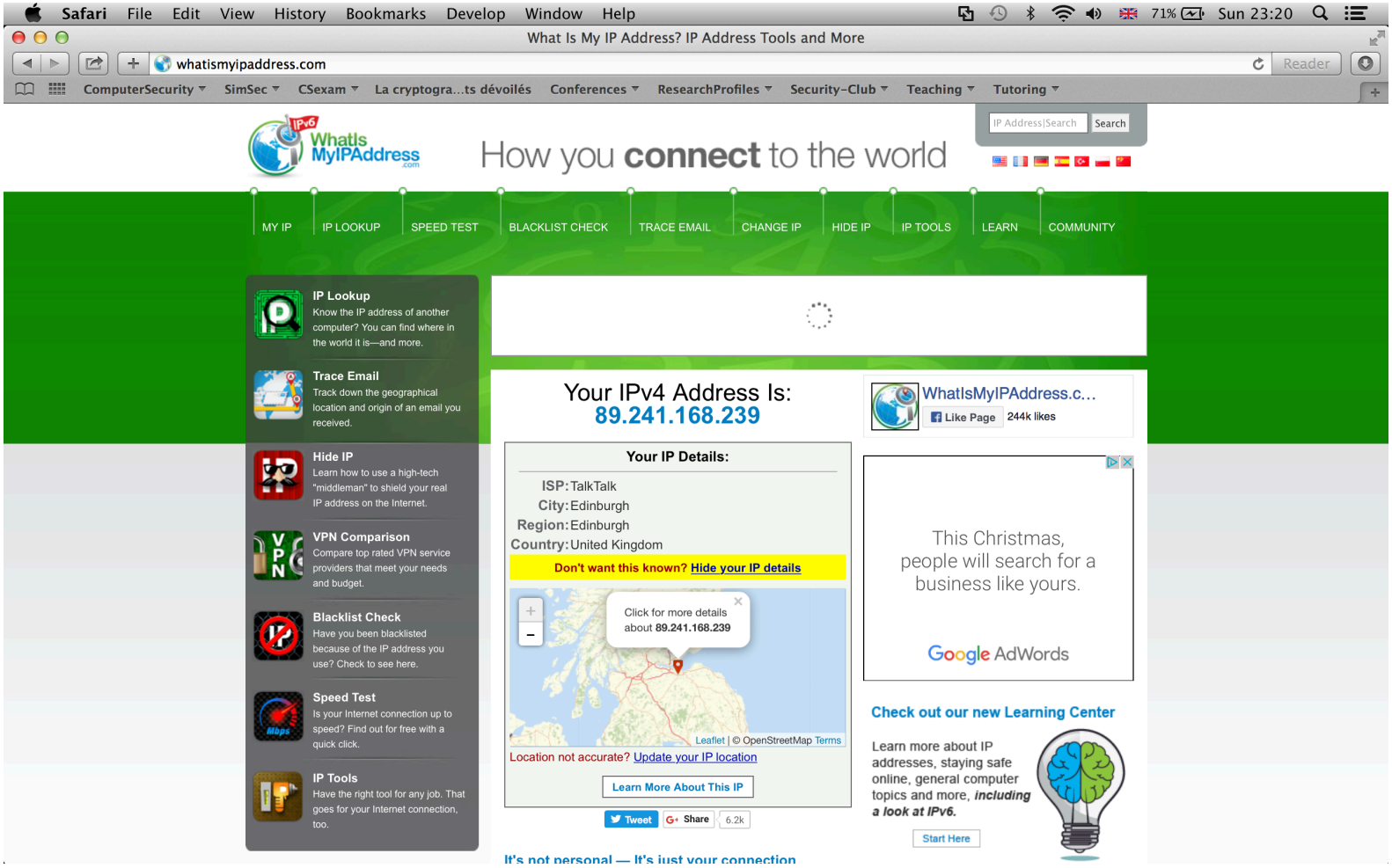
Routing information can reveal who you are!



Routing information can reveal who you are!



Routing information can reveal who you are!



Routing information can reveal who you are!

"With your permission, you give us more information about you, about your friends, and we can improve the quality of your searches. We don't need you to type at all. We know where you are. We know where you've been. We can more or less know what you're thinking about."

Eric Schmidt, CEO Google, 2010

Your IP address is your ID

Your IP address leaves behind digital tracks that can be used to identify you and invade your privacy

The McNealy argument

"You have zero privacy anyway. Get over it"
Scott McNealy, CEO Sun Microsystems, 1999

The Schmidt argument

"If you have something that you don't want anyone to know maybe you shouldn't be doing it in the first place"

Eric Schmidt, CEO Google, 2009

Anonymity

Definition (ISO/IEC standard 15408)

A user may use a service or resource without disclosing the users identity.

Anonymity

Definition (ISO/IEC standard 15408)

A user may use a service or resource without disclosing the users identity.

$\longrightarrow$ this can be achieved by hiding one's activities among others' similar activities

- Dinning cryptographers
- Crowds
- Chaum's mix
- Onion routing

Three-party dinning cryptographers (3DC) protocol

Three cryptographers are having dinner. Either NSA paid for the dinner, or one of the cryptographers. They want to know if it is the NSA that paid, but without revealing the identity of the cryptographer that paid in the case the NSA did not pay.

3DC protocol:

1. Each cryptographer flips a coin and shows it to his left neighbor:
   - each cryptographer will see his own coin and his right neighbor's
2. Each cryptographer announces whether the two coins he saw are the same. If he is the payer, he lies
3. odd number of "same" $\Rightarrow$ the NSA paid
   even number of "same" $\Rightarrow$ one of the cryptographers paid
   - only the payer knows he is the one who paid

Superposed sending

- 3DC protocol generalises to any group size n (nDC)
- Sender wants to anonymously broadcast a message m:
  1. for each bit of the m, every user generates a random bit and sends it to his left neighbor
     - every user learns two bits: his own, and his right neighbor's
  2. each user (except the sender) announces (own_bit XOR neighbor's_bit)
  3. the sender announces (own_bit XOR neighbor's_bit XOR message_bit)
  4. XOR of all announcements = message_bit
     - every randomly generated bit occurs in this sum twice (and is canceled by XOR)
     - message_bit occurs only once

Limitations of the DC protocol

The DC protocol is impractical:

- Requires pair-wise shared secret keys (secure channels) between the participants (to share random bits)
- Requires large amounts of randomness

Crowds

[M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions". ACM Transactions on Information and System Security.]

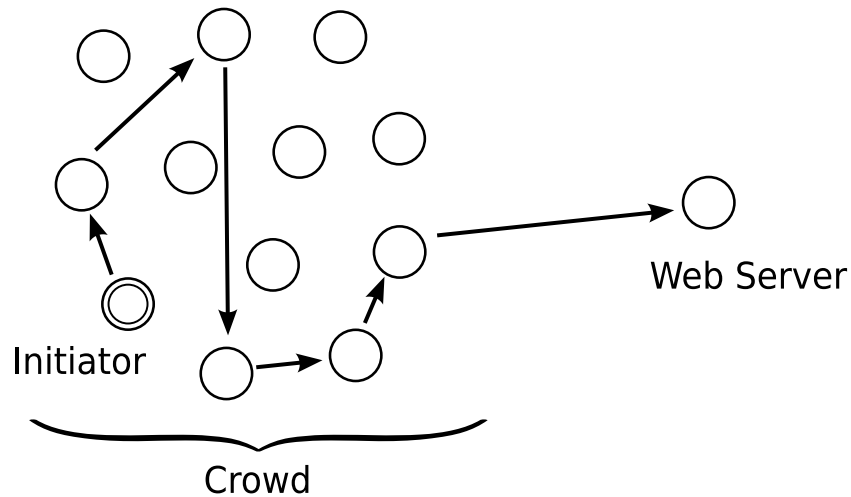
Idea: randomly route the request through a crowd of users

Crowds

[M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions". ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of m users; c out of m users may be corrupted

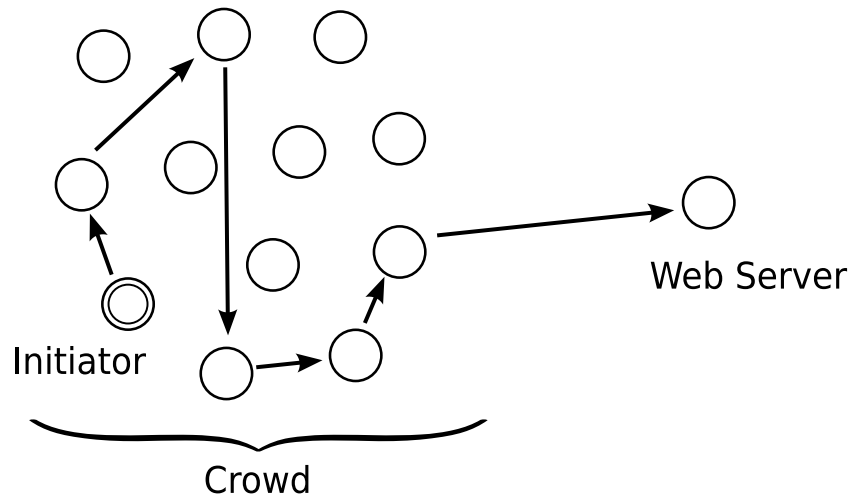


Crowds

[M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions". ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of m users; c out of m users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:

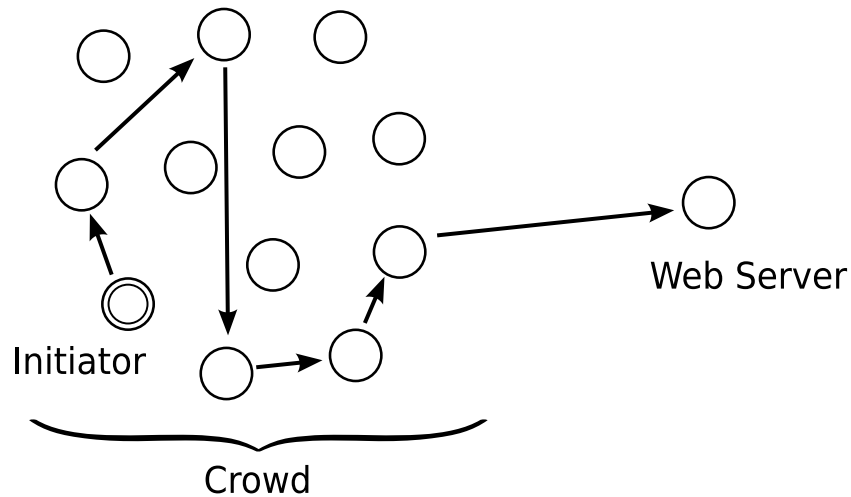


Crowds

[M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions". ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of m users; c out of m users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:
  1. the initiator selects a forwarder from the crowd and sends him his request

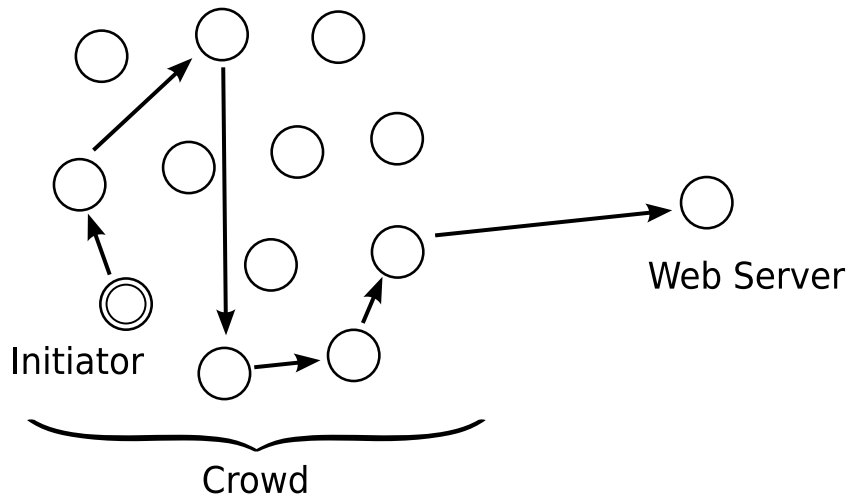


Crowds

[M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions". ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of m users; c out of m users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:
  1. the initiator selects a forwarder from the crowd and sends him his request
  2. a forwarder delivers the request directly to the server with probability $1 - p_f$; he forwards the request to a randomly selected new forwarder from the crowd with probability p_f; the new forwarder repeats the procedure

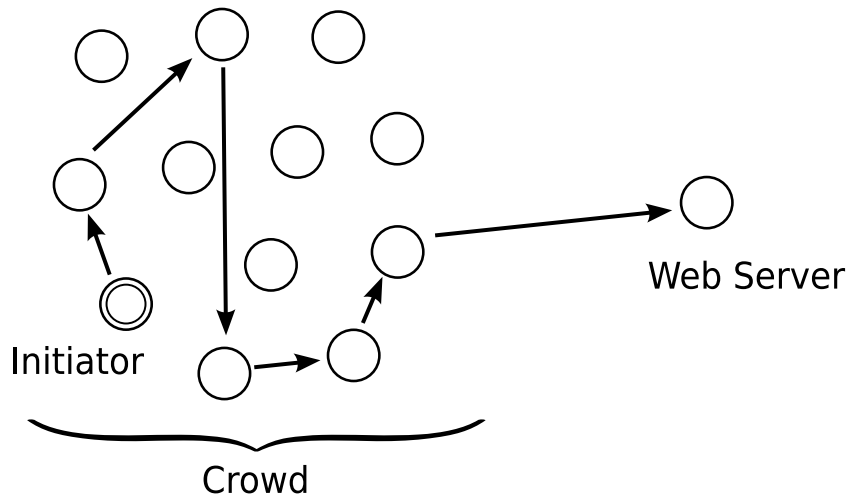


Crowds

[M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions". ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of m users; c out of m users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:
  1. the initiator selects a forwarder from the crowd and sends him his request
  2. a forwarder delivers the request directly to the server with probability $1 - p_f$; he forwards the request to a randomly selected new forwarder from the crowd with probability p_f; the new forwarder repeats the procedure
  3. the response from the server follows same route in opposite direction

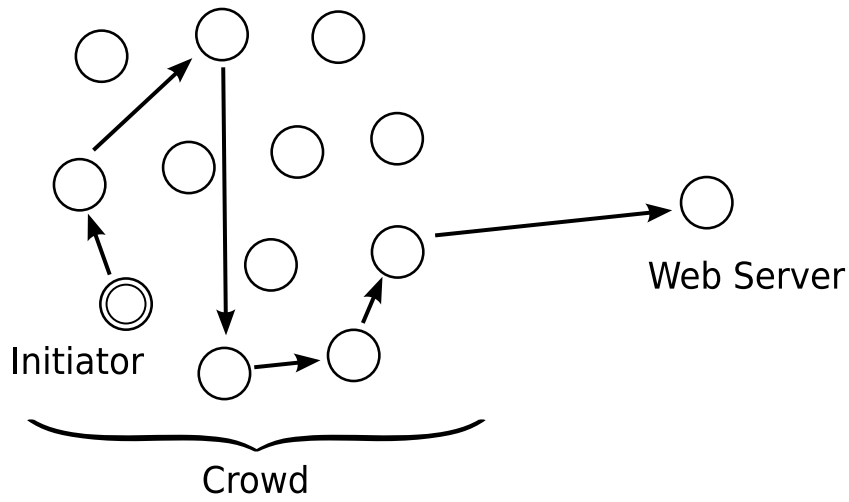


Crowds

[M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions". ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

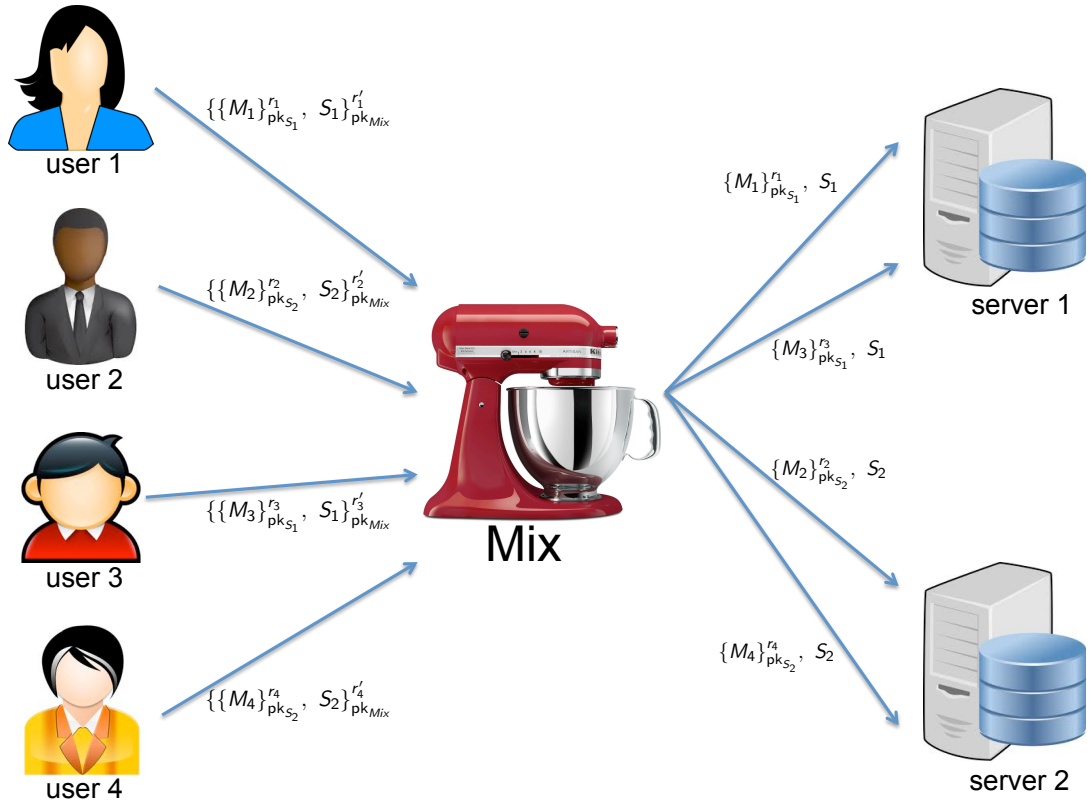
- a crowd is a group of m users; c out of m users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:
  1. the initiator selects a forwarder from the crowd and sends him his request
  2. a forwarder delivers the request directly to the server with probability $1 - p_f$; he forwards the request to a randomly selected new forwarder from the crowd with probability p_f; the new forwarder repeats the procedure
  3. the response from the server follows same route in opposite direction



Crowd IS NOT resistant against an attacker that sees the whole network traffic!

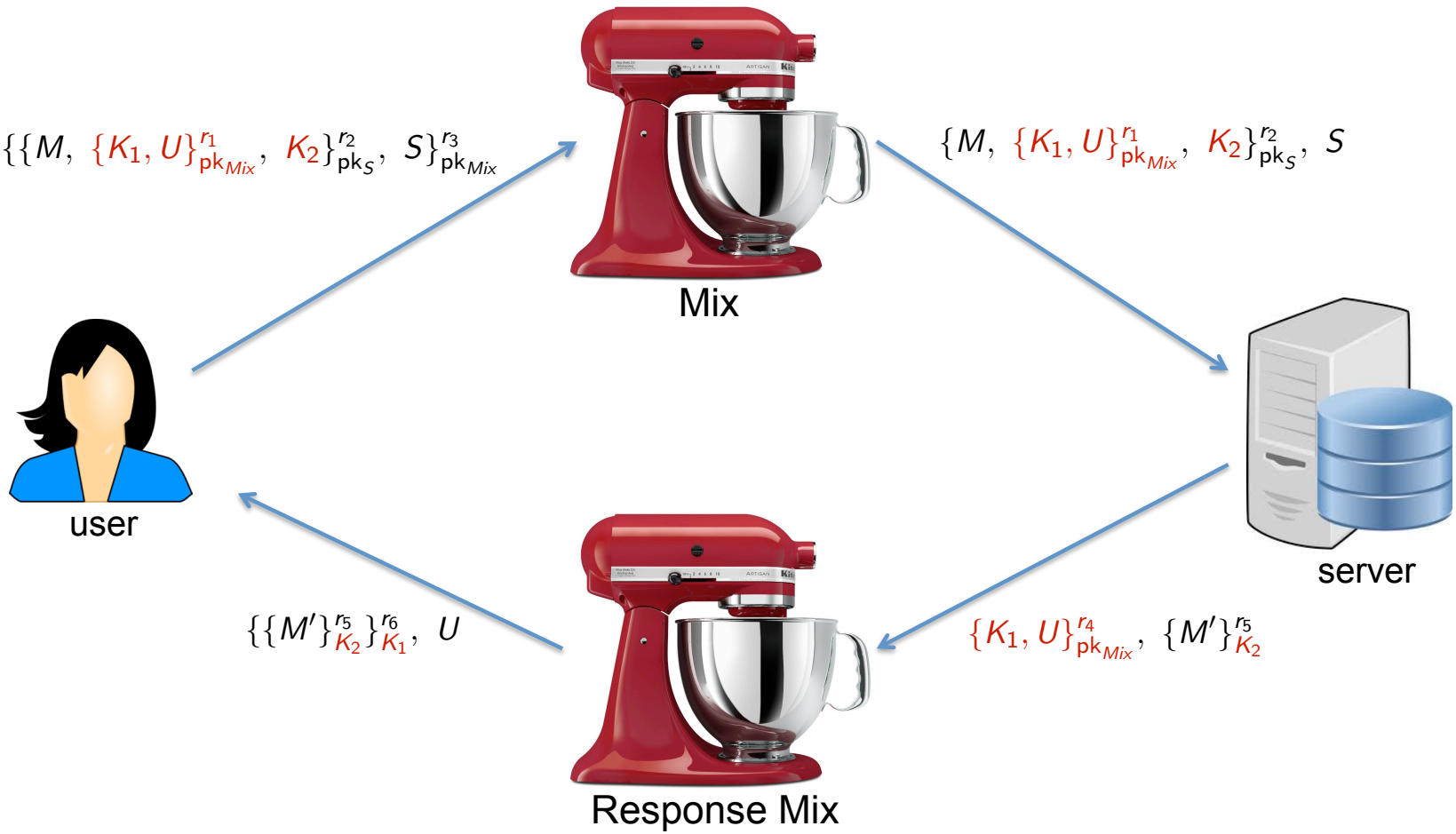
Chaum's mix

[D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", Communications of the ACM, February 1981.]

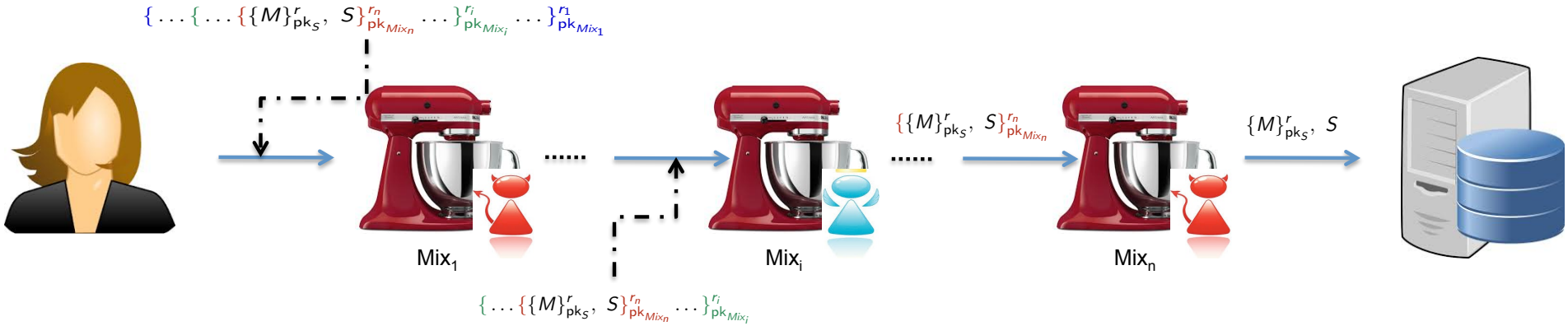


- message padding and buffering to avoid time correlation attacks
- dummy messages are generated by the mixes themselves to prevent an attacker sending $n-1$ messages to a mix with capacity n, allowing him to then link the sender of the n^{th} message with its recipient

Anonymous return addresses



Mix cascade



- messages are sent through a sequence of mixes
- some of the mixes may be corrupted
- a single honnest mix guarantees anonymity against an attacker controlling the whole network provided it applies:
  - message padding
  - buffering
  - dummy messages

Limitations of Chaum's mixnets

- Asymmetric encryption is not efficitent
- Dummy messages are innefficient
- Buffering is not efficient

Onion routing

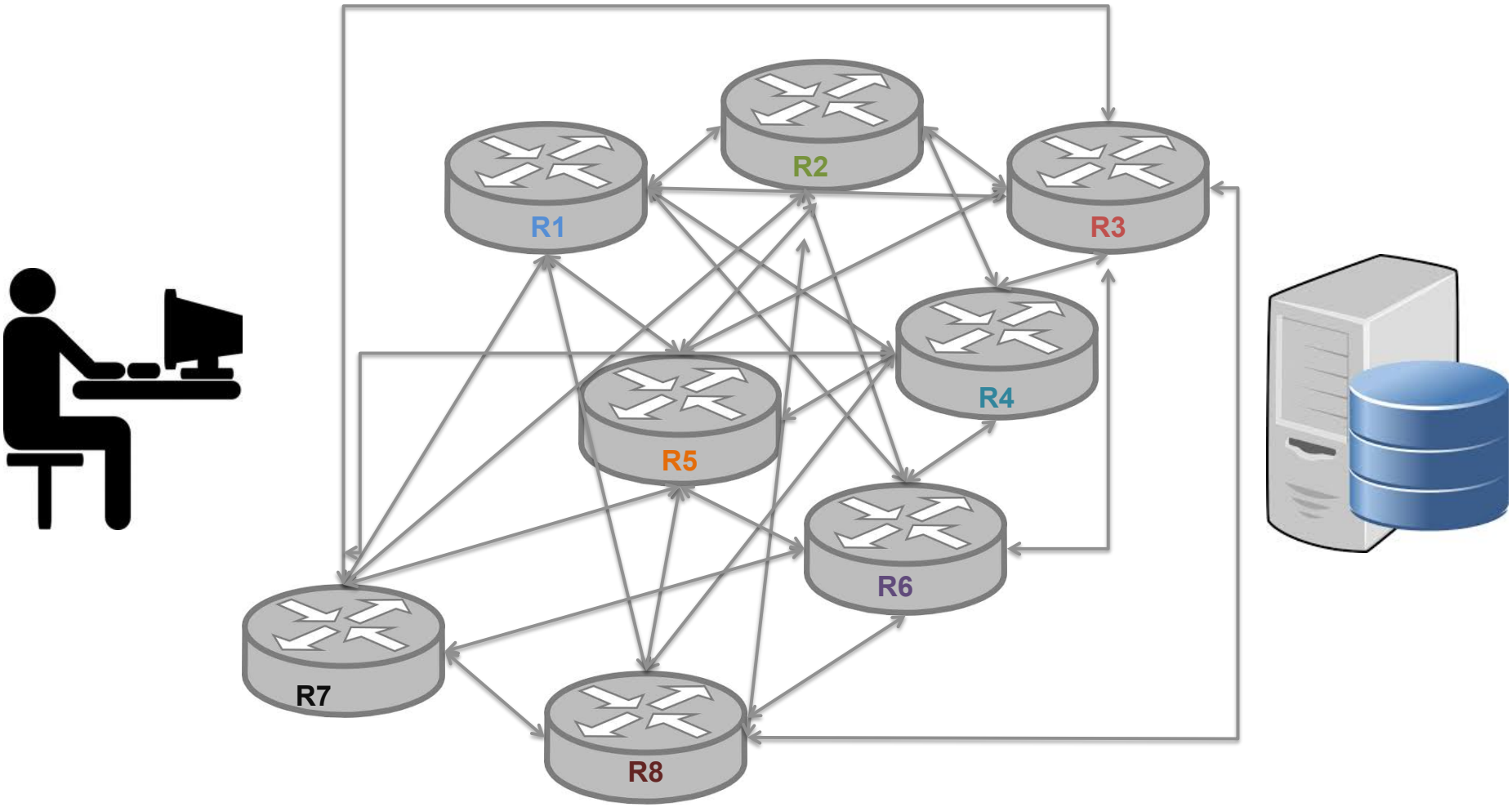
[R. Dingledine, N. Mathewson, and P. F. Syverson: "Tor: The Second-Generation Onion Router", USENIX Security Symposium 2004]

Idea: combine advantages of mixes and proxies

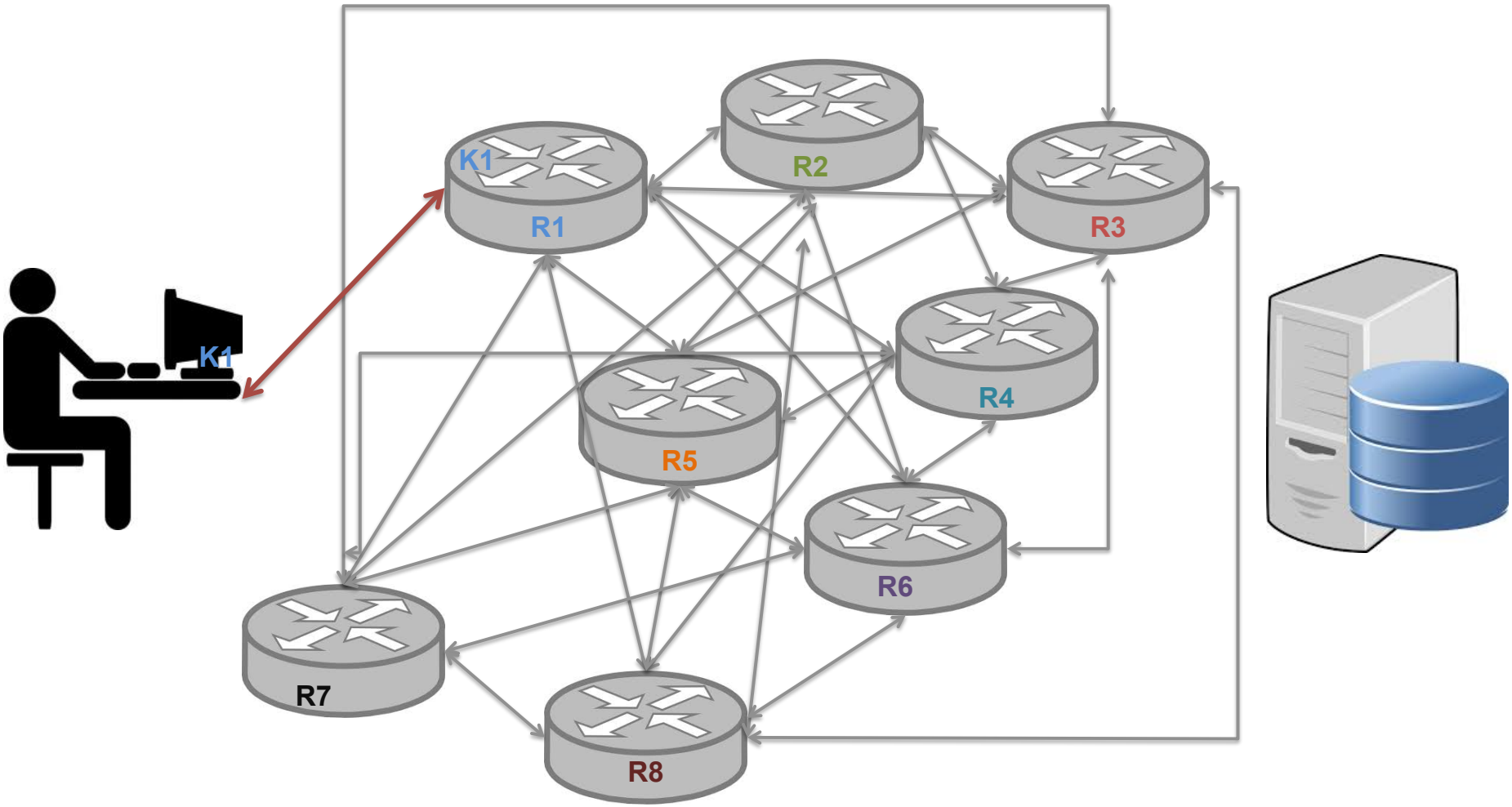
- use public-key crypto only to establish circuit
- use symmetric-key crypto to exchange data
- distribute trust like mixes

But does not defend against attackers that controle the hole network

TOR circuit setup

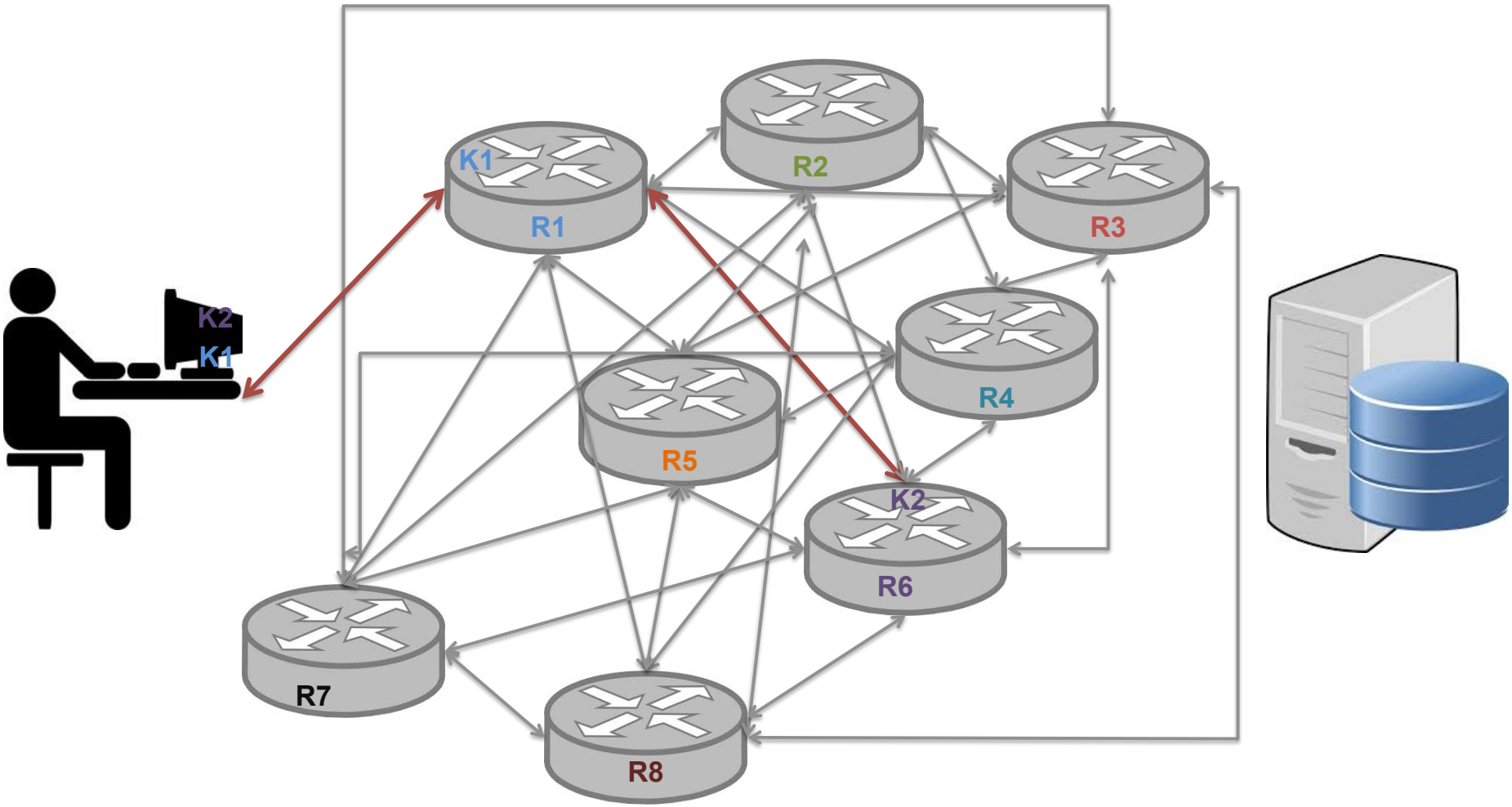


TOR circuit setup



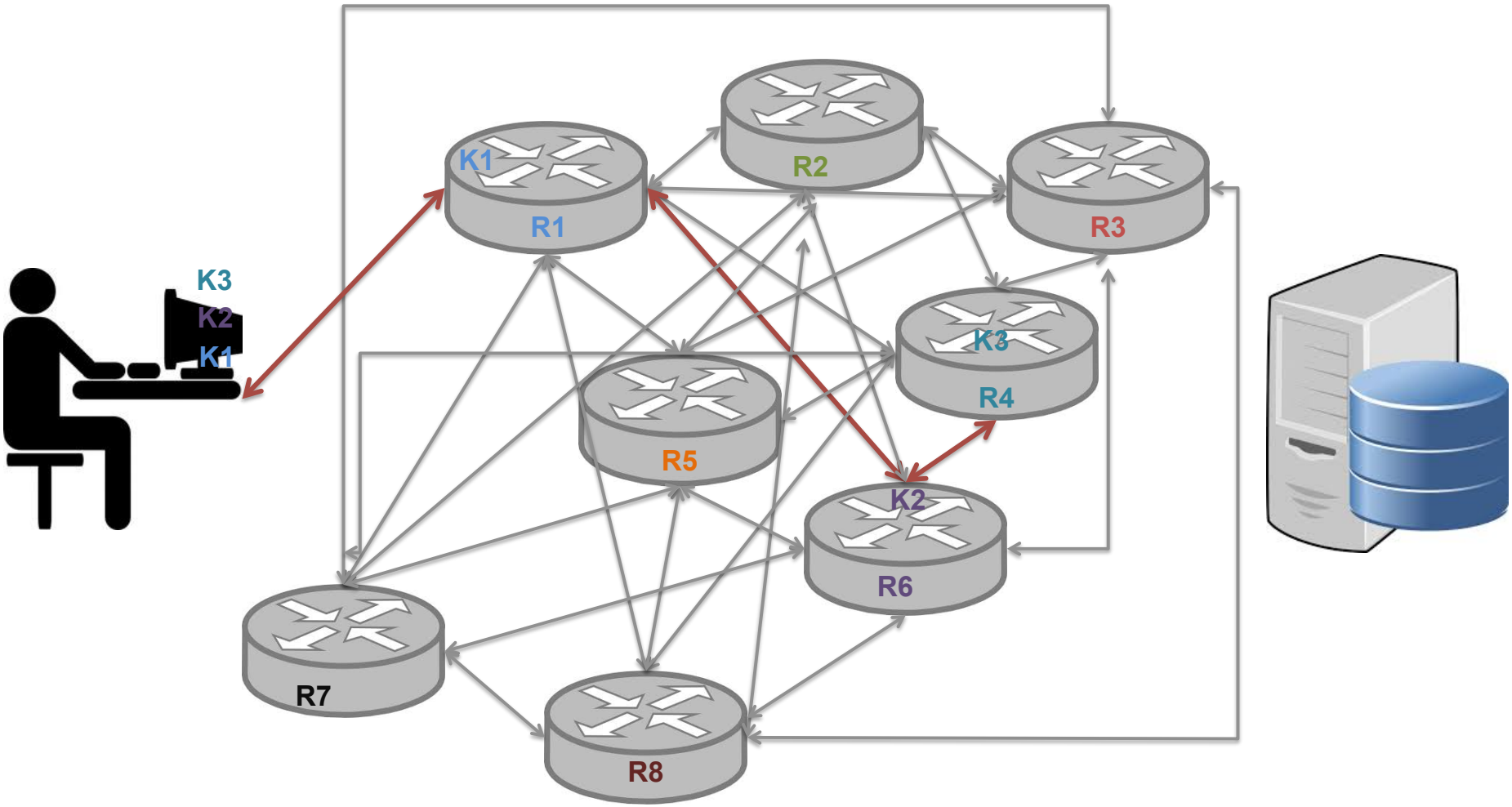
- client establishes session key **K1** and circuit with Onion Router **R1**

TOR circuit setup



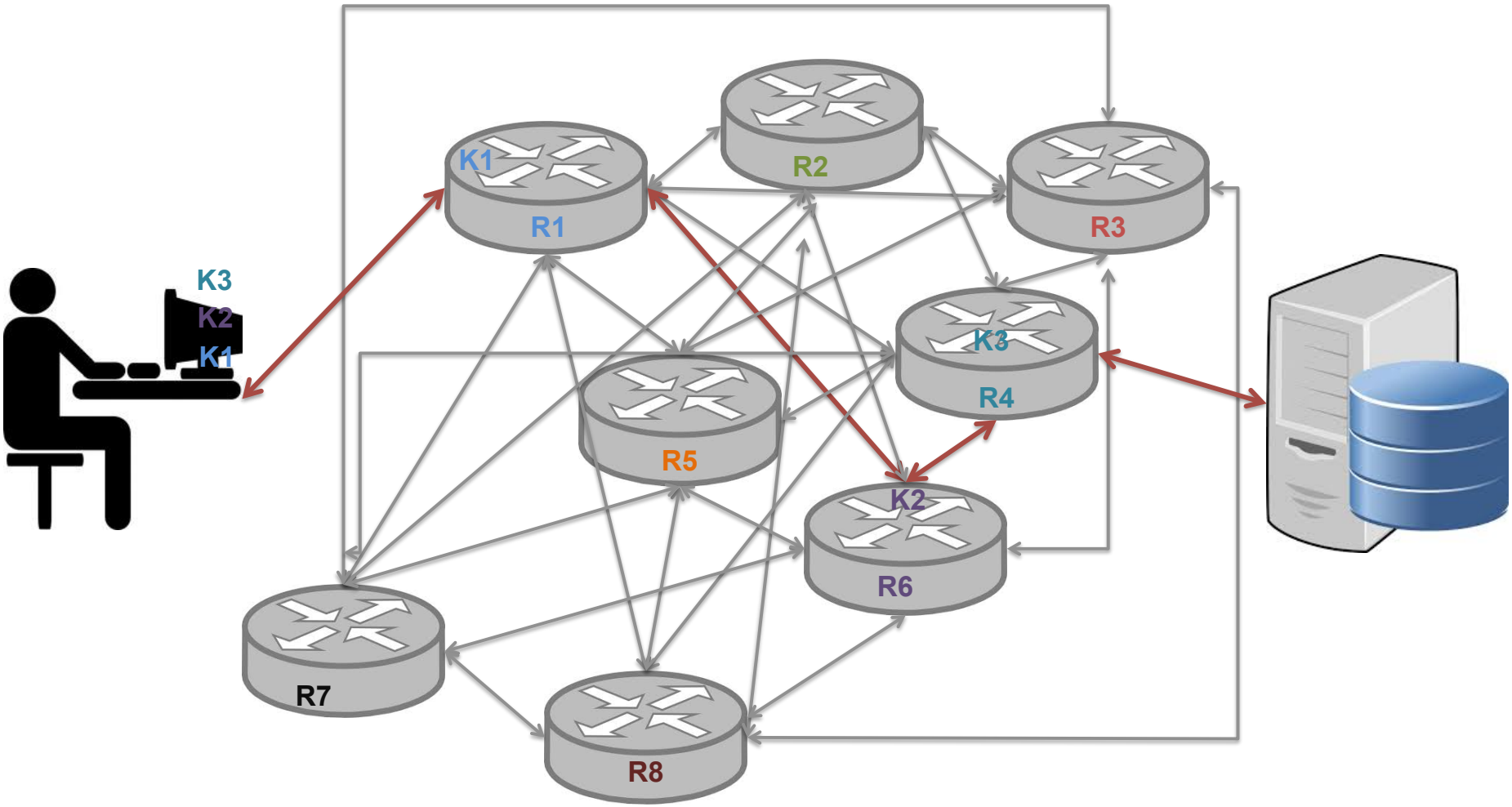
- client tunnels through that circuit to extend to Onion Router **R6**

TOR circuit setup



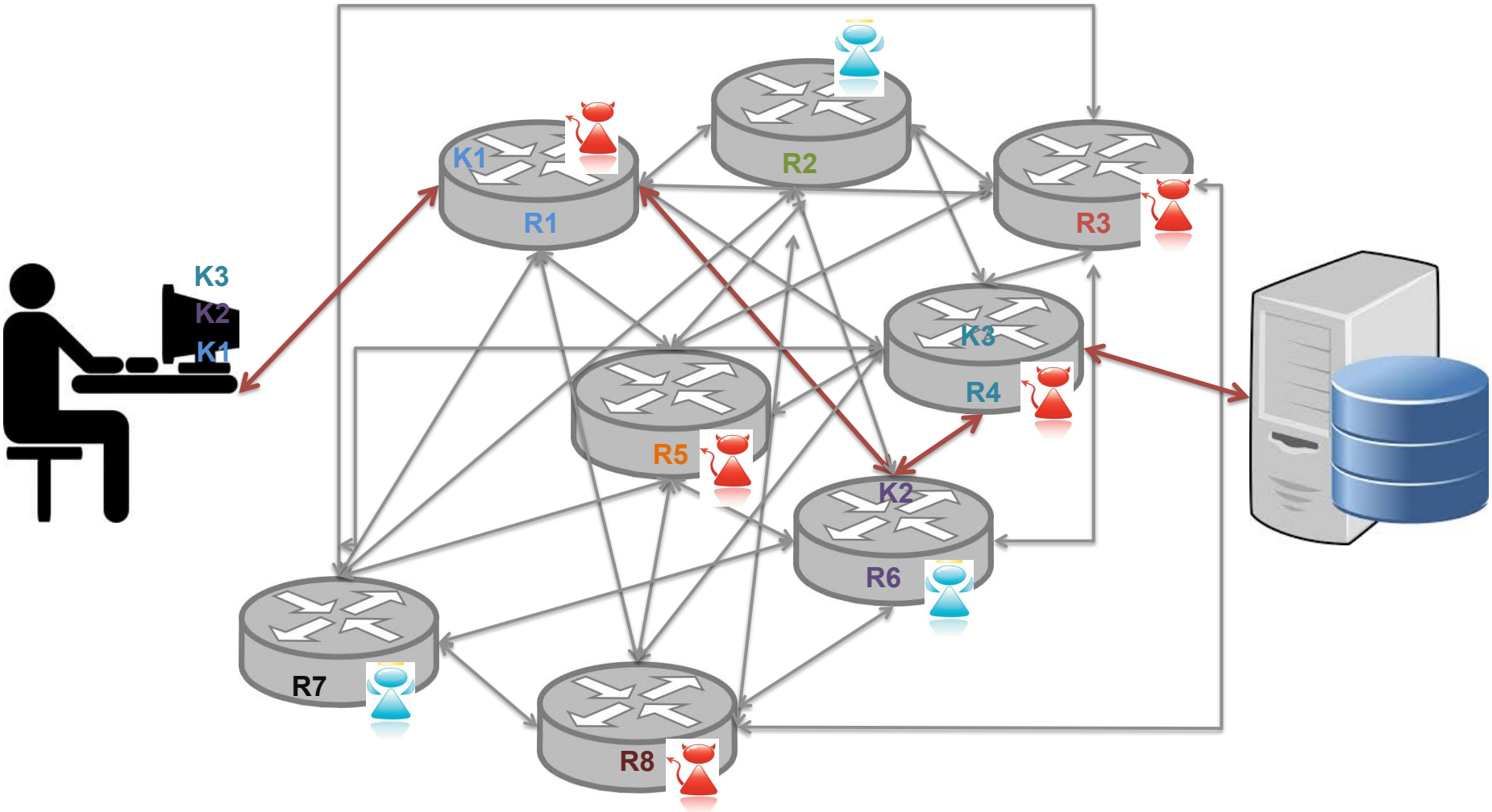
- client tunnels through that extended circuit to extend to Onion Router **R4**

TOR circuit setup



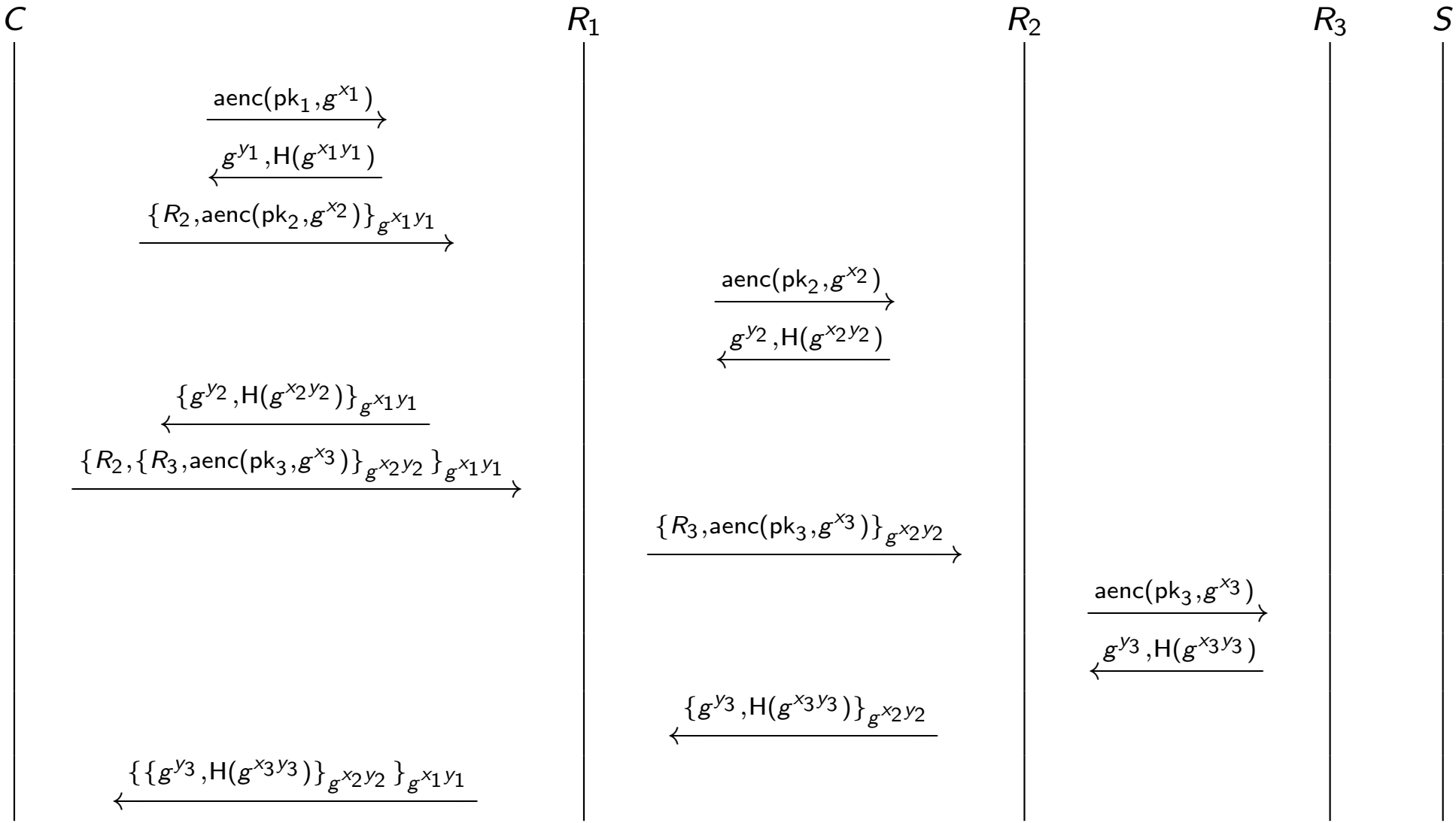
- client applications connect and communicate of established TOR circuit

TOR circuit setup

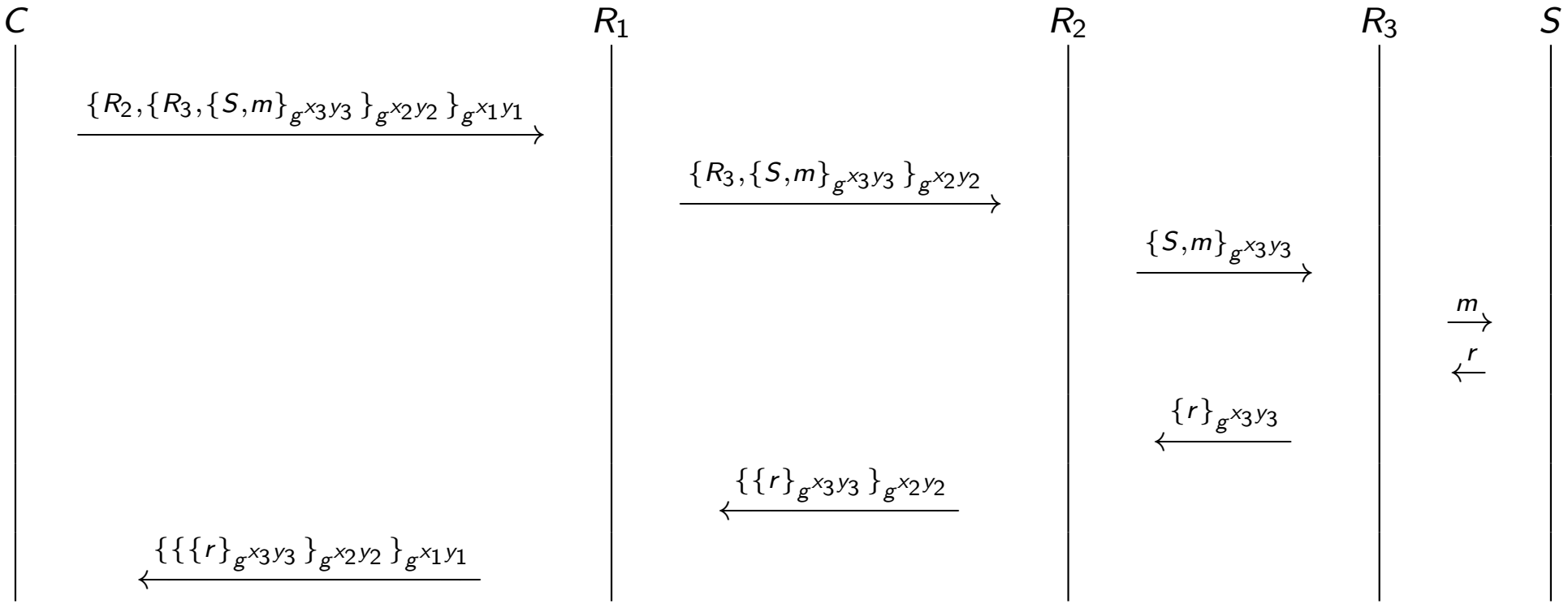


a single honnest Onion Router on the TOR circuit guarantees anonymity against an attacker controlling some Onion Routers

The (simplified) TOR message flow - circuit setup



The (simplified) TOR message flow - actual communication



TOR only provides privacy - not confidentiality

- TOR anonymises the origin of the traffic
- TOR encrypts everything inside the TOR network
- but TOR DOES NOT encrypt all traffic through the Internet
- for confidentiality you still need to use end-to-end encryption such as SSL/TLS

TOR takes care of DNS resolution

- TOR only anonymises TCP streams
- But, DNS resolution is executed over UDP
- So, DNS resolution if handled by the client browser defeats the purpose of using TOR
- To avoid privacy breaches due to DNS resolution, the TOR browser delegates DNS resolution to the exit node

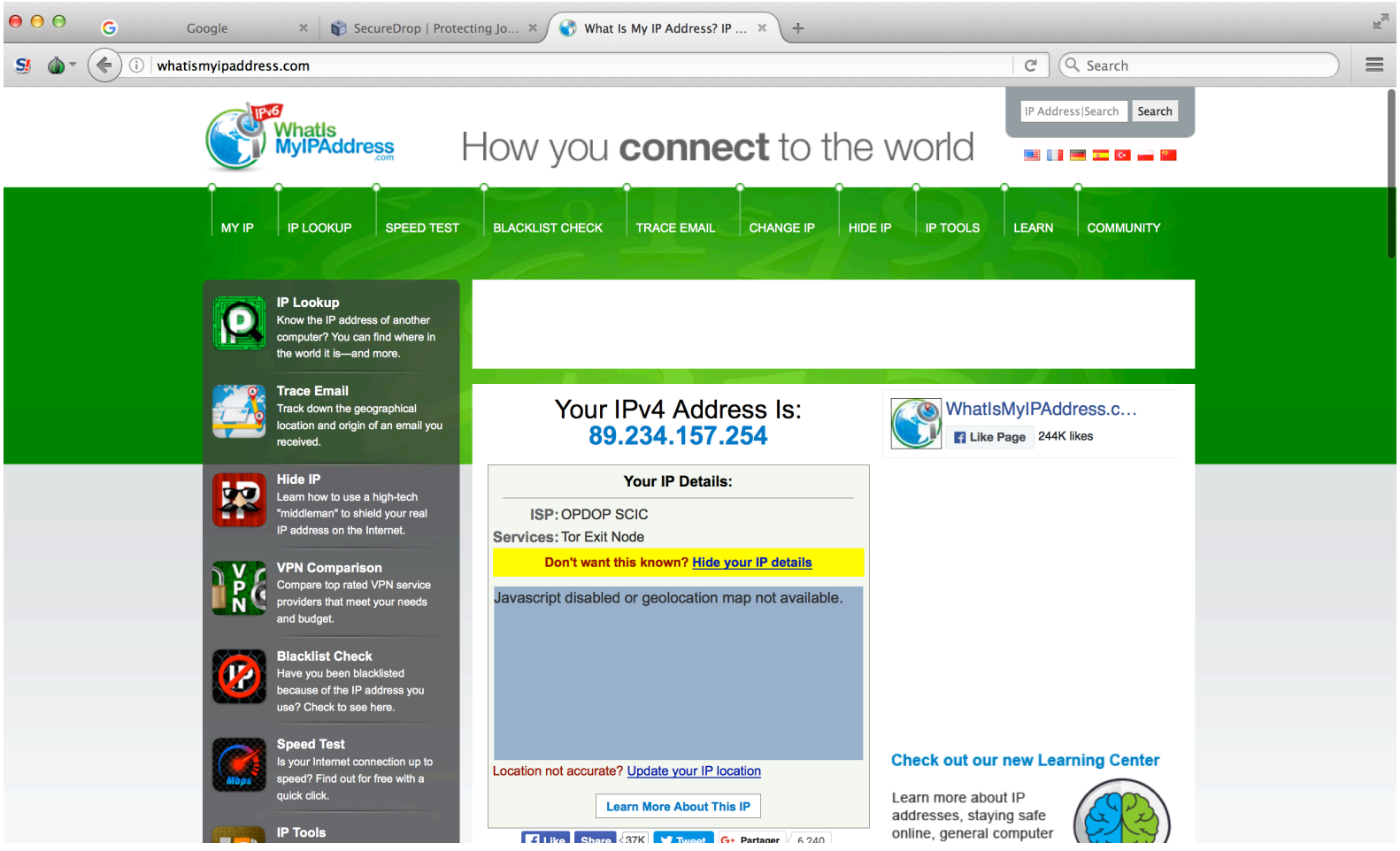
Avoiding censorship

- TOR relays are listed on the public TOR directory
- So your local ISP can observe that you are communicating with TOR nodes
- ISPs and governments can try to block access to the TOR network by blocking TOR relays
- TOR bridge relays are relays not listed on the public TOR directory
- Entering the TOR network thtough a TOR bridge relay can prevent ISPs and governments blocking access to the TOR network

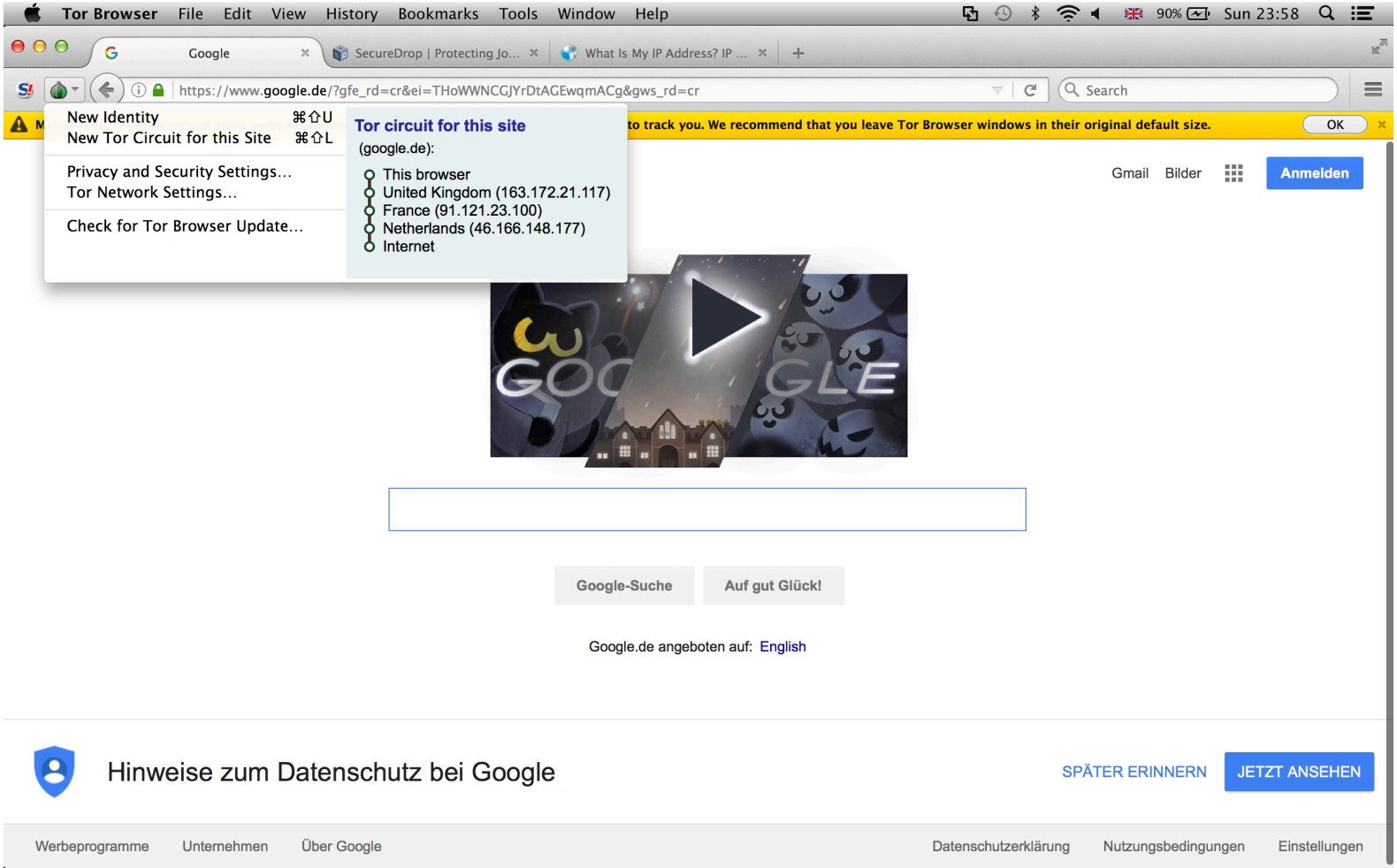
Limitations of TOR

- TOR does not provide protection against end-to-end timing attacks
- If the attacker can see both ends of the communication channel, he can correlate volume and timing information on the two sides

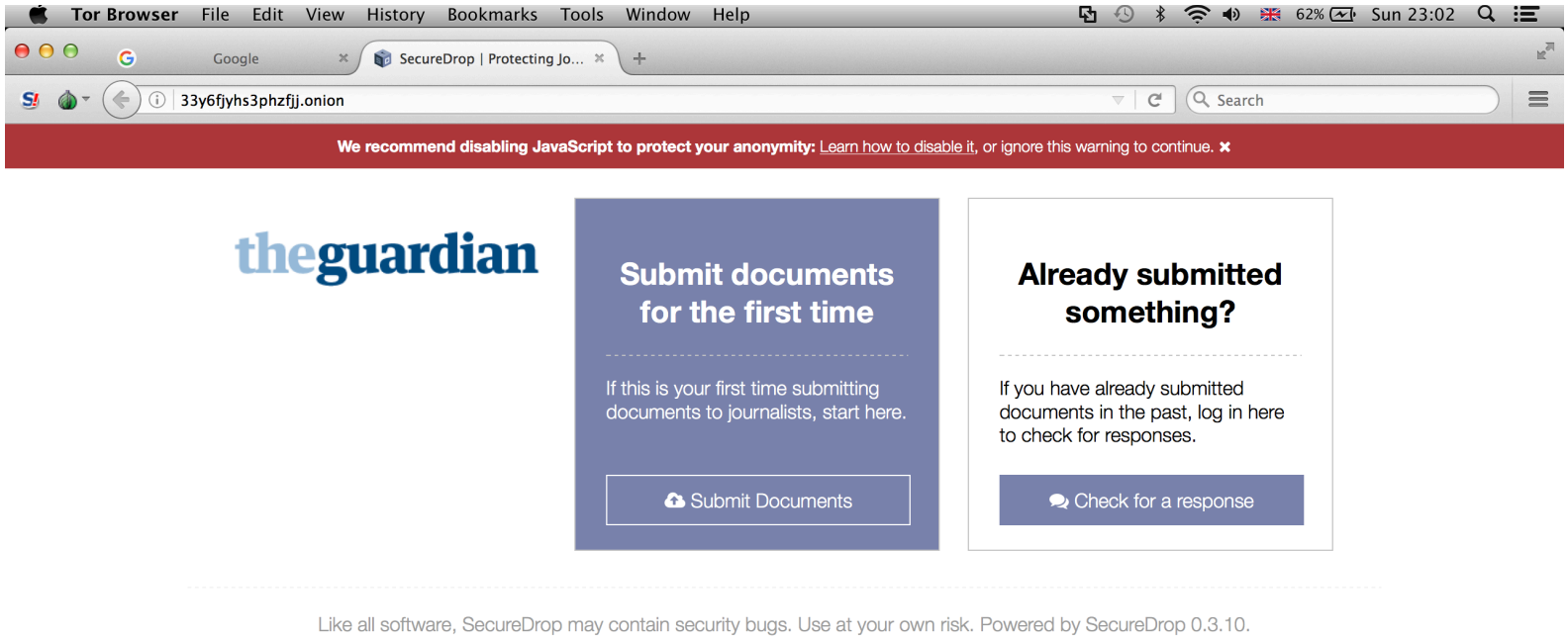
whatismyipaddress.com cannot tell where am I using TOR



google.com thinks I'm in the Netherlands using TOR



TOR hidden services



- TOR can also provide anonymity to websites and servers
- `www.torproject.org/docs/hidden-services.html`