Making Anonymous Communication

Paul Syverson

Center for High Assurance Computer Systems

Naval Research Laboratory

http://www.onion-router.net/

syverson@itd.nrl.navy.mil

Presented at the National Science Foundation,

June 8, 2004

Talk Outline

- Motivation: Why anonymous communication?
  - Personal privacy
  - Corporate and governmental security
  - Note: Anonymous comm. = Traffic analysis resistant comm.
- Characterizing anonymity: Properties and Types
- Mixes and proxies: Anonymity building blocks
- Onion Routing: Lower latency, Higher Security
- Features of Tor: 2nd Generation Onion Routing
- Hidden Servers and Rendezvous Points
- Summary and Future Work

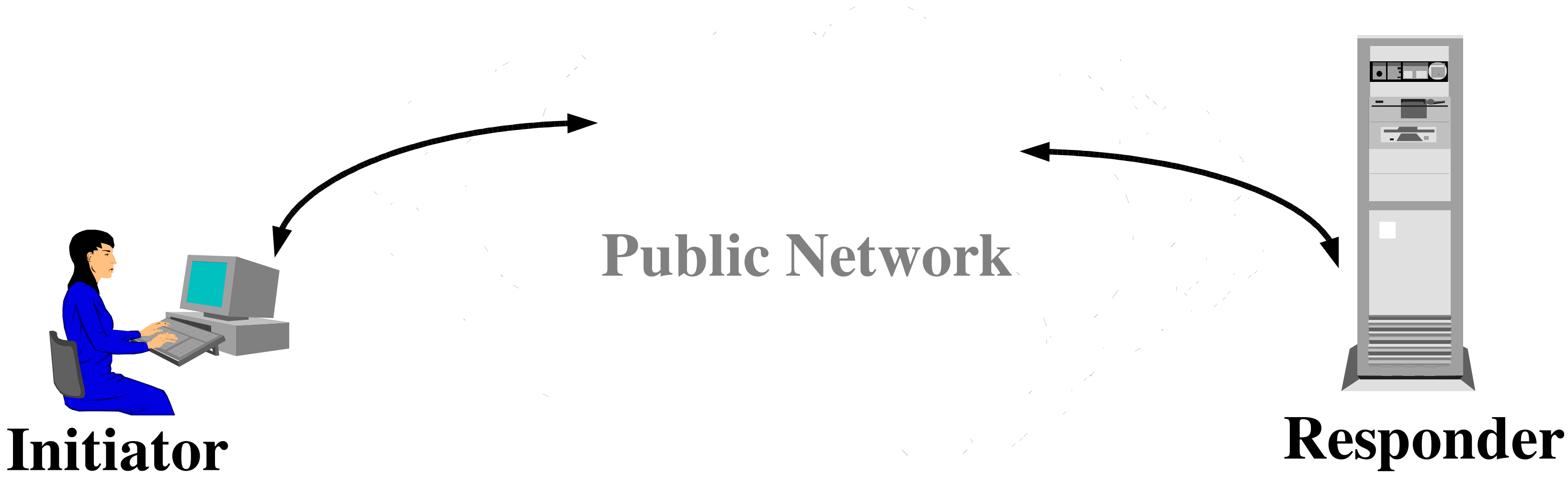
Credits

- Some slides on mixes cribbed from Ari Juels (with permission)
- Original Onion Routing concept, design, research, patent, etc.
  - with David Goldschlag and Michael Reed
- Tor, second generation Onion Routing
  - with Roger Dingledine and Nick Mathewson

- Numerous research predecessors and contemporaries
- Lots of volunteers contributing to open source Tor code, running nodes, etc.

Public Networks are Vulnerable to Traffic Analysis

- In a Public Network (Internet):
- Packet (message) headers identify recipients
- Packet routes can be tracked



Encryption does *not* hide routing information.

Who Needs Anonymity?

- Socially sensitive communicants:
  - Disease or crime victim chat rooms
- Law Enforcement:
  - Anonymous tips or crime reporting
  - Surveillance and honeypots (sting operations)
- Corporations:
  - Hiding collaborations of sensitive business units or partners
  - Hide procurement suppliers or patterns
- Political Dissidents
- Censorship resistant publishers
- Whistleblowers

Who Needs Anonymity?

- You:
  - Who are you sending email (who is sending you email)
  - What Web sites are you browsing
  - Where do you work, where are you from
  - What do you buy, what kind of physicians do you visit, what books do you read, ...

Who Needs Anonymity?

- Government

Government Needs Anonymity? Yes, for…

- Open source intelligence gathering
  - Hiding individual analysts is not enough
  - That a query was from a govt. source may be sensitive
- Defense in depth on open and *classified* networks
  - Networks with only cleared users (but a million of them)
- Dynamic and semitrusted international coalitions
  - Network can be shared without revealing existence or amount of communication between all parties
- Elections and Voting

Government Needs Anonymity? Yes, for…

- Networks partially under known hostile control
  - To attack comm. enemy must take down whole network
- Politically sensitive negotiations
- Road Warriors
- Protecting procurement patterns
- Homeland Security Information to/from municipalities, industry,…
- Anonymous tips (national security, congressional investigations, etc. In addition to law enforcement)

Existing Protections Can be Improved by Anonymity

- Virtual Hidden Networks
  - Traditional VPNs are not private
  - Anyone can see the network
  - Often adversary can see amount of communication
  - Onion Routing can provide anonymity to hide existence of private network and reduce countermeasure cost

Existing Protections Improved by Anonymity

- Location Hidden Survivable Services for
  - Homeland Security info to/from every town and industry
  - Censorship resistant publishers
  - Businesses with high value customers
- Hidden Server Properties
  - Servers accessible from anywhere
  - Resist attacks from authorized users
  - Resist Distributed DoS
  - Resist physical attack
  - Minimize redundancy, Reduce costs
  - Provide the above better than: firewalls, multiple redundant servers, physically hardened sites, IP filter, IP traceback

Who Needs Anonymity?

- And yes criminals

Who Needs Anonymity?

- And yes criminals

But they already have it.

We need to protect everyone else.

One Moral: For communication the real question is not,

How much privacy would you give up for security?

One Moral: For communication the real question is not,

How much privacy would you give up for security?

The question is,

How much security would you give up for <security?> ?

One Moral: For communication the real question is not,
How much privacy would you give up for security?

The question is,
How much security would you give up for <security?> ?

(and for anonymity there is no option to keep it to yourself)

Anonymity Loves Company

- You can't be anonymous by yourself
  - Can have confidentiality by yourself
- A network that protects only DoD network users won't hide that connections from that network are from Defense Dept.
- Need to share creates interesting incentive issues for network users and servers

Anonymous From Whom? Adversary Model

- Recipient of your message
- Sender of your message

Need Channel and Data Anonymity

- Observer of network from outside
- Network Infrastructure (Insider)

Need Channel Anonymity

- Note: Anonymous authenticated communication makes perfect sense
- Communicant identification should be in the basic channel not of the channel

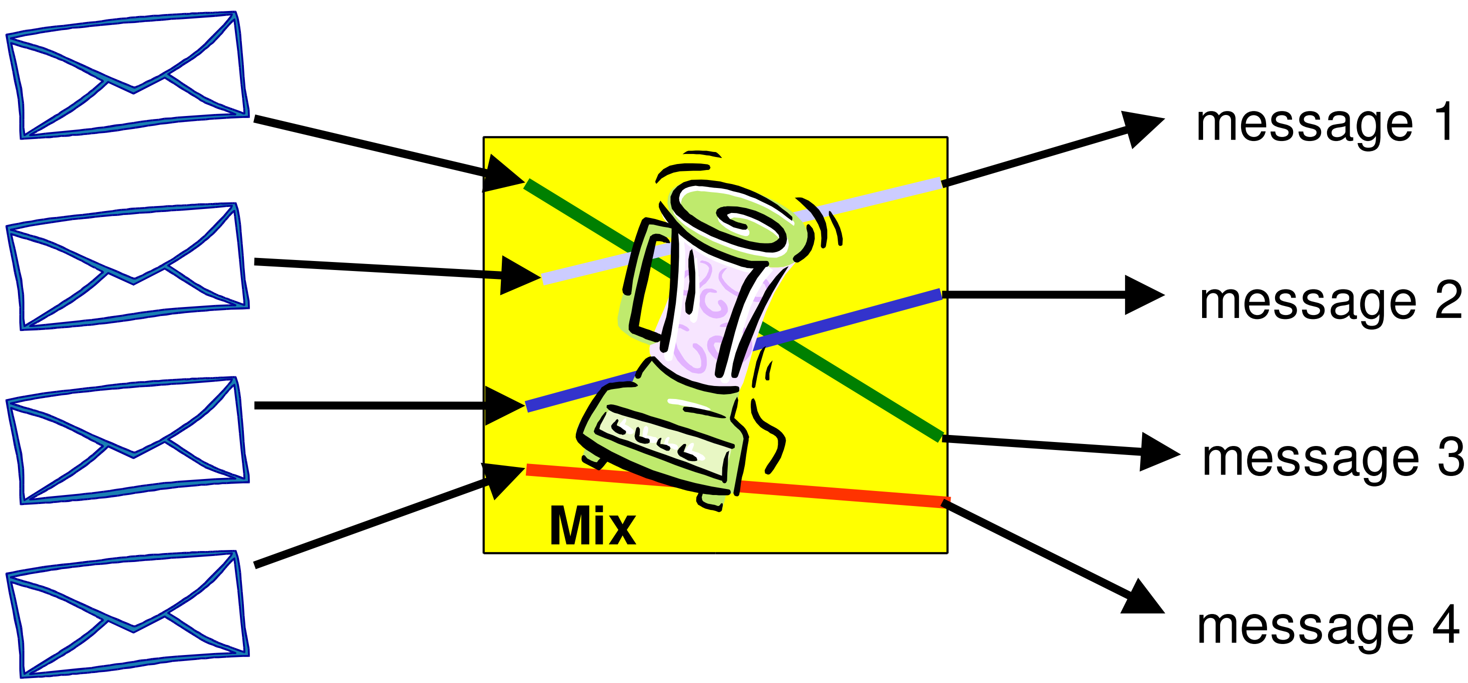
Focus of this work is anonymity of the communication pipe,

not what goes through it

How Do You Get Communication Anonymity?

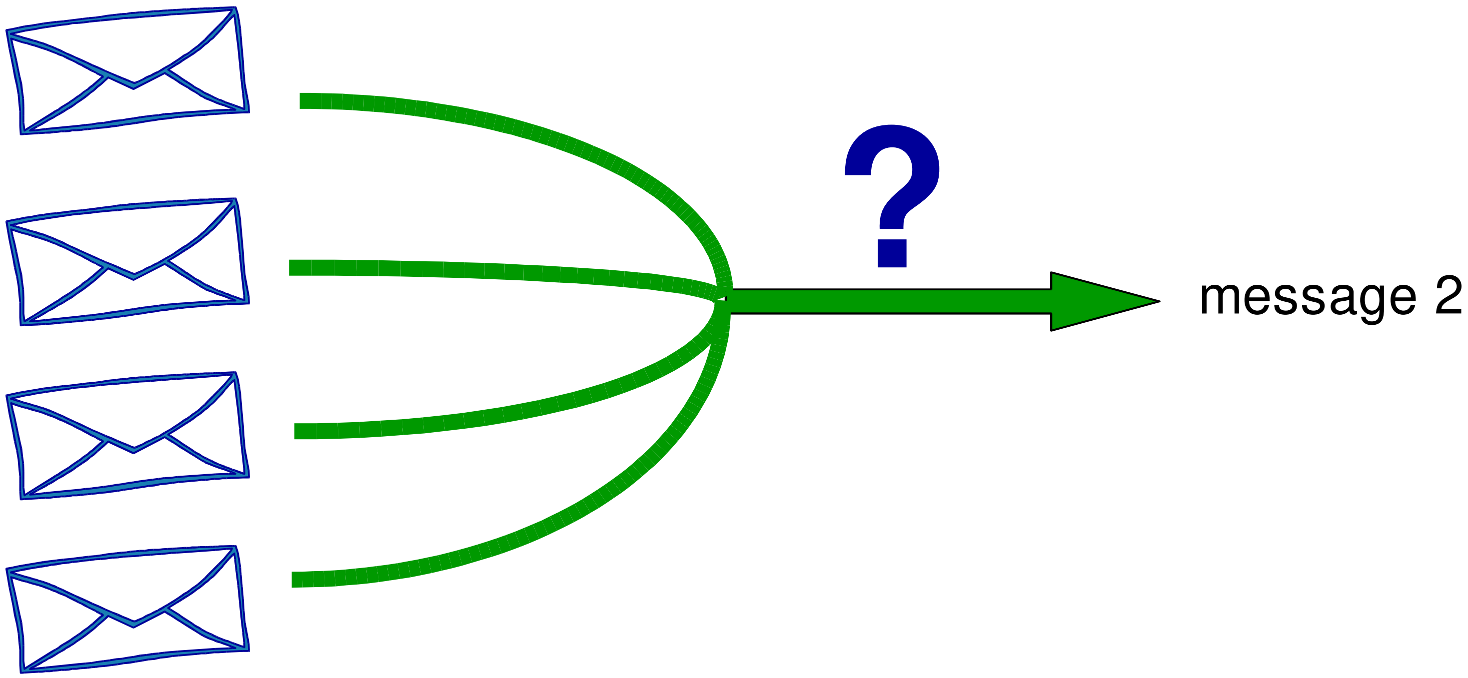
- Many technical approaches
- Overview of two extensively used approaches
  - Mixes
  - Proxies

What does a mix do?



Randomly permutes and decrypts inputs

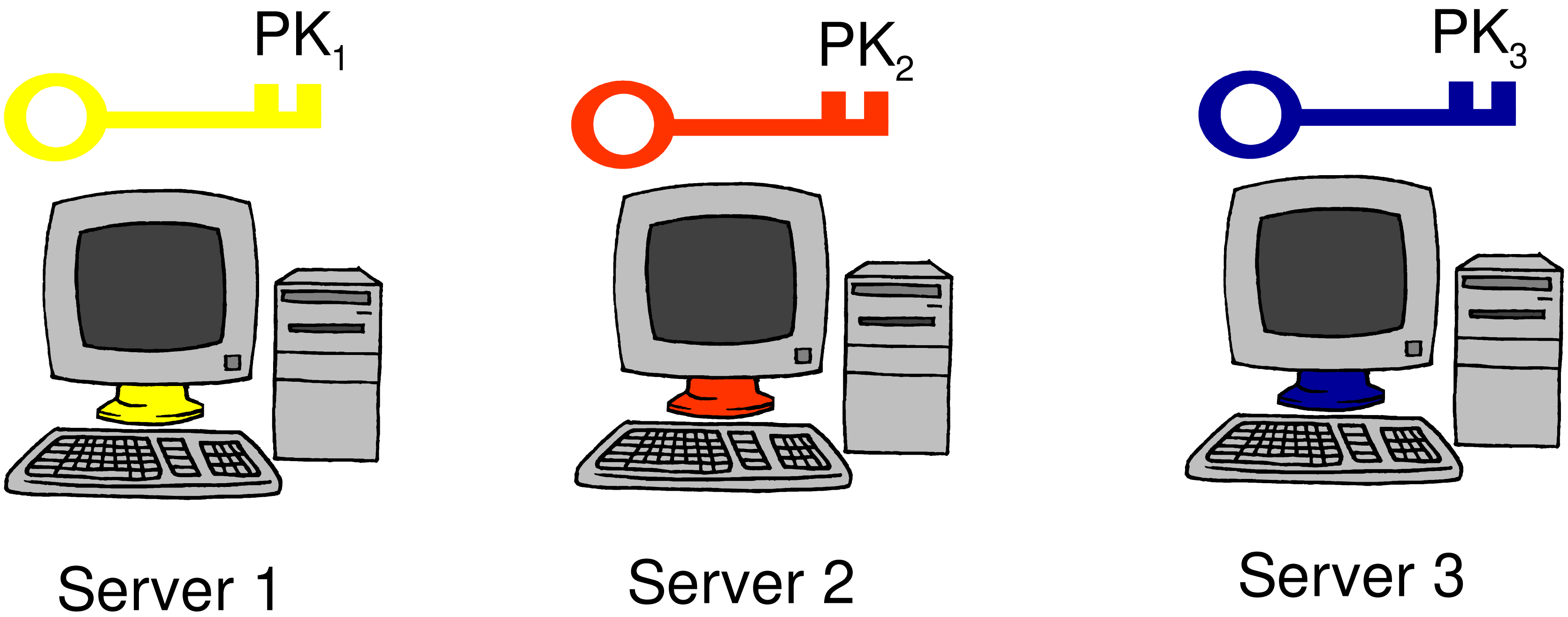
What does a mix do?



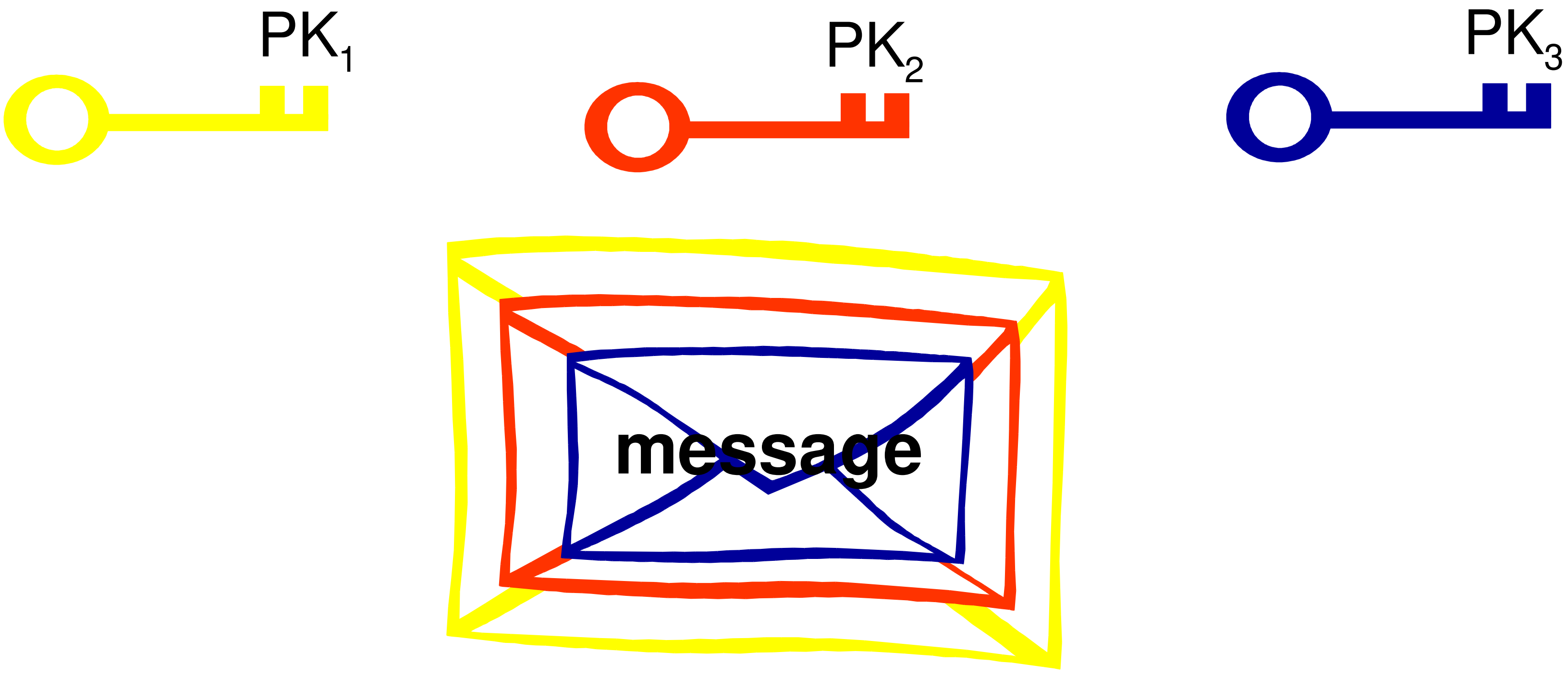
Key property: Adversary can't tell which ciphertext corresponds to a given message

A look under the hood

Basic Mix (Chaum '81)

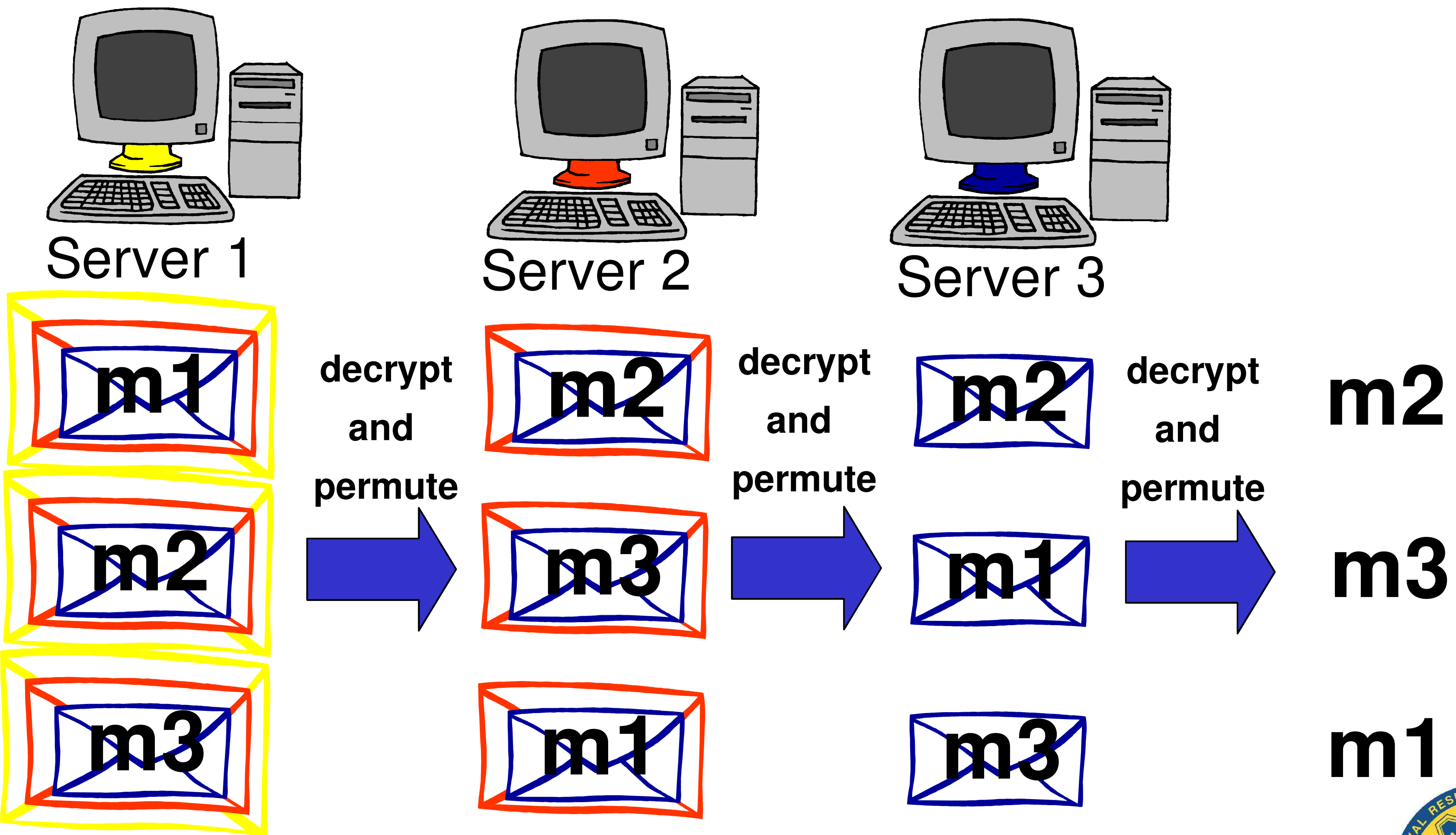


Encryption of Message

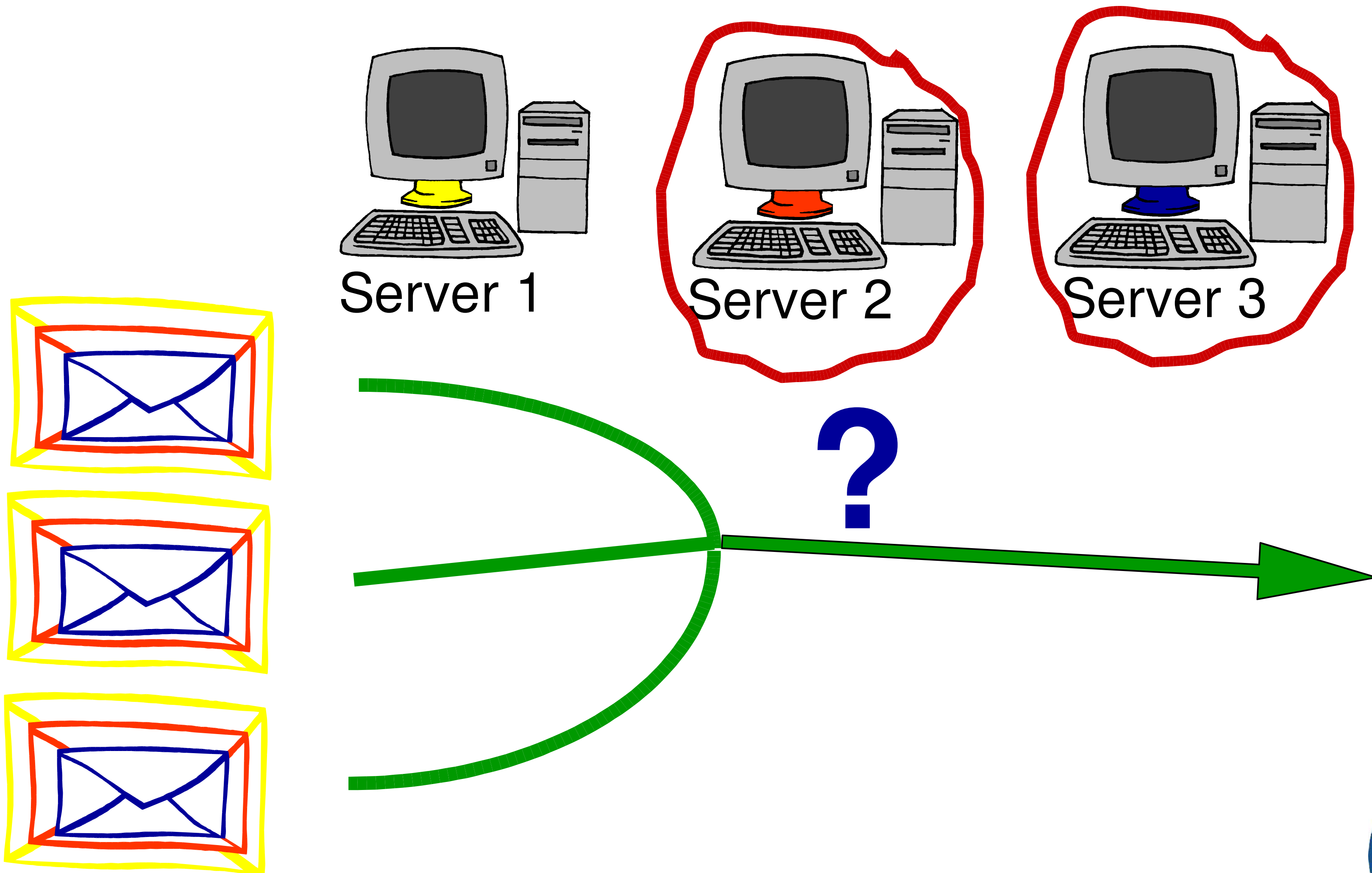


Ciphertext = $E_{PK1}[E_{PK2}[E_{PK3}[\text{message}]]]$

Basic Chaum-type Mix

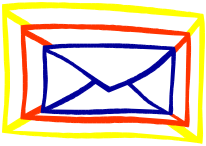


One honest server preserves privacy

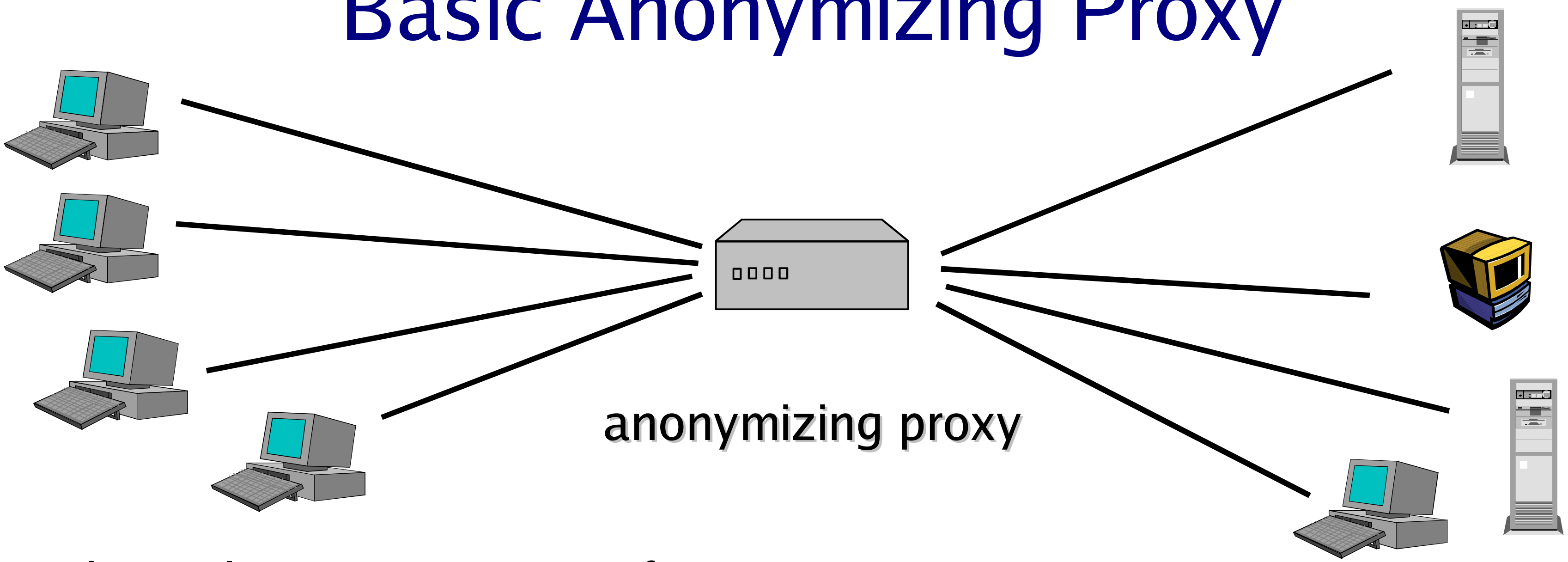


What if you need quick interaction?

- Web browsing, Remote login, Chat, etc.
- Mixnets introduced for email and other high latency apps
- Each layer of message requires expensive public-key crypto



Basic Anonymizing Proxy



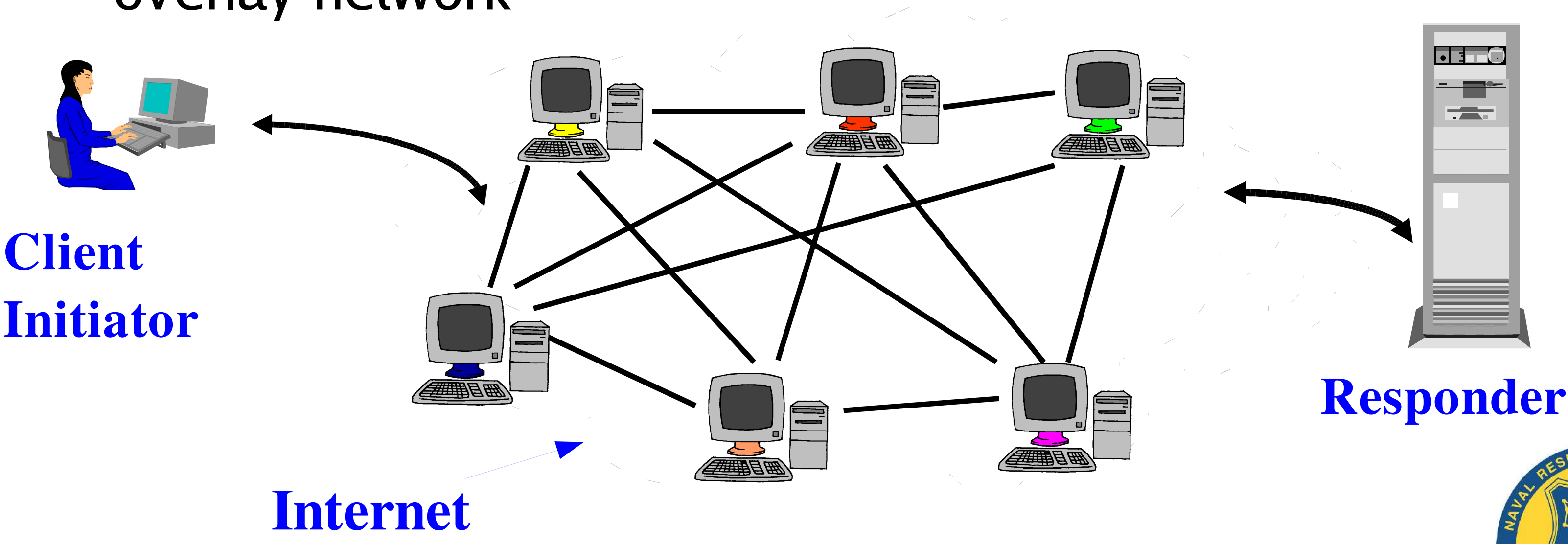
- Channels appear to come from proxy, not true originator
- Appropriate for Web connections, etc.:
  SSL, TLS, SSH (lower cost symmetric encryption)
- Examples: The Anonymizer
- Advantages: Simple, Focuses lots of traffic for more anonymity
- Main Disadvantage: Single point of failure, compromise, attack

Onion Routing
Traffic Analysis Resistant Infrastructure

- Main Idea: Combine Advantages of mixes and proxies
- Use (expensive) public-key crypto to establish circuits
- Use (cheaper) symmetric-key crypto to move data
  - Like SSL/TLS based proxies
- Distributed trust like mixes
- Related Work (some implemented, some just designs):
  - ISDN Mixes
  - Crowds, JAP Webmixes, Freedom Network
  - Tarzan, Morphmix

Network Structure

- Onion routers form an overlay network
  - Clique topology (for now)
  - Longstanding TLS encrypted connections (thick pipes)
- Proxy interfaces between client machine and onion routing overlay network



Tor

Tor

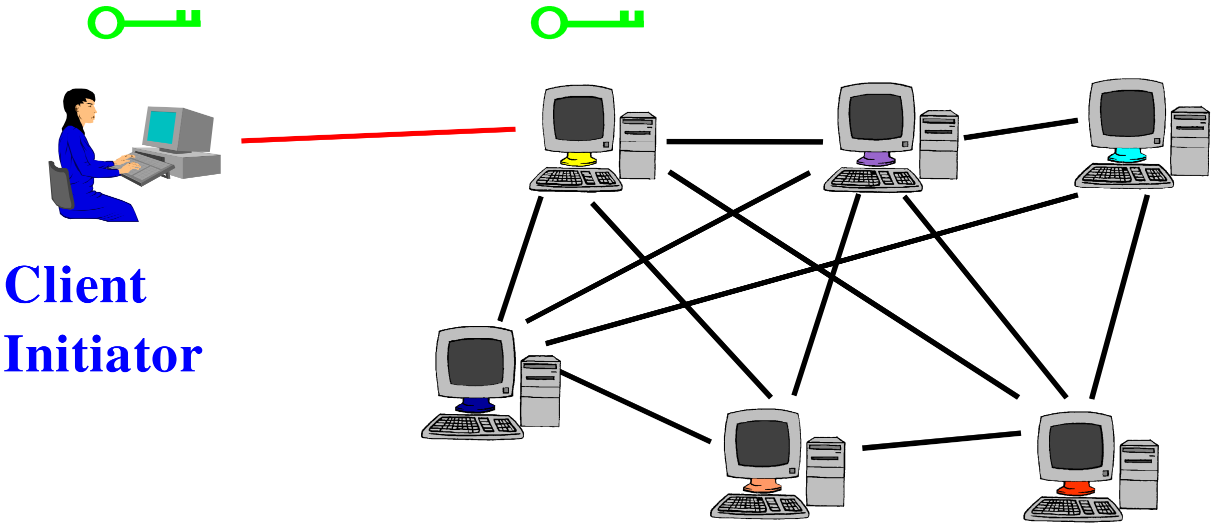
The Onion Routing

Tor

Tor's Onion Routing

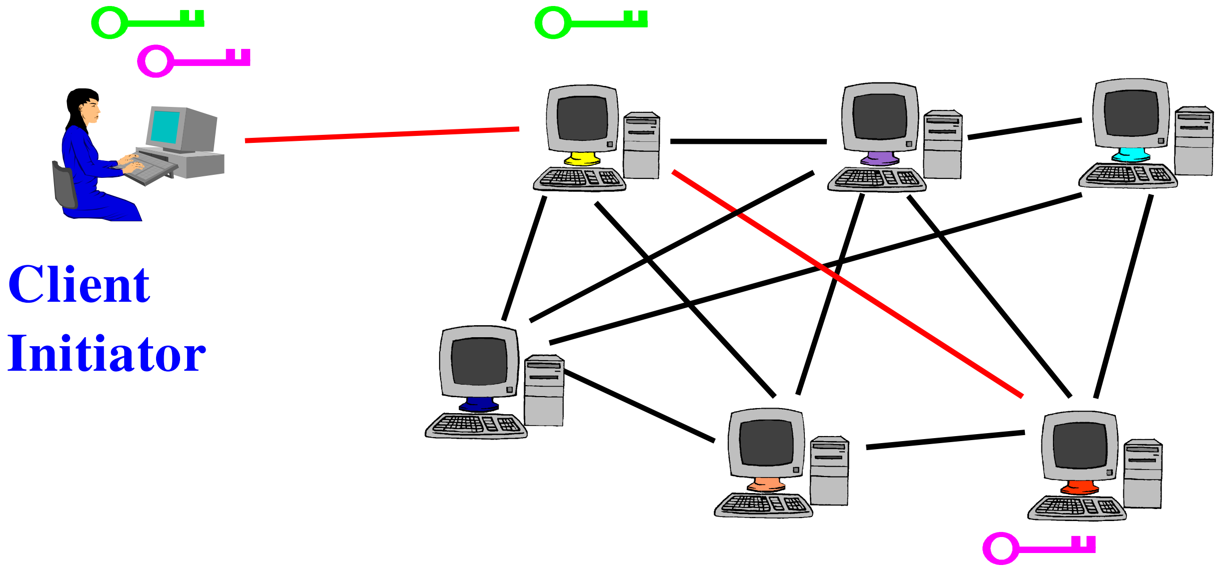
Tor Circuit Setup

- Client Proxy establishes session key and circuit w/ Onion Router 1



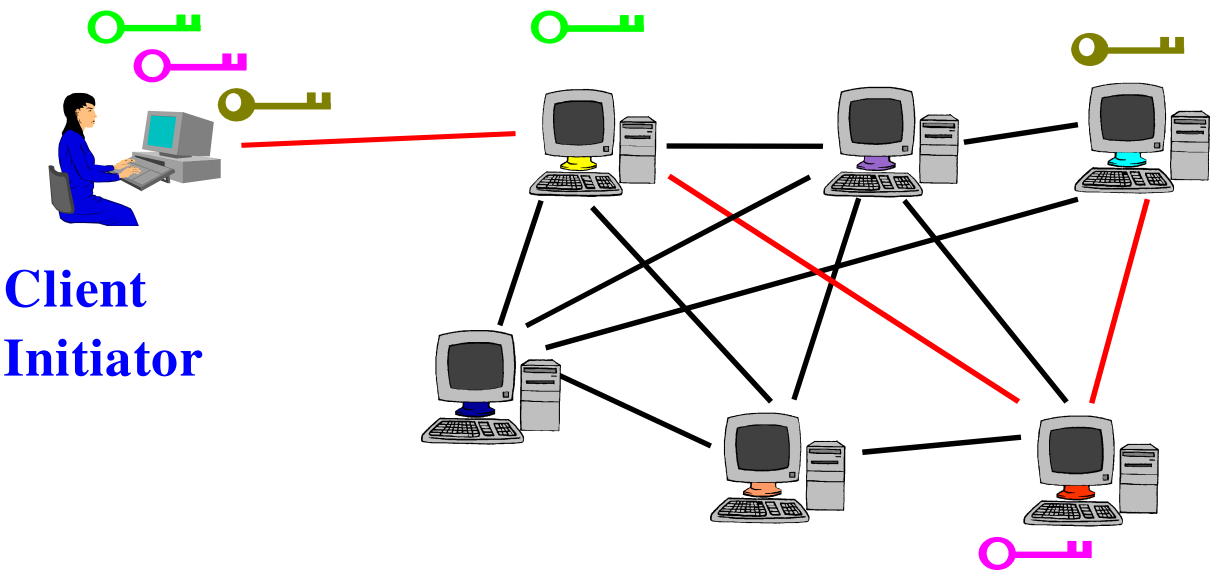
Tor Circuit Setup

- Client Proxy establishes session key and circuit w/ Onion Router 1
- Proxy tunnels through that circuit to extend to Onion Router 2



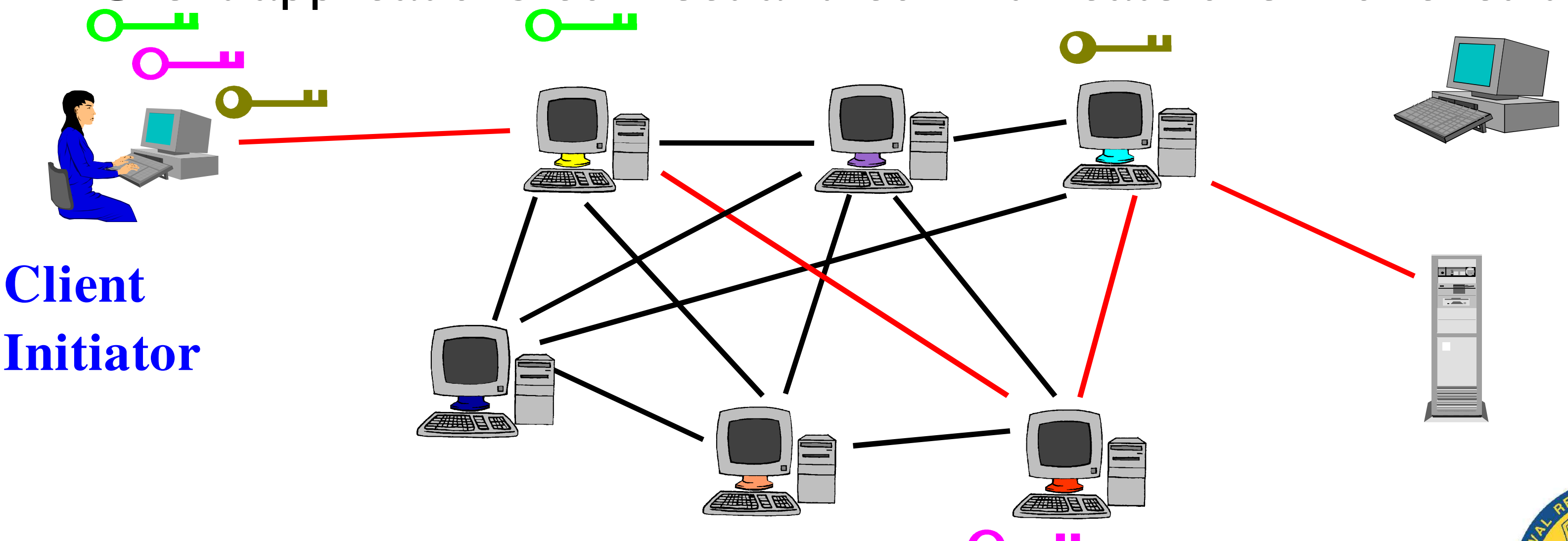
Tor Circuit Setup

- Client Proxy establishes session key and circuit w/ Onion Router 1
- Proxy tunnels through that circuit to extend to Onion Router 2
- Etc



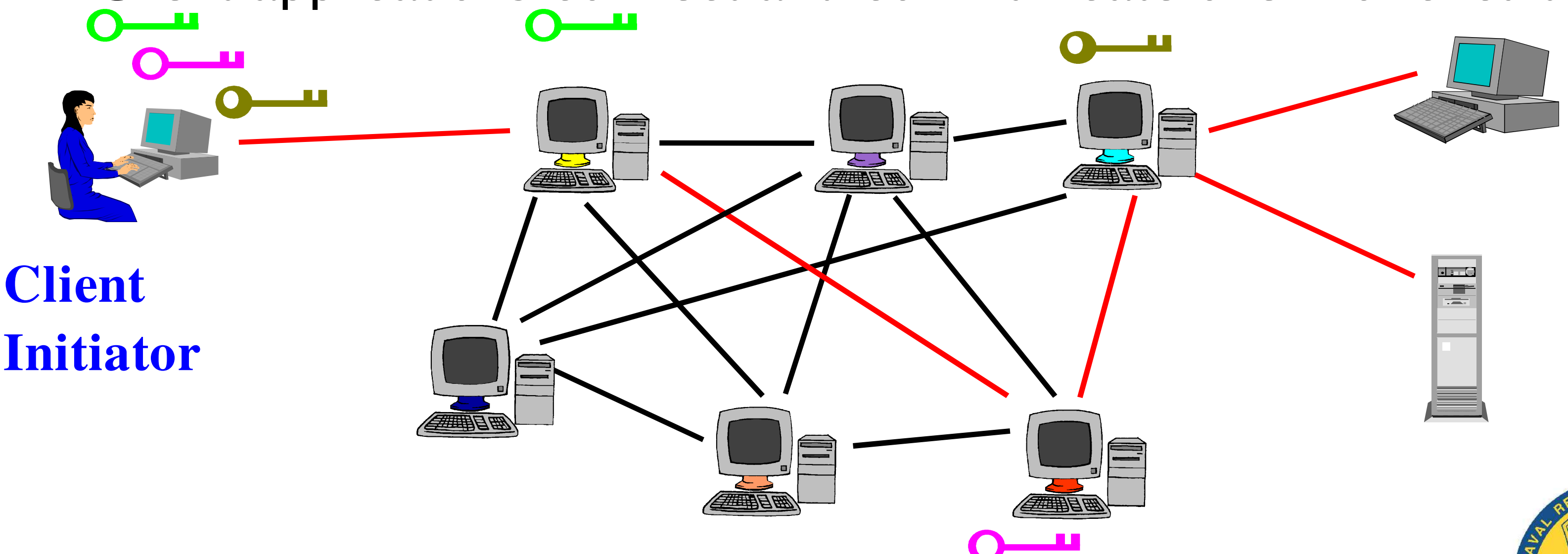
Tor Circuit Usage

- Client Proxy establishes session key and circuit w/ Onion Router 1
- Proxy tunnels through that circuit to extend to Onion Router 2
- Etc
- Client applications connect and communicate over Tor circuit



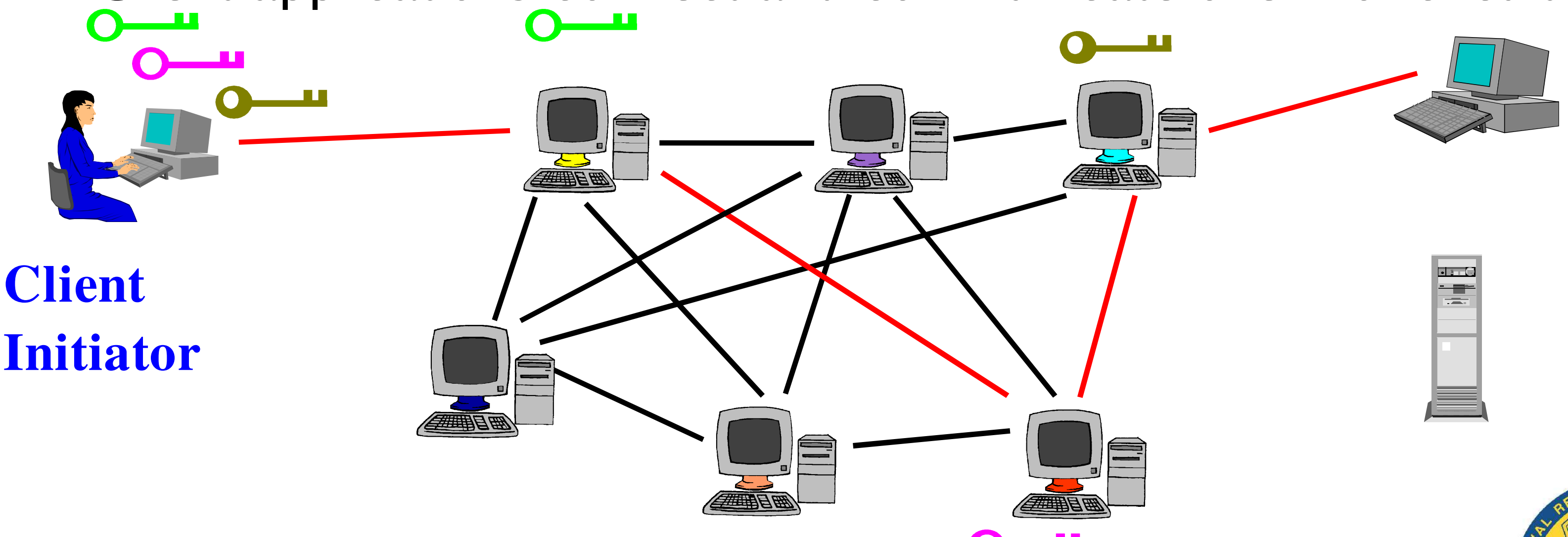
Tor Circuit Usage

- Client Proxy establishes session key and circuit w/ Onion Router 1
- Proxy tunnels through that circuit to extend to Onion Router 2
- Etc
- Client applications connect and communicate over Tor circuit



Tor Circuit Usage

- Client Proxy establishes session key and circuit w/ Onion Router 1
- Proxy tunnels through that circuit to extend to Onion Router 2
- Etc
- Client applications connect and communicate over Tor circuit



Where do I go to connect to the network?

- Directory Servers
  - Maintain list of which onion routers are up, their locations, current keys, exit policies, etc.
  - Control which nodes can join network
    - Important to guard against pseudospoofing attack and related problems

Some Tor Properties

- Simple modular design, Restricted ambitions
  - Circa 20K lines of C code
  - Even servers run in user space, no need to be root
  - Just anonymize the pipe
    - Can use, e.g., privoxy as front end if desired to anonymize data
  - SOCKS compliant TCP: includes Web, remote login, mail, chat, more
    - No need to build proxies for every application
  - Flexible exit policies, each node chooses what applications/destinations can emerge from it

Some Tor Properties

- Lots of supported platforms:

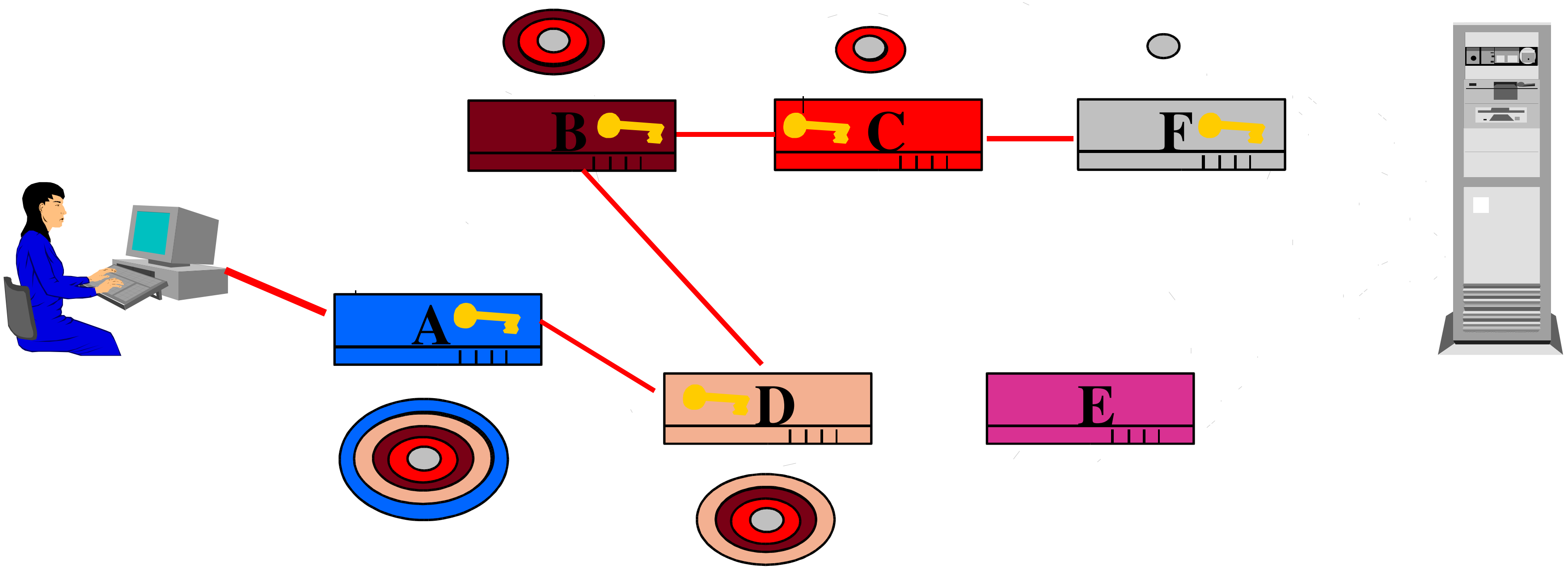
  Linux, BSD, MacOS, Solaris, Windows
- Many TCP streams (application connections) share one anonymous circuit
  - Less public-key encryption overhead than prior designs
  - Reduced anonymity danger from opening many circuits

More Tor Properties

- Thick pipe bandwidth rate limiting
  - Limits how much one OR can send to a neighbor
  - Token bucket approach limits average but permits burstiness
- Circuit and stream level throttling
  - Controls congestion
  - Mitigates denial of service that a single circuit can do
- Stream integrity checks
  - Onion Routing uses stream ciphers
  - Checks prevent, e.g., reasonable guess attack

    XOR out of '`dir`' and XOR in '`rm *`'

Generations 0 and 1 Circuit Setup



- The initial proxy knows the Onion Routing network topology, selects a route, and generates the onion
- Each layer of the onion identifies the next hop in the route and contains the cryptographic keys to be used at that node.

More Tor Advantages

- No need to keep track of onions to prevent replay
  - There are no onions anymore
  - Even a replayed create cell will result in a new session key at an honest onion router
- Perfect Forward Secrecy
  - Storing all traffic sent to a node and later breaking its public key will not reveal encrypted content
- Can adapt to network dynamics better
  - Down exit node or unusable exit policy does not require building whole new circuit

Numbers and Performance

- Prototype ran for two years (1998 - 2000)
  - 4 nodes running at a single location
  - During final months processed over 50K Web connections/day from a total of 60K IP addresses worldwide
- Current 2nd generation design running since October 2003
  - c. 25 nodes scattered through US (18) and Europe (7)
  - Hundreds (thousands?) of users
  - Average node processes 1 GB / day application cells
  - Up from .5 GB / week a month or two ago
  - Network has never been down

Latency Tests

- 4 node test network on single heavily loaded 1 GHz Athlon
  - Download 60MB file (108 times over 54 hours)
  - Avg. 300 sec/download vs. 210 sec/download without Tor
- Beta network test
  - Download cnn.com (55KB)
  - Median of 2.7 sec through Tor vs. 0.3 sec direct

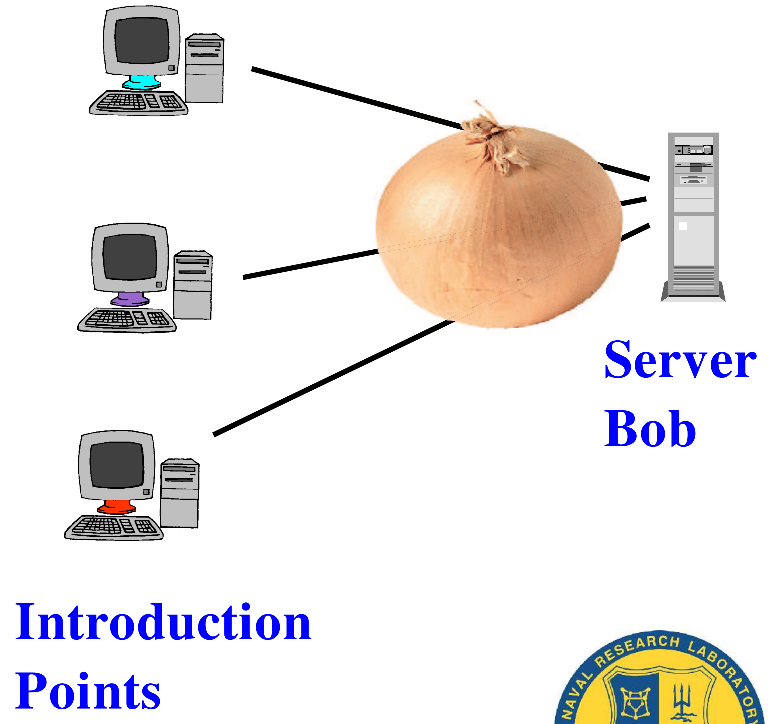
    Fastest through Tor was 0.6 sec

Location Hidden Servers

- Alice can connect to Bob's server without knowing where it is or possibly who he is
- Can provide servers that
  - Are accessible from anywhere
  - Resist censorship
  - Require minimal redundancy for resilience in denial of service (DoS) attack
  - Can survive to provide selected service even during full blown distributed DoS attack
  - Resistant to physical attack (you can't find them)
- How is this possible?

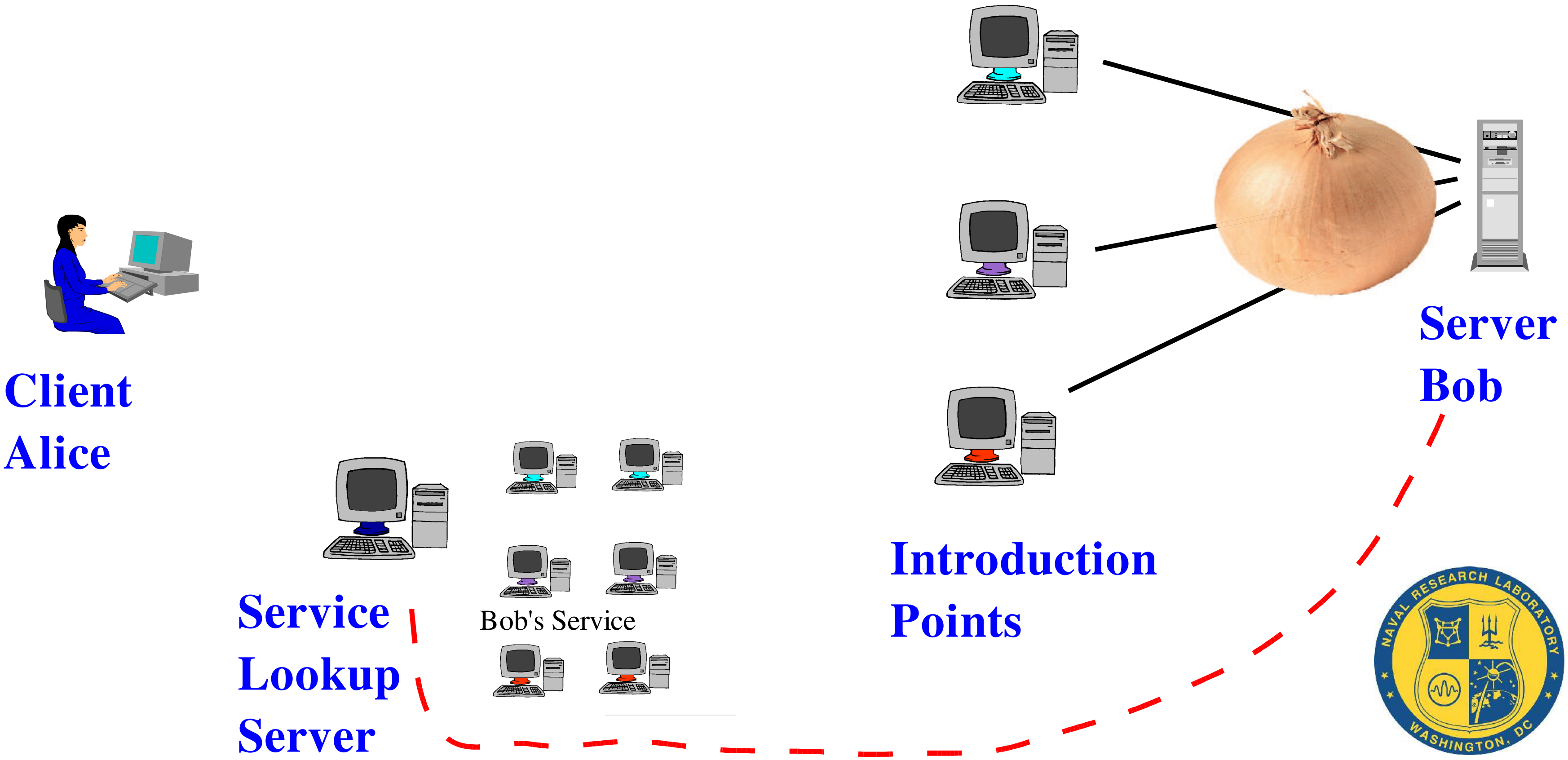
Location Hidden Servers

1. Server Bob creates onion routes to Introduction Points (IP)



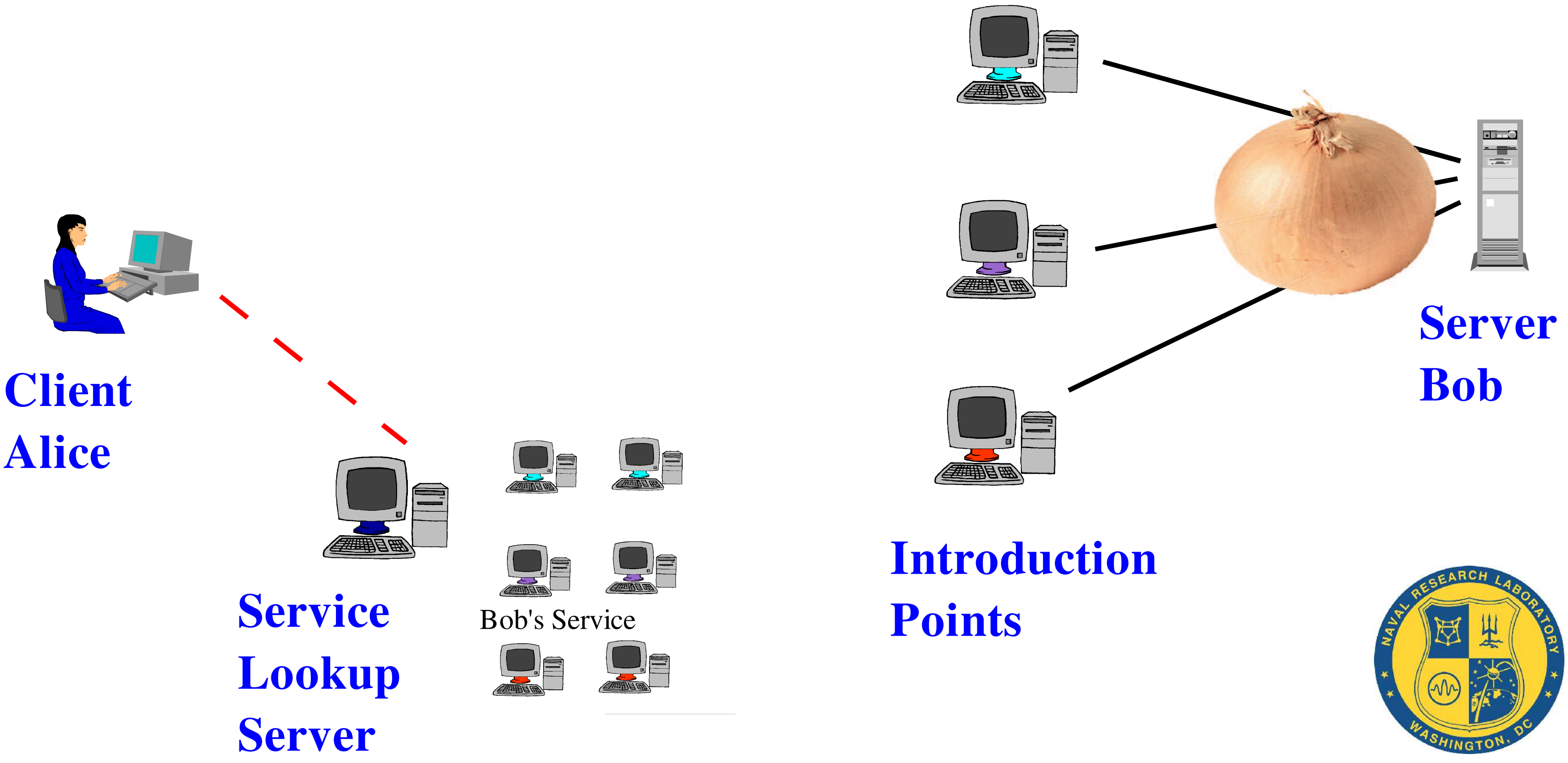
Location Hidden Servers

1. Server Bob creates onion routes to Introduction Points (IP)
2. Bob gets Service Descriptor incl. Intro Pt. addresses to Alice
   - In this example gives them to Service Lookup Server



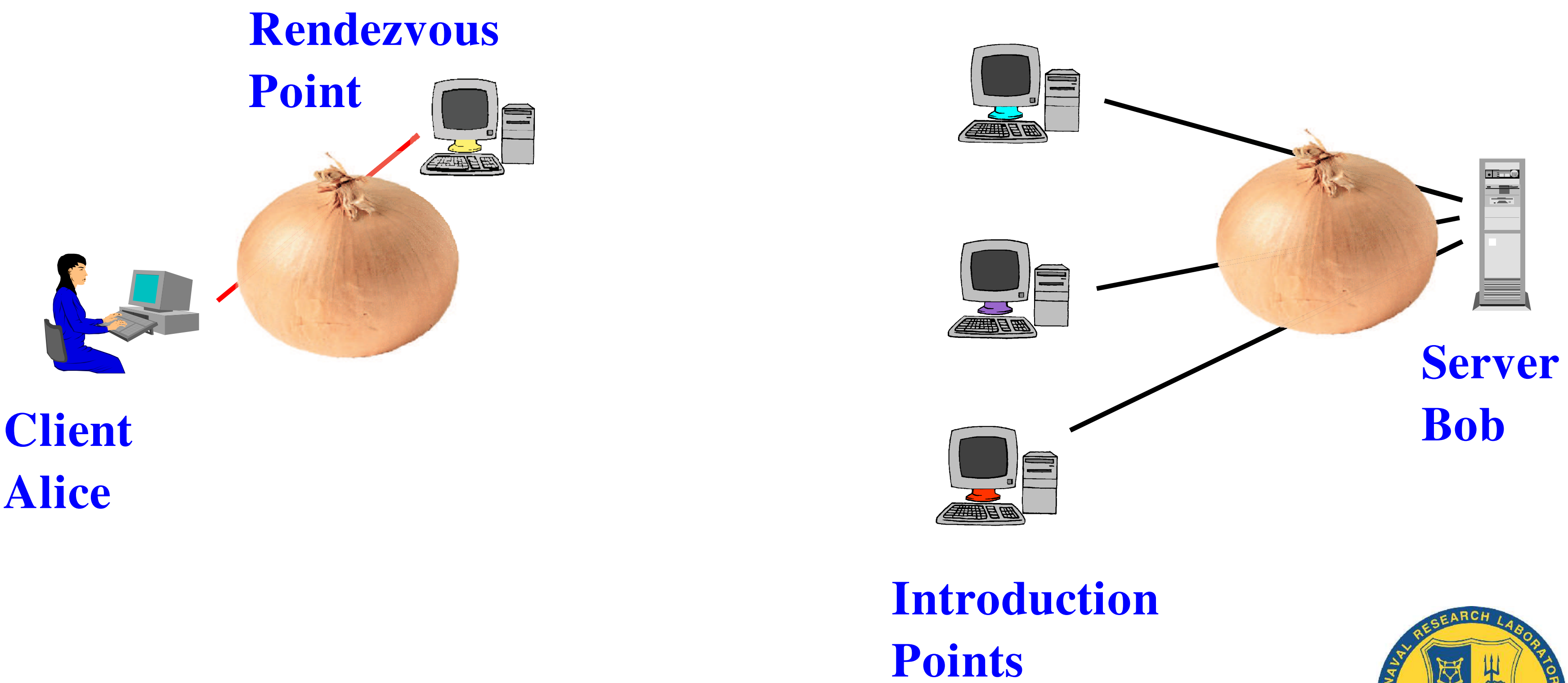
Location Hidden Servers

2'. Alice obtains Service Descriptor (including Intro Pt. address) at Lookup Server



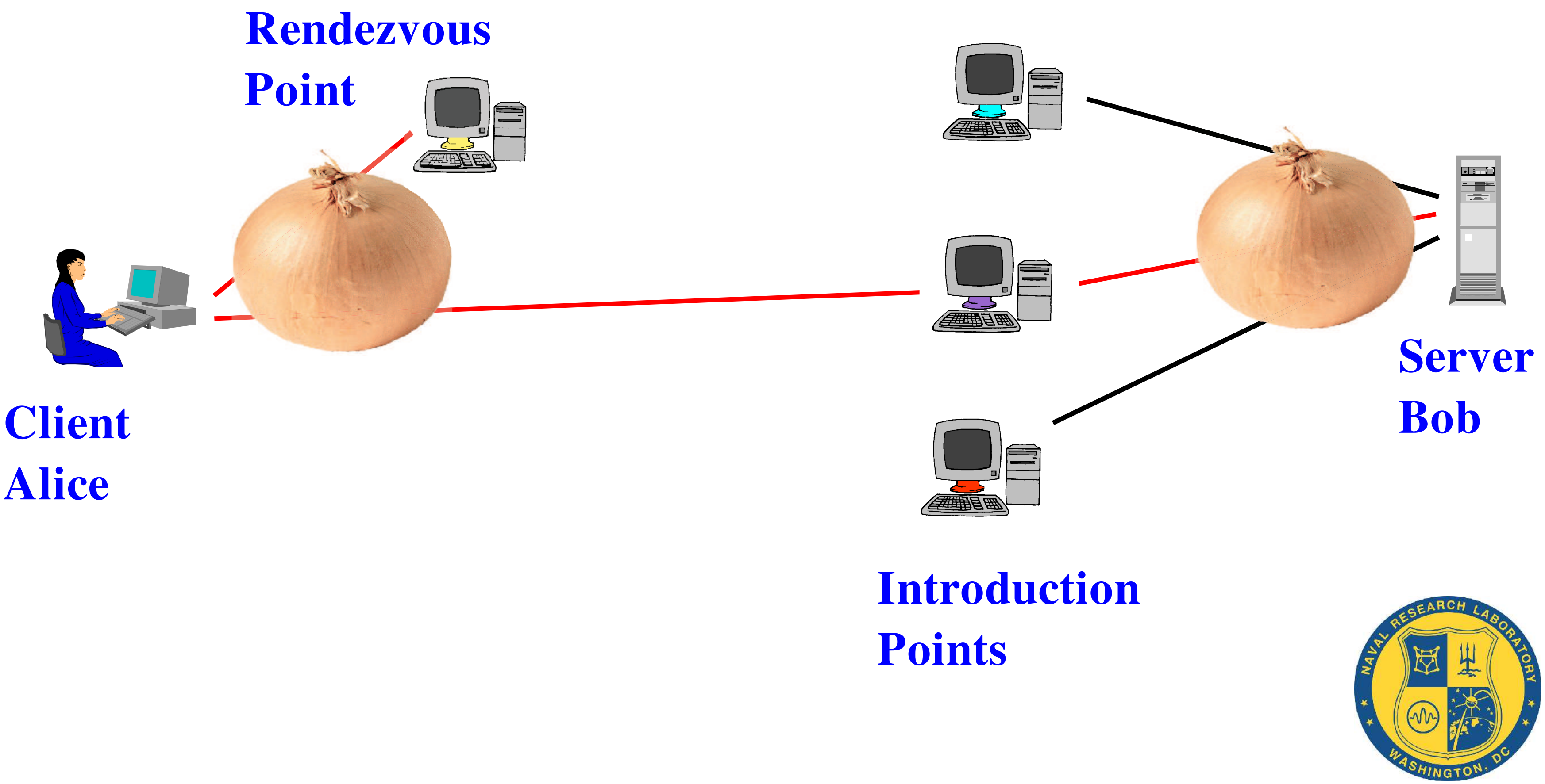
Location Hidden Servers

3. Client Alice creates onion route to Rendezvous Point (RP)



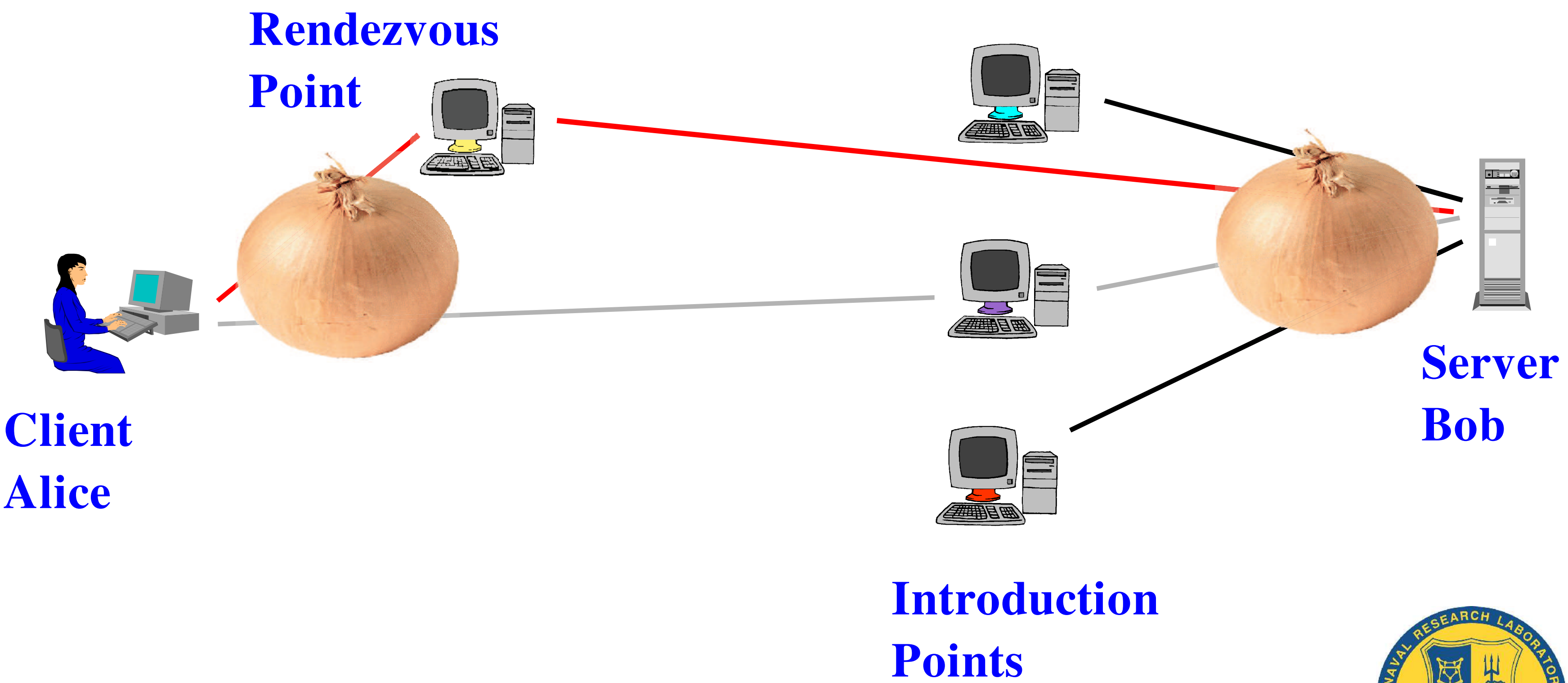
Location Hidden Servers

3. Client Alice creates onion route to Rendezvous Point (RP)
4. Alice sends RP addr. and any authorization through IP to Bob



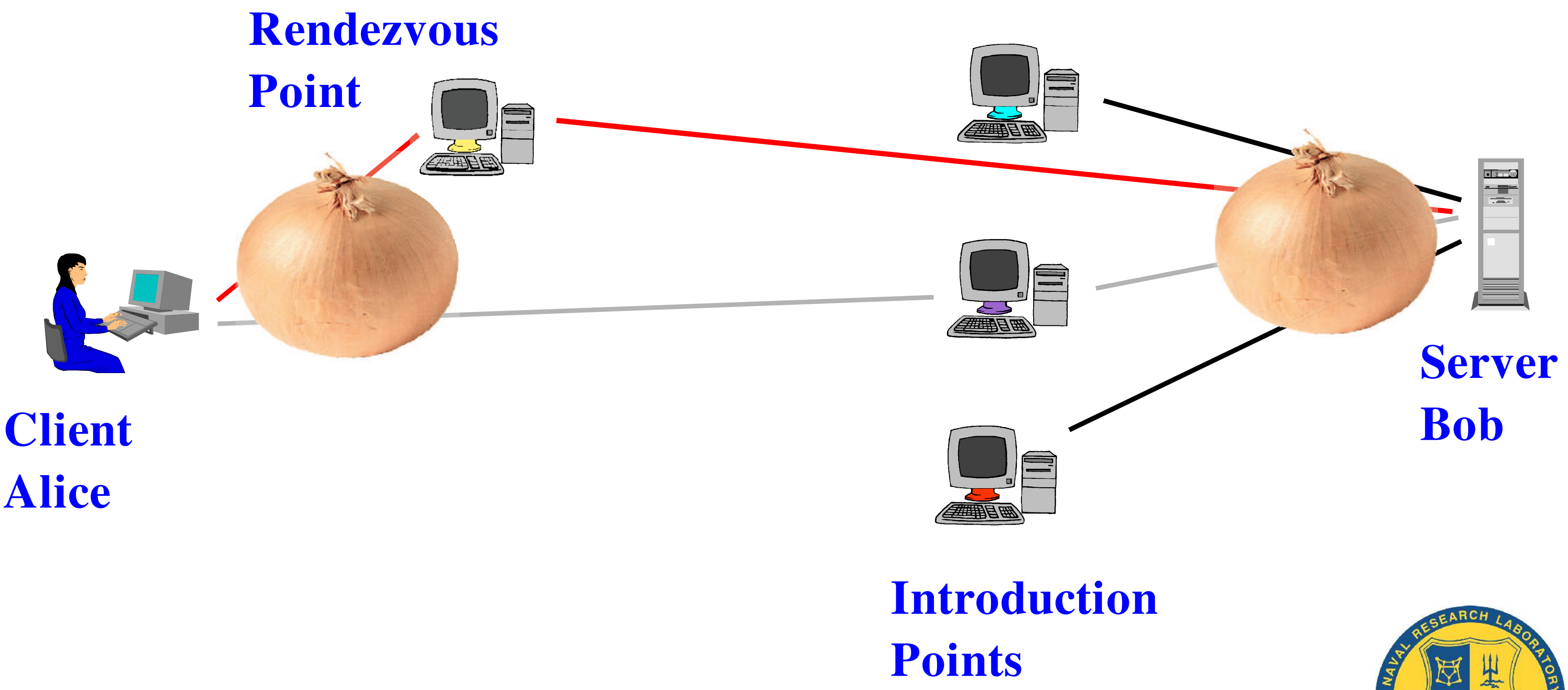
Location Hidden Servers

5. If Bob chooses to talk to Alice, connects to Rendezvous Point



Location Hidden Servers

5. If Bob chooses to talk to Alice, connects to Rendezvous Point
6. Rendezvous point mates the circuits from Alice and Bob



Rendezvous Point Protocol Overview (Location Hidden Server Building Block)

1. Bob (server) opens OR connections to introduction points
2. Bob makes these known to Alice (client)
   – Anonymously or not
3. Alice creates connection to rendezvous point
4. Alice connects to Bob via introduction point, gives rendezvous location possible authorization to Bob
5. Bob decides if he will contact Alice
6. If so, Bob anonymously routes to rendezvous point
7. Rendezvous point mates connection

Future Work

- Design and build distributed directory management
- Implement Location Hidden Servers
- Design and build Virtual Hidden Networks
- Restricted-route (non-clique) topology
  - To scale beyond hundreds of nodes and 10Ks of users

    (We should have such problems)
- Make it all work better
- Certification and Accreditation: Common Criteria
- More theoretical work
  - Midlatency synchronous batch netmixes?!?

Get the Code, Run a Node!
(or just surf the web anonymously)

- Original Onion Routing design is patented
  - 2001 Edison Patent Award
- Current system code freely available (mod. BSD license)
- Visit official site http://www.onion-router.net
- Visit http://freehaven.net/tor/ to download design paper, system spec, code, see the list of current nodes, etc.

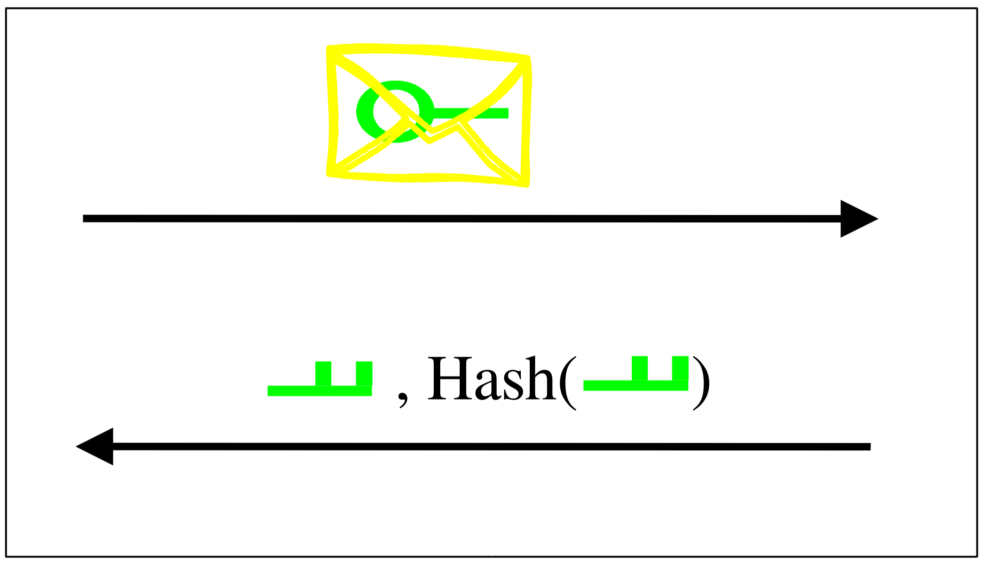
That's It

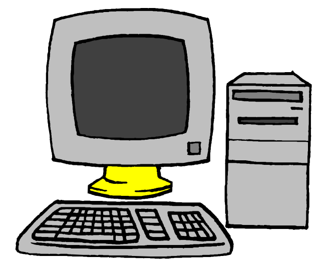
Questions?

Circuit Setup (Create)

Client chooses first node, establishes session key over TLS connection

Client
Initiator



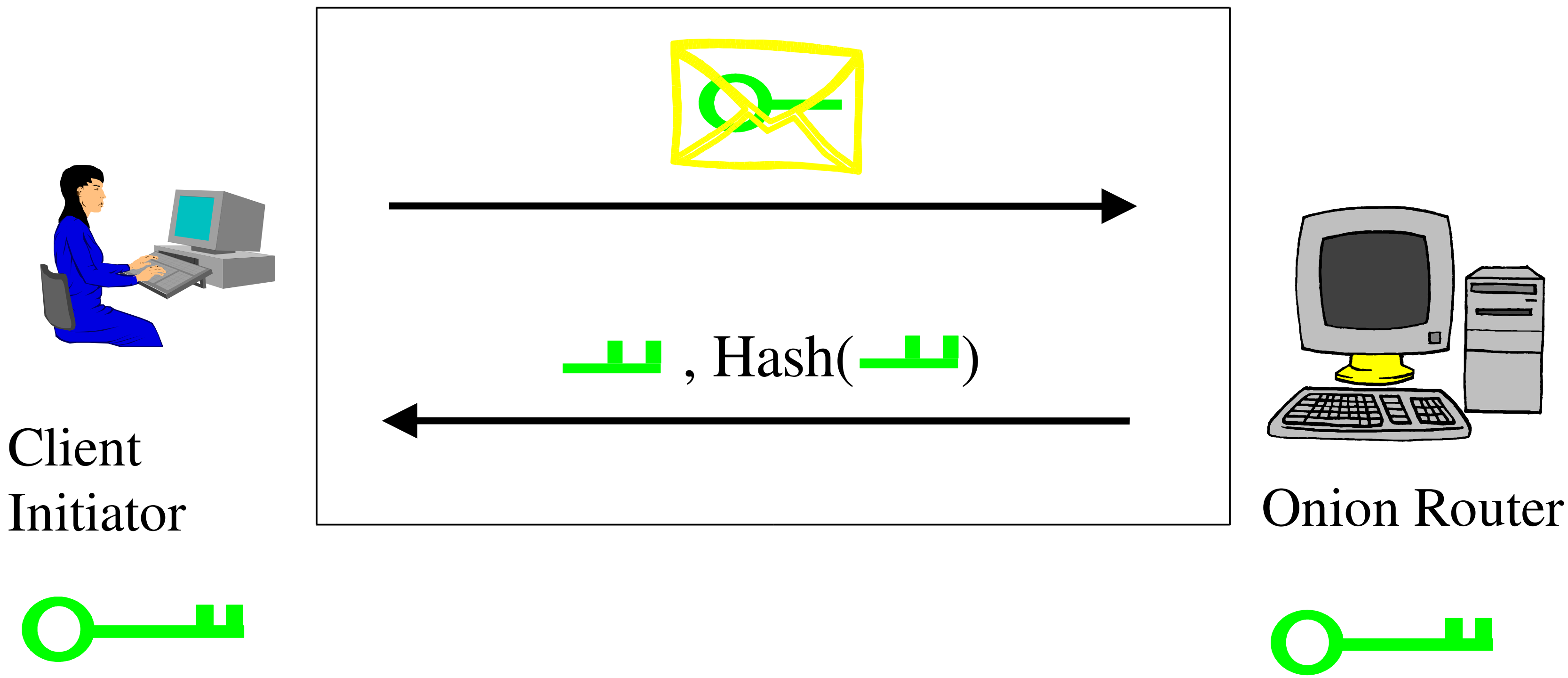


Onion Router

TLS connection

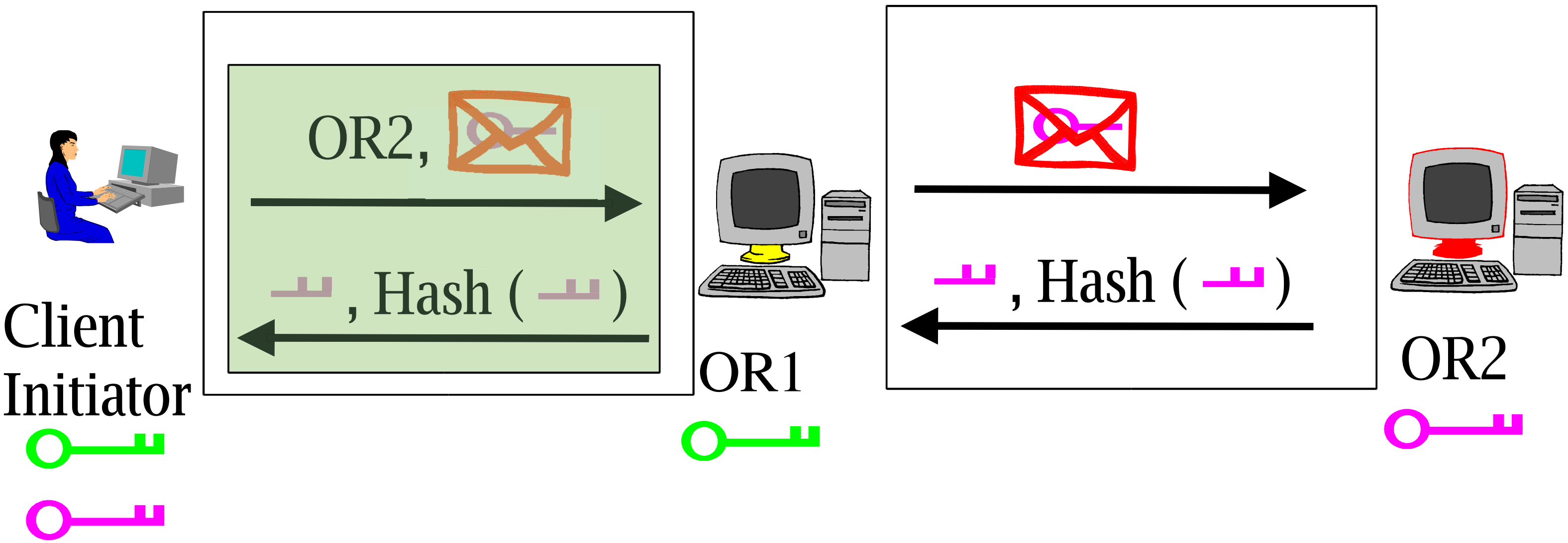
Circuit Setup (Create)

Client chooses first node, establishes session key over TLS connection



Circuit Setup (Extend)

Client chooses first node, establishes session key over TLS connection



Circuit Setup (Begin)and Data Flow

Slight simplification of actual protocol