CSE 484 / CSE M 584: Computer Security and Privacy

Anonymity

Spring 2016

Franziska (Franzi) Roesner
franzi@cs.washington.edu

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Admin

- **Project Checkpoint #2:** today at 8pm
- **Lab #3:** Friday 8pm
- **Final Project:** Monday 8pm

Privacy on Public Networks

- Internet is designed as a public network
  - Machines on your LAN may see your traffic, network routers see all traffic that passes through them
- Routing information is public
  - IP packet headers identify source and destination
  - Even a passive observer can easily figure out who is talking to whom
- Encryption does not hide identities
  - Encryption hides payload, but not routing information
  - Even IP-level encryption (tunnel-mode IPSec/ESP) reveals IP addresses of IPSec gateways

Questions

Q1: What is anonymity?

Q2: Why might people want anonymity on the Internet?

Q3: Why might people **not** want anonymity on the Internet?

Applications of Anonymity (I)

- Privacy
  - Hide online transactions, Web browsing, etc. from intrusive governments, marketers and archivists
- Untraceable electronic mail
  - Corporate whistle-blowers
  - Political dissidents
  - Socially sensitive communications (online AA meeting)
  - Confidential business negotiations
- Law enforcement and intelligence
  - Sting operations and honeypots
  - Secret communications on a public network

Applications of Anonymity (II)

- Digital cash
  - Electronic currency with properties of paper money (online purchases unlinkable to buyer's identity)
- Anonymous electronic voting
- Censorship-resistant publishing

What is Anonymity?

- Anonymity is the state of being not identifiable within a set of subjects
  - You cannot be anonymous by yourself!
    - Big difference between anonymity and confidentiality
  - Hide your activities among others' similar activities
- Unlinkability of action and identity
  - For example, sender and email he/she sends are no more related after observing communication than before
- Unobservability (hard to achieve)
  - Observer cannot even tell whether a certain action took place or not

Part 1: Anonymity in Datasets

How to release an anonymous dataset?

- Possible approach: remove identifying information from datasets?

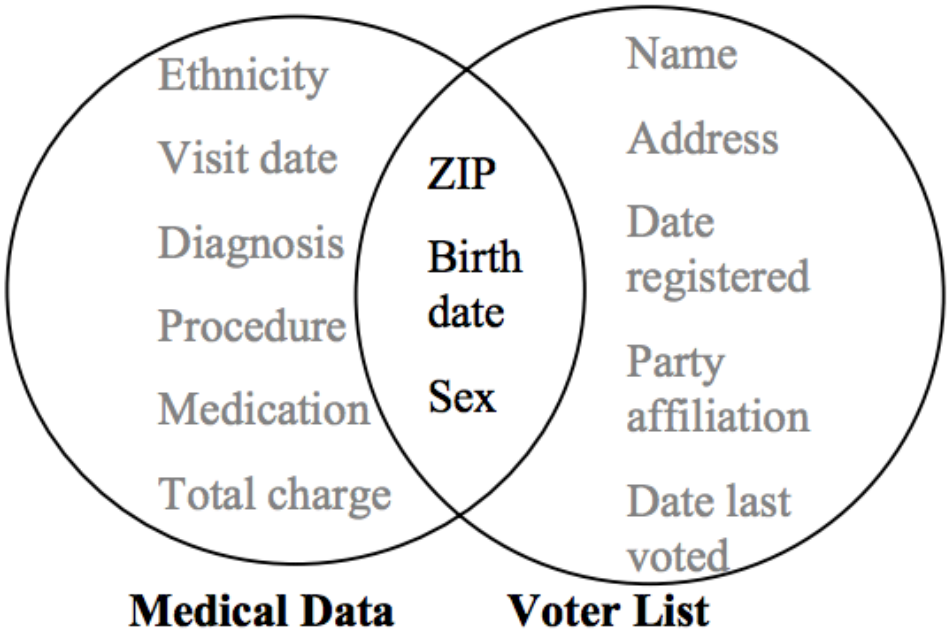


Figure 1 Linking to re-identify data

Massachusetts medical+voter data [Sweeney 1997]

k-Anonymity

- Each person contained in the dataset cannot be distinguished from at least k-1 others in the data.

Name	Age	Gender	State of domicile	Religion	Disease
*	20 < Age ≤ 30	Female	Tamil Nadu	*	Cancer
*	20 < Age ≤ 30	Female	Kerala	*	Viral infection
*	20 < Age ≤ 30	Female	Tamil Nadu	*	TB
*	20 < Age ≤ 30	Male	Karnataka	*	No illness
*	20 < Age ≤ 30	Female	Kerala	*	Heart-related
*	20 < Age ≤ 30	Male	Karnataka	*	TB
*	Age ≤ 20	Male	Kerala	*	Cancer
*	20 < Age ≤ 30	Male	Karnataka	*	Heart-related
*	Age ≤ 20	Male	Kerala	*	Heart-related
*	Age ≤ 20	Male	Kerala	*	Viral infection

Doesn't work for high-dimensional datasets (which tend to be **sparse**)

[Dwork et al.]

Differential Privacy

- **Setting:** Trusted party has a database
- **Goal:** allow queries on the database that are useful but preserve the privacy of individual records
- **Differential privacy intuition:** add noise so that an output is produced with similar probability whether any single input is included or not
- Privacy of the computation, not of the dataset

Part 2: Anonymity in Communication

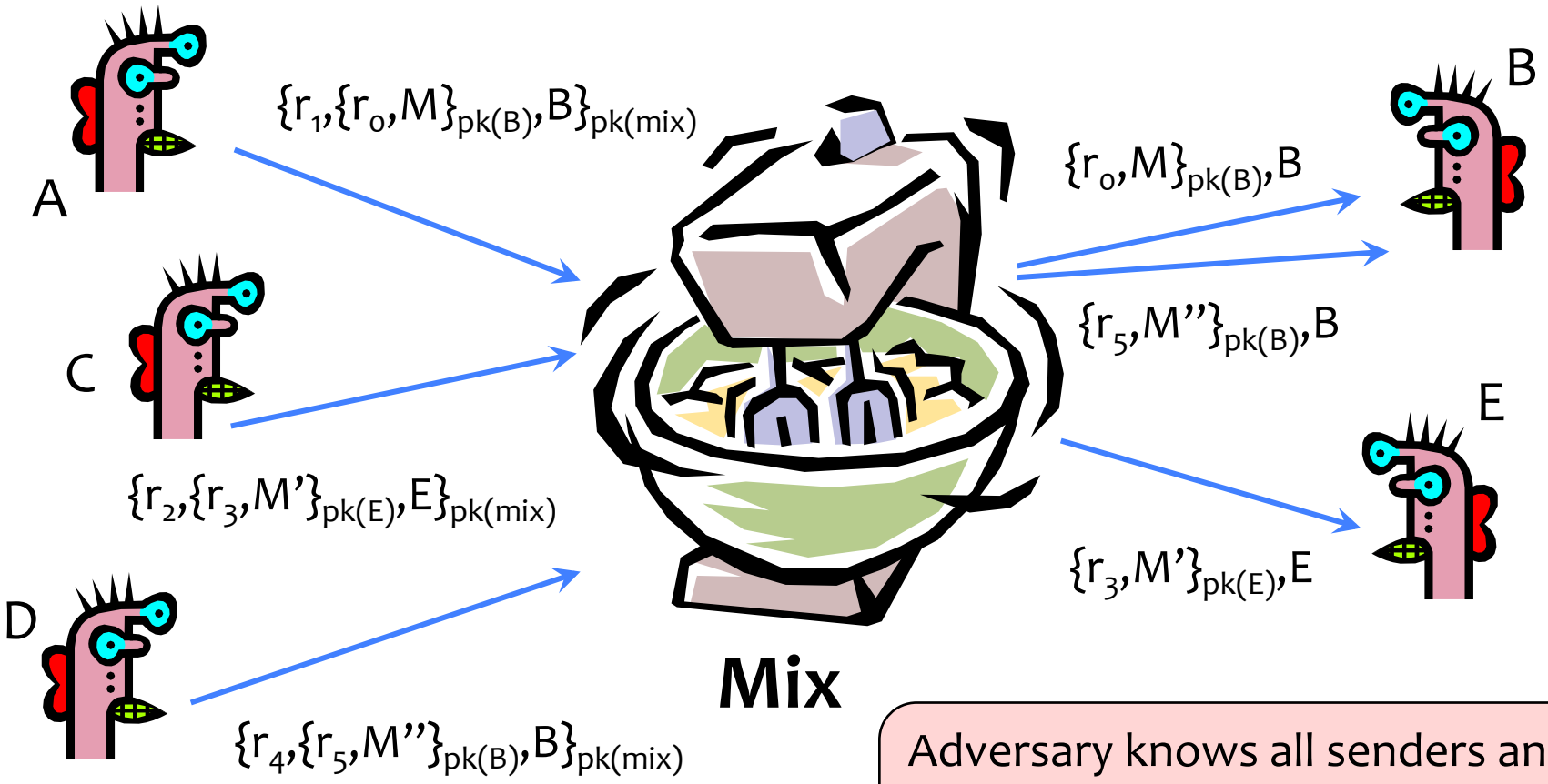
Chaum's Mix

- Early proposal for anonymous email
  - David Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms". Communications of the ACM, February 1981.

Before spam, people thought anonymous email was a good idea ☺

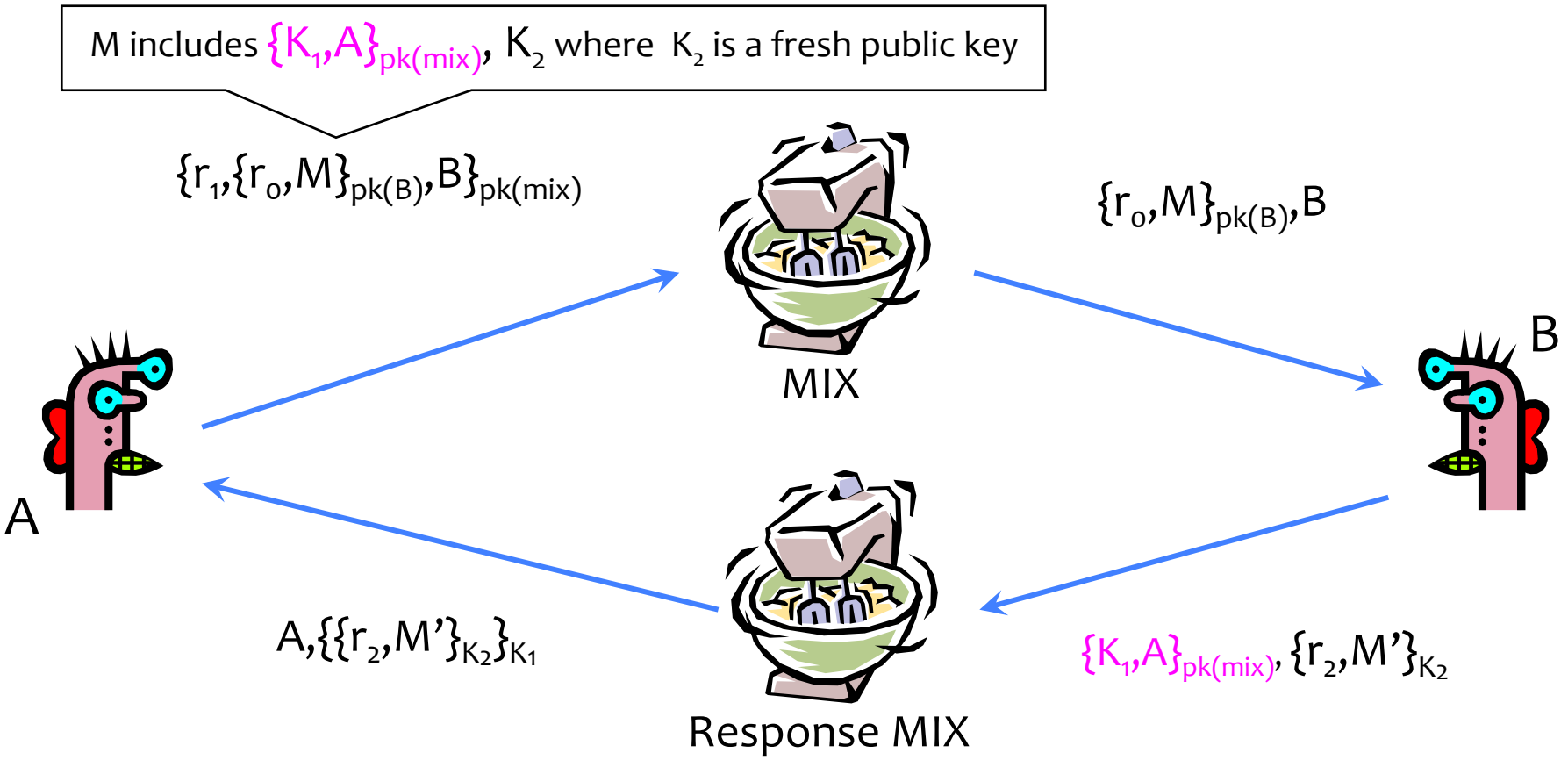
- Public key crypto + trusted re-mailer (Mix)
  - Untrusted communication medium
  - Public keys used as persistent pseudonyms
- Modern anonymity systems use Mix as the basic building block

Basic Mix Design



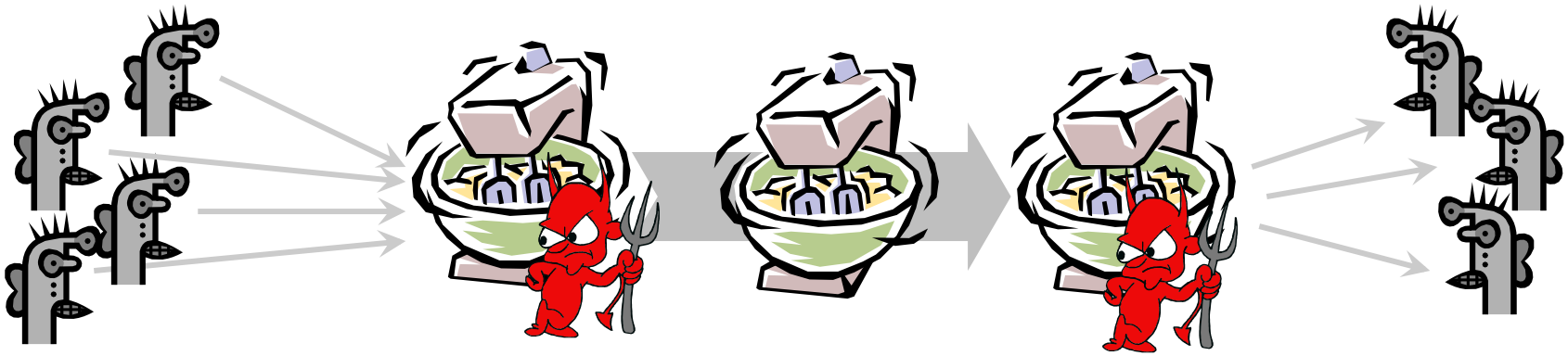
Adversary knows all senders and all receivers, but cannot link a sent message with a received message

Anonymous Return Addresses



Secrecy without authentication
(good for an online confession service ☺)

Mix Cascades and Mixnets

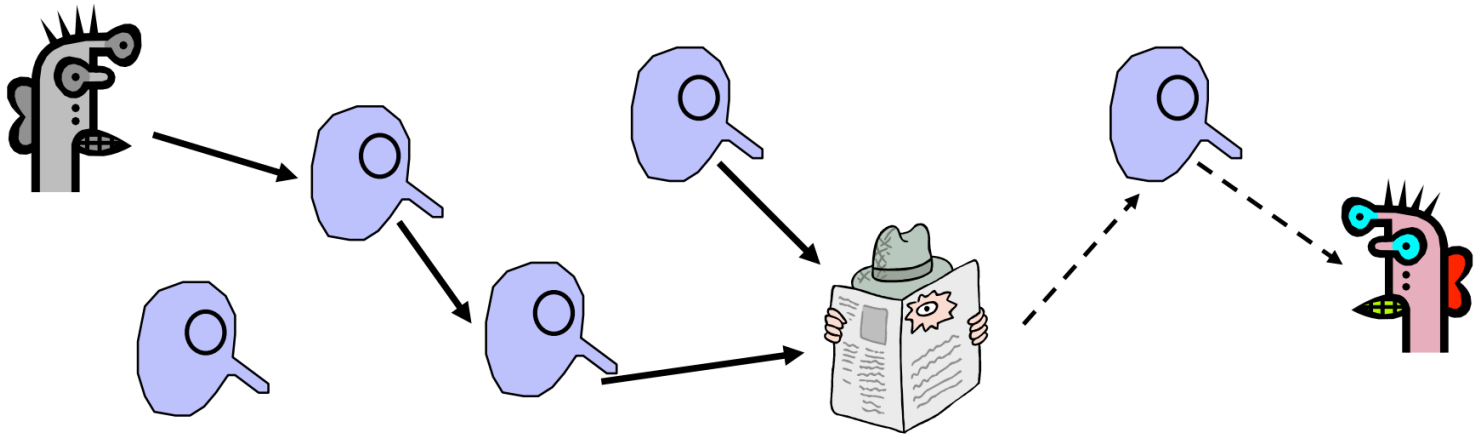


- Messages are sent through a sequence of mixes
  - Can also form an arbitrary network of mixes ("mixnet")
- Some of the mixes may be controlled by attacker, but even a single good mix ensures anonymity
- Pad and buffer traffic to foil correlation attacks

Disadvantages of Basic Mixnets

- Public-key encryption and decryption at each mix are computationally expensive
- Basic mixnets have high latency
  - OK for email, not OK for anonymous Web browsing
- Challenge: low-latency anonymity network

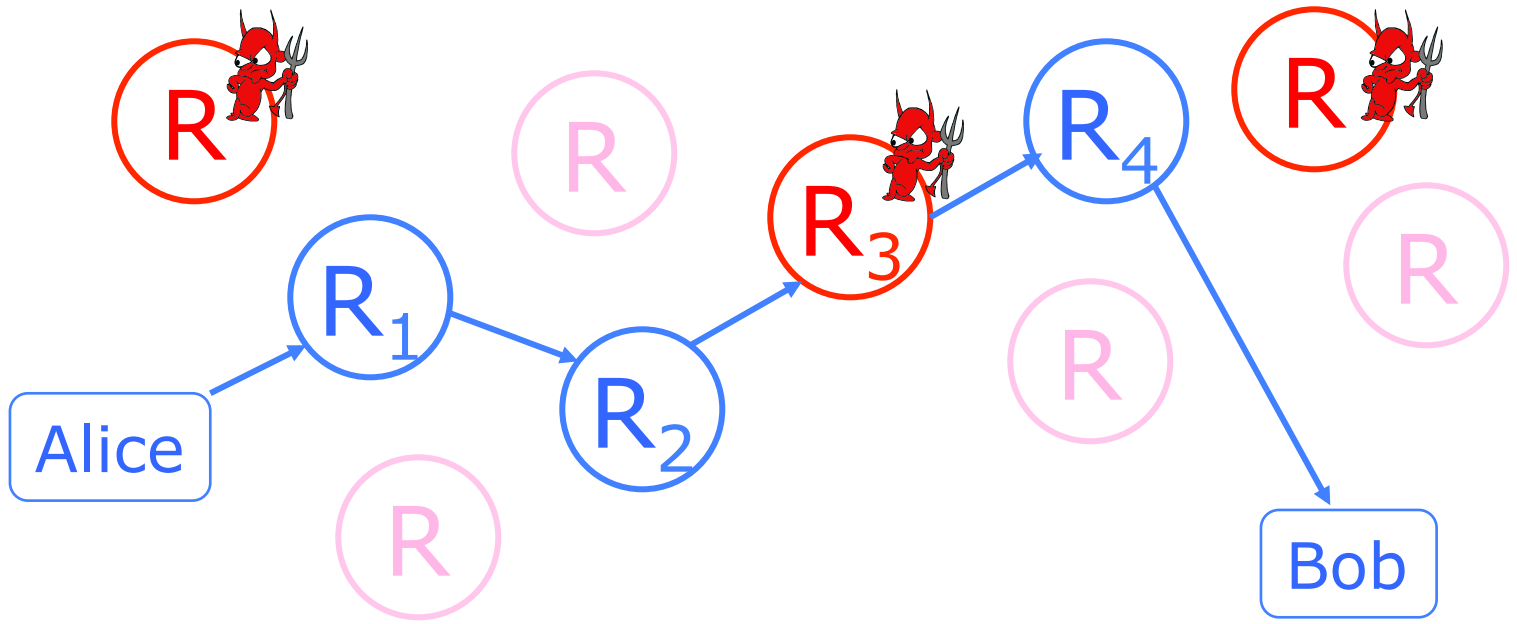
Another Idea: Randomized Routing



- Hide message source by routing it randomly
  - Popular technique: Crowds, Freenet, Onion routing
- Routers don't know for sure if the apparent source of a message is the true sender or another router

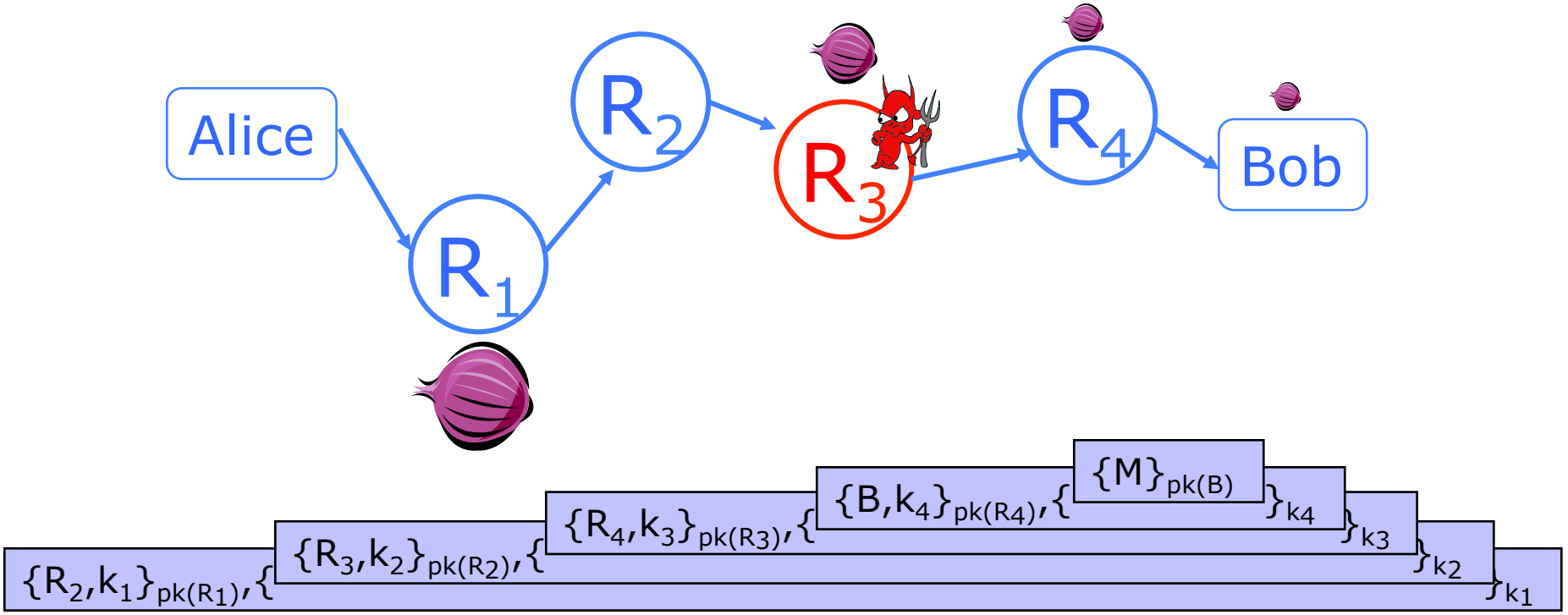
[Reed, Syverson, Goldschlag 1997]

Onion Routing



- Sender chooses a random sequence of routers
  - Some routers are honest, some controlled by attacker
  - Sender controls the length of the path

Route Establishment



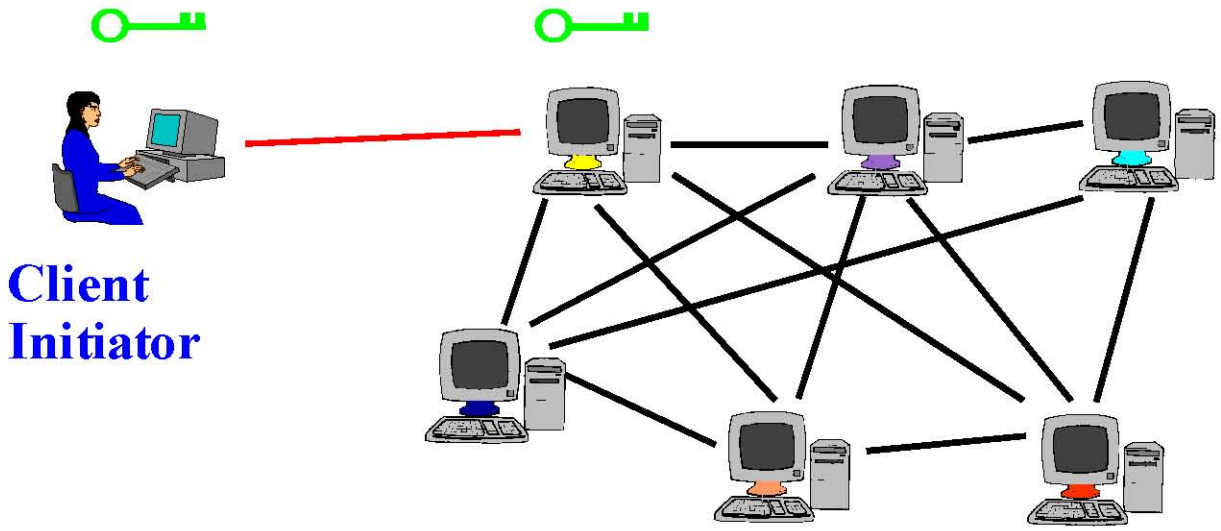
- Routing info for each link encrypted with router's public key
- Each router learns only the identity of the next router

Tor

- Second-generation onion routing network
  - http://tor.eff.org
  - Developed by Roger Dingledine, Nick Mathewson and Paul Syverson
  - Specifically designed for low-latency anonymous Internet communications
- Running since October 2003
- "Easy-to-use" client proxy
  - Freely available, can use it for anonymous browsing

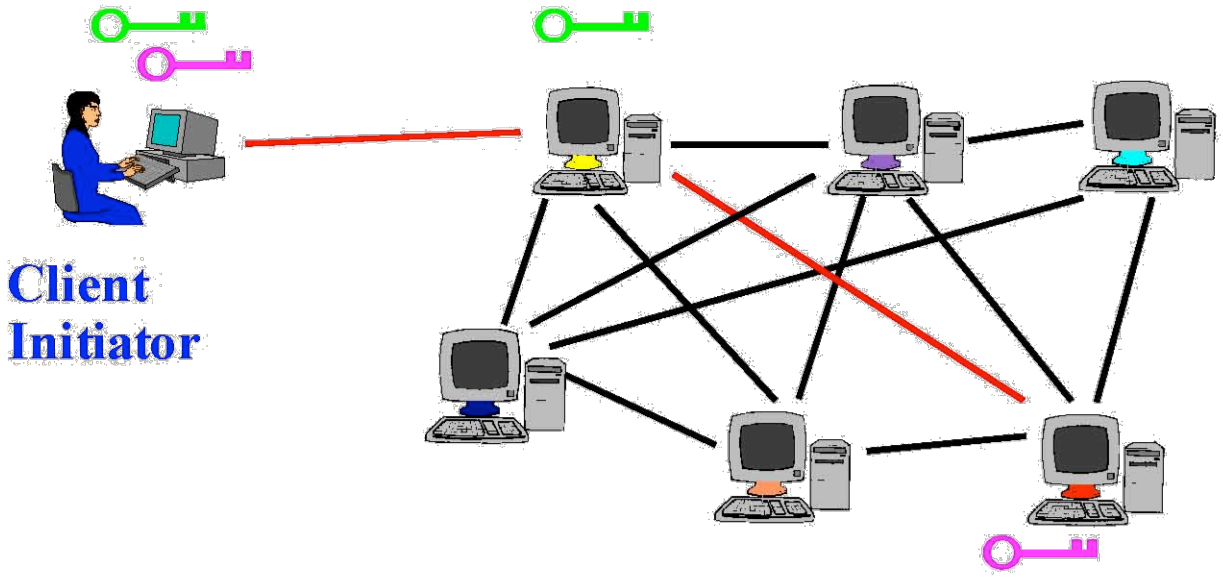
Tor Circuit Setup (1)

- Client proxy establishes a symmetric session key and circuit with Onion Router #1



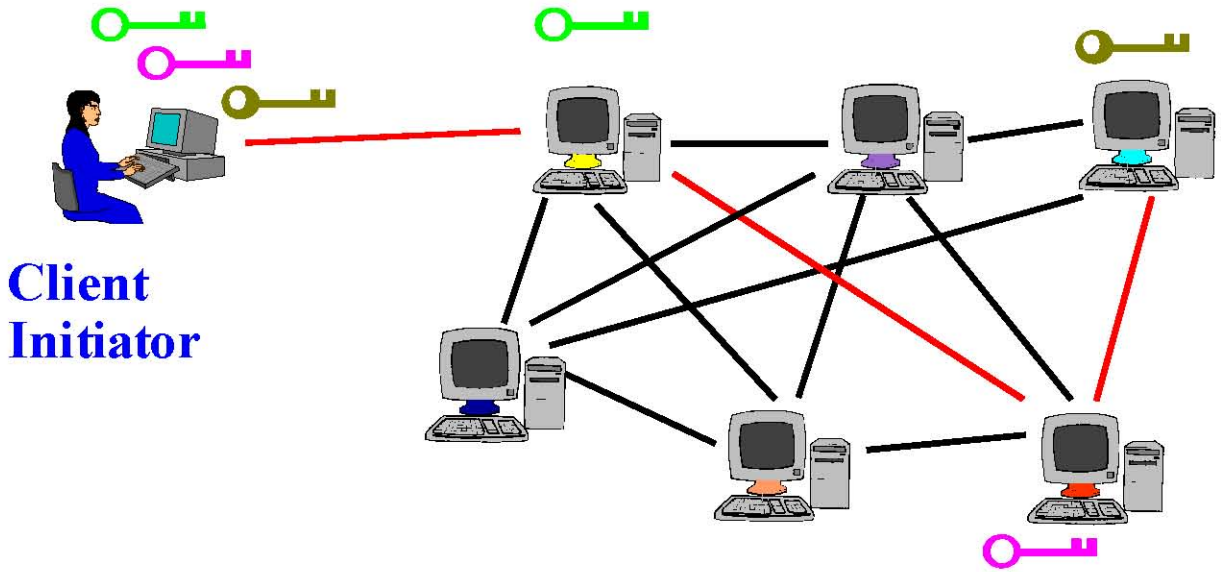
Tor Circuit Setup (2)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #2
  - Tunnel through Onion Router #1



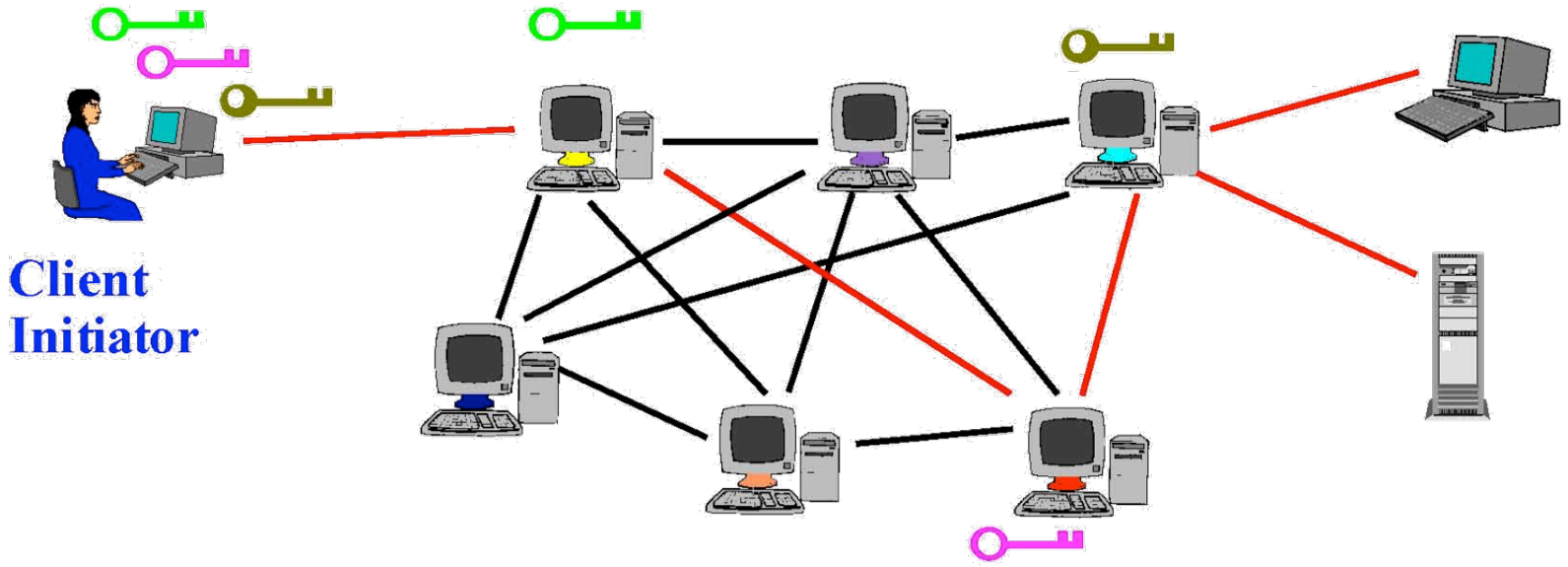
Tor Circuit Setup (3)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #3
  - Tunnel through Onion Routers #1 and #2



Using a Tor Circuit

- Client applications connect and communicate over the established Tor circuit.



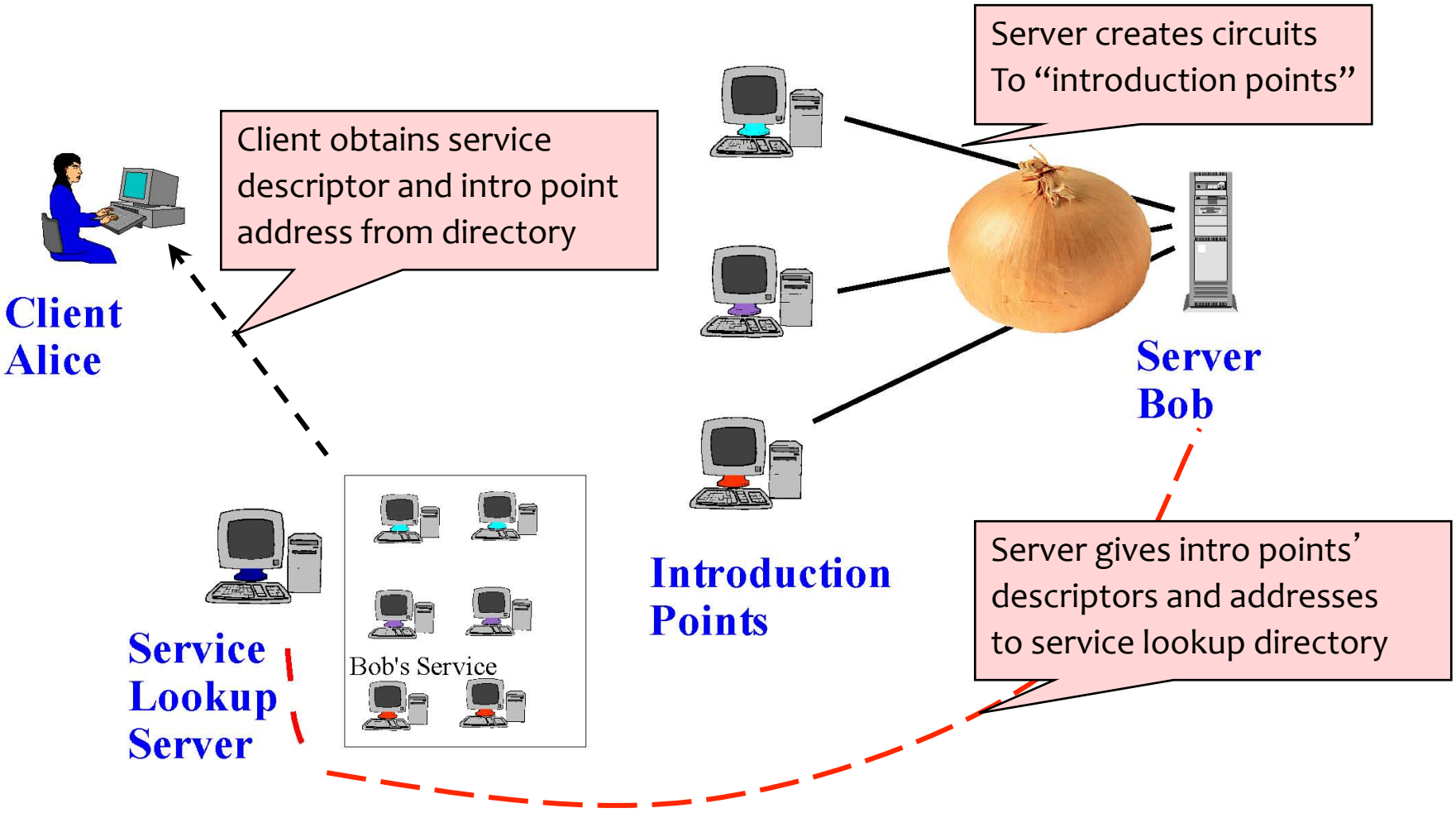
Tor Management Issues

- Many applications can share one circuit
  - Multiple TCP streams over one anonymous connection
- Tor router doesn't need root privileges
  - Encourages people to set up their own routers
  - More participants = better anonymity for everyone
- Directory servers
  - Maintain lists of active onion routers, their locations, current public keys, etc.
  - Control how new routers join the network
    - "Sybil attack": attacker creates a large number of routers
  - Directory servers' keys ship with Tor code

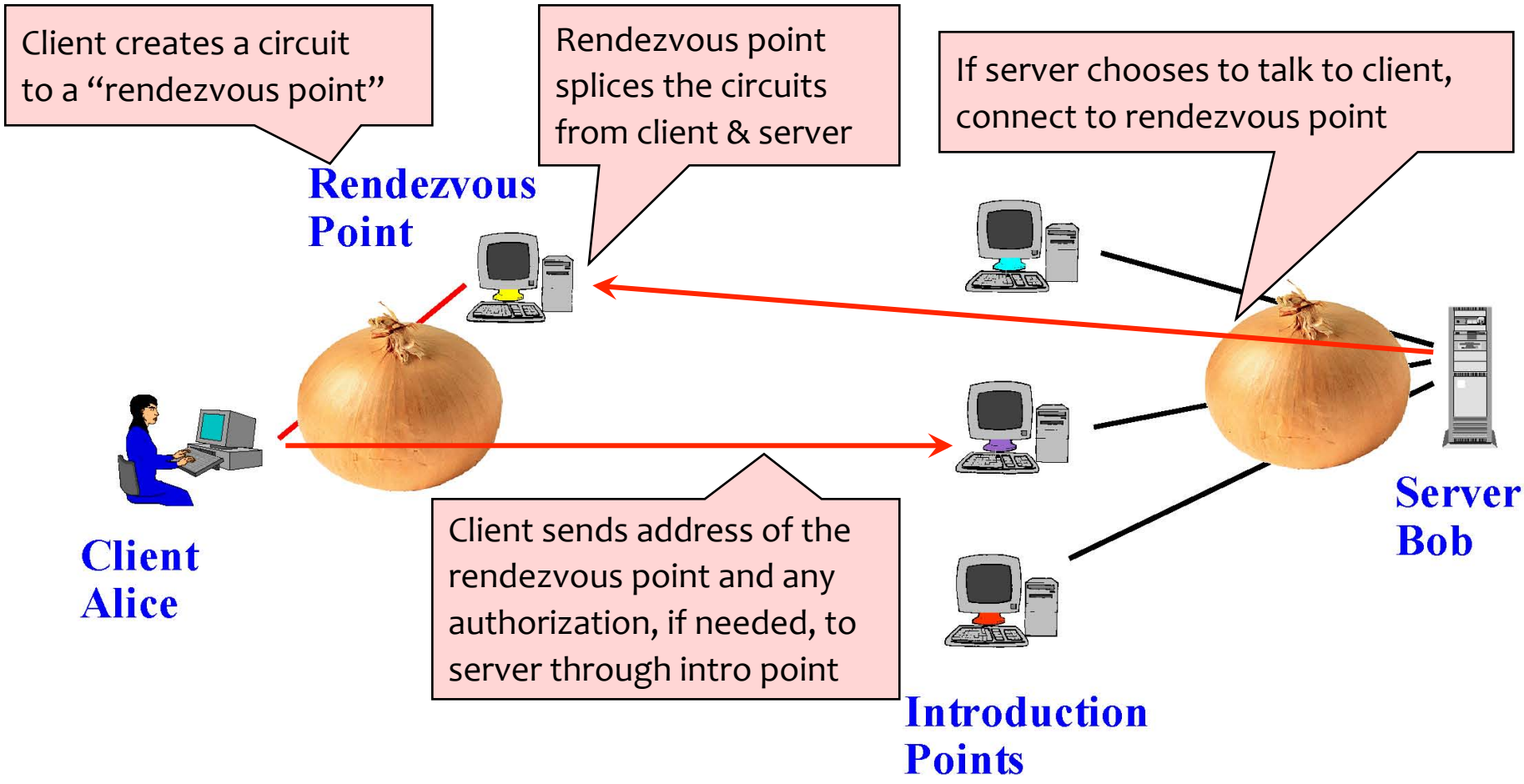
Location Hidden Service

- **Goal:** deploy a server on the Internet that anyone can connect to without knowing where it is or who runs it
- Accessible from anywhere
- Resistant to censorship
- Can survive a full-blown DoS attack
- Resistant to physical attack
  - Can't find the physical server!

Creating a Location Hidden Server



Using a Location Hidden Server



Attacks on Anonymity

- Passive traffic analysis
  - Infer from network traffic who is talking to whom
  - To hide your traffic, must carry other people's traffic!
- Active traffic analysis
  - Inject packets or put a timing signature on packet flow
- Compromise of network nodes
  - Attacker may compromise some routers
  - It is not obvious which nodes have been compromised
    - Attacker may be passively logging traffic
  - Better not to trust any individual router
    - Assume that some fraction of routers is good, don't know which

Deployed Anonymity Systems

- Tor (http://tor.eff.org)
  - Overlay circuit-based anonymity network
  - Best for low-latency applications such as anonymous Web browsing
- Mixminion (http://www.mixminion.net)
  - Network of mixes
  - Best for high-latency applications such as anonymous email
- Not: YikYak ☺

Some Caution

- Tor isn't completely effective by itself
  - Tracking cookies, fingerprinting, etc.
  - Exit nodes can see everything!