Web Technology 2015

Lecture 7. Encrypted and anonymous communication (part 2)

Staas de Jong

Universiteit Leiden > LIACS > Media Technology

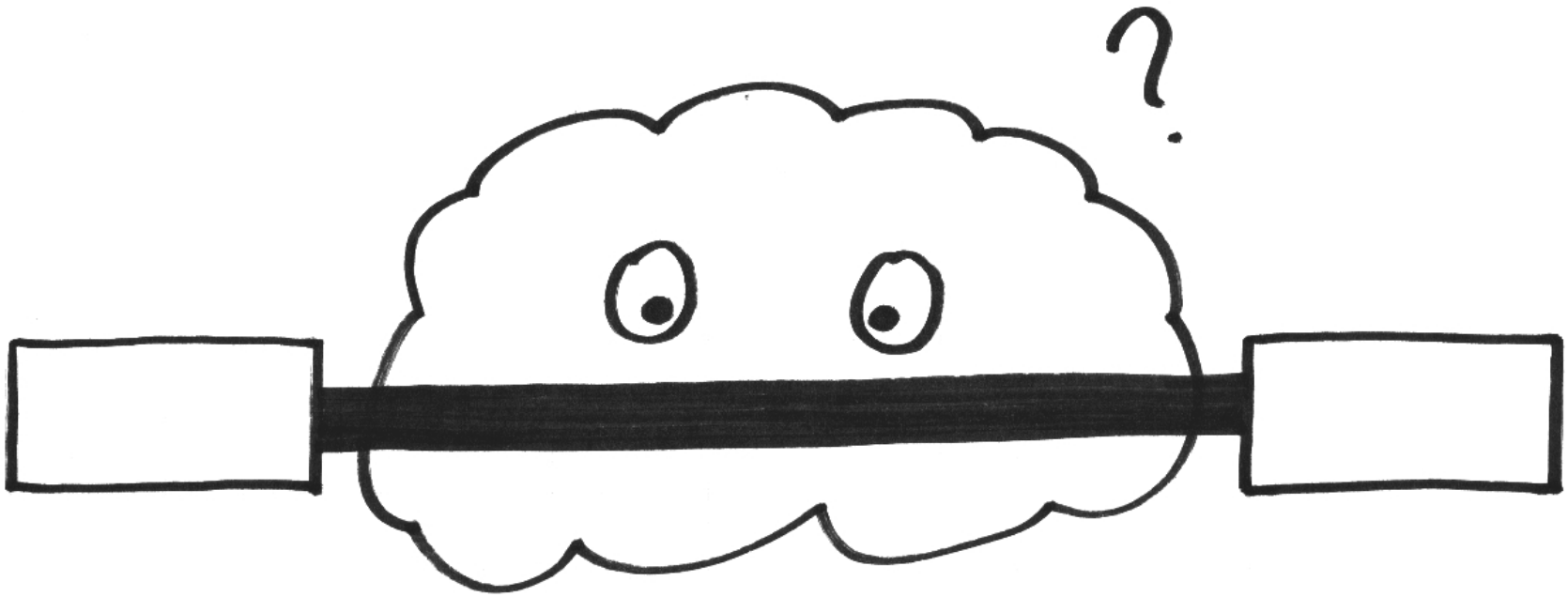
Previous lecture:

When using Internet technologies, we are confronted with two fundamental questions:

- How to hide *what* is communicated?
- How to hide *who* communicates? ◀

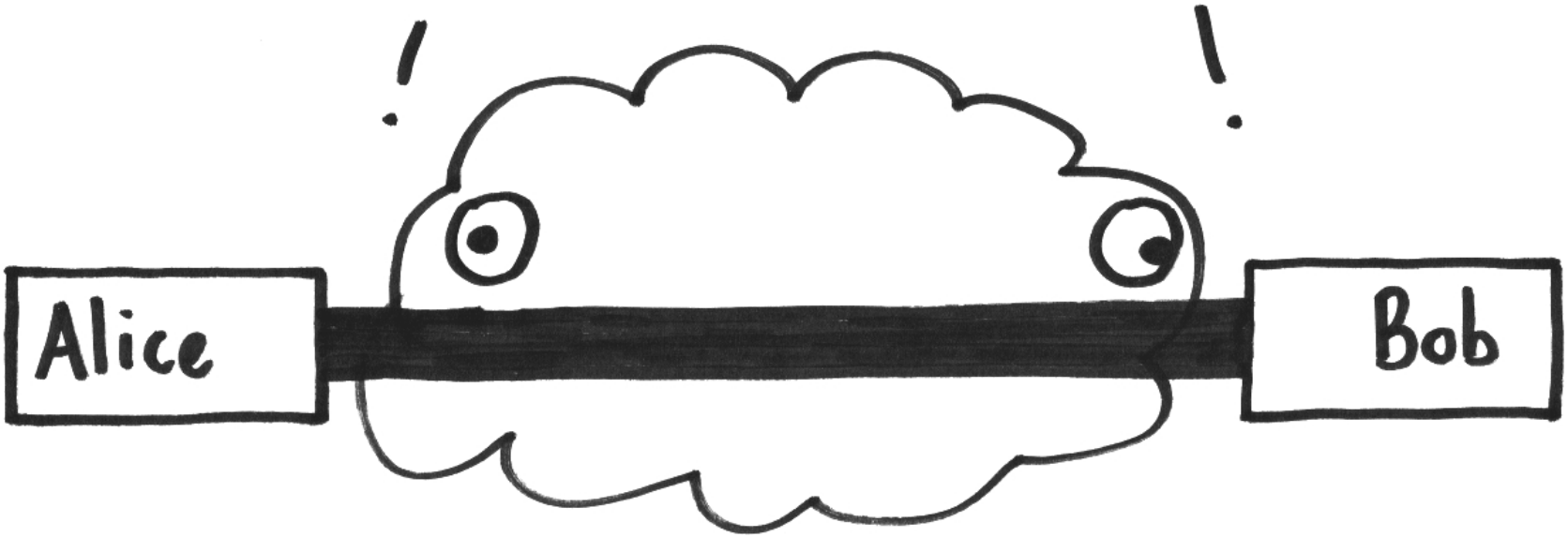
...in the face of an opponent that has *total knowledge of all the IP traffic involved*.

Traffic analysis



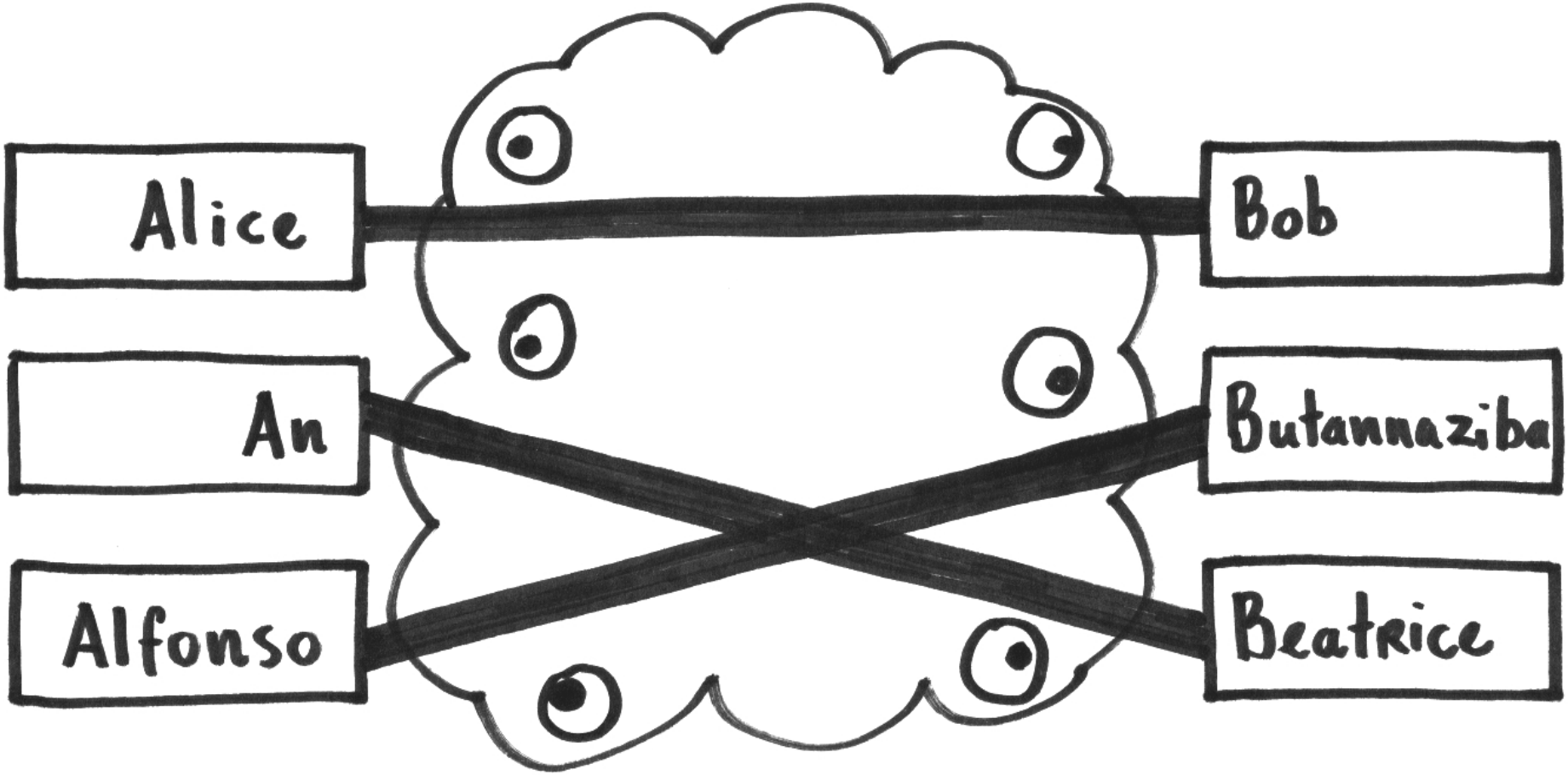
Thanks to key exchange algorithms: The *contents* of Internet connections can be encrypted, end-to-end, on the fly.

Traffic analysis



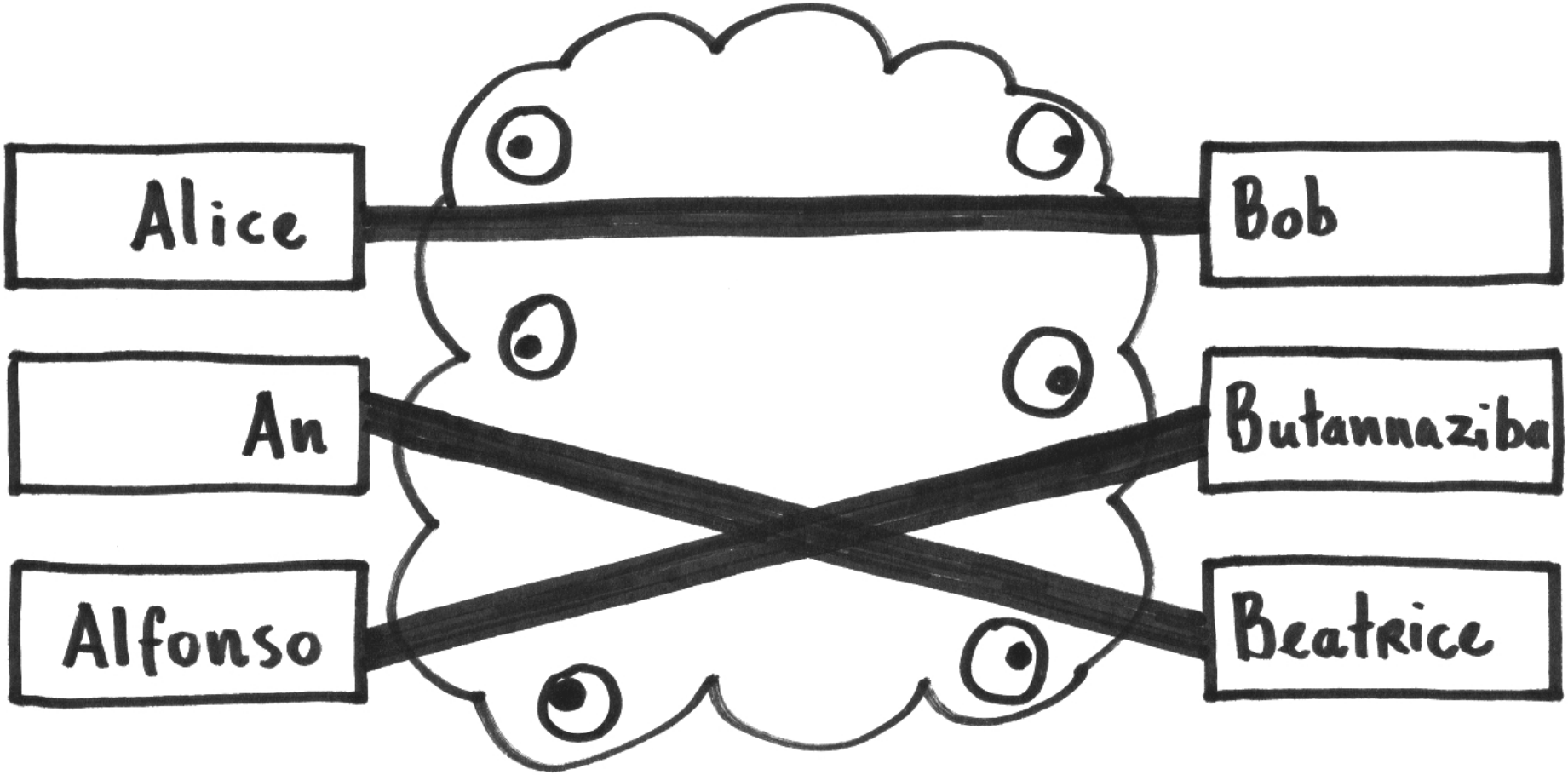
But traffic analysis can still monitor the *participants* and *timing* of communication!

Traffic analysis



↑ And this is not just the case for Alice and Bob...

Traffic analysis

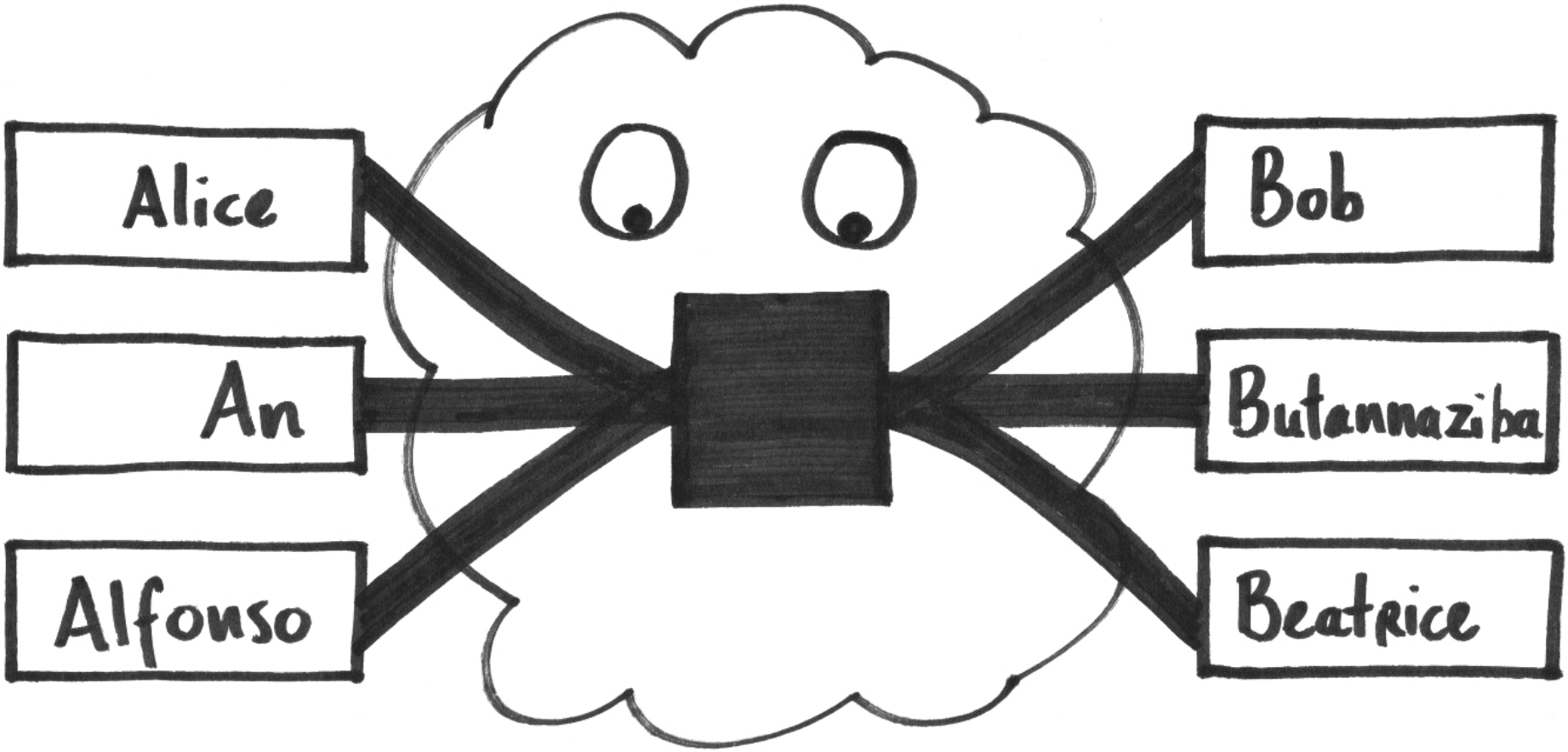


↑ ***But*** this is not just the case for Alice and Bob!

Countering traffic analysis: context

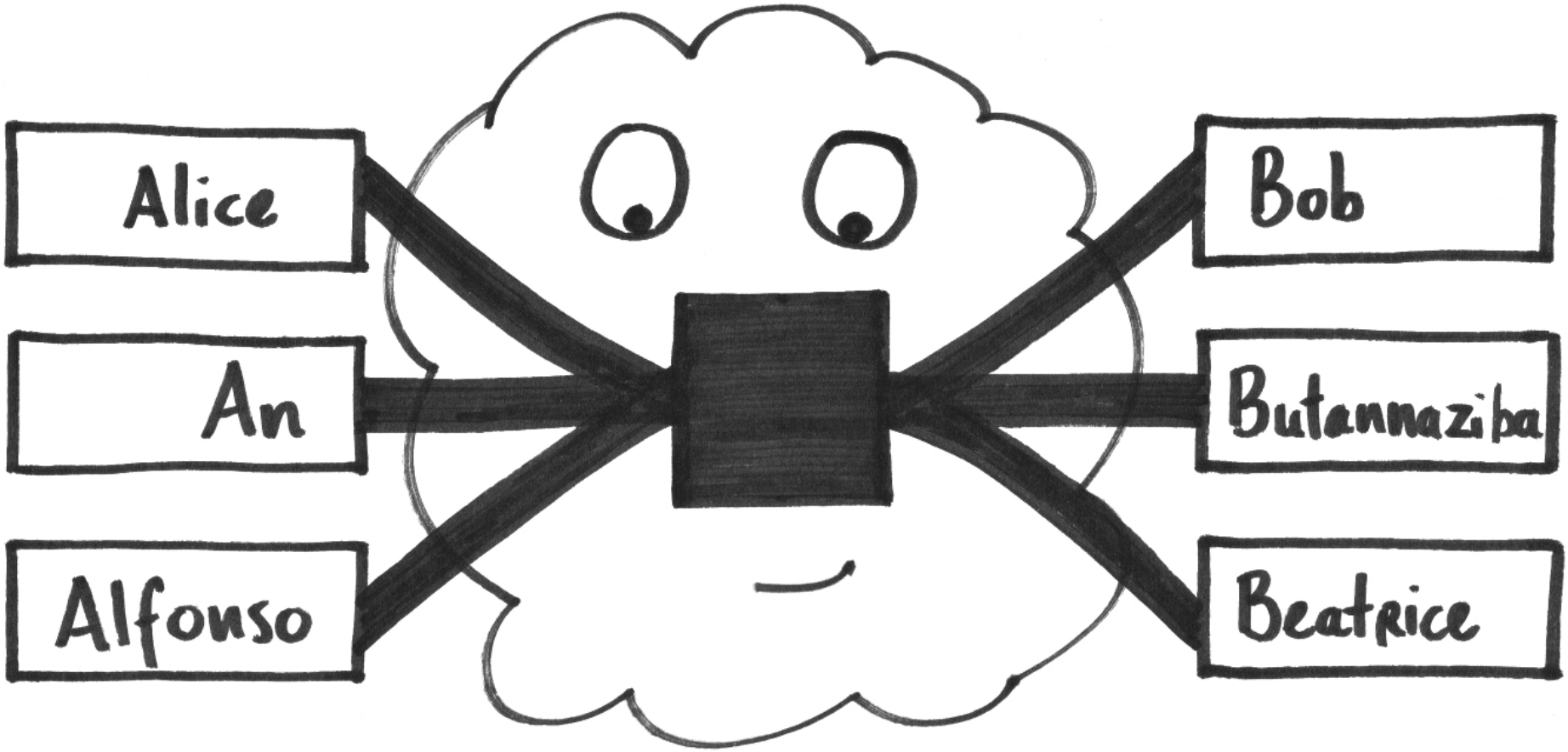
- As we know from public life, *acting in groups* can *anonymize* the transactions that occur.
  - E.g. who actually shot in a firing squad?
  - E.g. clashing groups of hooligans.
  - ...but there can also be safety in crowds.
- For the Internet: consider the following idea...

Hypothetical: the Internet Anonymization Server



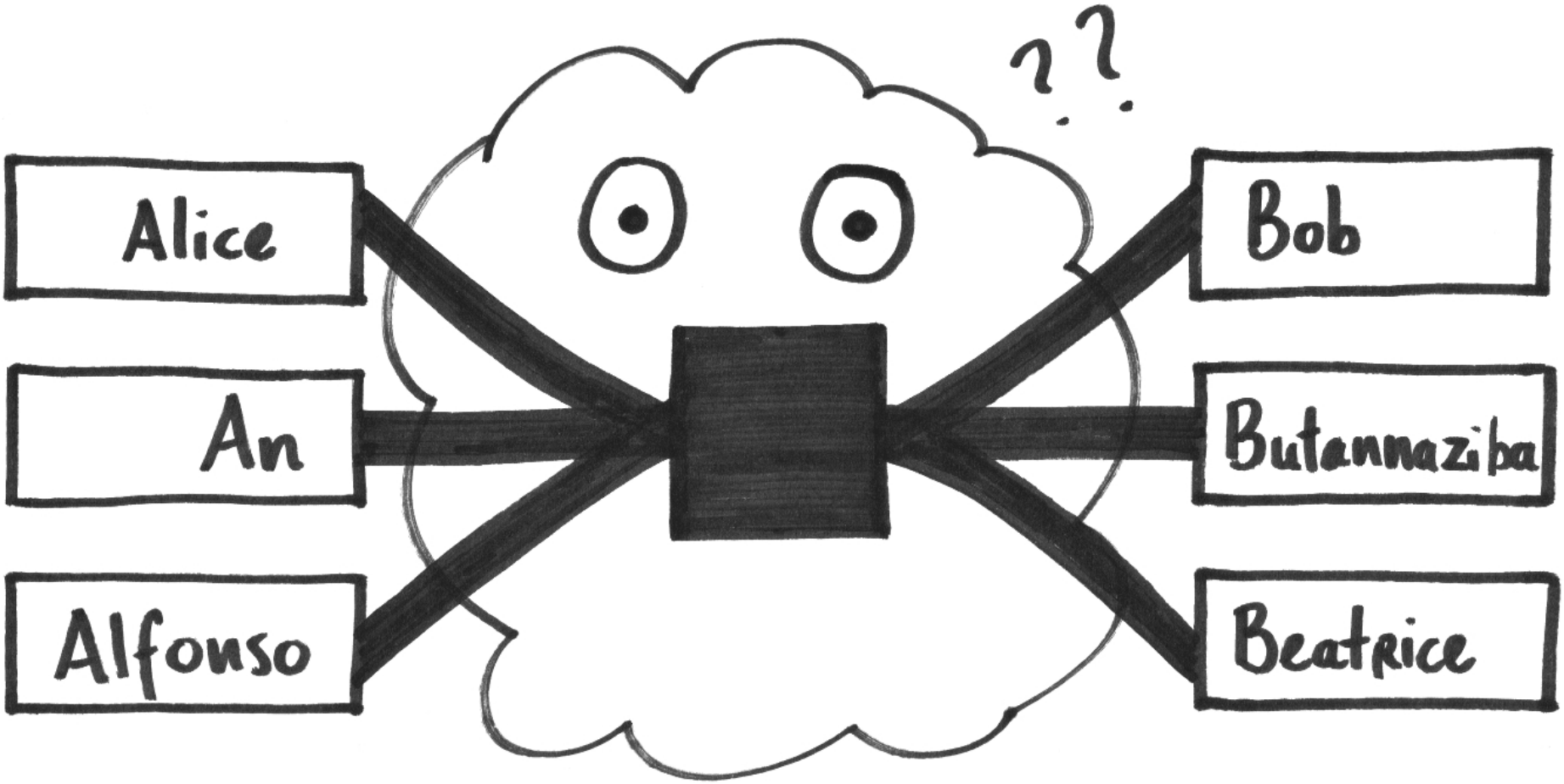
↑ Each connection to and from the black box is encrypted;
and it passes incoming data to the designated destination.

Hypothetical: the Internet Anonymization Server



↑ If (at a given moment) only Alice and Bob are using this, *it makes no sense*.

Hypothetical: the Internet Anonymization Server



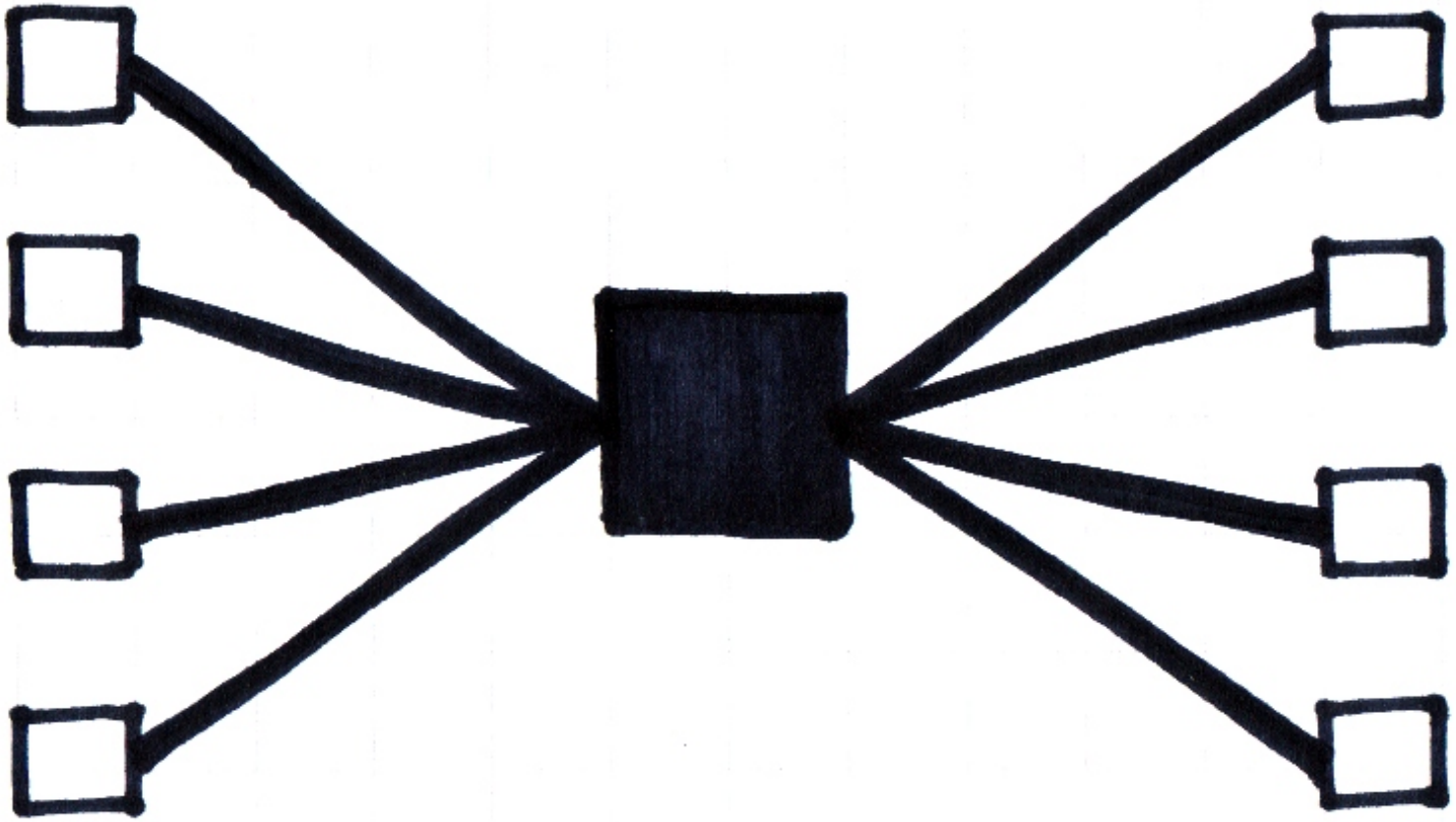
↑ But *the more* hosts use the "IAS", *the harder* it becomes to identify end-to-end connections from traffic patterns!

Some major remaining issues...

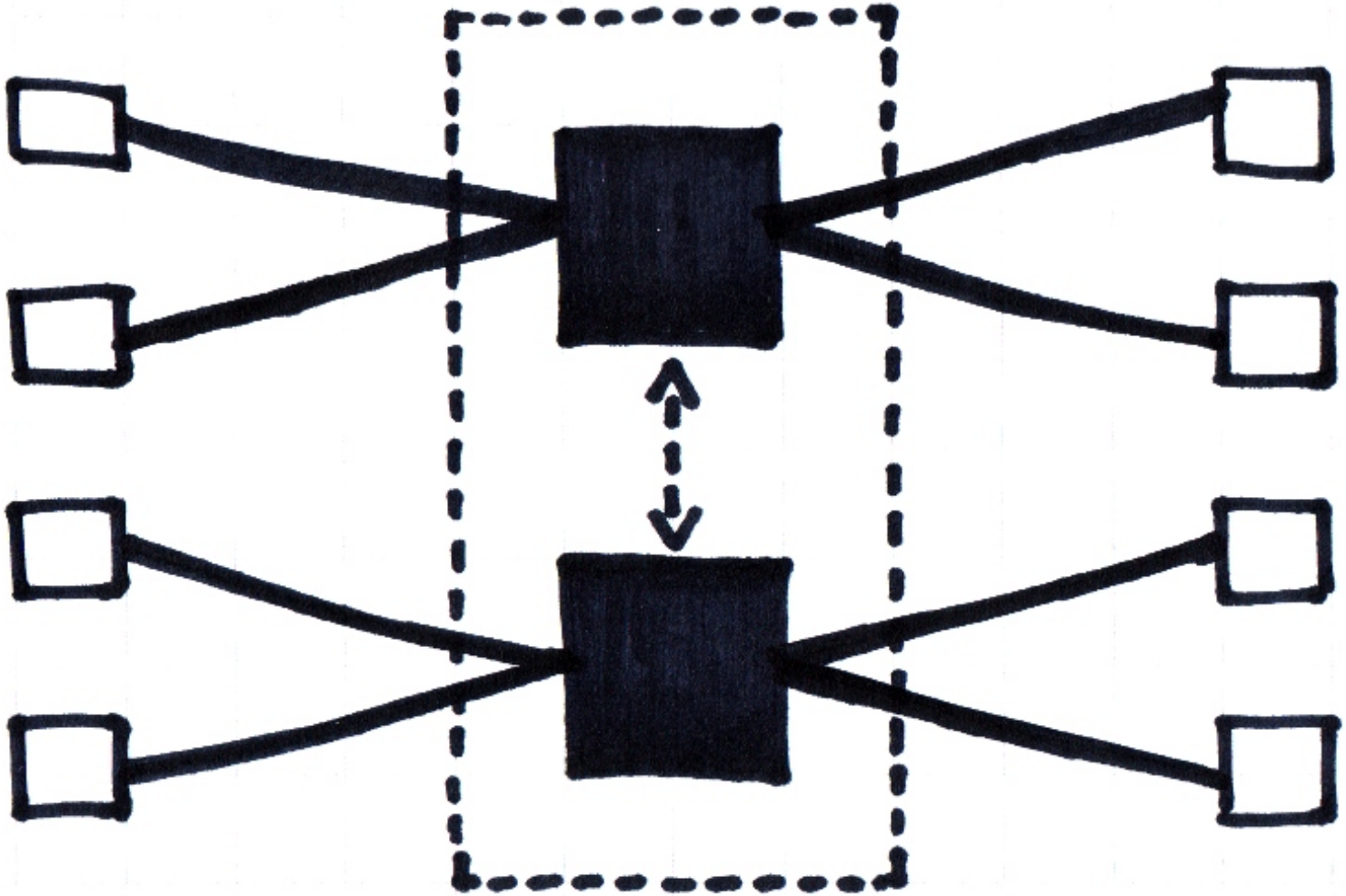
- *Performance*:
  - One server processing everyone's IP packets / TCP segments is not practical (scalability!). *
- *Trust*:
  - What if our anonymization server becomes compromised?
  - Who gets to control the hardware?
- *Backward compatibility*:
  - The existing Internet, e.g. the web, does not work like this.
  - Consider trying to request a webpage using this system.

* Still, some services do use this topology. See the sheets of Roger Dingledine's 2010 talk at Stanford University: *http://freehaven.net/~arma/slides-26c3.pdf* .

Performance: bad relay topology

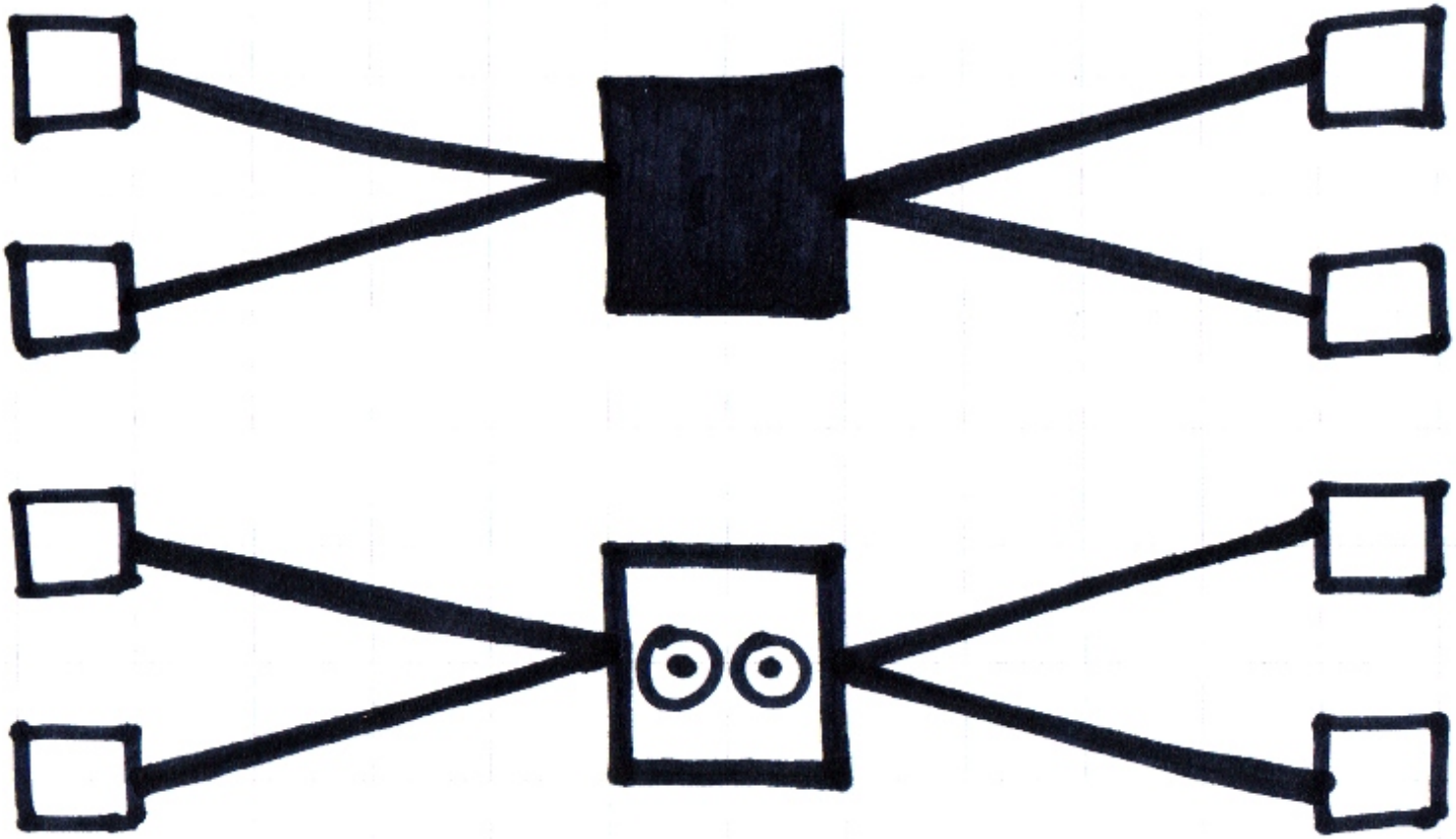


Performance: better relay topology



↑ Use a *series* of machines (“relays”), and distribute the connections load among them.

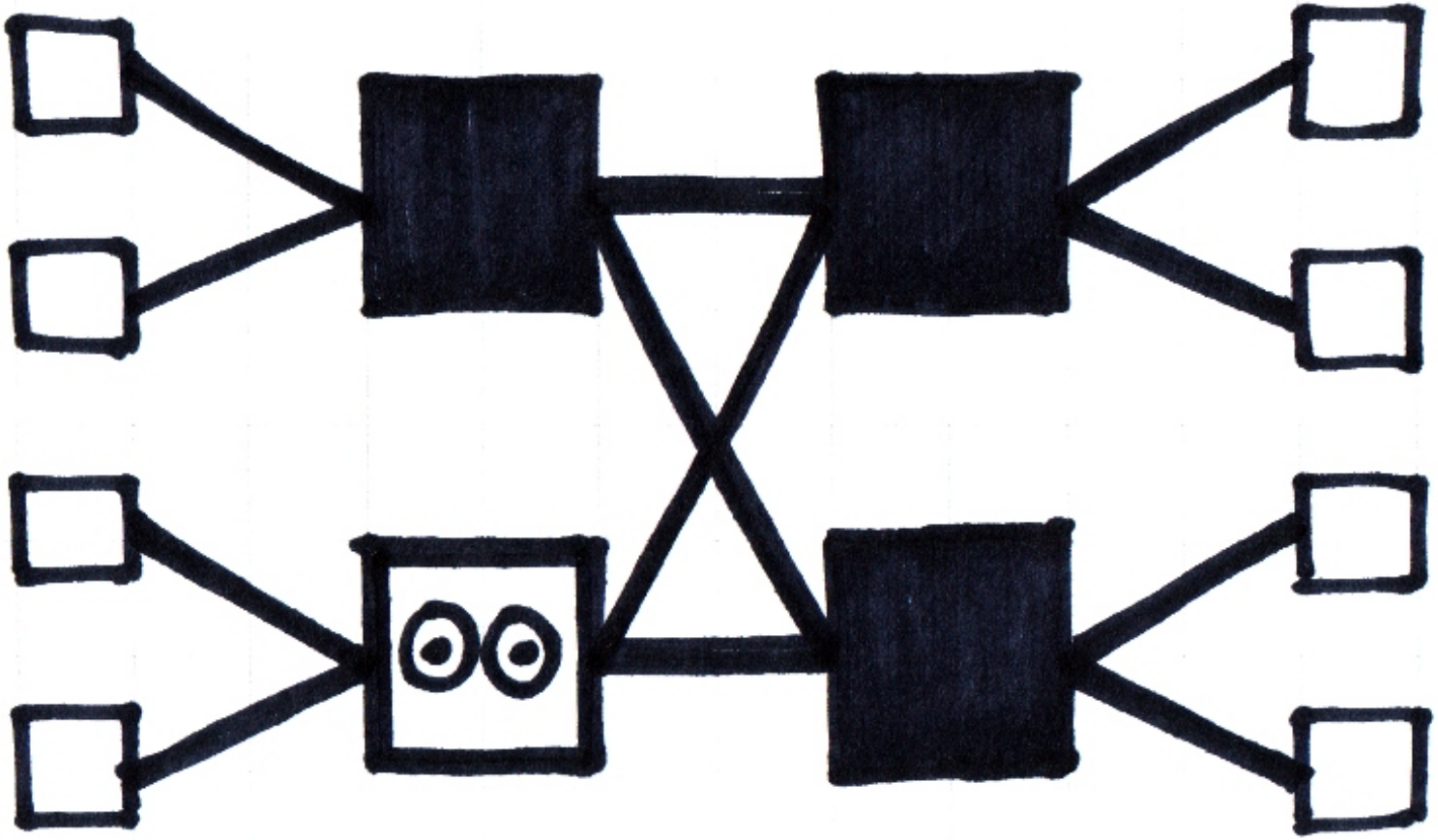
Trust: bad relay topology



↑ If a relay is compromised, all its users are de-anonymized!

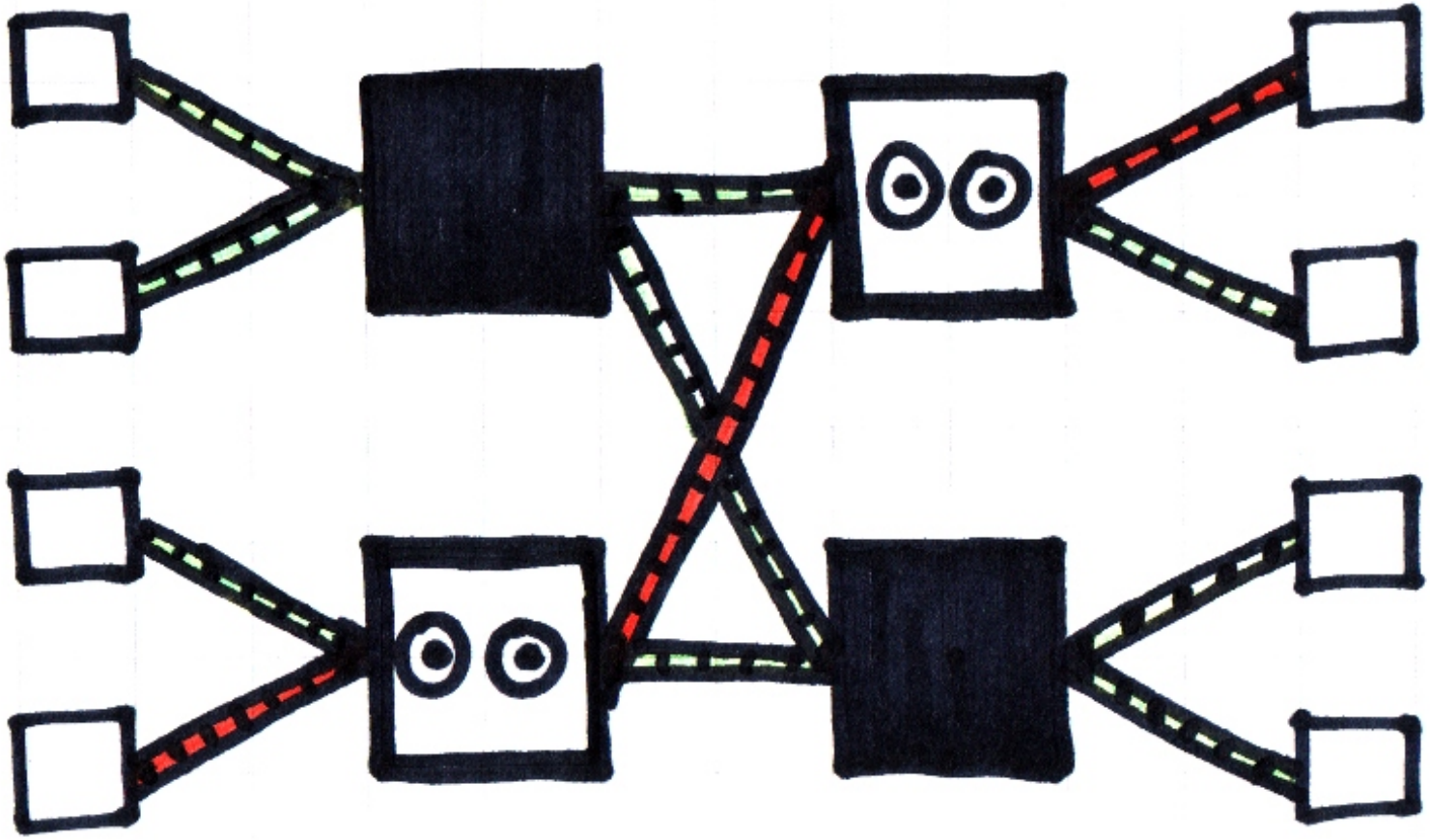
⇒ *No relay should know both a connection's origin & destination...*

Trust: better relay topology



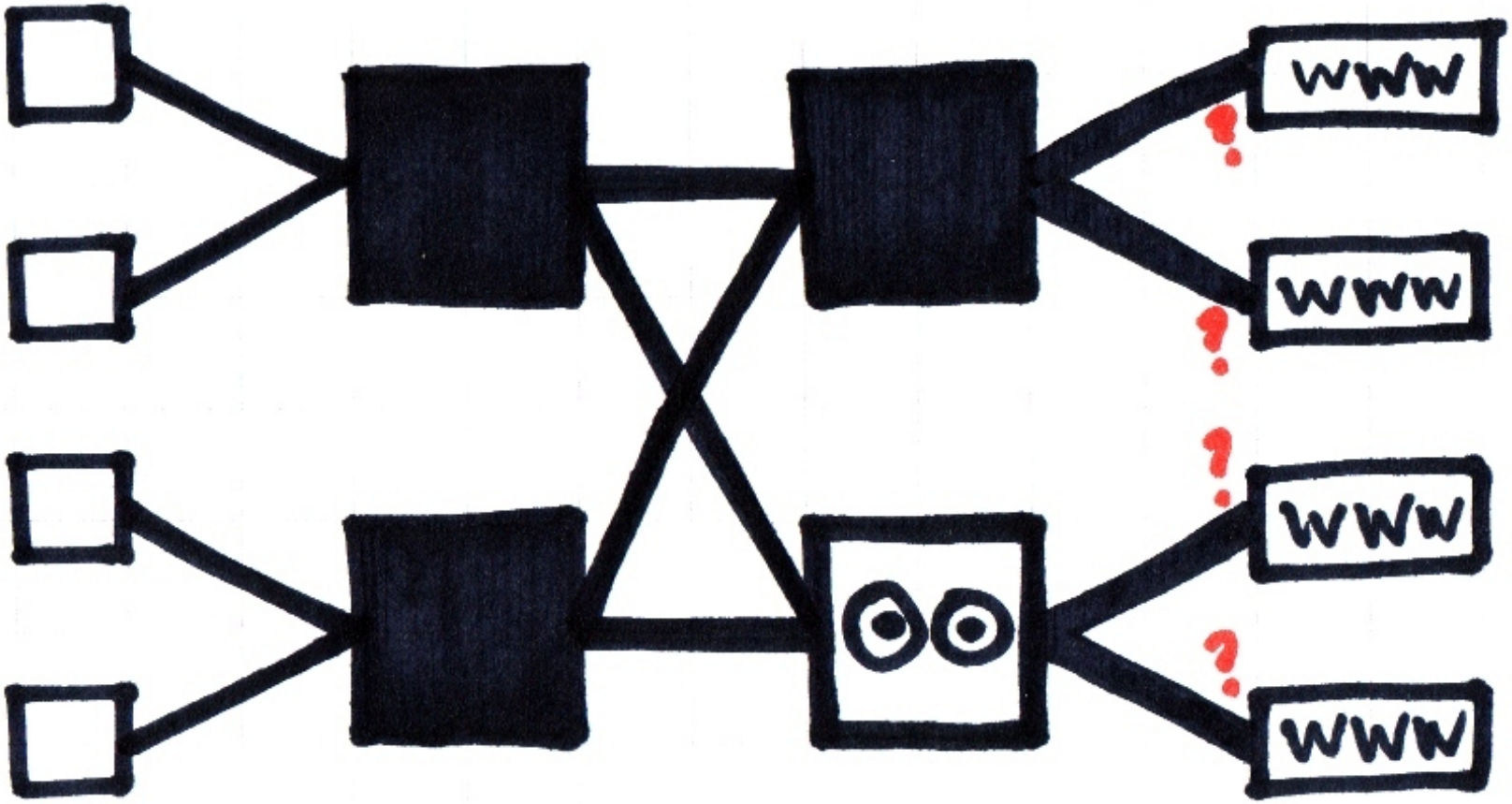
↑ *Multi-hop circuit:* Alice connects to an entry node first; an exit node then connects to Bob.

Trust: compromised relay nodes should be an exception!



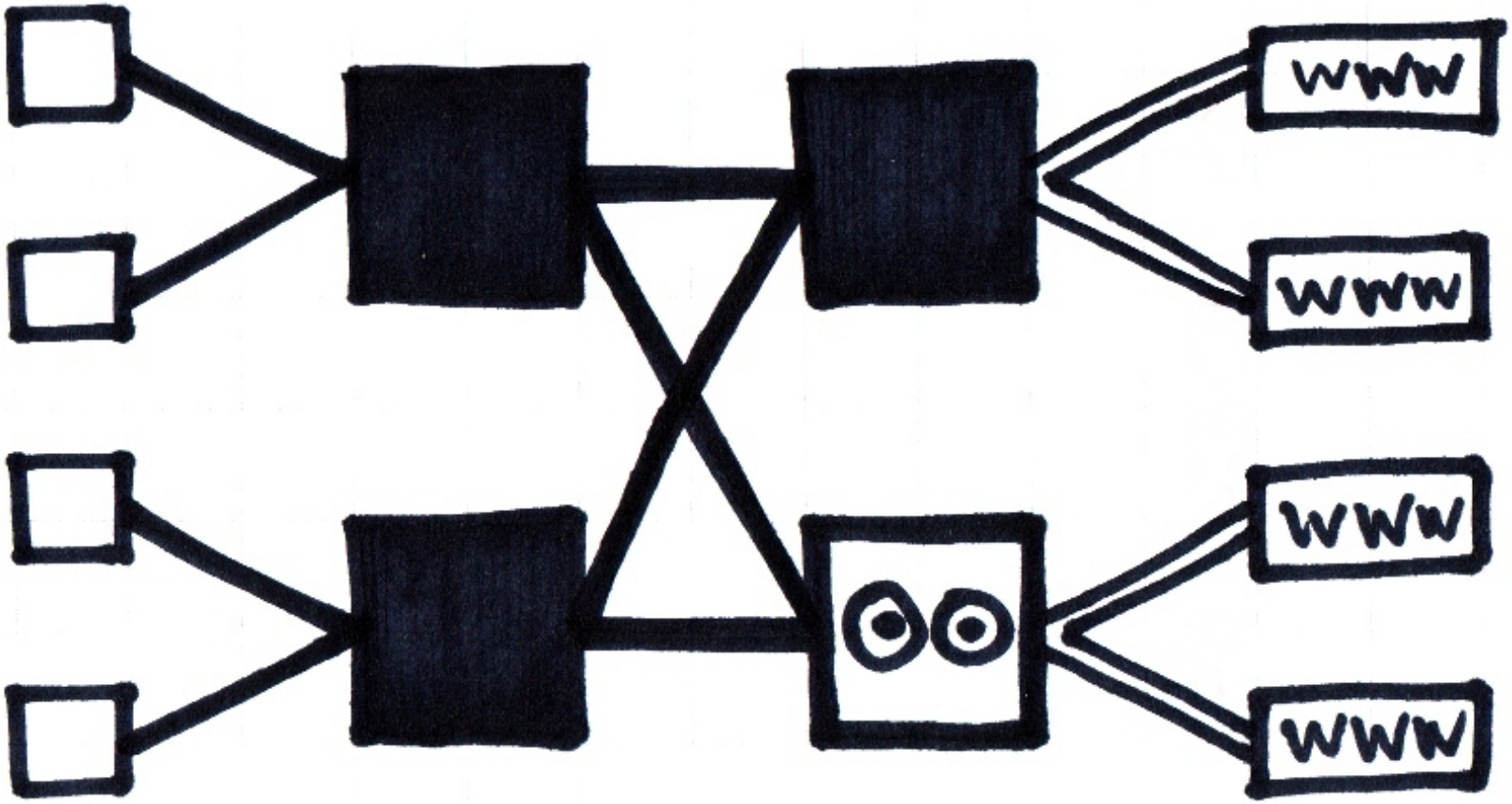
↑ Too many, and de-anonymization could still happen...

Backward compatibility: at the exit nodes



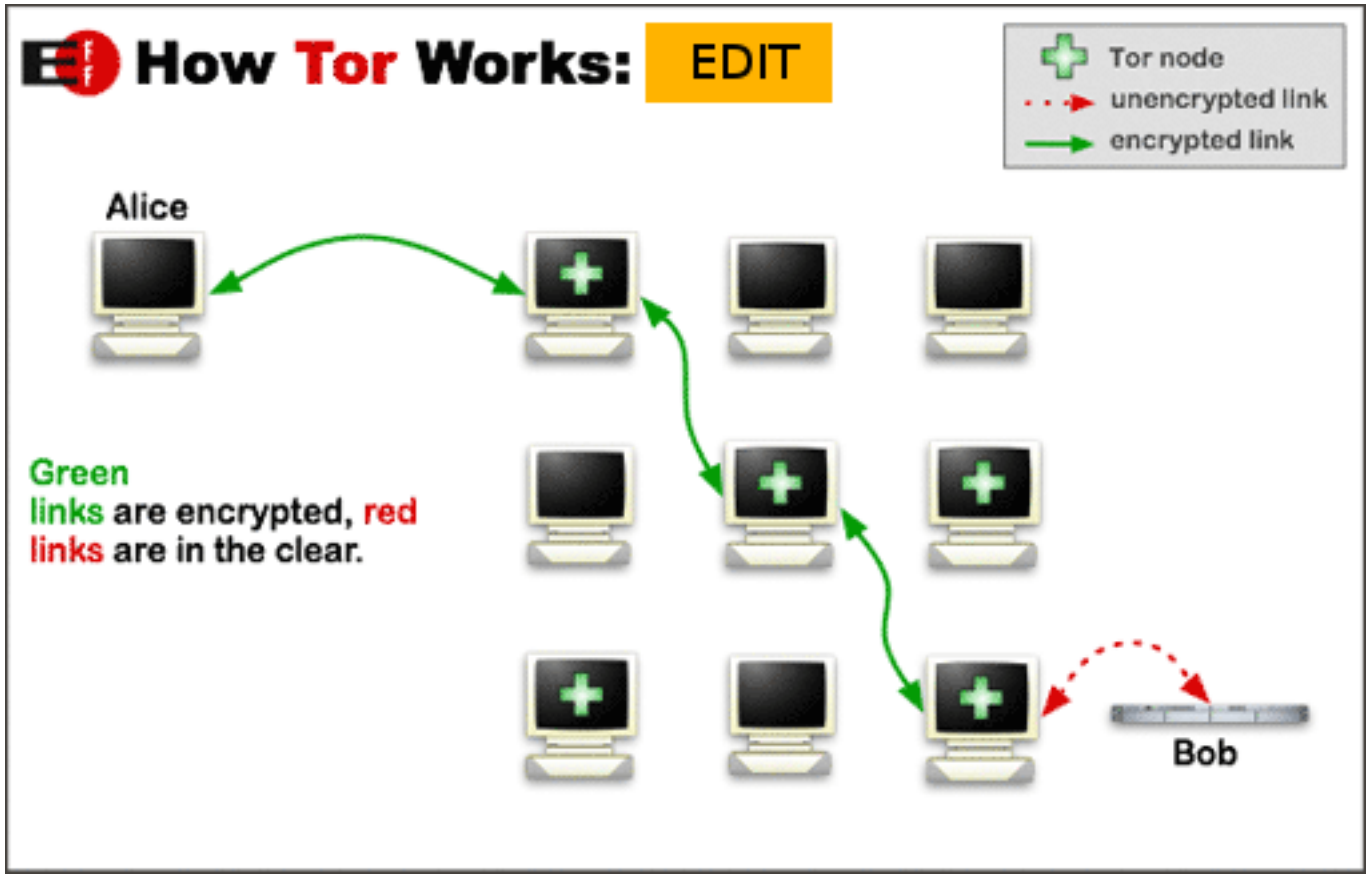
⇒ *Exit nodes should connect to destinations using ordinary, unencrypted connections.*

Backward compatibility: at the exit nodes



↑ A realistic, distributed architecture
for anonymous communication.

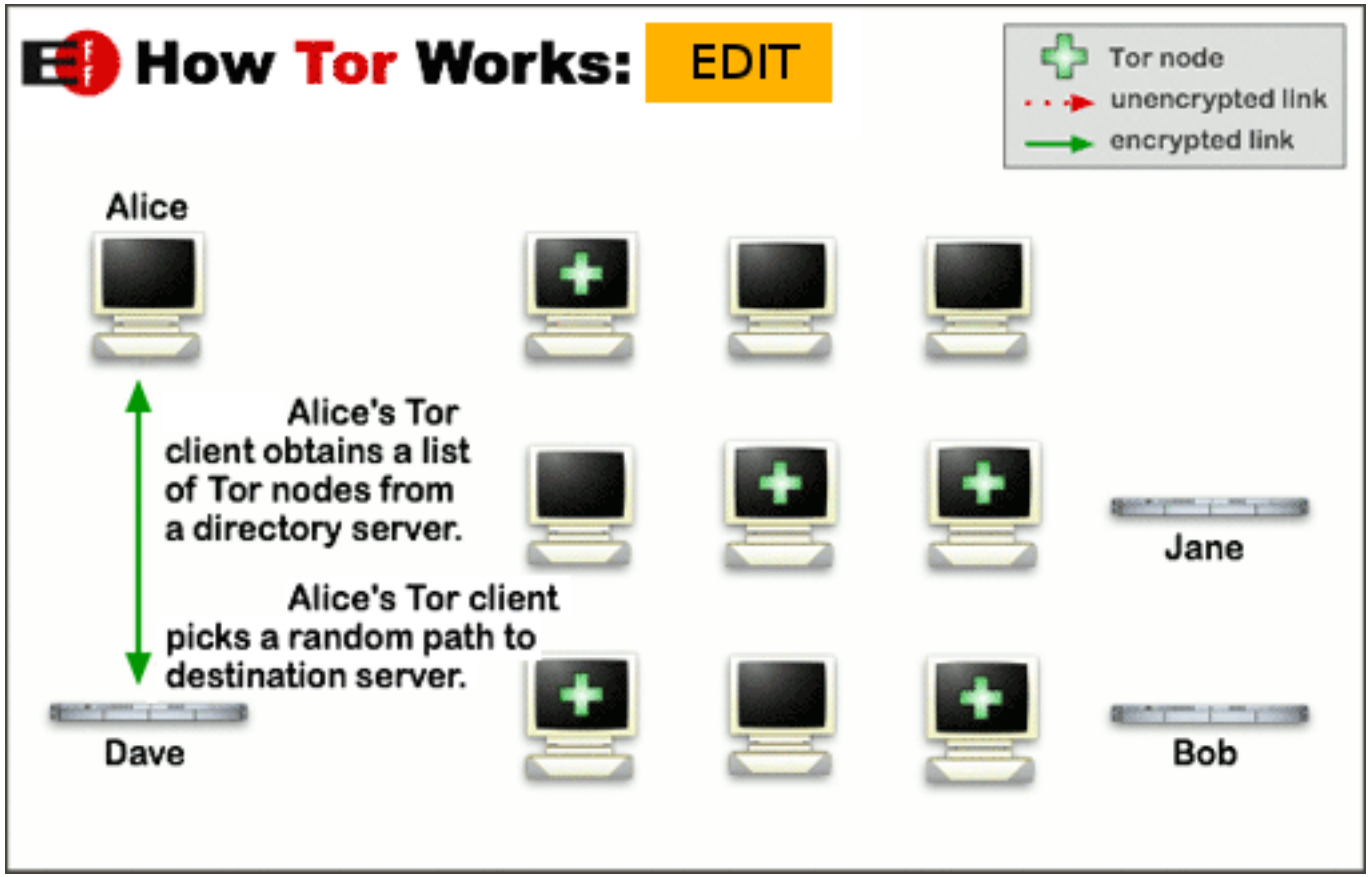
Tor: a distributed architecture for anonymous communication



↑ *Compare to Tor...*

...it *also* uses intermediate nodes.

Tor: a distributed architecture for anonymous communication

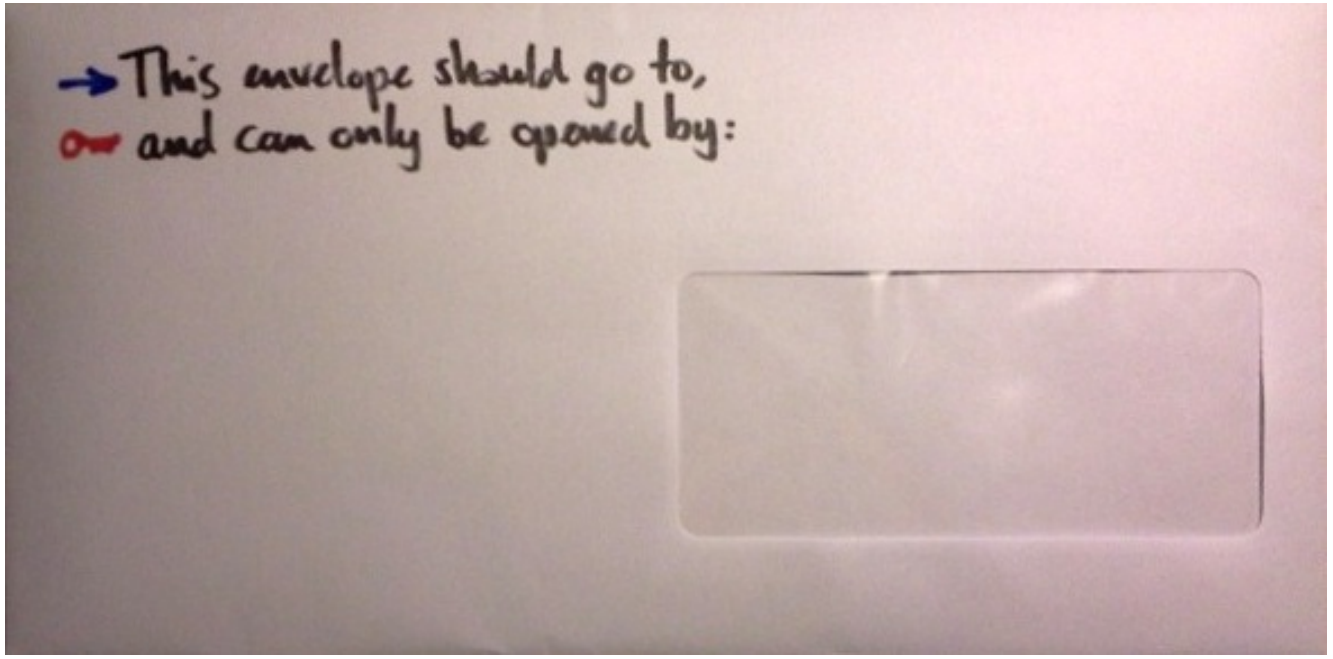


↑ A (temporary) multi-hop circuit first needs to be set up...

Tor: a distributed architecture for anonymous communication

- *On circuit setup:*
  - Alice chooses an entry node & sets up an encrypted connection with it.
  - Alice uses this connection to negotiate a second encrypted connection, to the intermediate node of her choice.
  - Alice finally uses the second connection to negotiate a third connection, to her chosen exit node.
  - Alice can now anonymously connect to Bob.
  - As her first message passes to Bob, each relay node along the circuit undoes & discards a layer of encryption.
  - *Trust*: because of the repeated encryption, each relay only has data about its two immediate neighbours in the chain.

Live example: manual onion routing



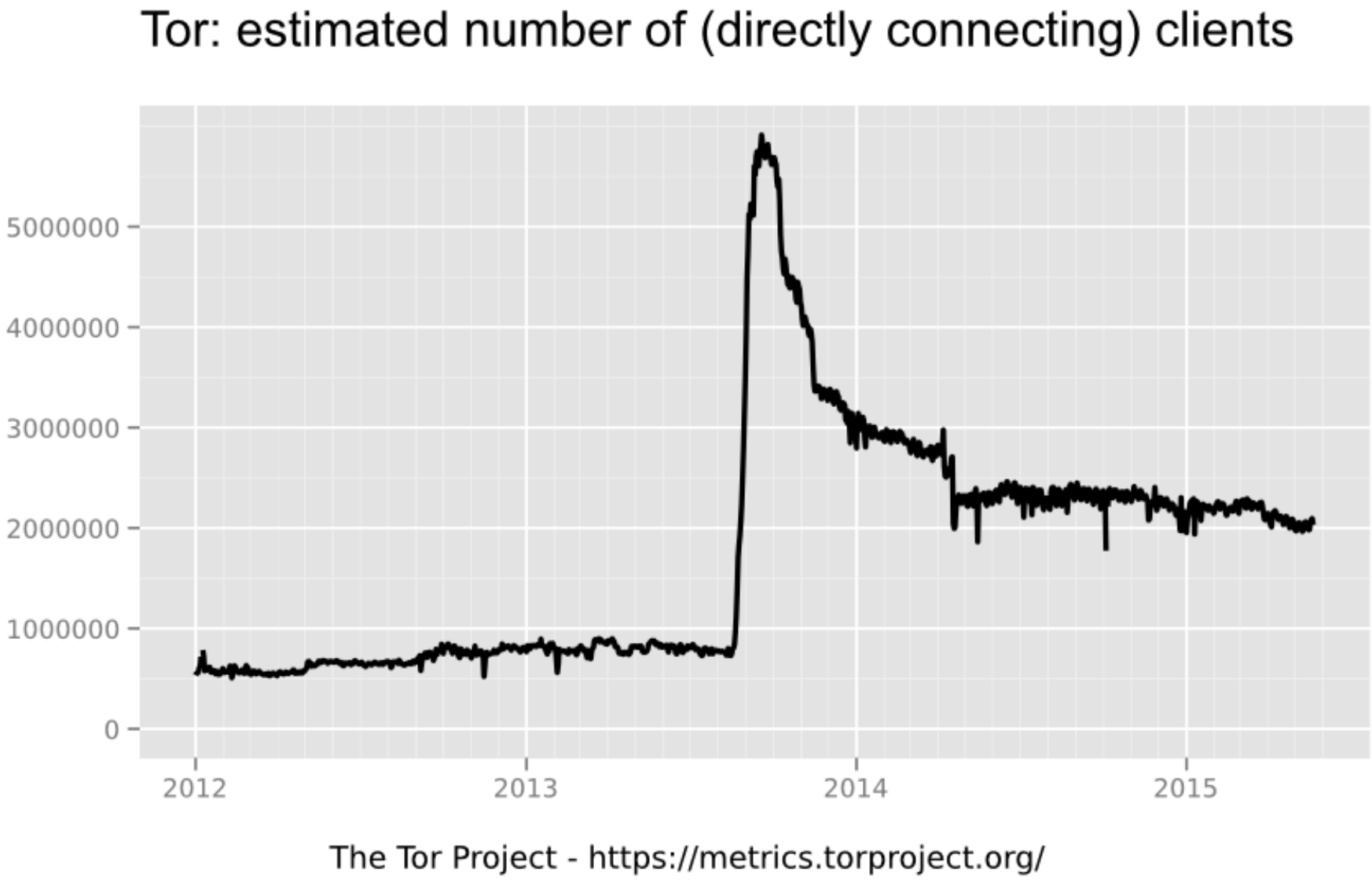
Tor: a distributed architecture for anonymous communication

- *Just illustrated:* why originally, Tor was an acronym for "The Onion Router".

- Tor uses TLS over TCP.

- *Interesting possible future direction:* Tor directly on IP (see their FAQ).

Tor: usage


Tor: estimated number of (directly connecting) clients
5000000
4000000
3000000
2000000
1000000
0
2012
2013
2014
2015
The Tor Project - https://metrics.torproject.org/

Tor: a testimonial...