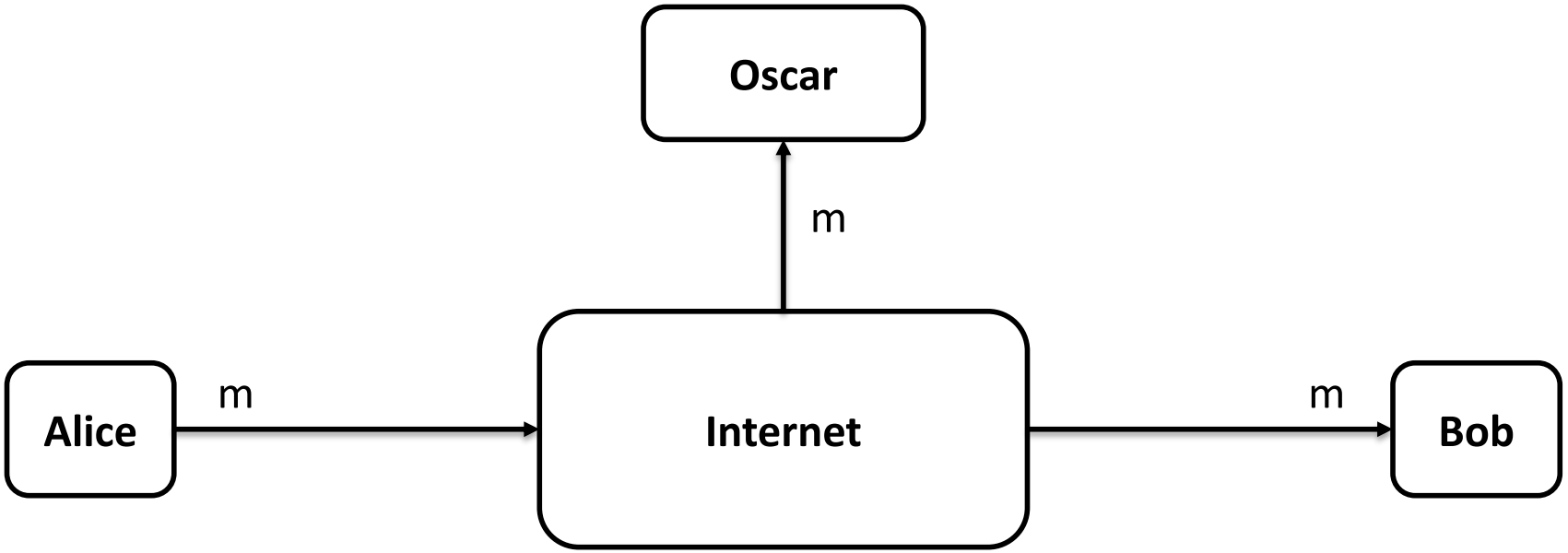
CS326– Systems Security

Lecture 23

The Onion Router (TOR)

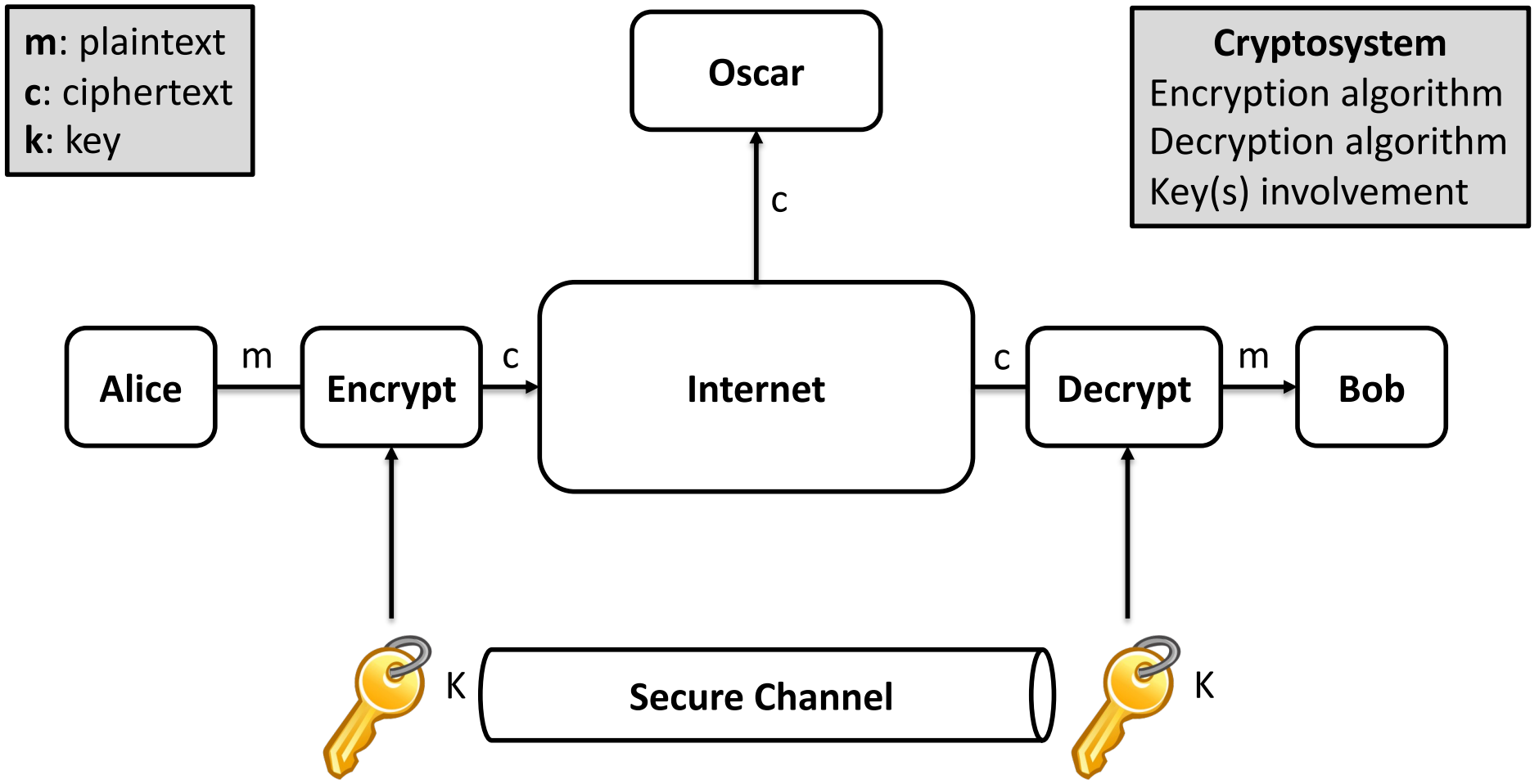
Elias Athanasopoulos
athanasopoulos.elias@ucy.ac.cy

Recall: Basic Problem

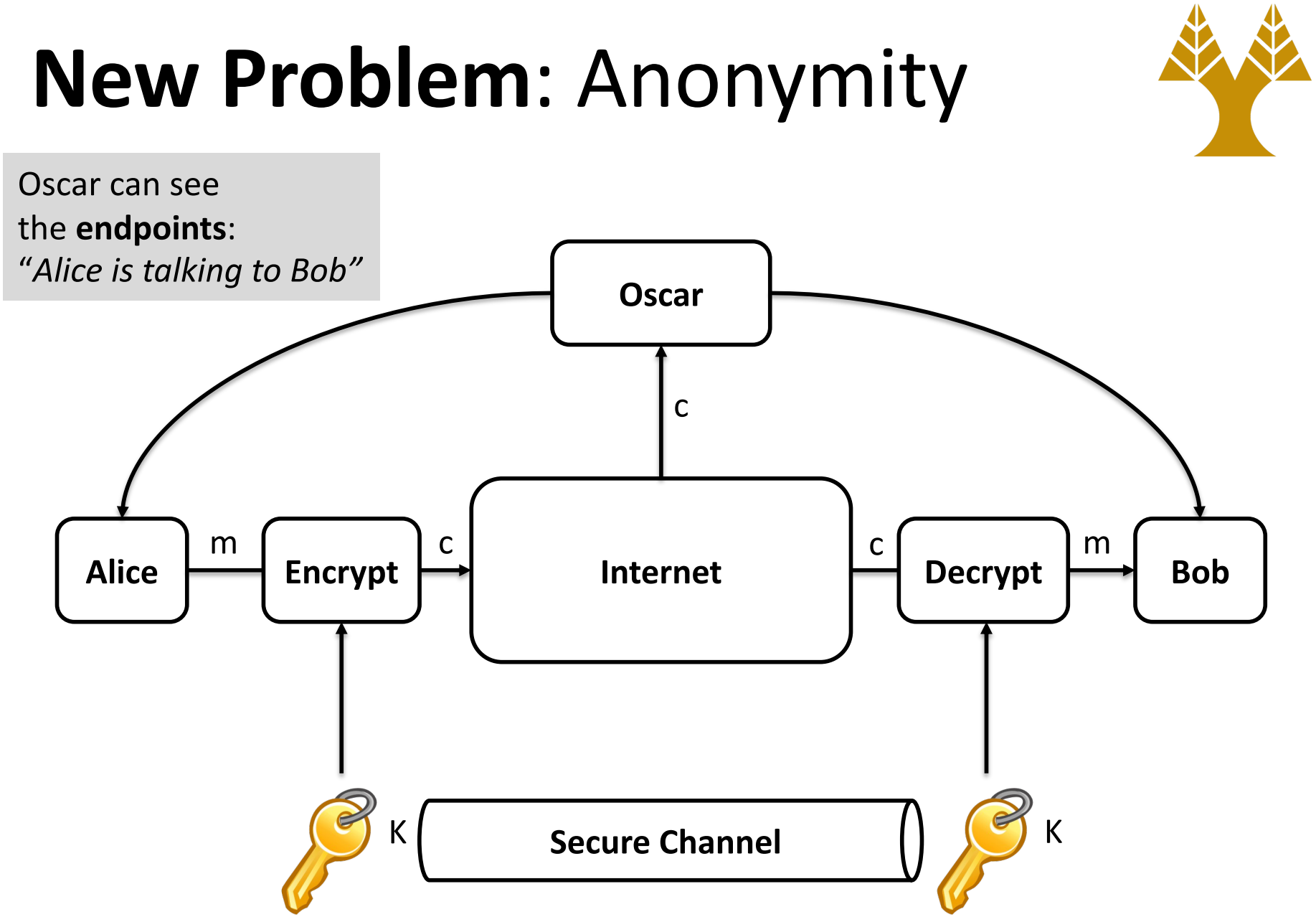


Oscar can see the message (confidentiality)
Oscar can modify the message (integrity)

Recall: Cryptography



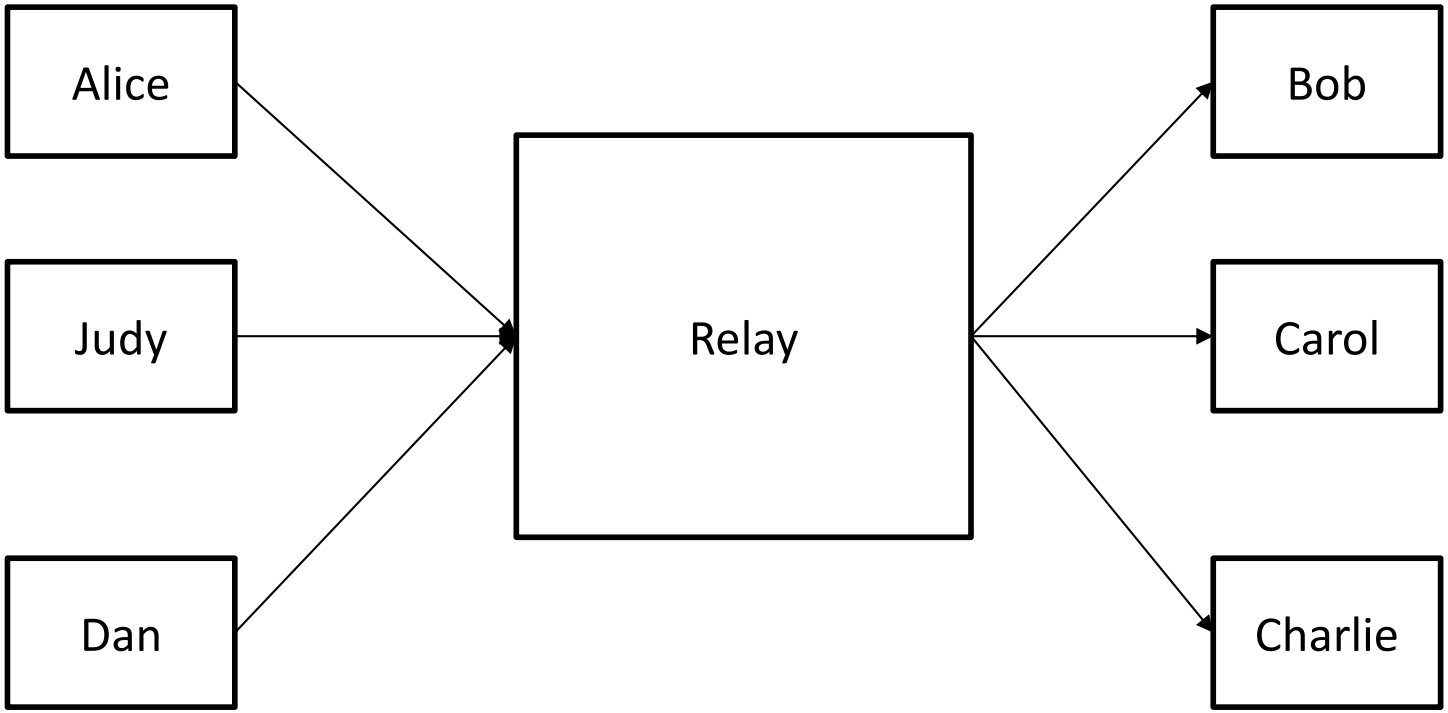
New Problem: Anonymity



Anonymous Communication

- Users can interact with other users over the Internet
  - MitM should not be able to infer who is talking to whom
- Is this a real problem?
  - Journalists and activists
  - Citizens of oppressive regimes
  - Minorities
  - People that do not want to be associated with certain activities (i.e., Alice belongs to political party *X*)

Simple Solution



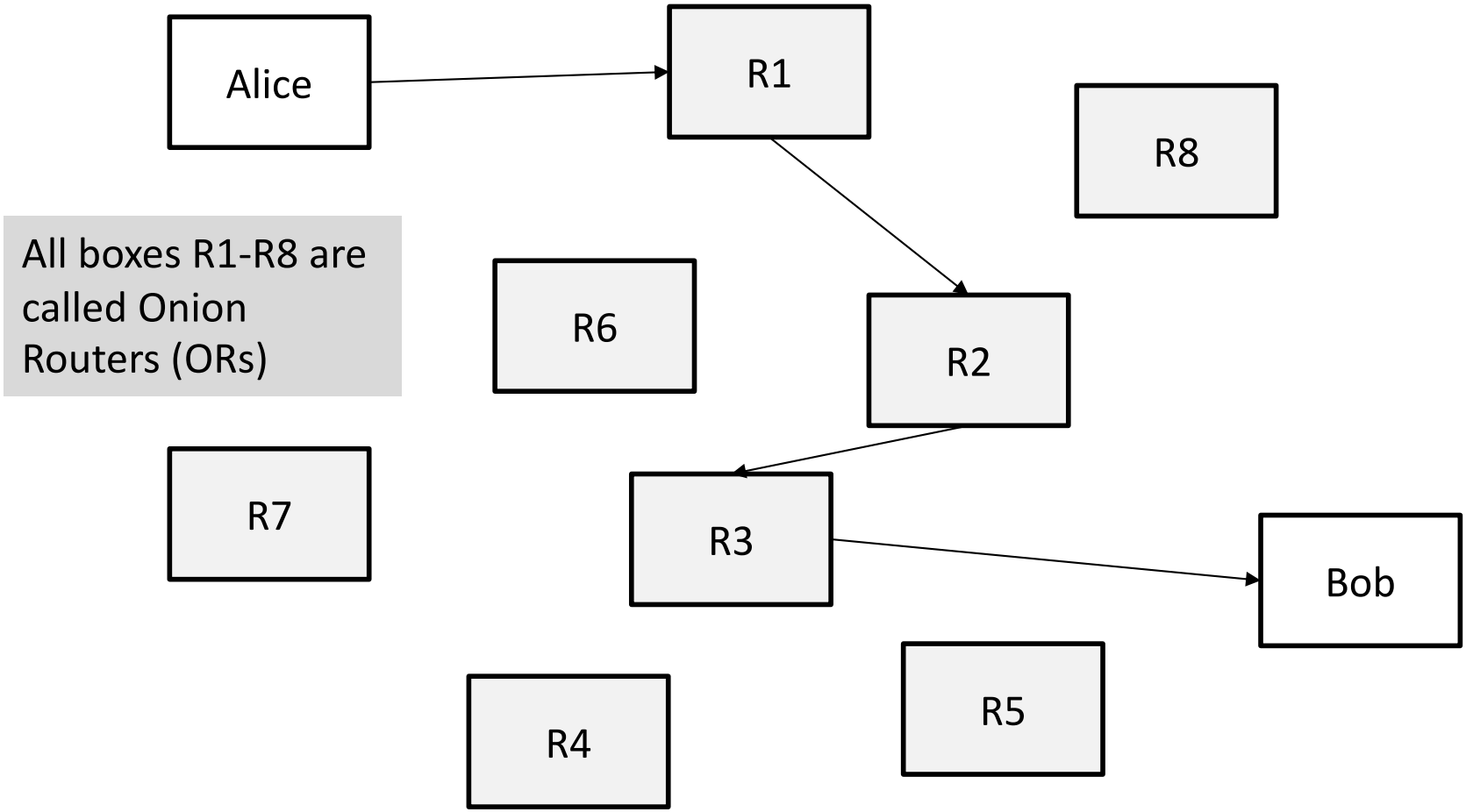
Properties of Relay

- Needs a big set of senders
- Needs a big set of receivers
  - The larger the sets, the better for anonymous communication
- Needs time to process the messages
  - The longer it takes for the relay to output the messages, the better for anonymous communication
- Single point of failure
  - If you compromise the relay, all communications are compromised

Goals

- Anonymity for practical low-latency communications
  - Web browsing, etc.
- Defend a realistic threat model
  - Attacker **cannot** monitor all Internet links (global passive attacker)
  - Can observe some fraction of network traffic
  - Can generate, modify, delete, or delay traffic
  - Can operate onion routers of their own
  - Can compromise some fraction of the onion routers

The Onion Router (TOR)

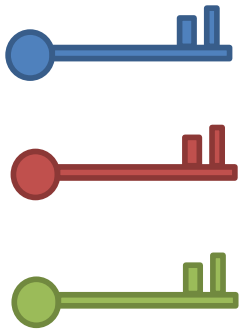


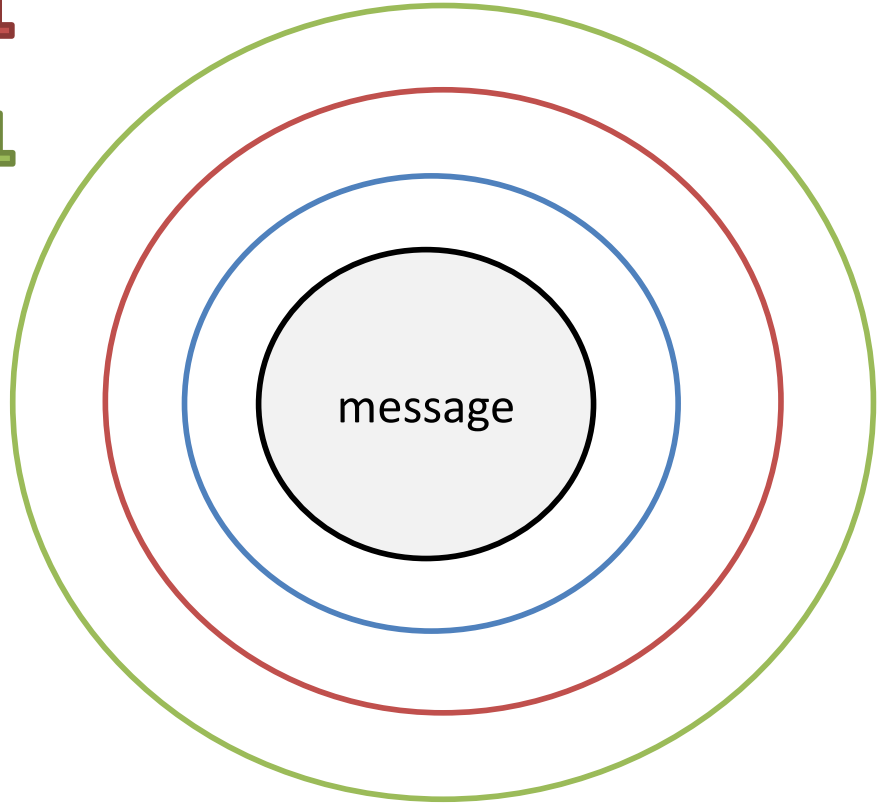
How it works?

- Alice builds circuits of Onion Routers
- A circuit includes *at least* three Onion Routers
  - Default is **three**, but longer circuits are allowed
  - Three is not magic, it is a compromise
  - An attacker must control the first and the exit node for breaking TOR
- A circuit is a number (by default **three**) of encapsulated TLS tunnels

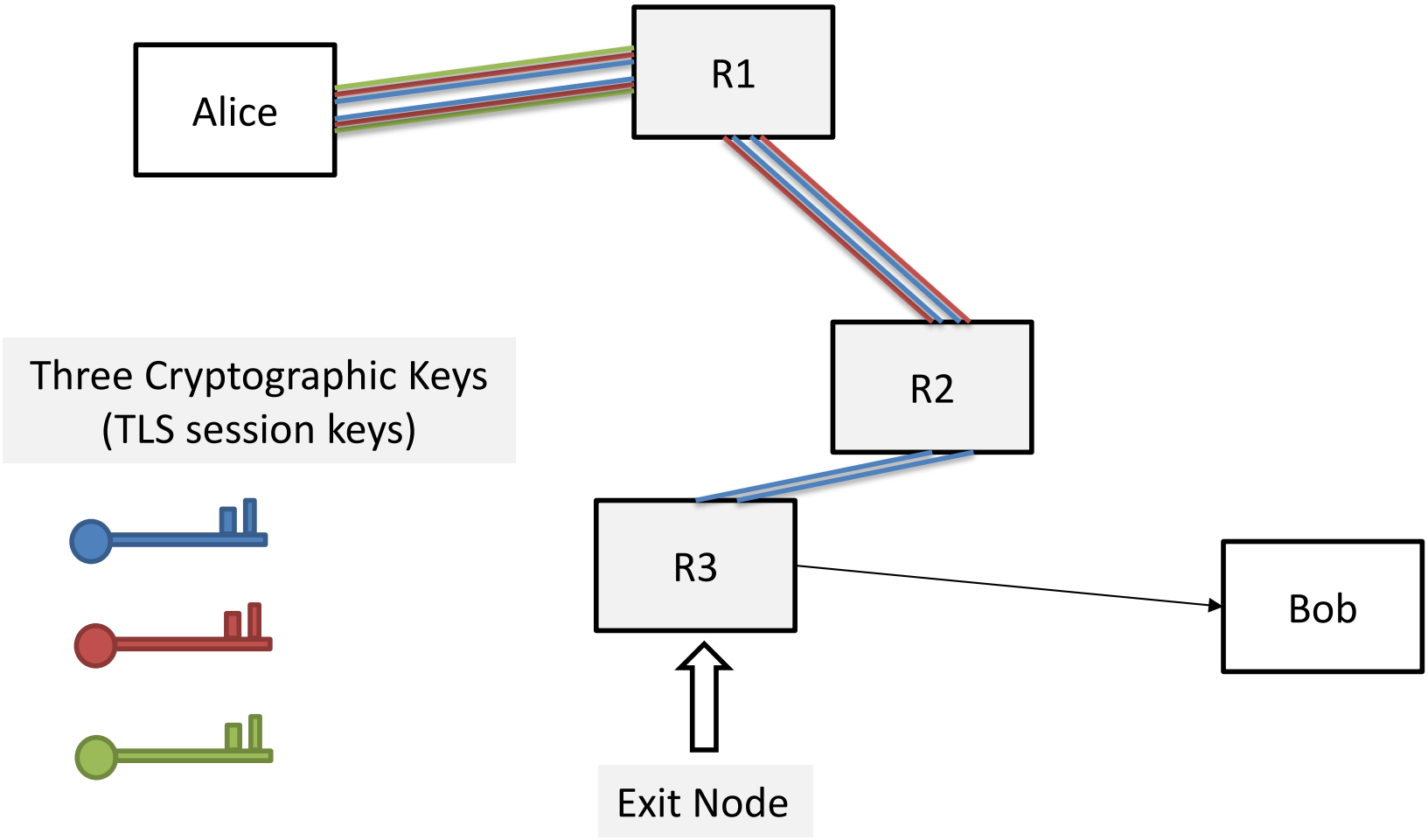
Onion

Three Cryptographic Keys
(TLS session keys)





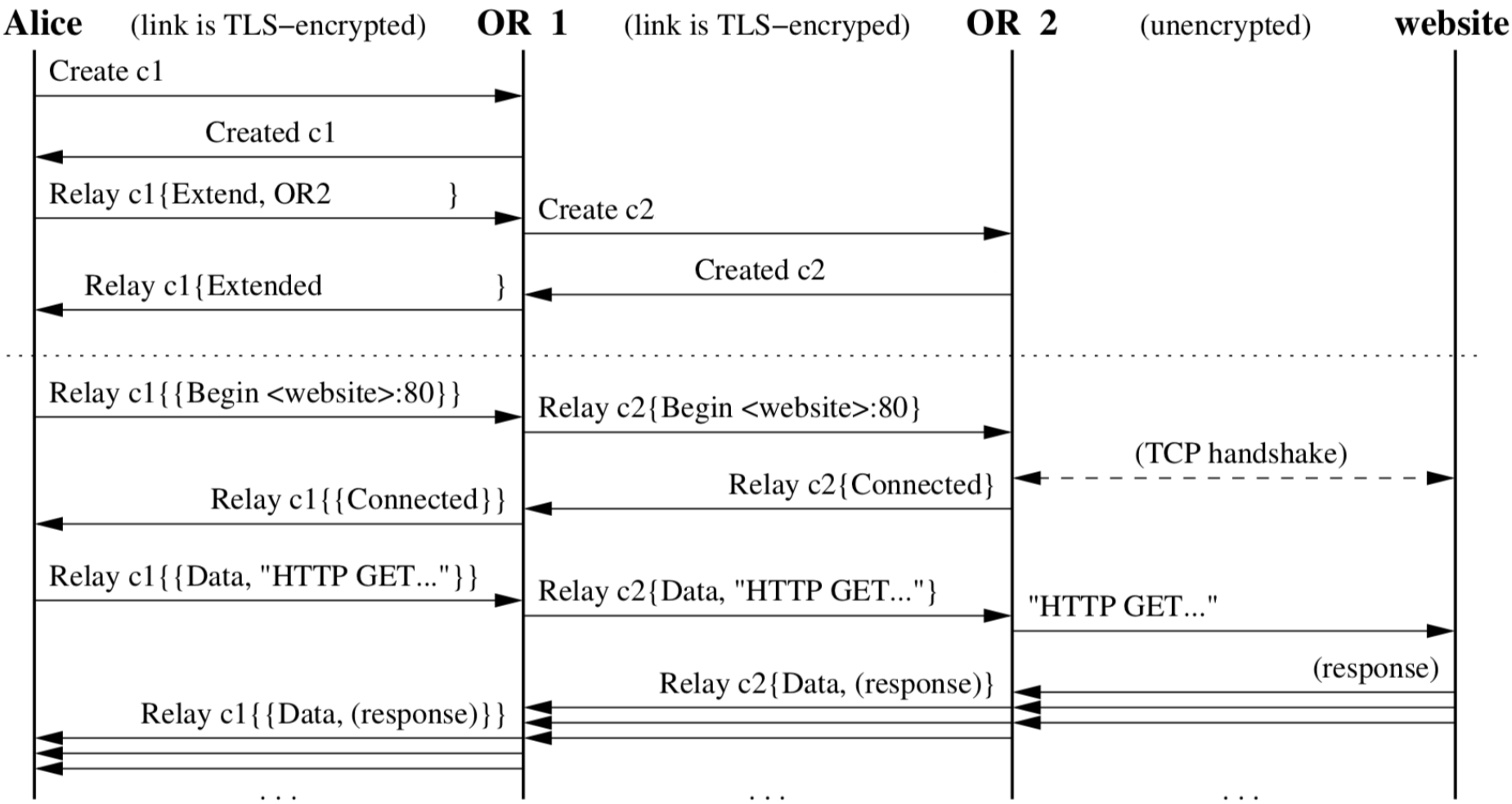
Onion Routing



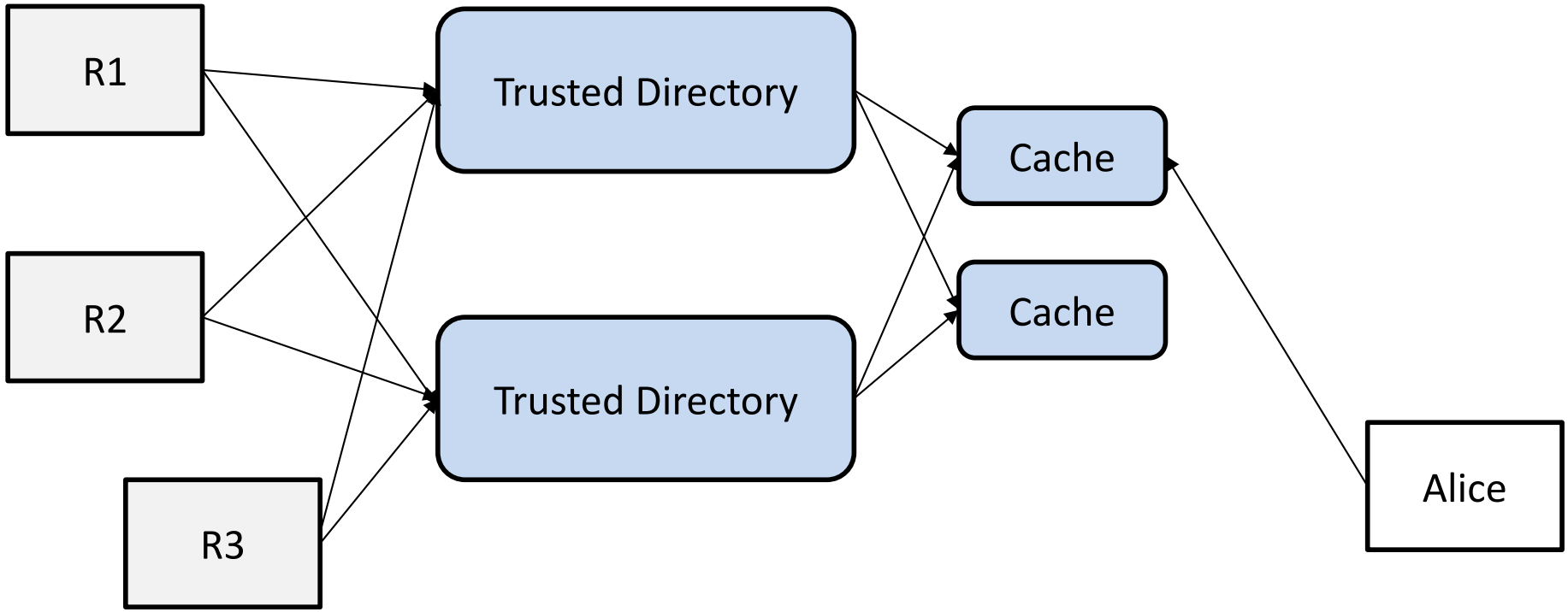
Cells and Circuits

- Alice builds *circuits* by chaining TOR onion routers
- TOR traffic is composed by *cells*
  - Each cell is 512 bytes (both for headers and payload)
  - Each cell header has a *circuit identifier (circID)*
  - Cells can contain just control-data (i.e., extend the circuit) or payload to be relayed

Building a two-hop Circuit (simplified)



Directory Servers



Explore the active Onion Routers: https://metrics.torproject.org/

Blocking TOR

- Blocking the directory authorities
- Blocking all the relay IP addresses in the directory
- Filtering based on Tor's network fingerprint
- Preventing users from finding the Tor software

Rendezvous Points

- Alice hides their identity when communicating with Bob
- It might be desirable for Bob to hide his identity, as well
- Bob can announce a *hidden service*
  - Announced in the directory servers (using cryptography)
  - Serviced by several TOR circuits that end up to Bob
- Alice can connect to the hidden service using TOR
  - Both parties are now anonymous
  - Alice must know about the service out of band

TOR Attacks

- Several active and passive attacks
- Traffic analysis
- Pollution with controlled ORs
- TOR is based on the voluntary effort of running legitimate ORs

Resources

- TOR project page
  - https://www.torproject.org
- TOR paper
  - https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf