Tor

The Onion Router

Sophie Kuna

@sophiiistika

Basics

How Tor works

Onion services

Implementations

Some Numbers

Basics

- Distributed, anonymous network to send encrypted data across the internet in multiple layers

- Free and open source, developed by *The Tor Project, Inc.*

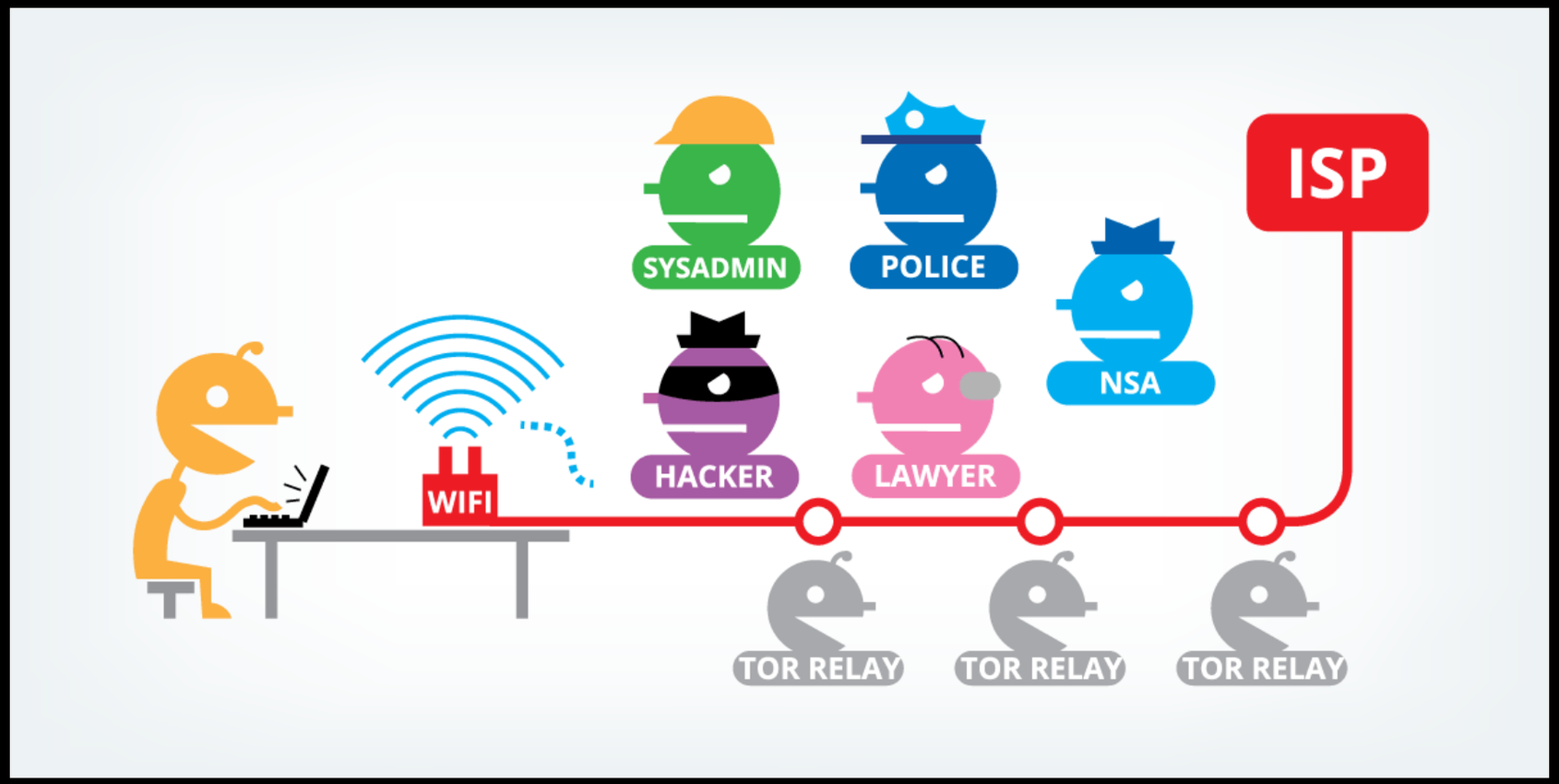
- Goals:

    - Disguise IP-Address

    - Prevent network surveillance and traffic analysis

    >> protection of user’s privacy

Traffic analysis?



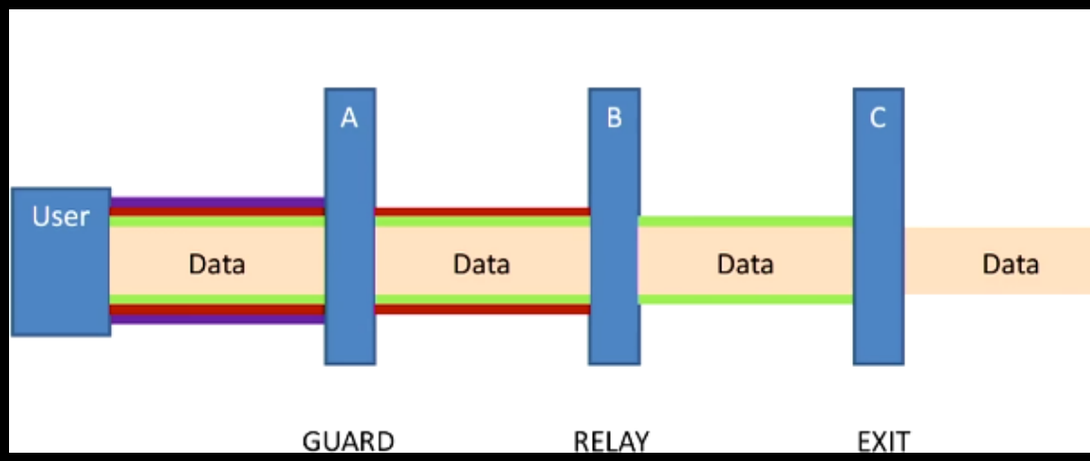
- Internet data packets: data payload and header for routing
- Encryption of data payload alone does not help against attack on communication patterns
- Traffic analysis focuses on the header, which discloses source, destination, size, timing, …

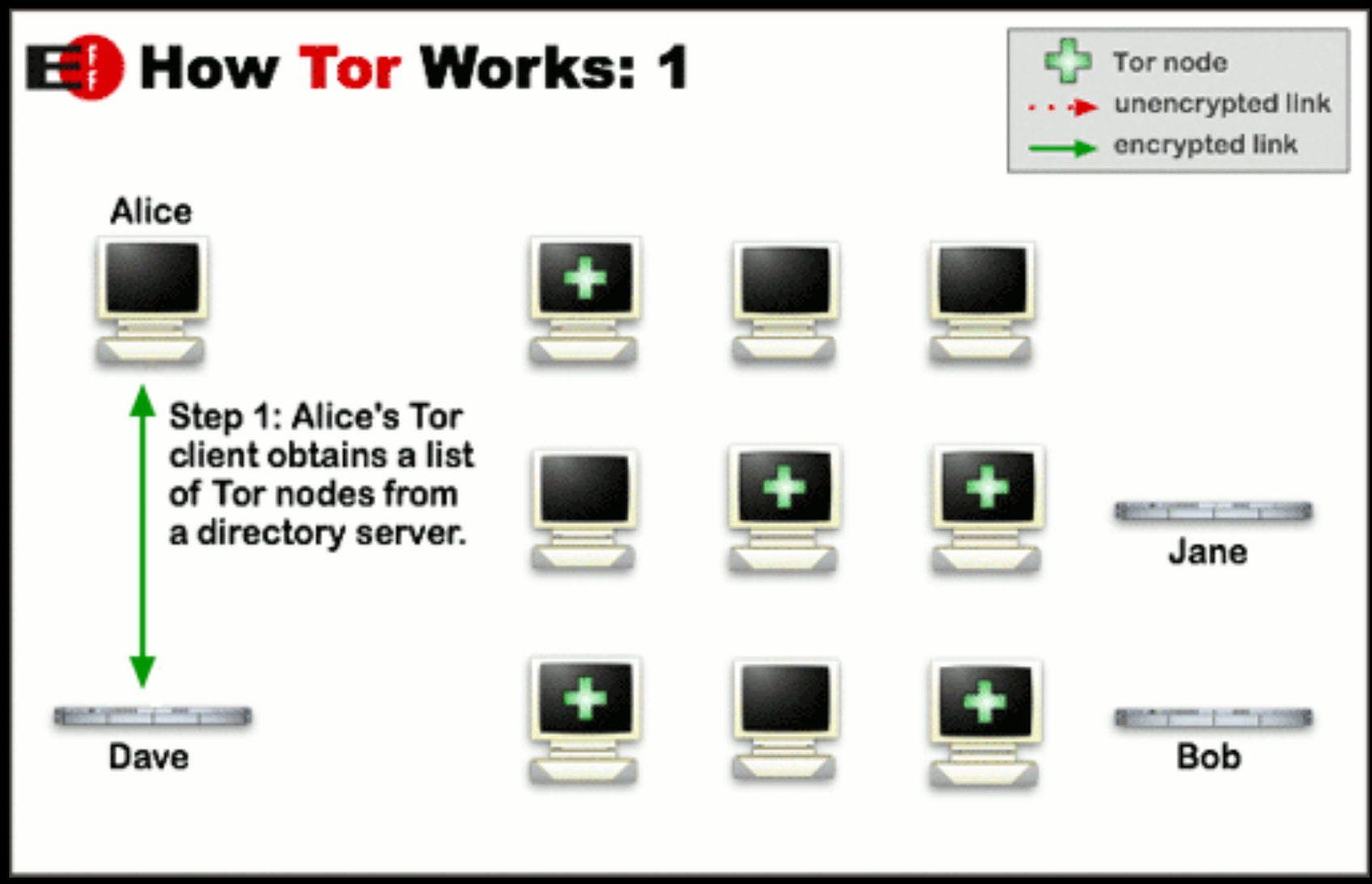
How does Tor work?

- **Idea**: Random pathway through relays that cover tracks
  >> No observer at any single point knows where data came from and where it is going

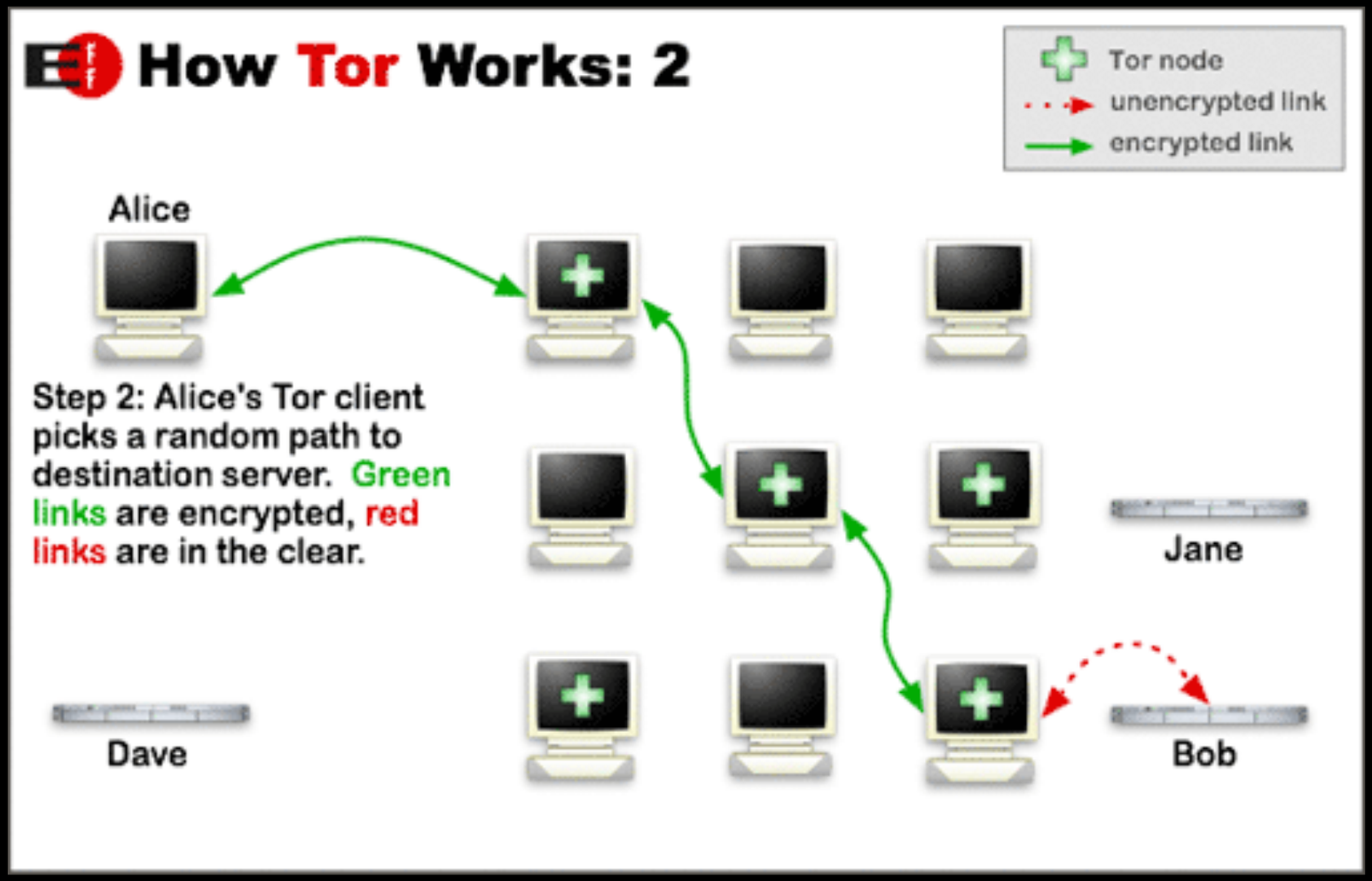
- **Onion routing**: data is encapsulated in layers of encryption and then transmitted through the relays
  >> each „peels“ away a single layer and so uncovers the data’s next destination

- Circuit of encrypted connections

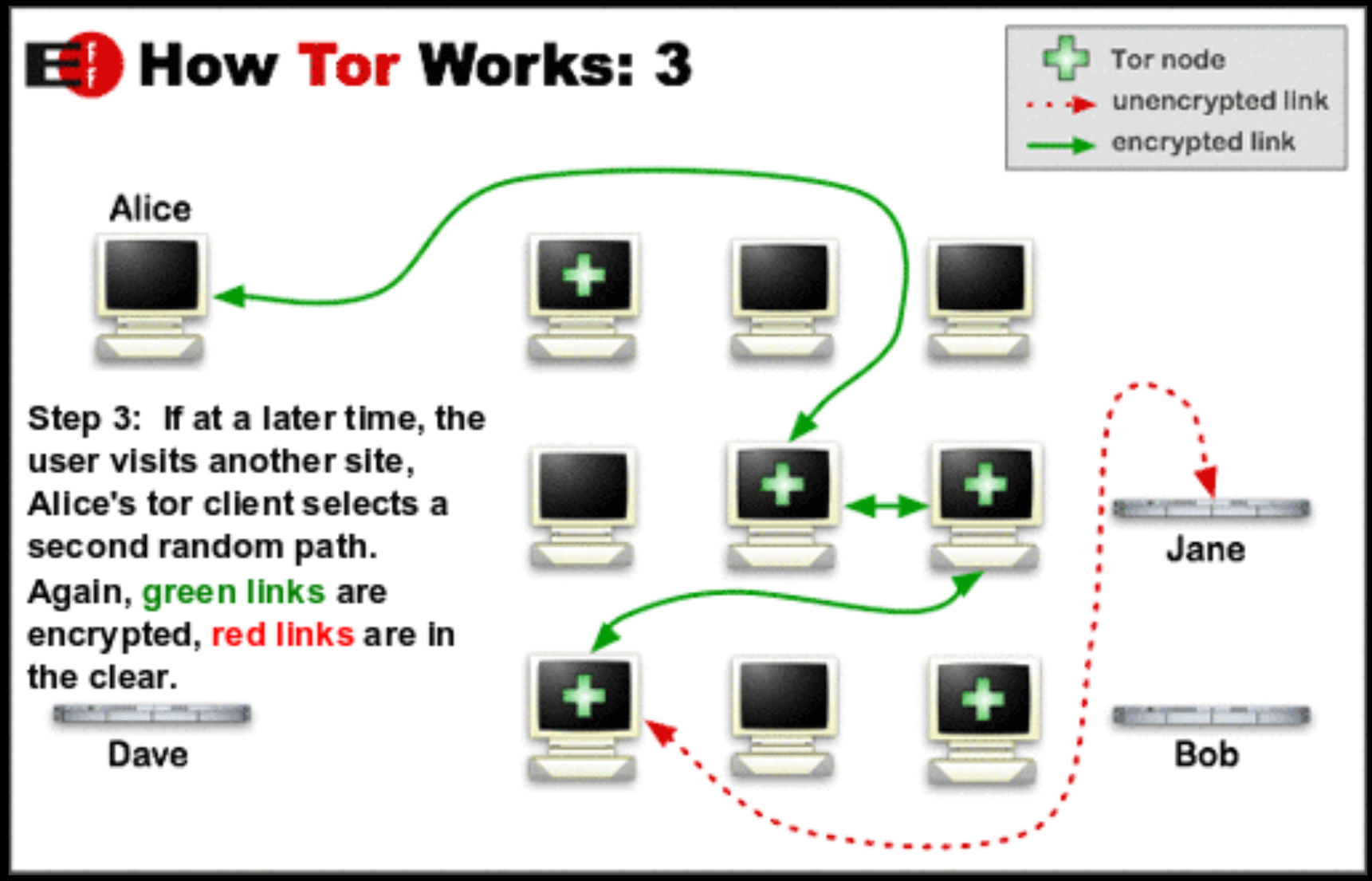




- Client gets list of Tor nodes / relays from directory server



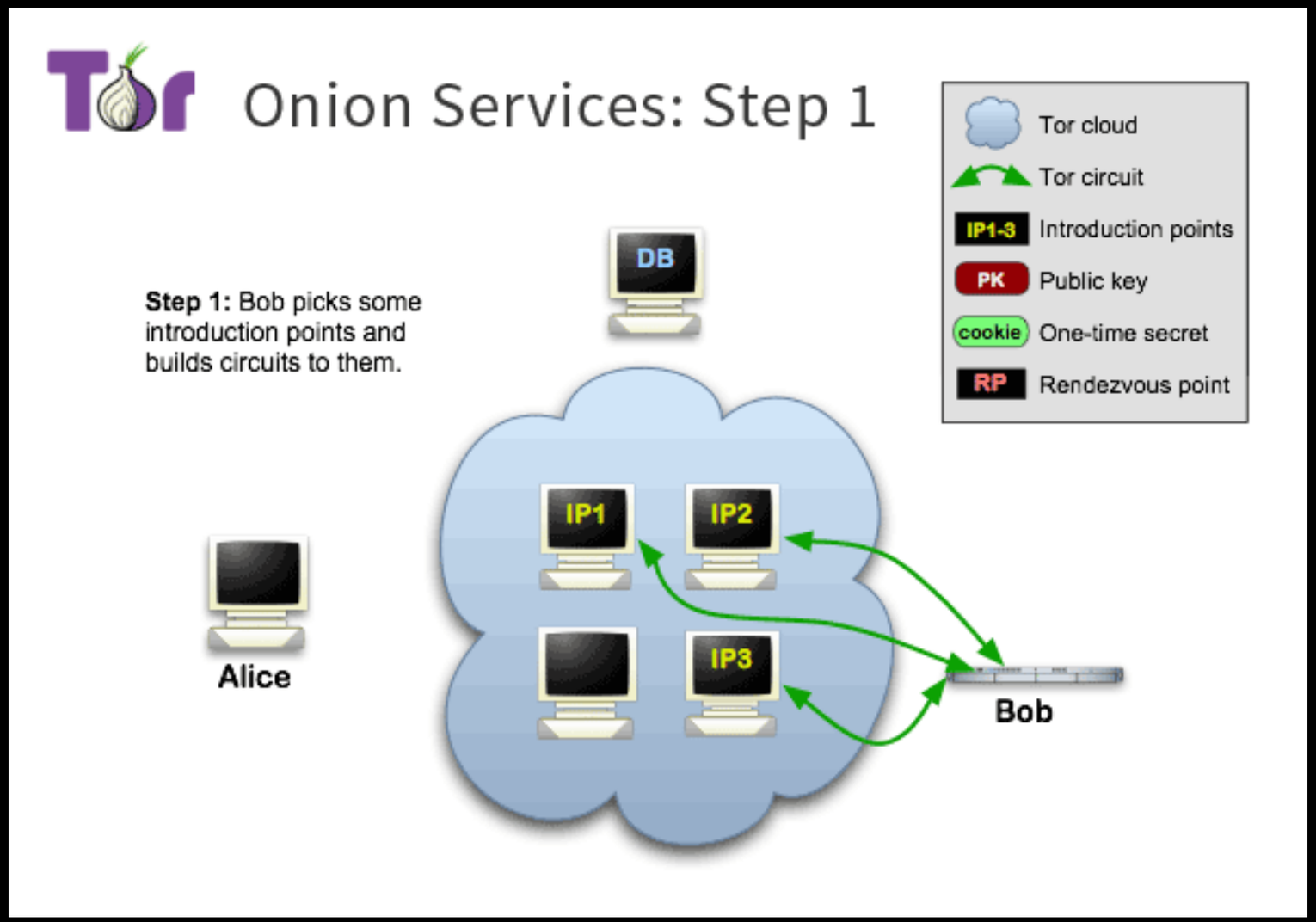
- Selects random pathway through relays
- Builds incrementally a circuit of encrypted connections >> circuit is extended one hop at a time
- One relay knows only the one before it and the one after >> no knot knows everything!



- Same circuit for connections that happen within 10 min
- Traffic from Exit to destination is unencrypted
- TLS-connections; separate set of encryption keys for each hop

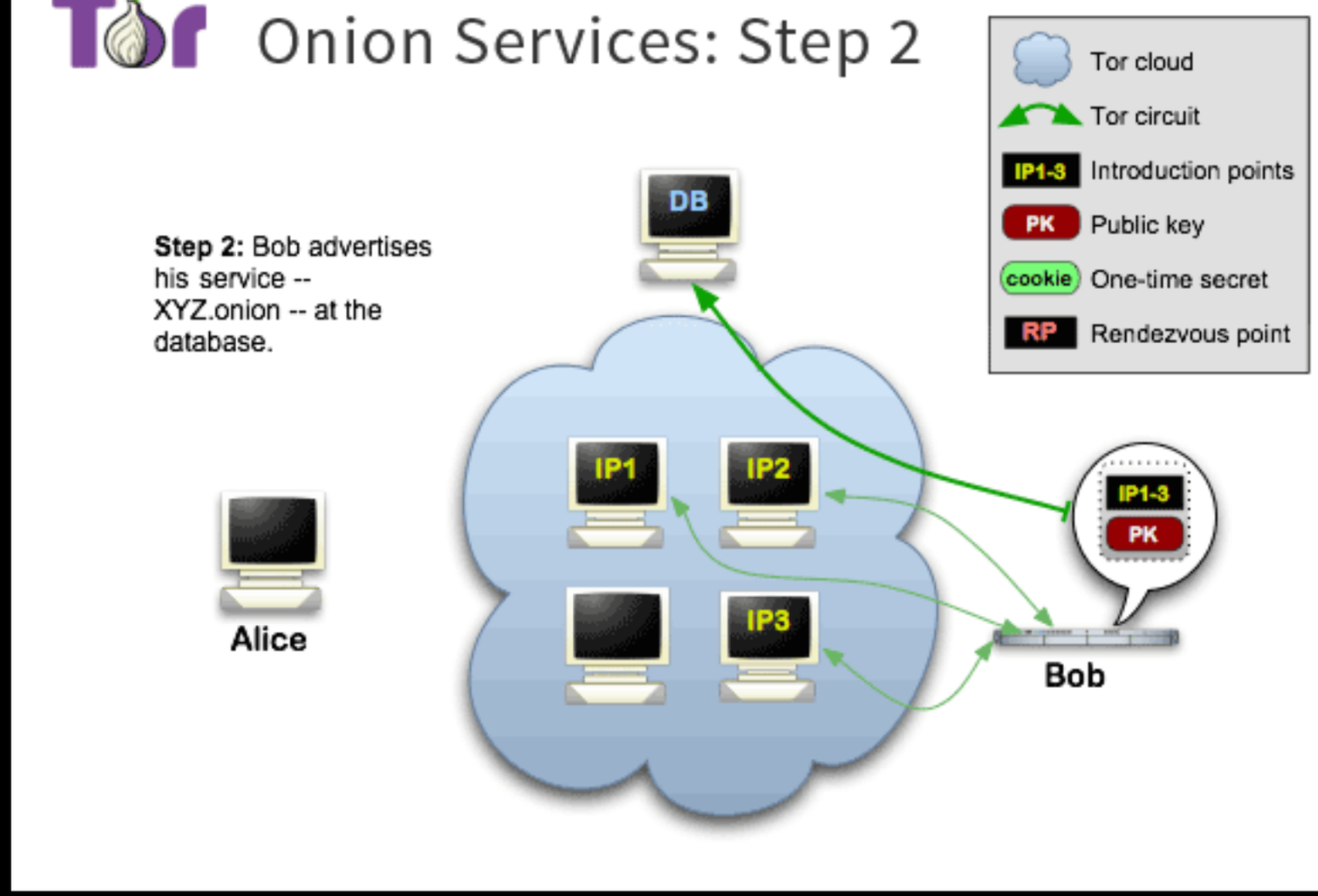
Onion Services

- Former „Hidden Services“
- Not only client but also server is anonymous
- Are only accessible via Tor
- Service gets .onion-address, like facebookcorewwwi.onion



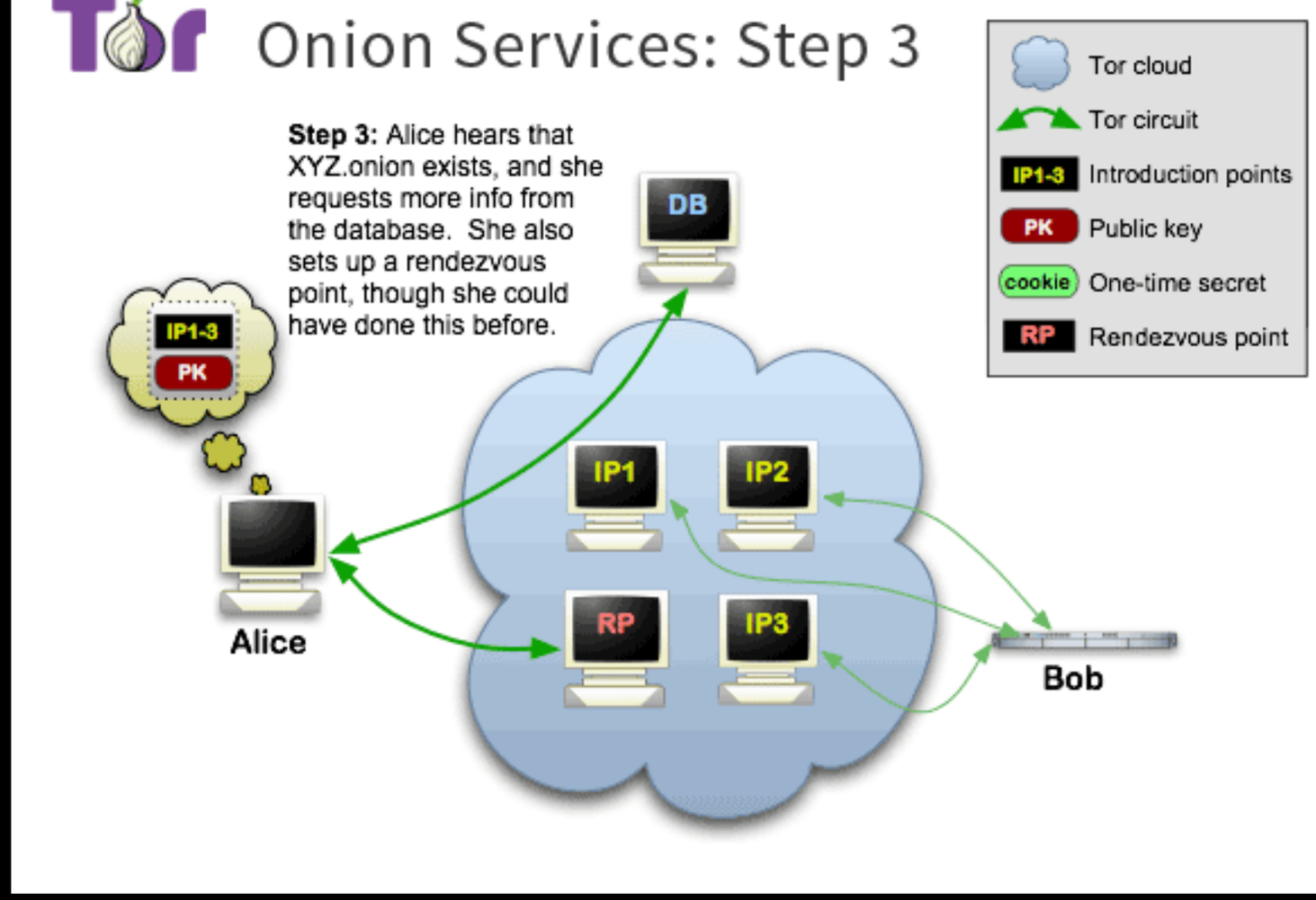
- Service picks introduction points by telling them its public key and builds circuit to them

Onion Services: Step 2



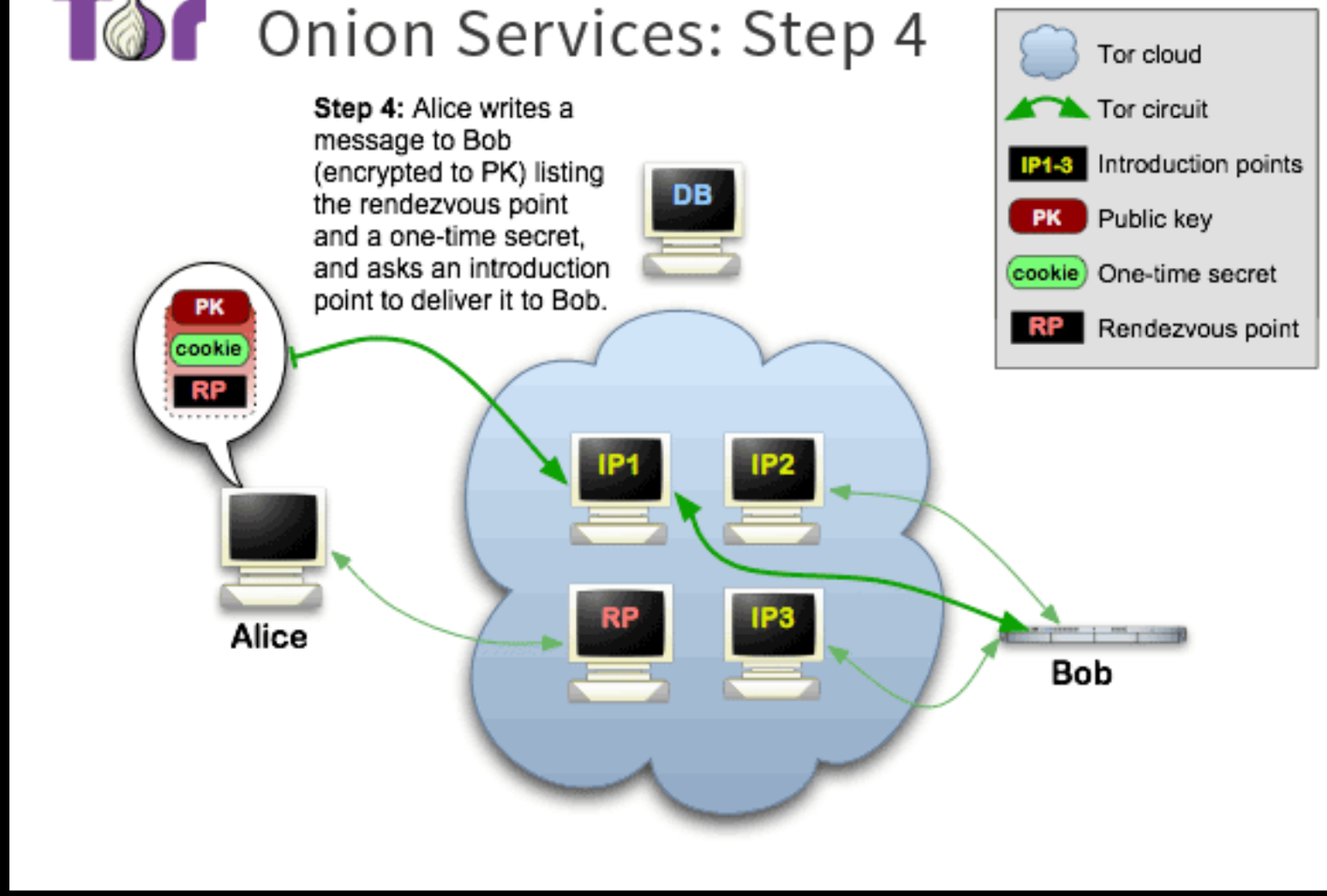
- Service compiles an *onion service descriptor* containing public key and Introduction points, signs it with private key >> uploads descriptor to distributed hash table, gets an **.onion**-address

Onion Services: Step 3



- Client knows set of Introduction points from descriptor
- Chooses Rendezvous point by telling it a one-time-secret

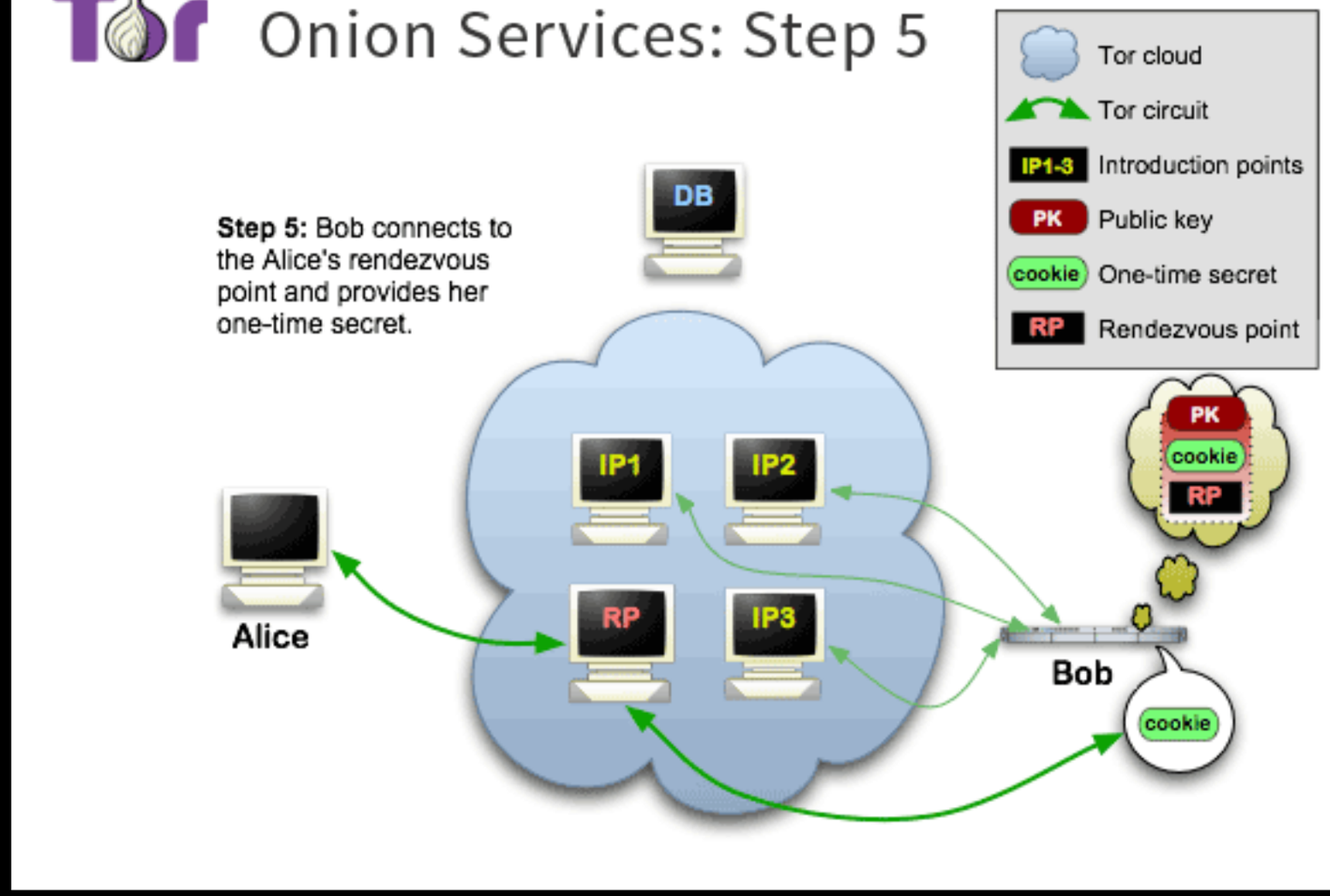
Onion Services: Step 4



- Client sends introduce message (encrypted with onion services public key) containing Rendezvous point and one-time-secret to one Introduction point >> requests to be delivered to onion service

- Client remains anonymous because of circuit

Onion Services: Step 5



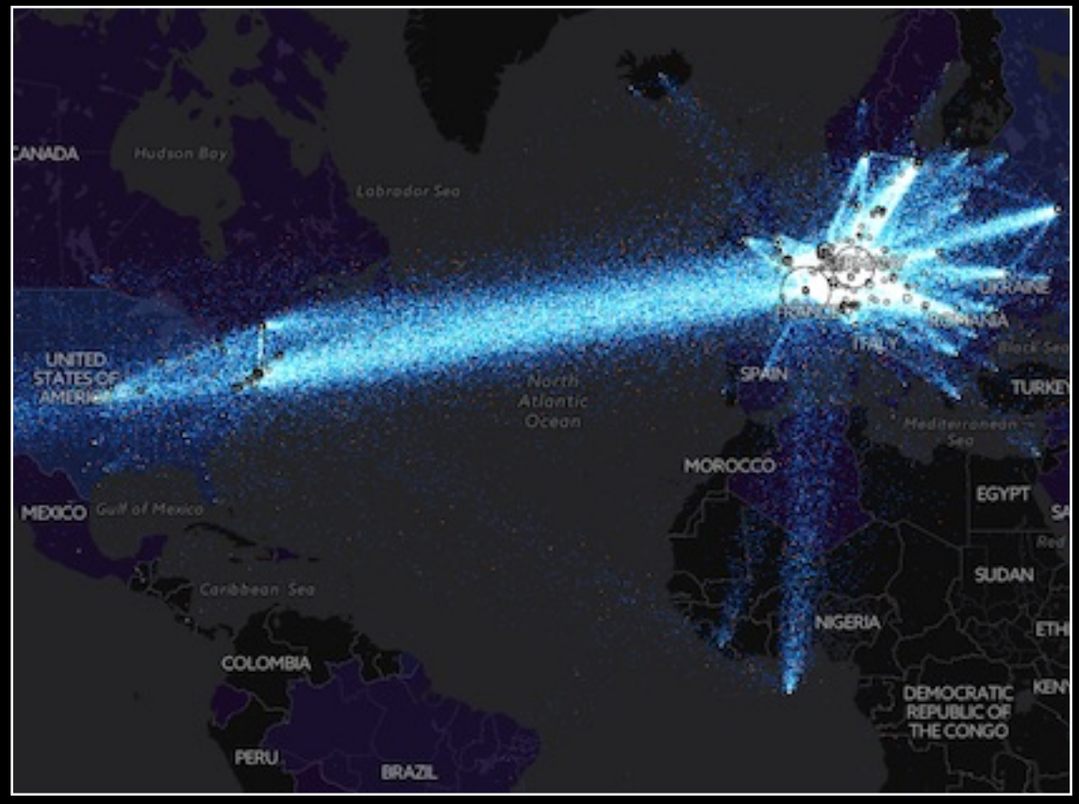
- Service decrypts message, finds and connects to Rendezvous point (with one-time-secret)
- Rendezvous point informs client about successful connection establishment

Implementations

- Tor Browser
    - Automatically starts Tor background processes, deletes cookies and browsing history after session
- Tor Messenger
- Security focused operation systems

Some numbers

- Between 2-3 Mio users in last two months
- Approximately 6500 relays
- 1000 Exits
- Tor data flow
- https://metrics.torproject.org/



The anonymous Internet

Daily Tor users per 100,000 Internet users

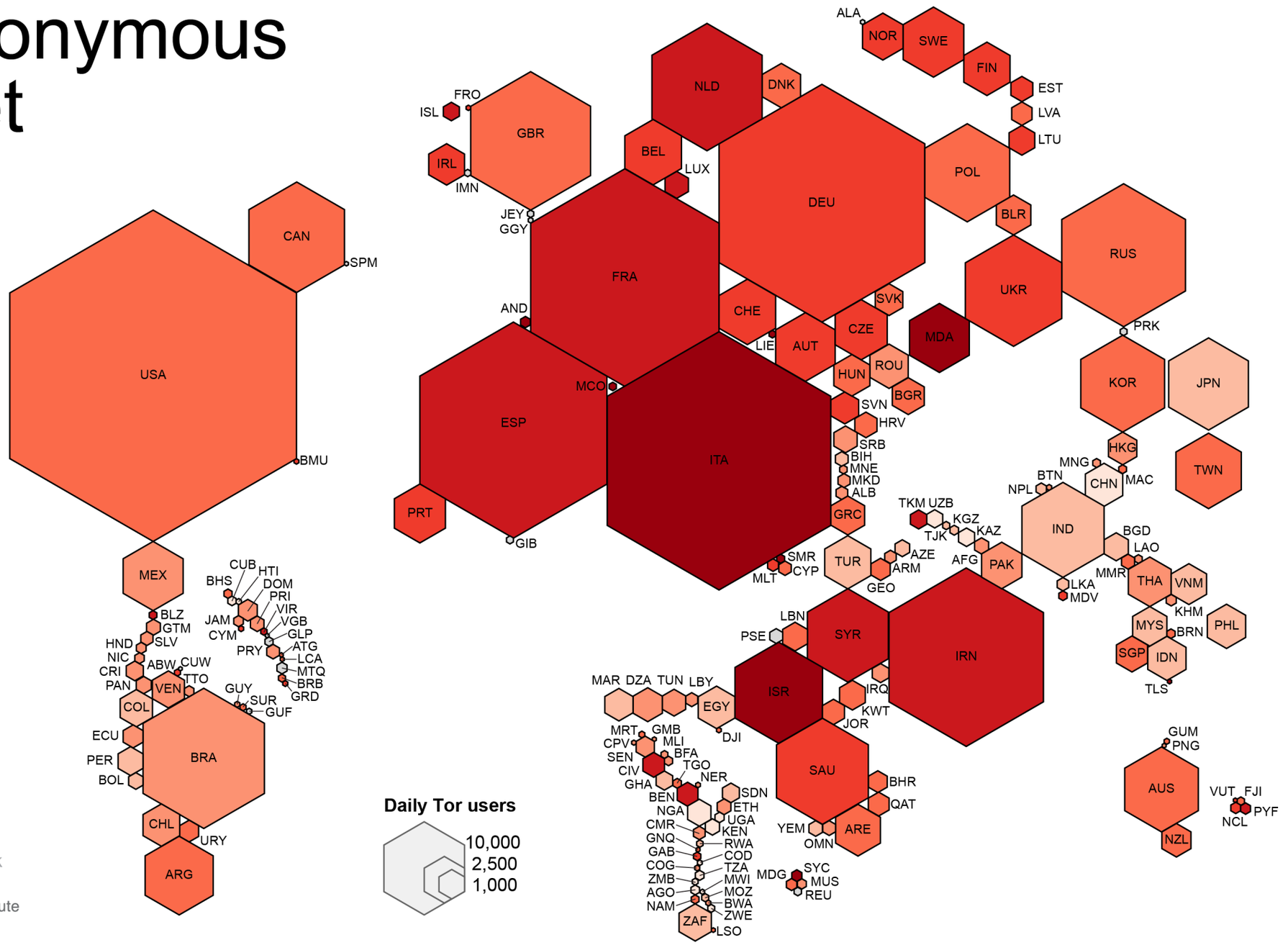
- > 200
- 100 - 200
- 50 - 100
- 25 - 50
- 10 - 25
- 5 - 10
- < 5
- no information

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

oiioiioi
oiioiioi
oiioiioi
Oxford Internet Institute
University of Oxford



Daily Tor users

10,000
2,500
1,000

- https://2019.www.torproject.org/index.html.en
- https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf
- https://www.torproject.org/download/
- https://www.eff.org/pages/tor-and-https

Happy anonymous browsing