



Anonymity on the Internet



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



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
- ◆ Crypto-anarchy
 - "Some people say 'anarchy won't work'. That's not an argument against anarchy; that's an argument against work." – Bob Black



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 his email are no more related after observing communication than they were before
- ◆ Unobservability (hard to achieve)
 - Any item of interest (message, event, action) is indistinguishable from any other item of interest



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



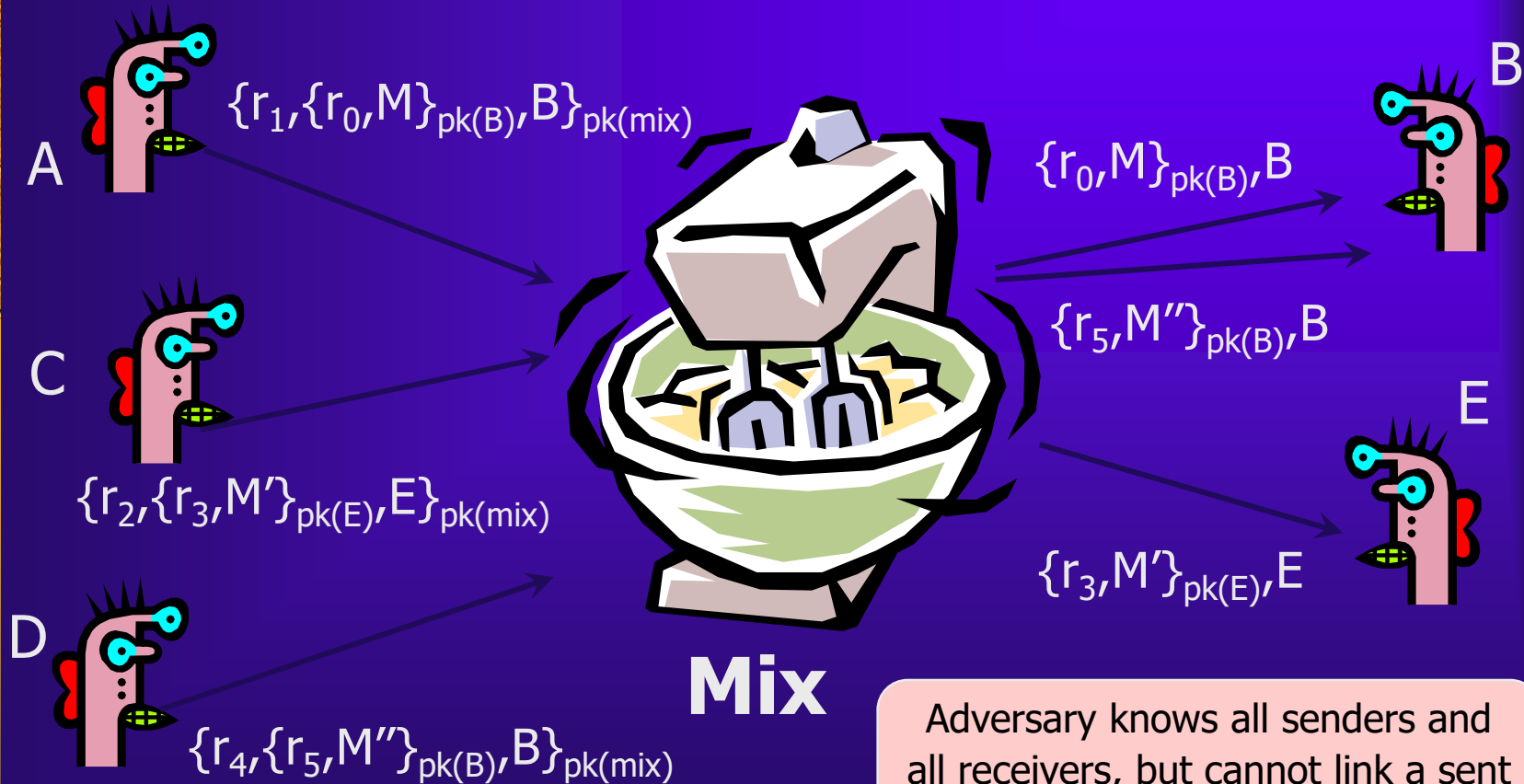
Chaum's Mix

- ◆ 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

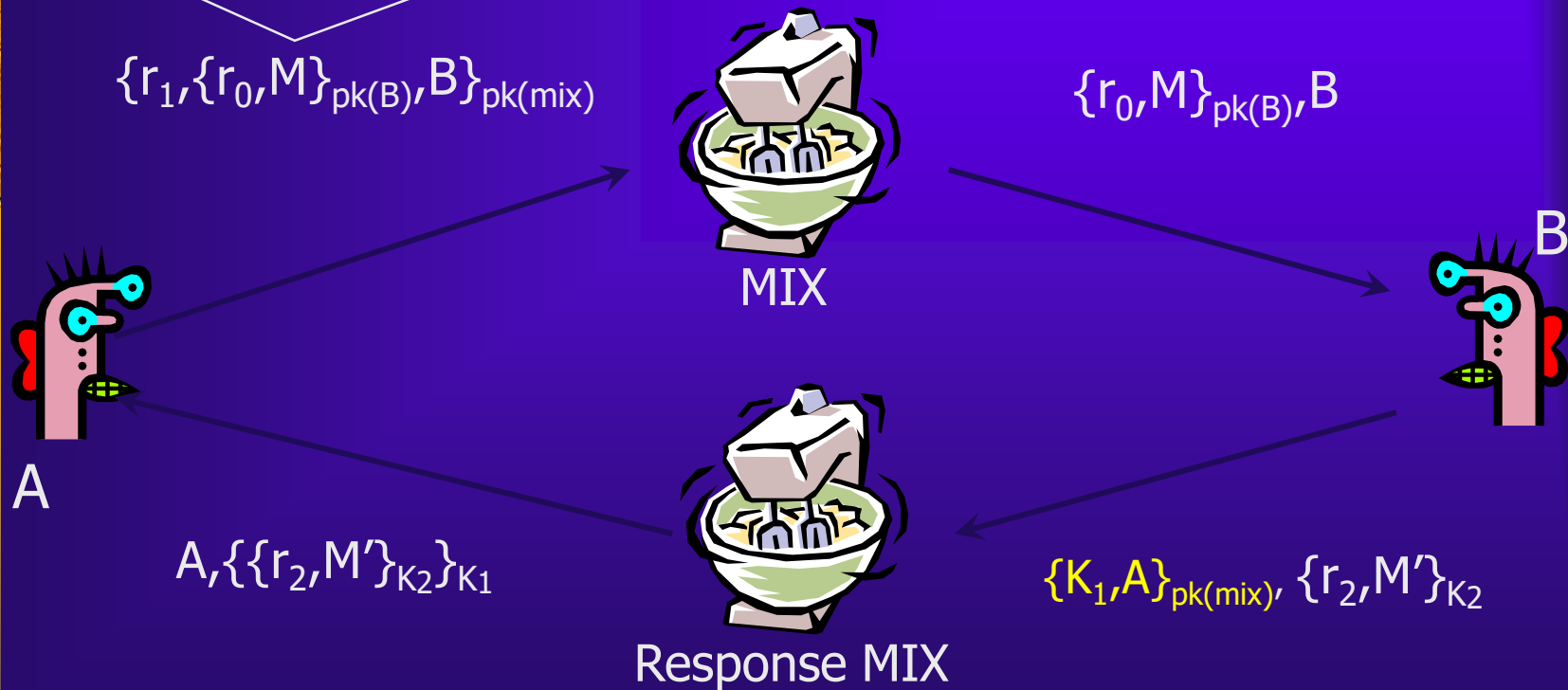
Basic Mix Design



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

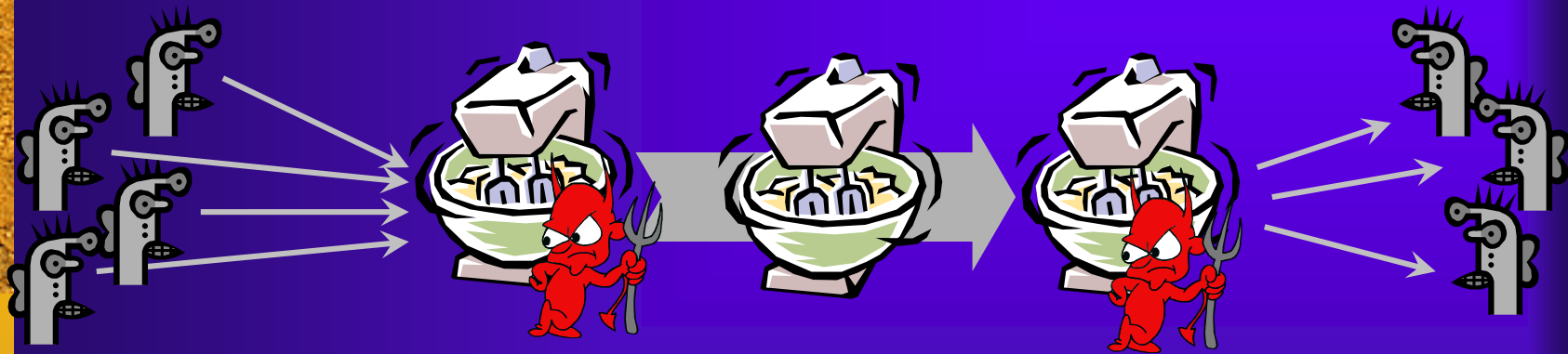
Anonymous Return Addresses

M includes $\{K_1, A\}_{pk(mix)}$, K_2 where K_2 is a fresh public key



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

Mix Cascade



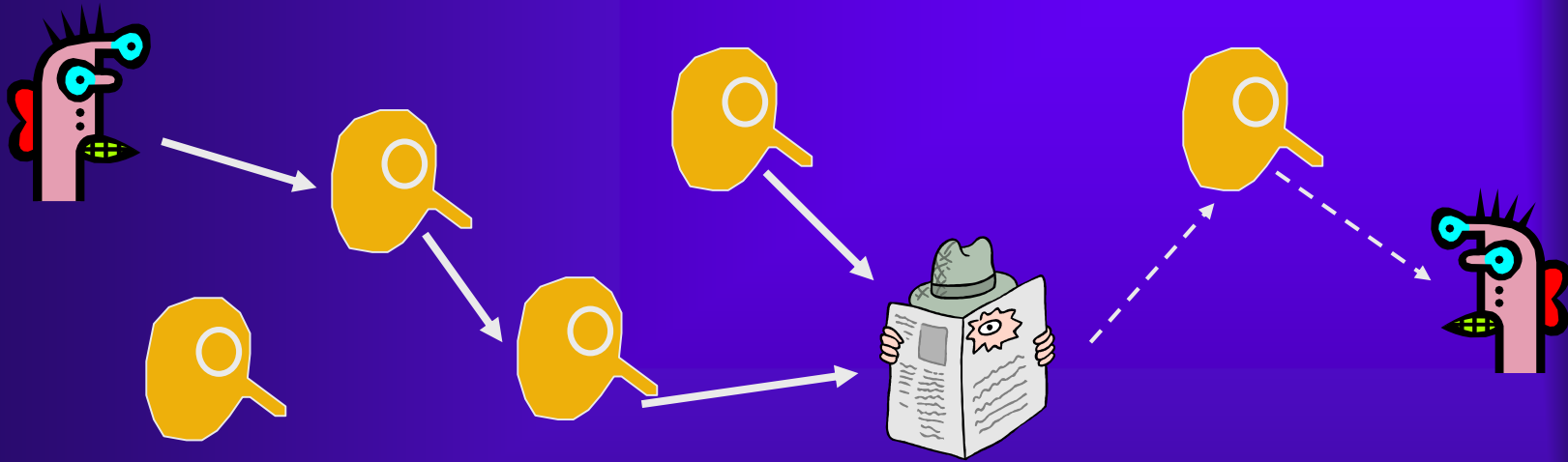
- ◆ Messages are sent through a **sequence of mixes**
 - Can also form an arbitrary network of mixes ("mixnet")
- ◆ Some of the mixes may be controlled by attacker, but even a single good mix guarantees anonymity
- ◆ Pad and buffer traffic to foil correlation attacks



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
 - Use public-key cryptography to establish a “circuit” with pairwise symmetric keys between hops on the circuit
 - Then use symmetric decryption and re-encryption to move data messages along the established circuits
 - Each node behaves like a mix; anonymity is preserved even if some nodes are compromised

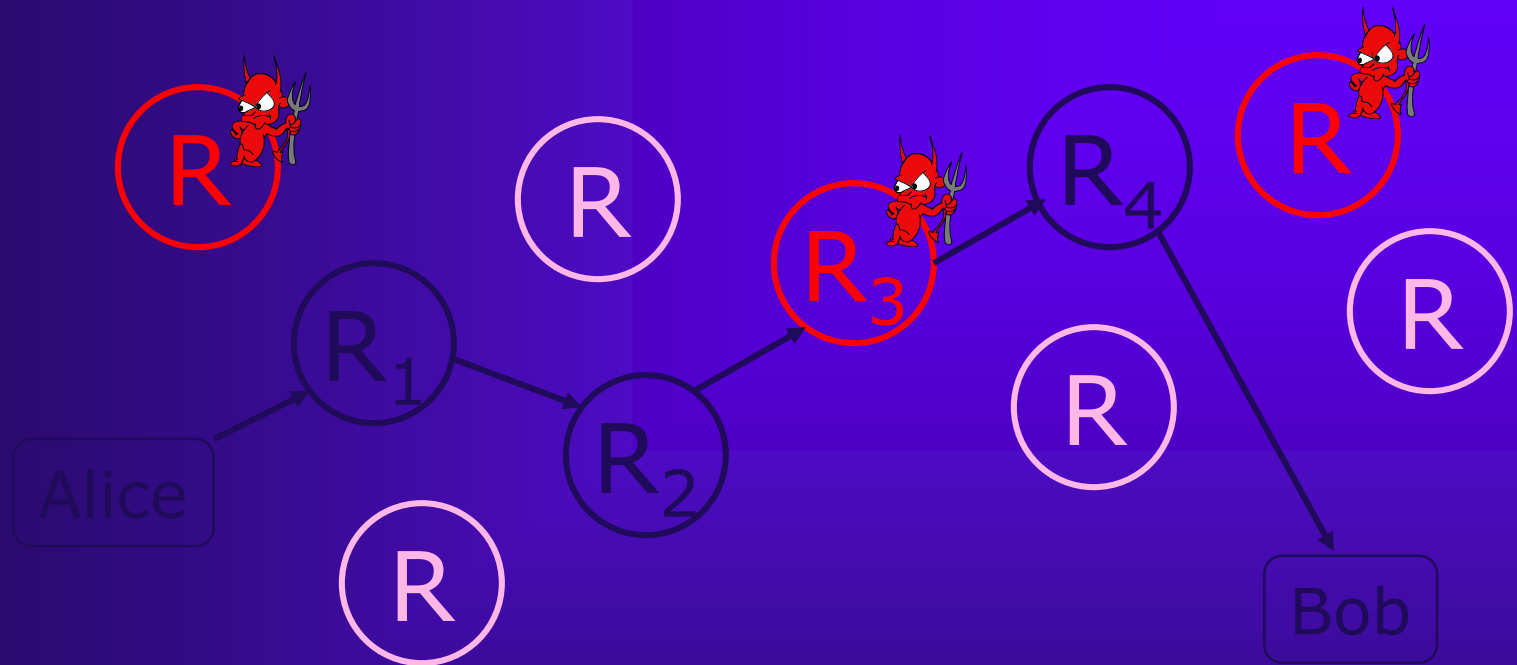
Another Idea: Randomized Routing



- ◆ 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

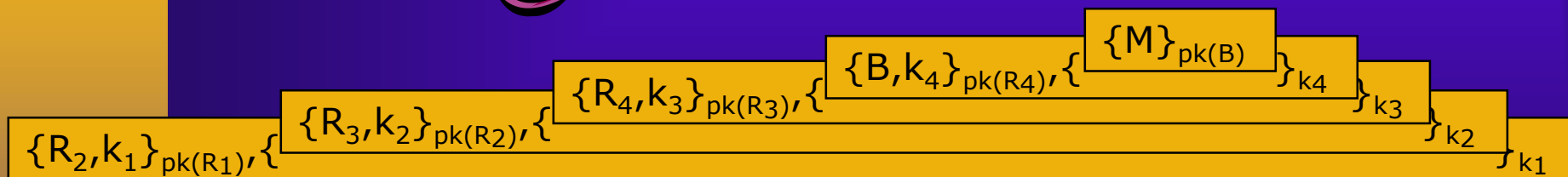
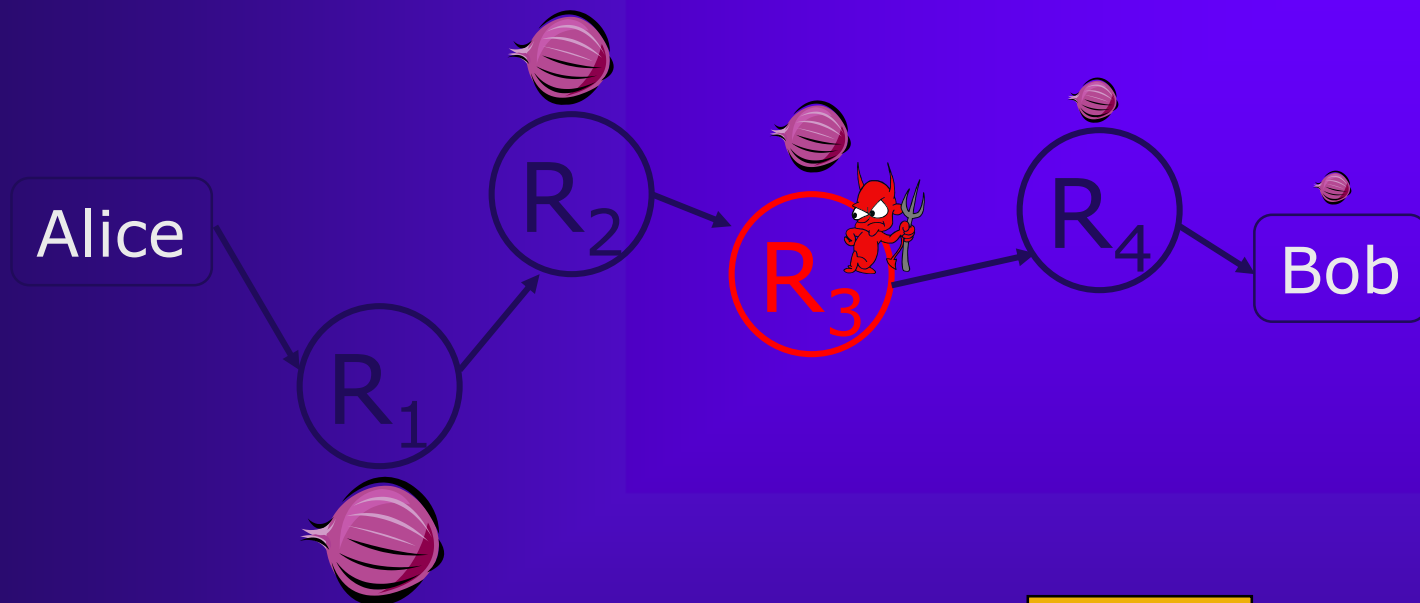
Onion Routing

[Reed, Syverson, Goldschlag '97]



- ◆ 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

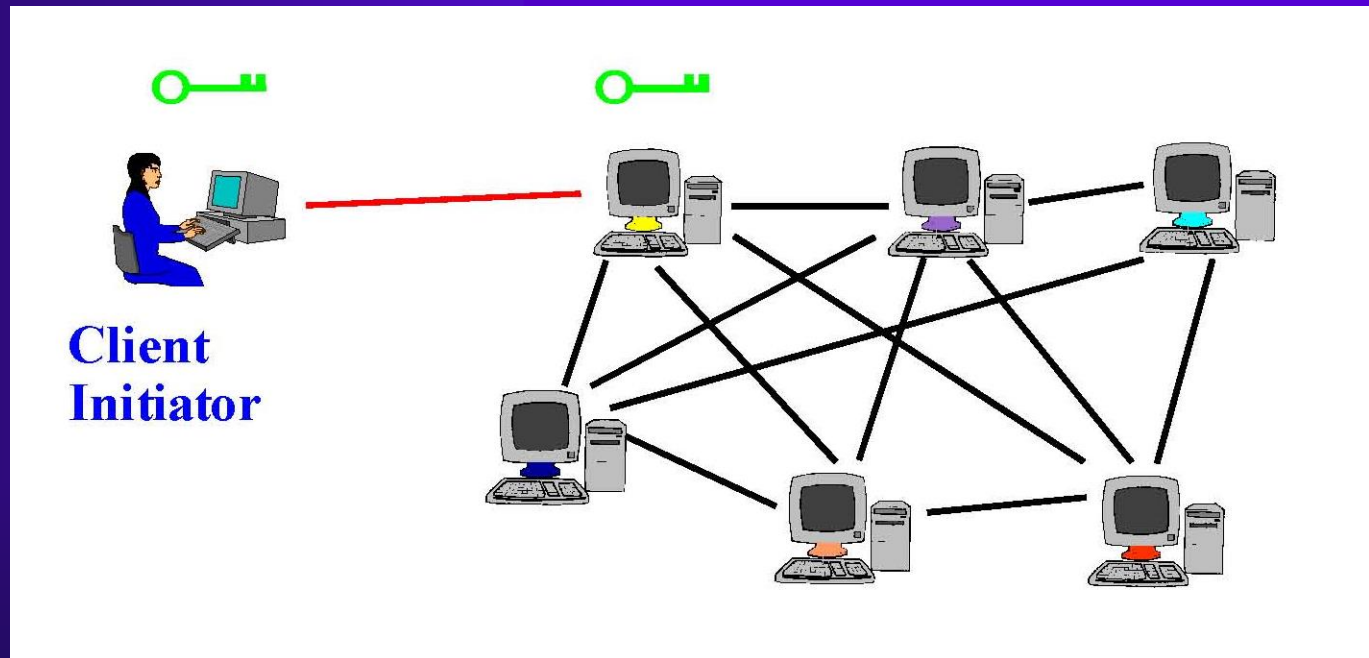


Tor

- ◆ 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
- ◆ 100 nodes on four continents, thousands of users
- ◆ “Easy-to-use” client proxy
 - Freely available, can use it for anonymous browsing


Tor Circuit Setup (1)

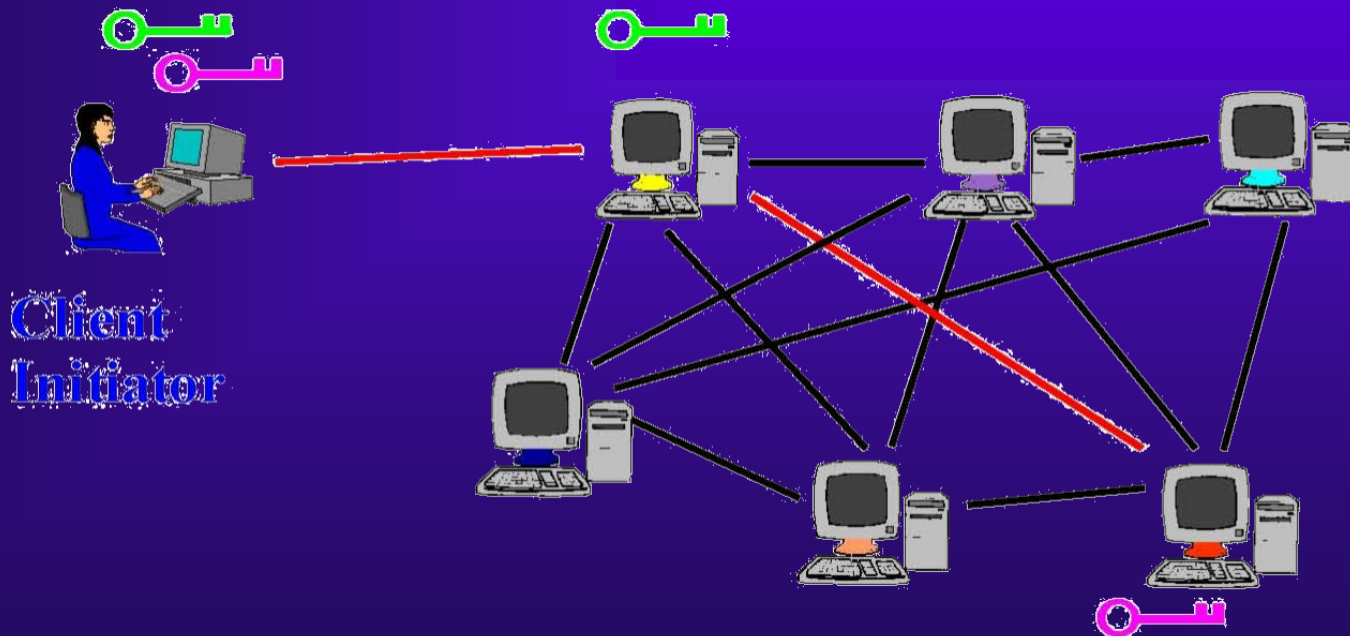
- ◆ Client proxy establish a symmetric session key and circuit with Onion Router #1





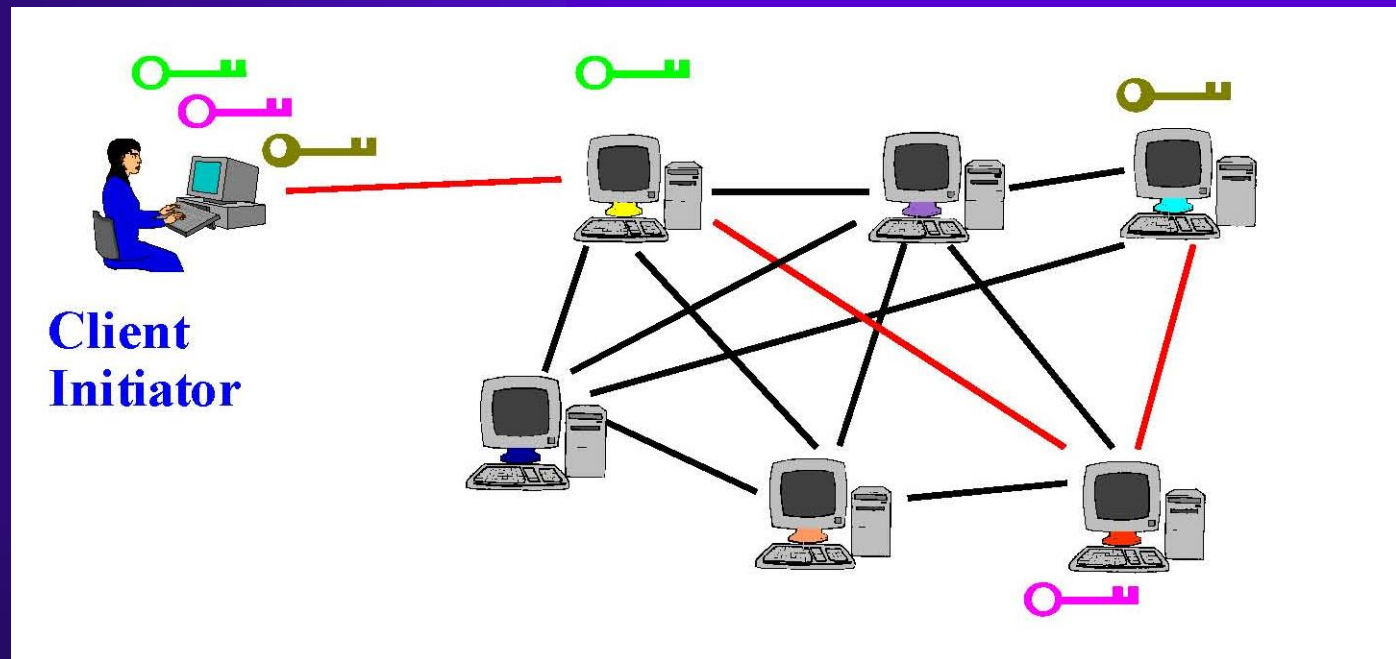
Tor Circuit Setup (2)

- ◆ Client proxy extends the circuit by establishing a symmetric session key with Onion Router #2
 - Tunnel through Onion Router #1 (don't need )



Tor Circuit Setup (3)

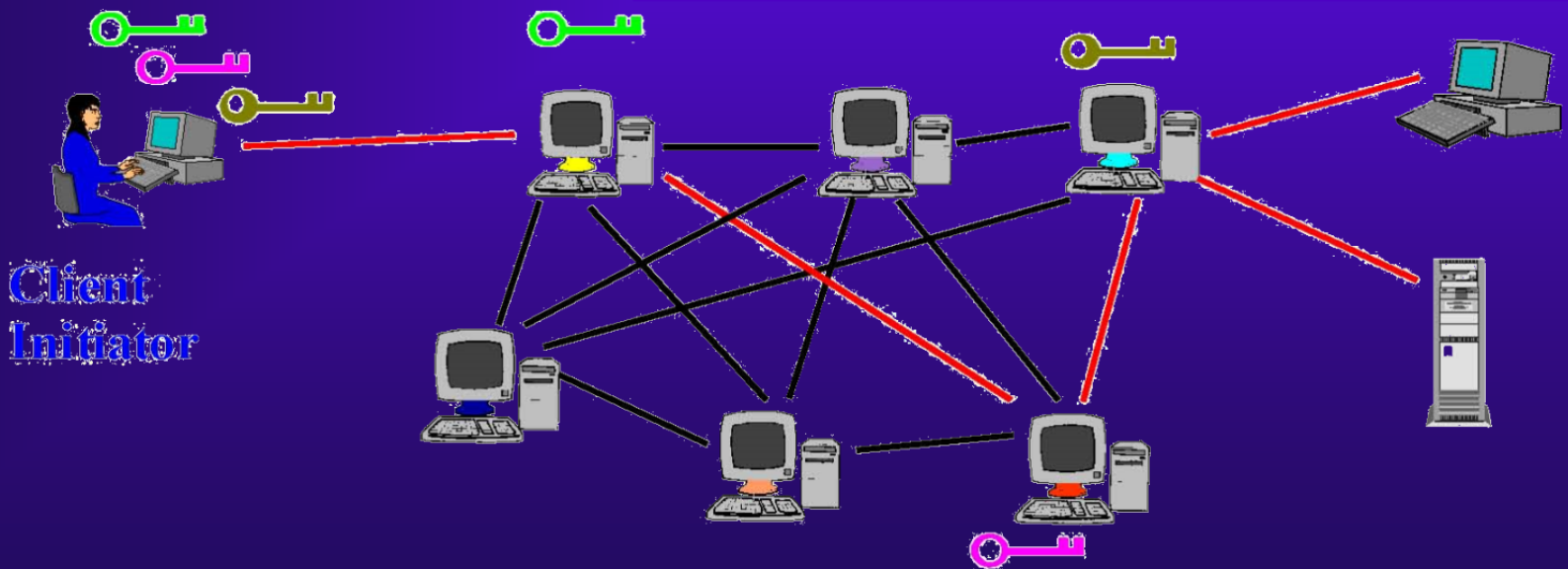
- ◆ Client proxy extends the circuit by establishing a symmetric session key with Onion Router #3
 - Tunnel through Onion Routers #1 and #2





Using a Tor Circuit

- ◆ Client applications connect and communicate over the established Tor circuit
 - Datagrams are decrypted and re-encrypted at each link





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 Servers

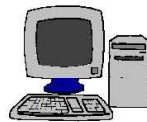
- ◆ Goal: deploy a server on the Internet that anyone can connect to without knowing where it is or who runs it
- ◆ Accessible from anywhere
- ◆ Resistant to censorship
- ◆ Can survive full-blown DoS attack
- ◆ Resistant to physical attack
 - Can't find the physical server!

Creating a Location Hidden Server

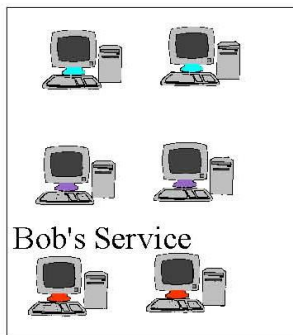


**Client
Alice**

Client obtains service descriptor and intro point address from directory

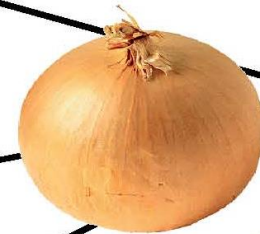


**Service
Lookup
Server**



**Introduction
Points**

Server creates onion routes to "introduction points"



**Server
Bob**

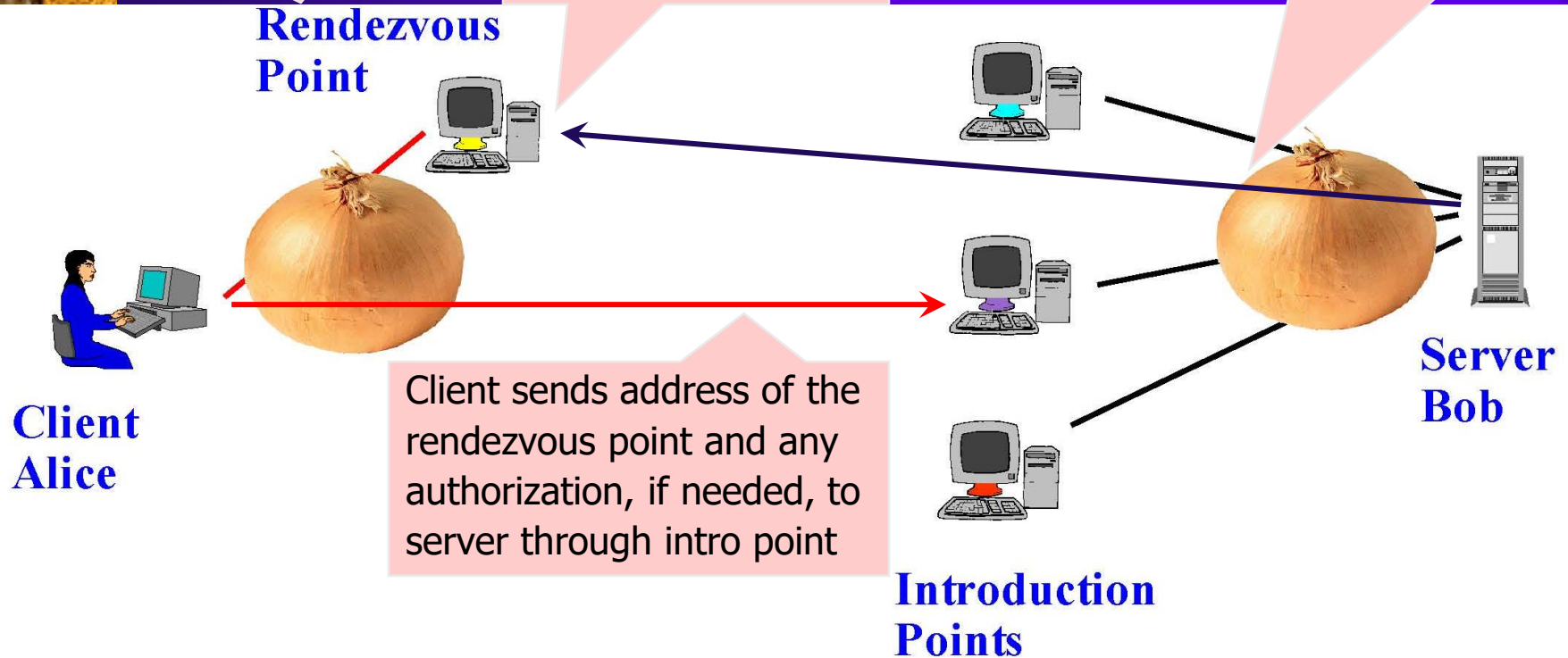
Server gives intro points' descriptors and addresses to service lookup directory

Using a Location Hidden Server

Client creates onion route to a "rendezvous point"

Rendezvous point mates the circuits from client & server

If server chooses to talk to client, connect to rendezvous point





Deployed Anonymity Systems

- ◆ Free Haven project has an excellent bibliography on anonymity
 - Linked from the reference section of course website
- ◆ 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



Dining Cryptographers

- ◆ Clever idea how to make a message public in a perfectly untraceable manner
 - David Chaum. "The dining cryptographers problem: unconditional sender and recipient untraceability." Journal of Cryptology, 1988.
- ◆ Guarantees information-theoretic anonymity for message senders
 - This is an unusually strong form of security: defeats adversary who has unlimited computational power
- ◆ Impractical, requires huge amount of randomness
 - In group of size N , need N random bits to send 1 bit



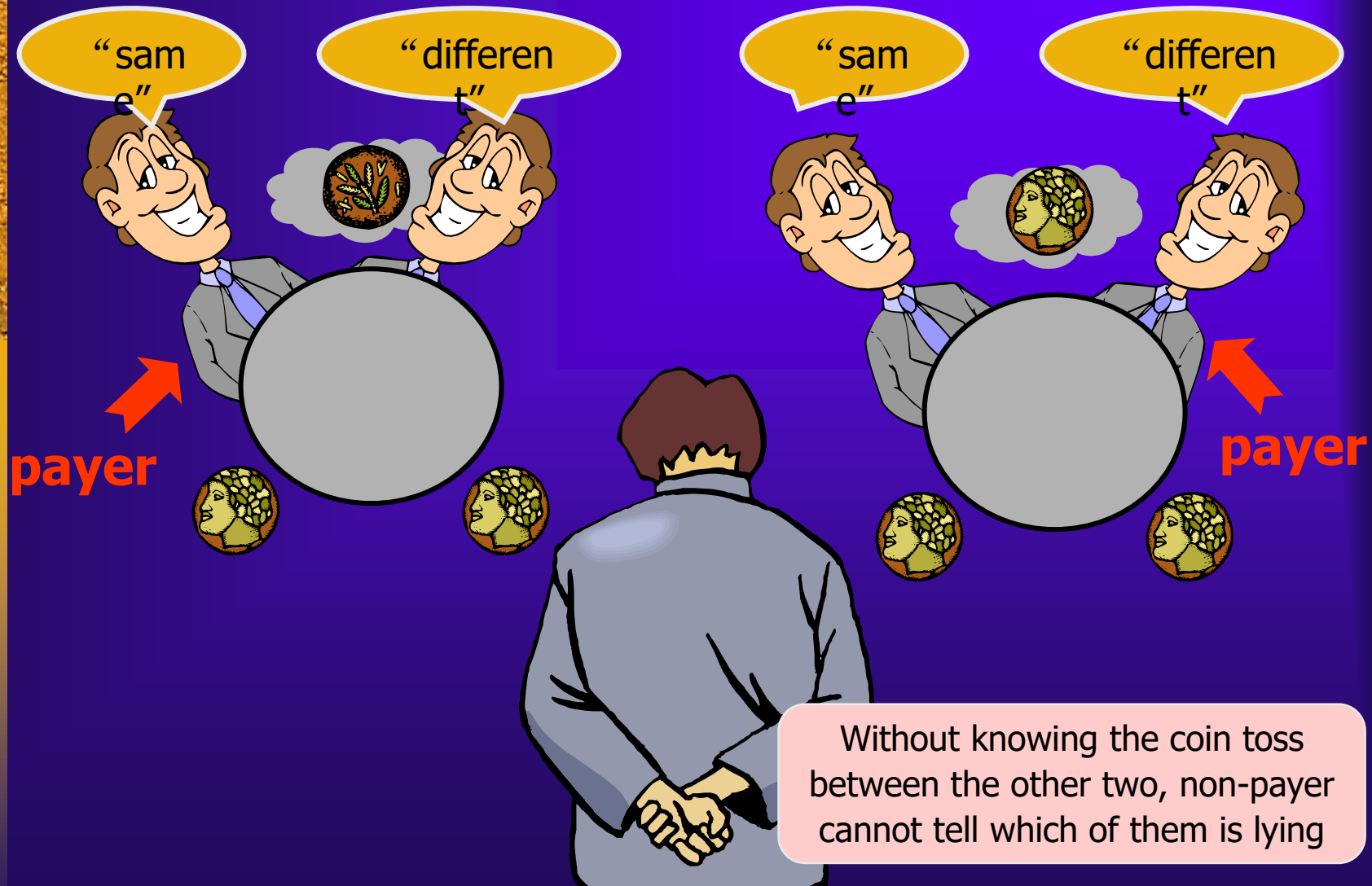
Three-Person DC Protocol

Three cryptographers are having dinner.

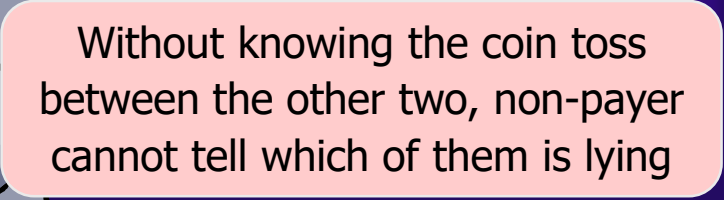
Either NSA is paying for the dinner, or
one of them is paying, but wishes to remain anonymous.

1. Each diner flips a coin and shows it to his left neighbor.
 - Every diner will see two coins: his own and his right neighbor's
2. Each diner announces whether the two coins are the same.
If he is the payer, he lies (says the opposite).
3. Odd number of "same" \Rightarrow NSA is paying;
Even number of "same" \Rightarrow one of them is paying
 - But a non-payer cannot tell which of the other two is paying!

Non-Payer's View: Same Coins



A large, ornate metal key with a circular bow and a notched bit, resting on a textured, yellowish-brown surface. The key is made of dark, possibly iron or steel, metal. The bow is a simple circle. The shaft is straight and ends in a bit with several notches. The background is a rough, granular surface, likely sand or gravel.





Superposed Sending

- ◆ This idea generalizes to any group of size N
- ◆ For each bit of the message, every user generates 1 random bit and sends it to 1 neighbor
 - Every user learns 2 bits (his own and his neighbor's)
- ◆ Each user announces own bit XOR neighbor's bit
- ◆ Sender announces own bit XOR neighbor's bit XOR message bit
- ◆ XOR of all announcements = message bit
 - Every randomly generated bit occurs in this sum twice (and is canceled by XOR), message bit occurs once



DC-Based Anonymity is Impractical

- ◆ Requires secure pairwise channels between group members
 - Otherwise, random bits cannot be shared
- ◆ Requires massive communication overhead and large amounts of randomness
- ◆ DC-net (a group of dining cryptographers) is robust even if some members collude
 - Guarantees perfect anonymity for the other members