Anonymity

With material from:

Dave Levin and Michelle Mazurek

http://www.sogosurvey.com/static/sogo_resp_images/tat_resp_images/designimg/guaranteed-anonymous-survey.png

- What is anonymity?
- Dining cryptographers
- Mixnets and Tor

What is anonymity?

- An observer/attacker cannot determine who is communicating
- **Sender** anonymity: Cannot distinguish true sender from set of potential senders
- **Receiver** anonymity: Cannot distinguish true receiver from set of potential receivers

Sender anonymity

- Ransom note
- Pass a note when teacher is not looking
- Hang fliers / chalk messages late at night
- etc.

Receiver anonymity

- Dedicate a book/song/etc. to “you know who”
- Codes in classified ads
- Cold war spies: Number stations
- etc.

Quantifying anonymity

- K-anonymity: Can’t distinguish sender/receiver from pool of K potential senders/receivers
- Most of these real-world examples are not “provably” anonymous
- We want something with stronger mathematical properties

Dining cryptographers

https://4.bp.blogspot.com/-uqRm-AGzCxk/T1vLveV9WMI/AAAAAAAAAYY/L5CbKs1fs4M/s1600/baby-girl-eating.jpg

Problem setup

- From David Chaum (optional reading: http://www.cs.ucsb.edu/~ravenben/classes/595n-s07/papers/dcnet-jcrypt88.pdf)
- Three cryptographers having dinner
  - Waiter says someone has paid
  - Was it one of them? Or a third party?
- Can one of them admit to paying without the others knowing which one it was?

How to do it

- Each pair of cryptographers flips one coin, hidden from the 3rd person
- Everyone reports “same” or “different” for the two coins they can see
- **Except**, person who paid reports the wrong answer

Why does this work?

```
A : (b_AB XOR b_AC) XOR m
B : (b_AB XOR b_BC)
C : (b_AC XOR b_BC)

All messages:
 (b_AB XOR b_AB) XOR (b_AC XOR b_AC)
   XOR (b_BC XOR b_BC) XOR m
 = m
```

Why is this secure?

- Suppose you did not pay
- If the result is 1 (odd “diff”)
  - You can tell one of the others is lying
  - But without coin they share, can’t tell which
- If result is 0 (even “diff”) then no anonymity issue
  - We all know the third party paid

Potential issues

- Unfair coins
- Not executing the protocol honestly

Generalizing the protocol

- More than 3 people:
  - Fine with one shared bit per pair of users
- More than 1 bit of data
  - Proceed in rounds, one bit per round
  - Now we need a shared **key** (one bit per round)
- What about collisions?

Pros and Cons

- Pro: Not interactive
  - After key establishment, no crosstalk by users
  - Make systems simpler, proofs easier
- Pro: Collusion is hard
  - Generally need everyone conspiring against you
- Cons:
  - Collisions / Jamming
  - N^2 shared keys

Mixnets

Problem setup

- One mail server, M
  - Lots of senders (S_i) and receivers (R_i)
- One global observer G
- Goal: Send messages without G being able to determine which sender -> which receiver

Strawman protocol

- Every sender sends a message to M
  - Encrypted with M’s pub key
  - Indicates intended receiver
- M waits for all messages; shuffles the order
- Send messages encrypted for recipient

Fixing this protocol (1)

- Problem: Mail server reads all messages
- Solution: Encryption layers
  - $E(k_M, R_i \,||\, E(k_{R_i}, m))$

Fixing this protocol (2)

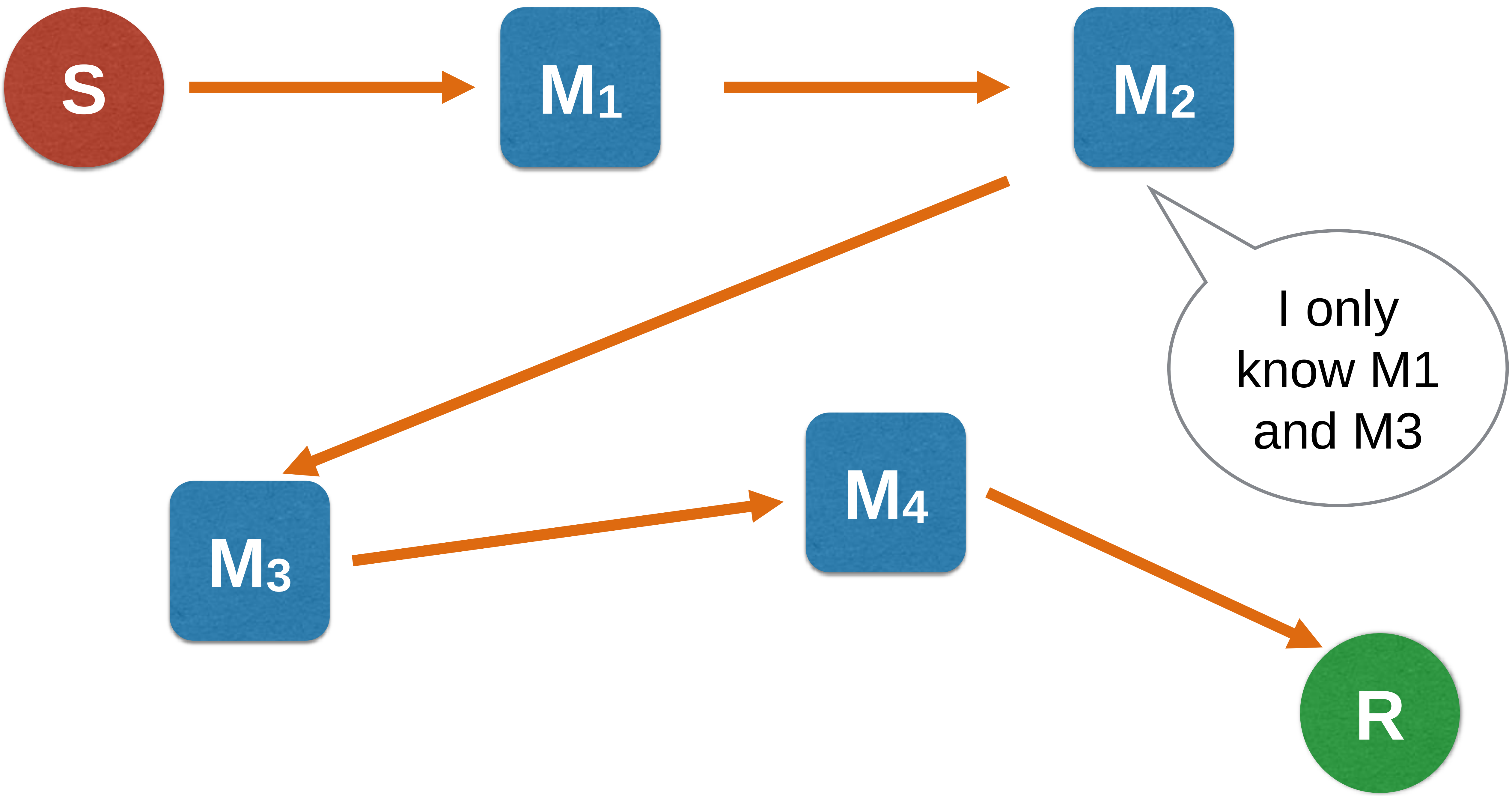
- Problem: What if not everyone has a message
  - Mail server might wait forever!

- Solution: Everyone sends every round
  - Some is labeled as junk
  - Wastes bandwidth/resources on junk

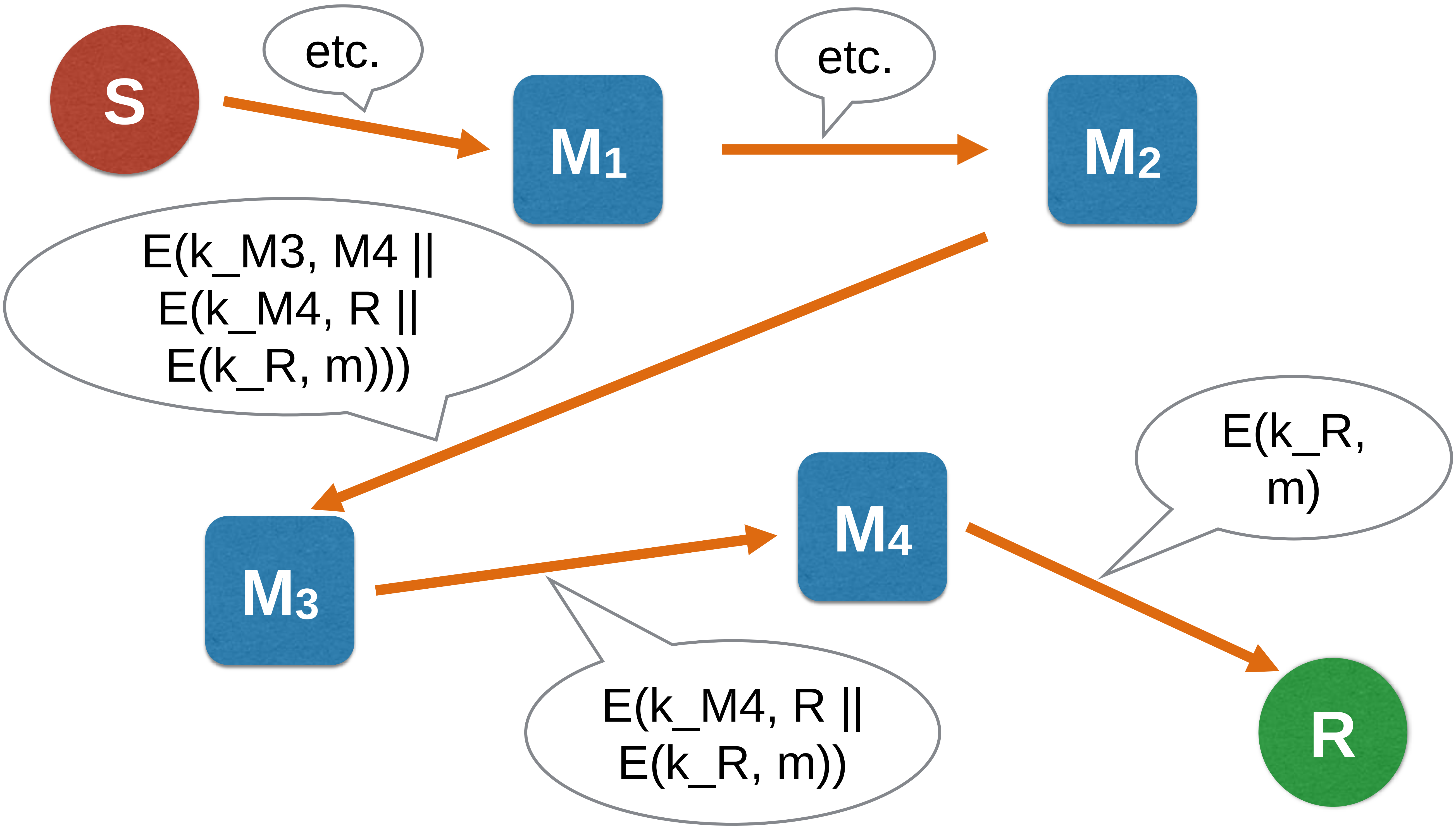
Fixing this protocol (3)

- Problem: Mail server knows who talks to who
- Solution: Chain of mail servers
- …. wrapped in layers
- …. like an ***onion***

Only know your links



Encryption layers



Tor: The Onion Router

- This layers idea is the basis for Tor
- End-to-end path = a circuit
  - Default = 3-hop circuits
- ***Exit node:*** last hop before destination
  - Nodes decide whether to exit, for where

Tor vs. Mix-nets

- Tor doesn't assume global observer
  - Instead, some (small) proportion of Tor nodes are assumed to be malicious
  - Instead, eavesdroppers on a fraction of links
- As a result, does not batch/delay packets
  - Which would not be very practical for many use-cases, e.g. web browsing
- Relies on lots of ***cover traffic***!

Confirmation vs. analysis

- If you suspect Alice is talking to Bob
  - Watch both ends
  - ***Confirm*** via timing, volume

- Tor instead aims to prevent analysis attacks
  - Figure out who Alice is talking to

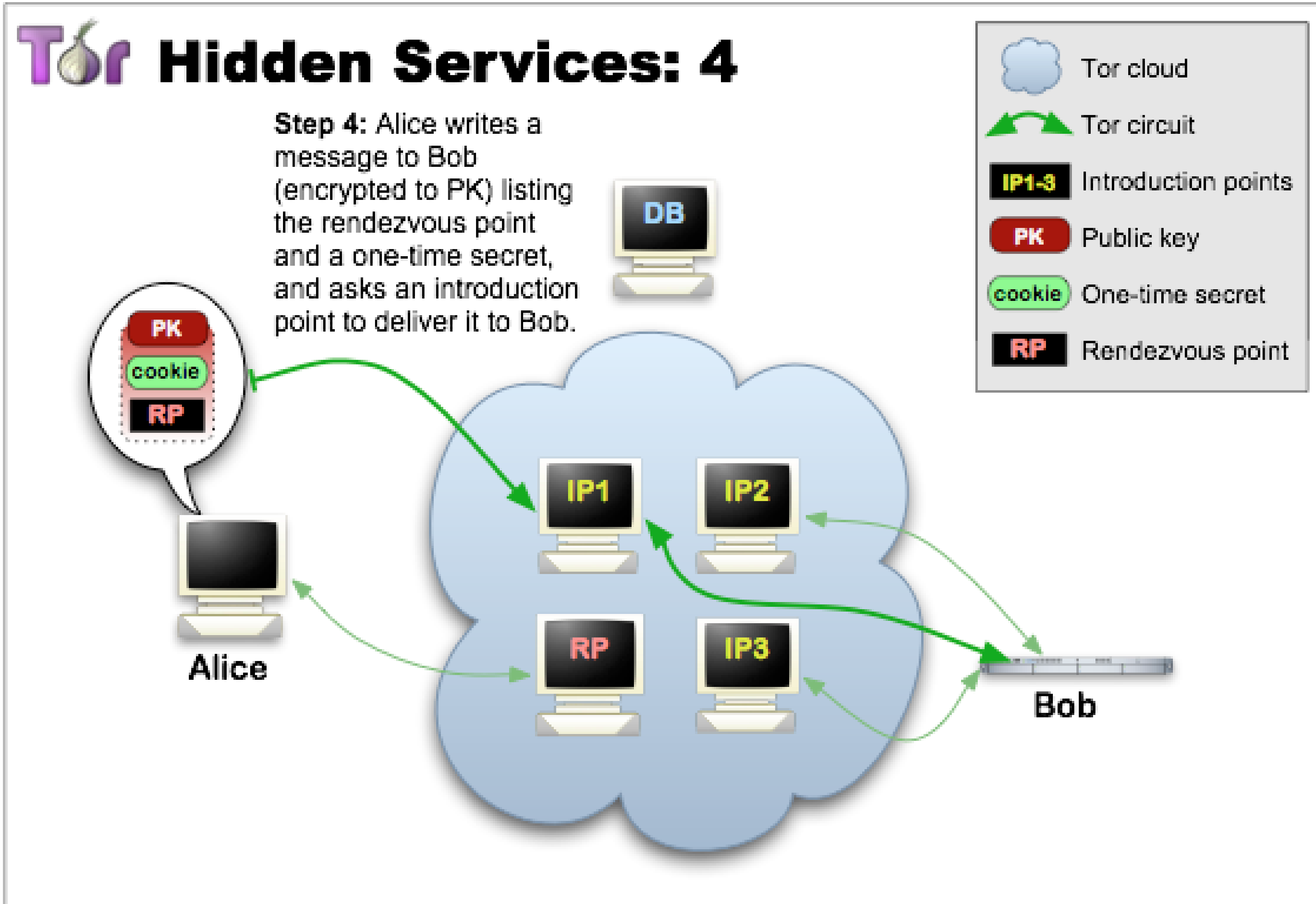
Something is still missing …

- We have disguised senders, what about receivers?
- Goal: Run service X on host D
  - Without anyone knowing D runs it
  - ***hidden service***
  - (aka, dark web)

Hidden services

- Bob creates his service
  - Set up circuits to *introduction points*
  - Create a directory listing that maps X to points

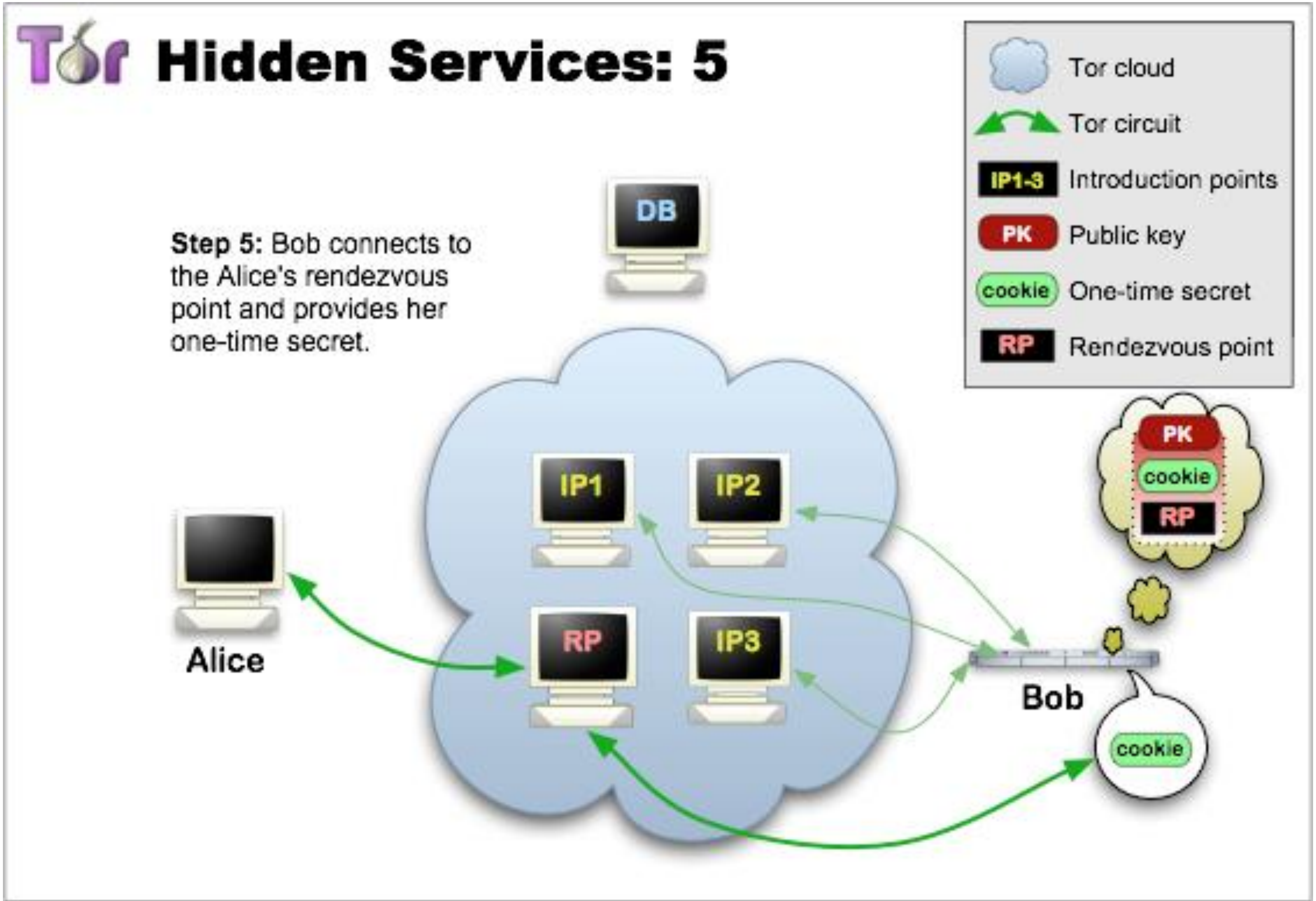
- Alice wants to connect
  - Set up circuit to *rendezvous point* R
  - Associate with unique token I
  - Set up circuit to one of the intro points
  - Send message: Please forward R, I to X



https://www.torproject.org/docs/hidden-services.html.en

Hidden services (2)

- Connection via R
  - Bob sends message containing I to R
  - R links the two circuits together (forwarding)
  - Alice and Bob can now talk anonymously



https://www.torproject.org/docs/hidden-services.html.en

Who knows what?

- Only Bob knows he runs service X
- Intro point knows someone accessed X, but not who
- R knows someone accessed a hidden service, but not who or what
- Alice knows she accessed X, but not who/where X is

Potential Tor attacks

- Insert malicious relays into the network
  - Or compromise legitimate ones
  - Generally need multiple to be useful
- DOS on trustworthy routers
  - Drive traffic toward your relay
- DOS more generally
  - Force relay to do expensive crypto a lot

More Tor problems

- Exit nodes can be blamed for abusive actions
  - Limits desire to be an exit node
  - Monitor exit nodes for traffic analysis