Playing Server Hide and Seek

Lasse Øverlier
Norwegian Defence Research Establishment

Paul Syverson
Naval Research Laboratory

lasse.overlier@ffi.no
http://www.syverson.org

Location Hidden Servers

- Alice can connect to Bob's server without knowing where it is or possibly who he is
- Who needs this?

Popular Pages
New Podcasts
Top 25 Podcasts
Top Rated Podcasts
Search for a Podcast
100 Most Recent Podcasts
Podcast News Archive
Podcasting Jobs

Check out Some Podcasts!
Beyond Jazz
Bicyclemark's Audiocommunique
CatFish Show
Chub Creek
Concert Blast!
Disposable Radio
Domestic Life
The DV Show
EarthCore
EZHelp
Florn.net
Gardner Writes
Hoboken Rock City
Incoherent Mumblings
Indie Airplay
Israelisms
Kickbiking Podcasts
Media Artist Secrets
Mike Tech Show
MotoGPod
The Mushroom Accelerator
Noise Inside My Head
On The Wing

« Odeo Beta Generates Controversy | Main | Microsoft Censoring Free Speech in China »

Torcasting Lets Podcasters Speak Their Minds Anonymously

June 18, 2005

UndergroundMedia.org has published an **introduction to Torcasting Article**, a method for publishing content to the web anonymously.

Recent events, including **a podcaster losing his job** over his show, Microsoft's censoring of messages in China and the revealing of the identity of Deep Throat, highlight the fact that **freedom of speech sometimes requires the freedom to speak**.

UndergroundMedia is an outlet for media information to help empower the average person to take an active role in media.

"In order to have a true voice, one has to also have the ability to be anonymous," notes UndergroundMedia. "Privacy is a key component of freedom of press and expression. This is where Tor comes in."

By combining Torcasting with anonymizing voice effects, podcasters can speak their minds while preserving their anonymity.

About Tor & Torcasting

Tor is an "onion routing" project. An onion router is a way of creating tunnels among a group of servers that encrypt traffic and create multiple layers of security, and in turn privacy, for those connecting to the network.

When you use Tor, you can set up hidden services, including a web server that is only available to other Tor users. This means that the server you are running and the services it offers are anonymous. By setting up a hidden Tor server, you can distribute MP3 files, while obscuring your location and identity.

Comments

Anonymous Employee - Communication without the risk.

Anonymous Login

Employee Login
Employer Login

I Need Action

Anonymous Message
Anonymous Dialogue
Report a Problem
Request a Mediator
Get Legal Advice
Anonymous Survey

Site Details

Employee Details
Recipient Details
Sarbanes-Oxley
Bill C-11 and C-46
Code of Ethics

Common Questions

How is it Anonymous?
How Does it Work?
Other F.A.Q.

Anonymous Employee provides a communication process for workplace issues.

Anonymous Employee provides employees with the opportunity to express anonymous problems and concerns in the workplace. Employees can inform their employer of the issues they face in the workplace, without needing to reveal their identity. As an employee, you can recommend solutions to problems so that your employer can better understand your needs. Registered members have the added benefit of being able to have a complete two way communication process. In short, you can not only just send a message, but you can also get a complete detailed response from your employer. The entire process can take place within the privacy of your own home, library, or friend's house.

Many times, your employer does not even know that a problem exists. Before a problem gets out of hand, take a moment to try to solve it. Instead of resorting to quitting your job or taking costly legal action, why not try to resolve your problem through anonymous communication? No matter what your issue is, you can get a confidential response from the person of your choosing, at any company. This can be your supervisor, manager, or third party. If your issue is serious enough, we can contact your company on your behalf so that they are doubly aware of the problem.

If your issue has escalated beyond the point where communication can solve the problem, then Anonymous Employee also offers you the ability to request assistance such as professional mediation or legal advice. Issues such as corporate fraud, harassment, workplace bullying, pay inequities, unjust termination, workplace safety and more should be resolved in a positive, constructive setting where you can speak out without jeopardizing your job. All great change comes from within, and this is your opportunity to express what changes need to happen at your job.

Are Hidden Servers Safe?

- Later on we'll tell you how hidden services, such as torcasting are set up
- But how safe are they?

Are Hidden Servers Safe?

- Later on we'll tell you how hidden services, such as torcasting are set up
- But how safe are they?

- We will show how someone with a single machine can quickly find any hidden server...
- Rather they previously could: before the countermeasures we will describe were implemented

Talk Outline

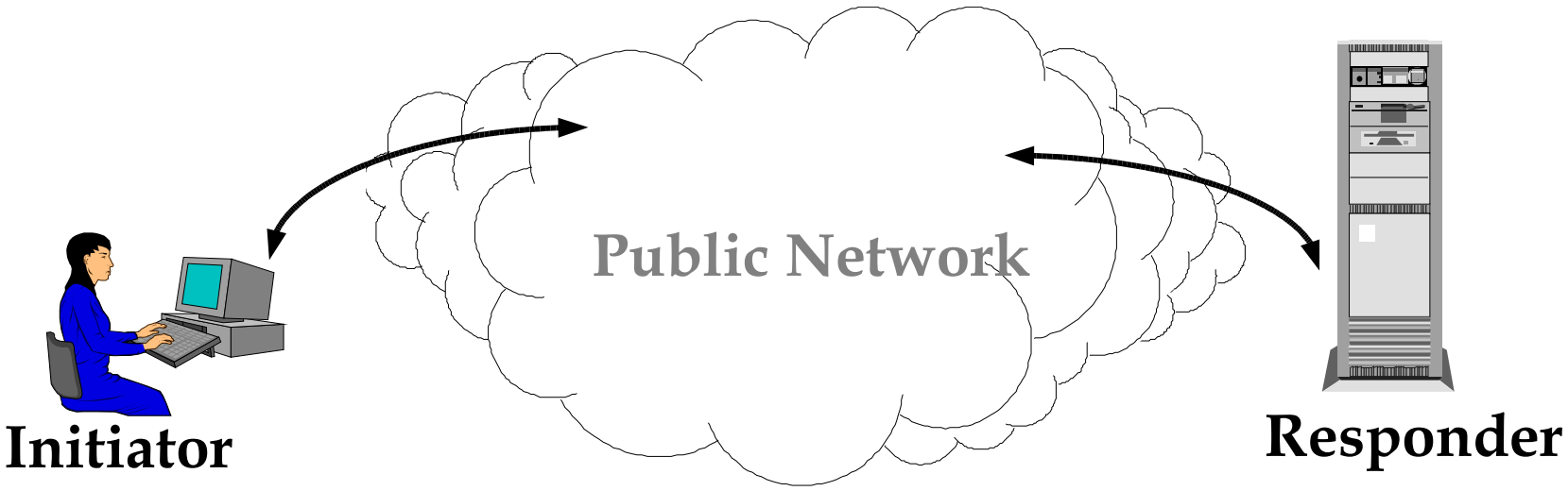
- Hidden servers in brief: Demo Setup
- Motivation: Why anonymous communication?
  - Personal privacy, Corporate and governmental security
- General overview of practical anonymous communication
- Overview of Tor: Third generation Onion Routing
- Hidden servers in more detail
- Description of our attacks on hidden servers
  - Timing, Round Trip Time, Client, Statistical, Two Nodes Attack
- Experimental results of our attacks
- Countermeasures
  - What's Implemented, what's not
- Demo Results and Failures
- Conclusions and Future Work

Credits

- Some slides on mixes cribbed from Ari Juels (with permission)
- Tor, Generation 2 Onion Routing and Hidden Service Design
  - with Roger Dingledine and Nick Mathewson
  - Roger and Nick did all the implementation
  - Based on original Onion Routing systems done with David Goldschlag and Michael Reed
- Numerous research predecessors and contemporaries: ZKS Freedom, JAP Webmixes,...
- Funding by ONR, DARPA, and EFF
- Lots of volunteers contributing to open source Tor code, running nodes, running hidden services, 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?

- Political Dissidents, Whistleblowers, Censorship resistant publishers
- Socially sensitive communicants:
  - Chat rooms and web forums for abuse survivors, people with illnesses (diabetes-people.de)
- Law Enforcement:
  - Anonymous tips or crime reporting
  - Surveillance and honeypots (sting operations)
- Corporations:
  - Fraud, abuse, safety reporting. Sarbanes-Oxley law etc.
  - Who's talking to the company lawyers? Are your employees looking at monster.com?
  - Hide procurement suppliers or patterns
  - Competitive analysis

Who Needs Anonymity?

- You:
  - Where are you sending email (who is emailing you)
  - 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

Location Hidden Servers

- 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
- Resist attack from authorized users
- Resistant to physical attack (you can't find them)

Who Needs Anonymity?

- And yes criminals

Who Needs Anonymity?

- And yes criminals

But they already have it.

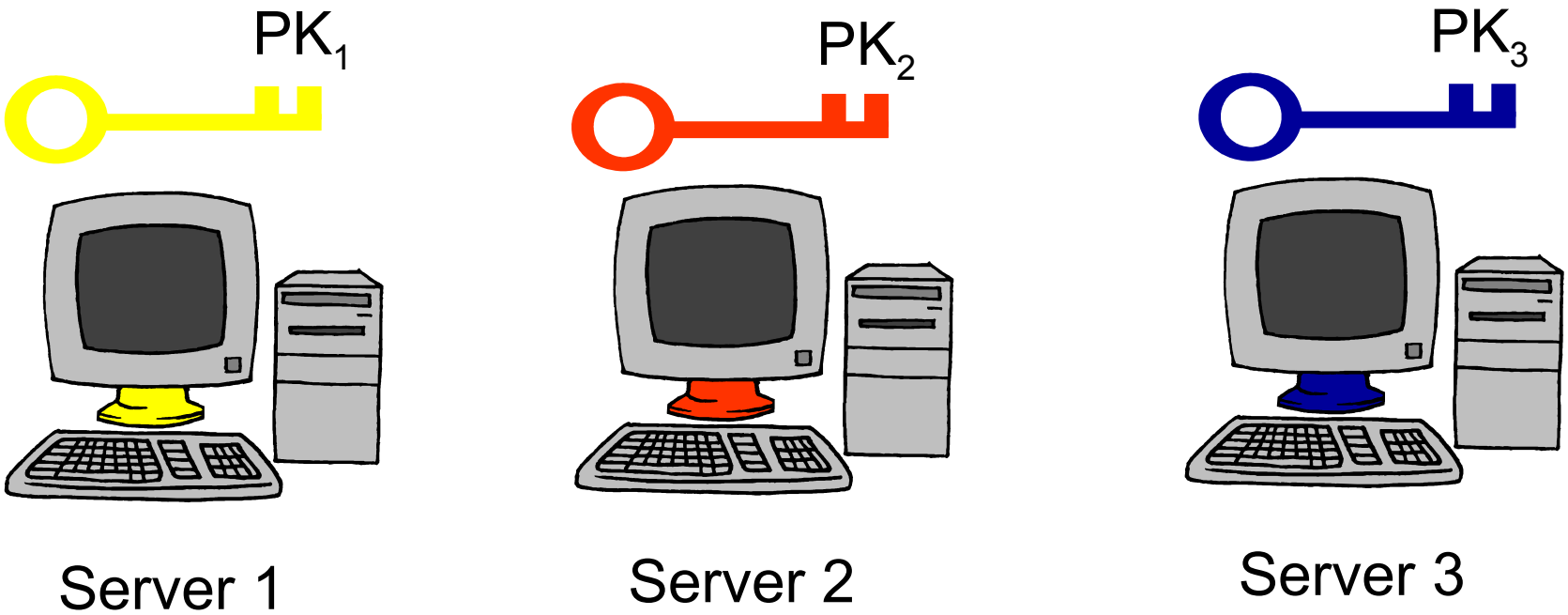
We need to protect everyone else.

Anonymity Loves Company

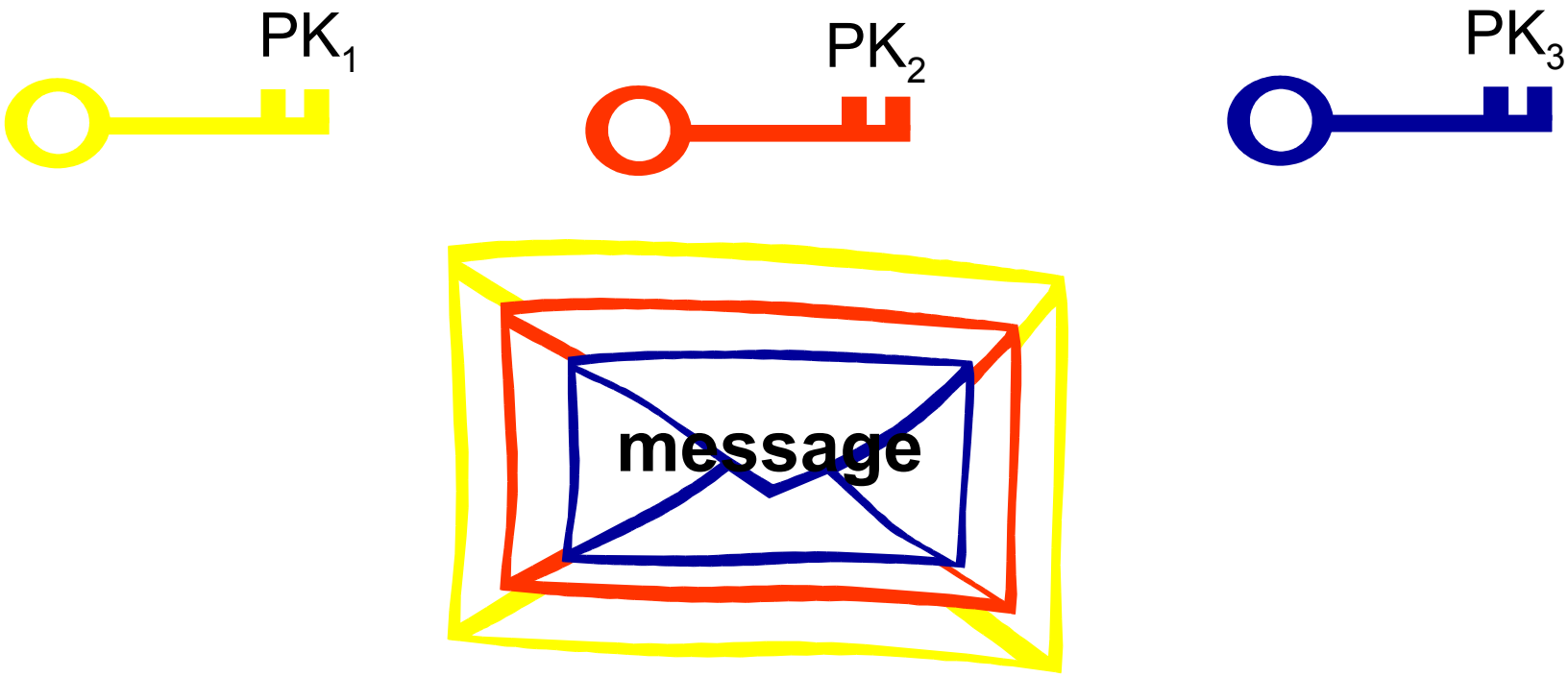
- You can't be anonymous by yourself
  - Can have confidentiality by yourself
- A network that protects only DoD network users won't hide that connections from that network are from Defense Dept.
- You must carry traffic for others to protect yourself
- But those others don't want to trust their traffic to just one entity either. Network needs *distributed trust*.

Focus of Tor is anonymity of the communication pipe,
not what goes through it

Basic Mix (Chaum ‘81)

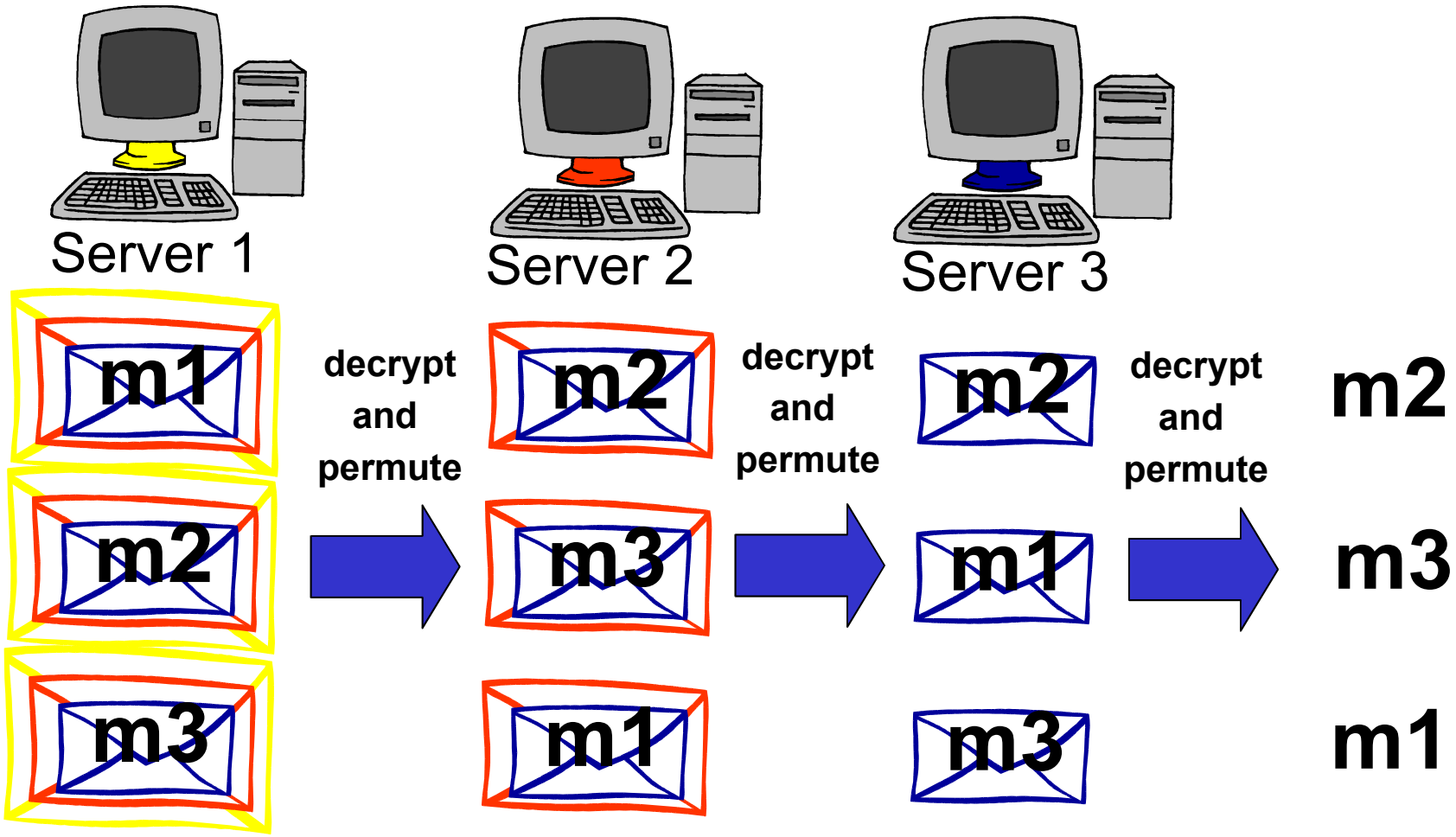


Encryption of Message

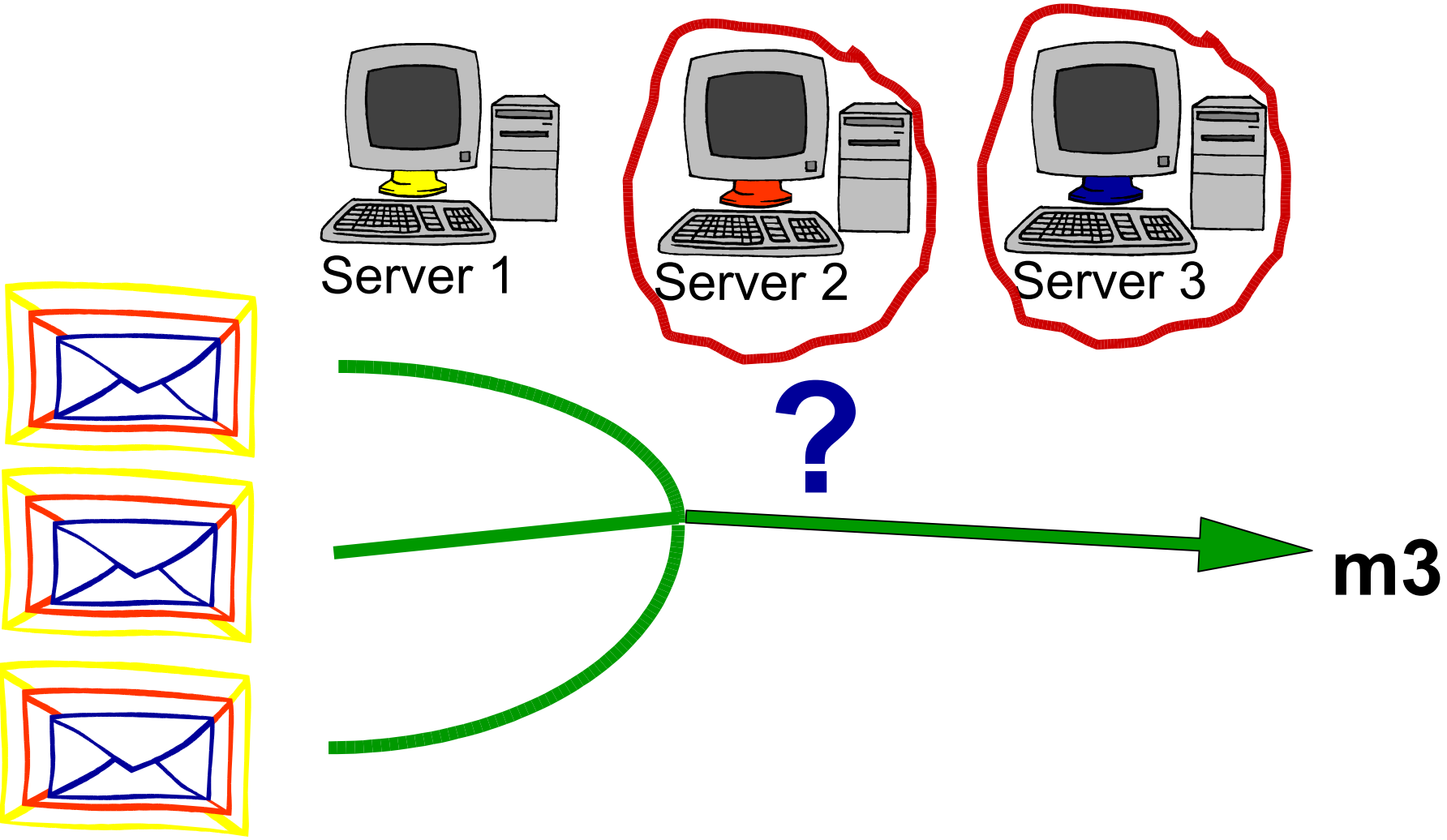


$$\text{Ciphertext} = E_{PK1}[E_{PK2}[E_{PK3}[\text{message}]]]$$

Basic Chaum-type Mix

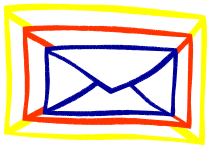


One honest server preserves privacy

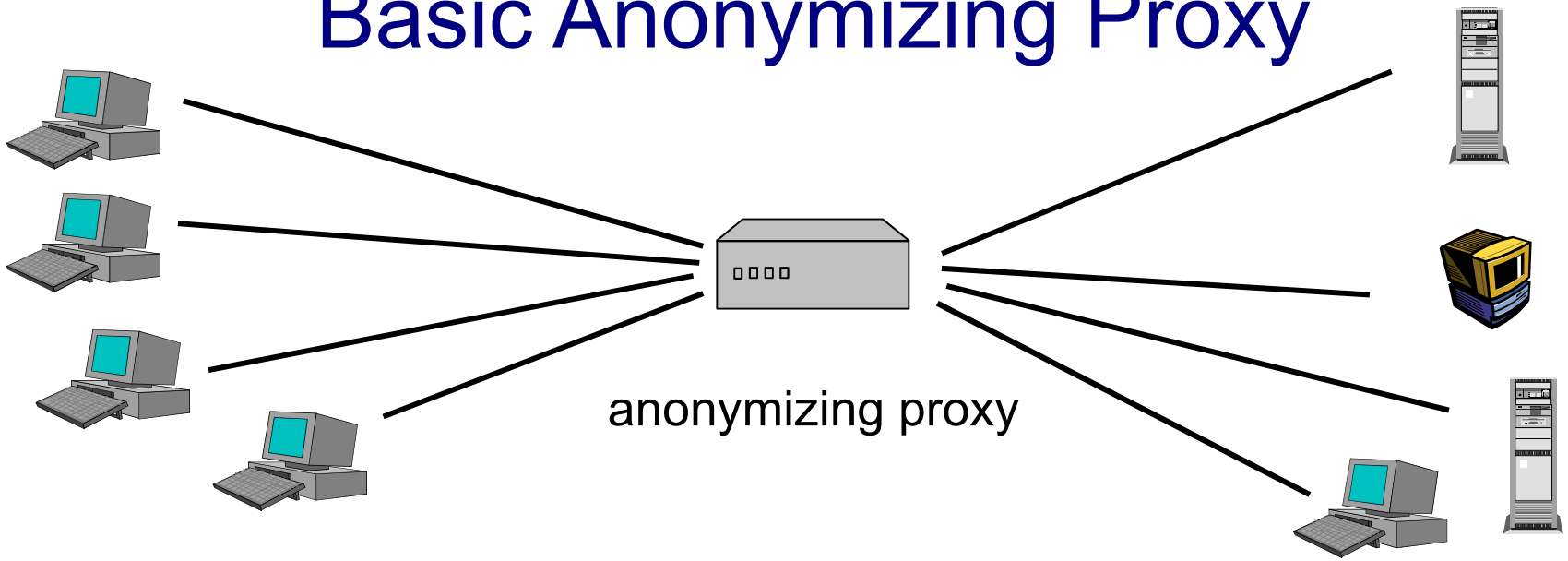


What if you need quick interaction?

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



Basic Anonymizing Proxy



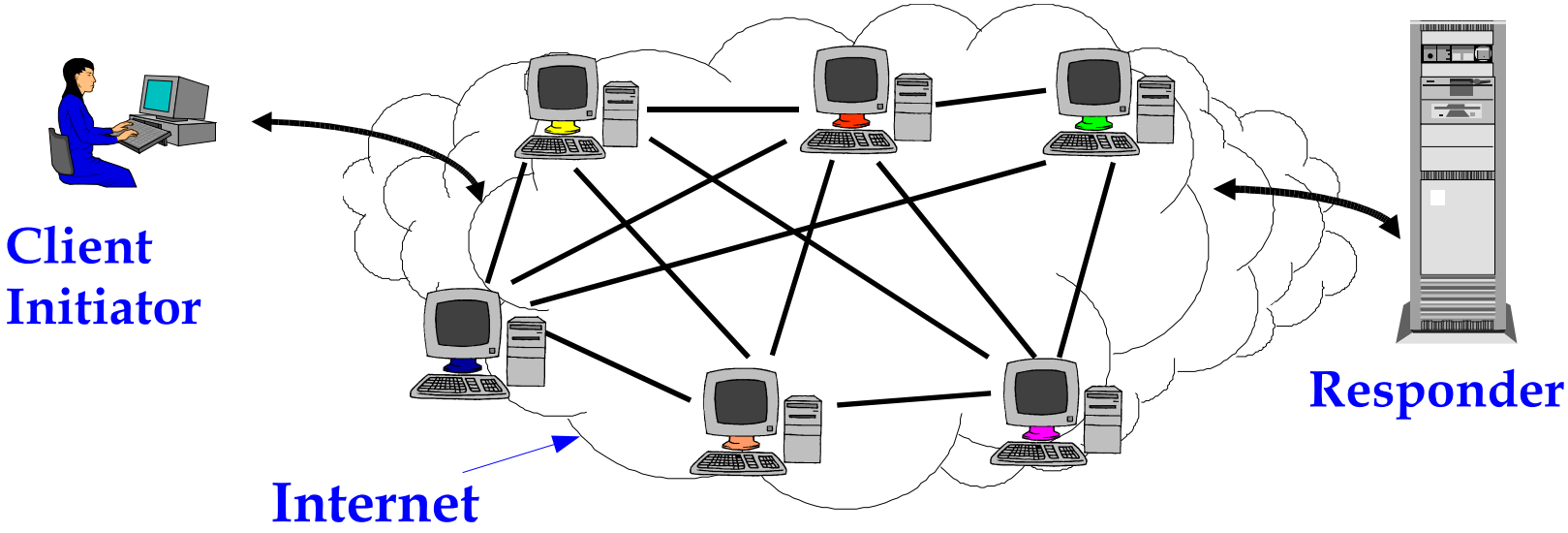
- Channels appear to come from proxy, not true originator
- Appropriate for Web connections, etc.:
  SSL, TLS, SSH (lower cost symmetric encryption)
- Examples: The Anonymizer
- Advantages: Simple, Focuses lots of traffic for more anonymity
- Main Disadvantage: Single point of failure, compromise, attack

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

Network Structure

- Onion routers form an overlay network
  - TLS encrypted/authenticated connections
- Proxy interfaces between client machine and onion routing overlay network



Tor

Tor

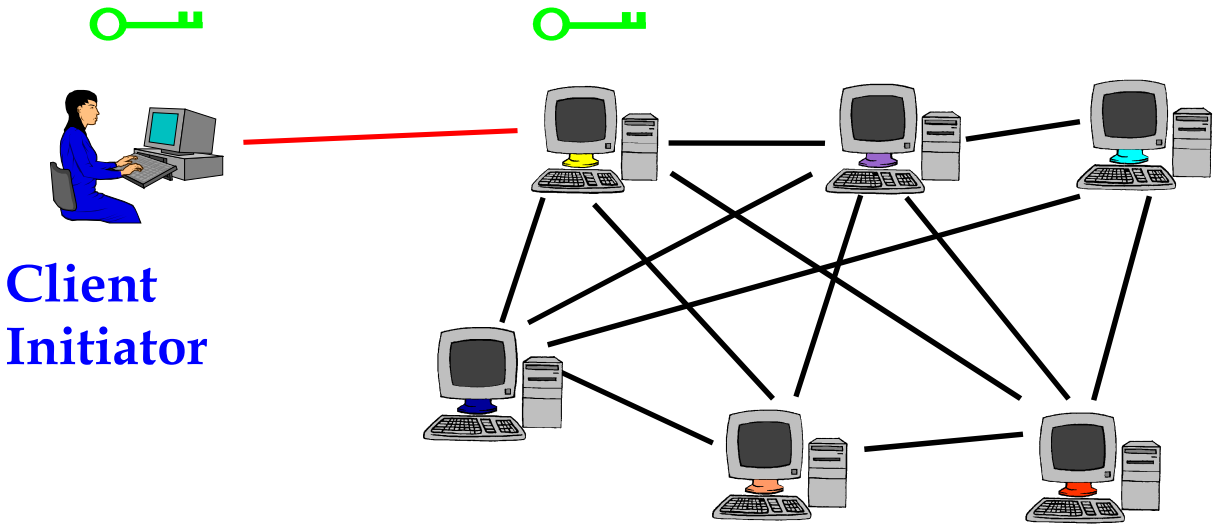
The Onion Routing

Tor

Tor's Onion Routing

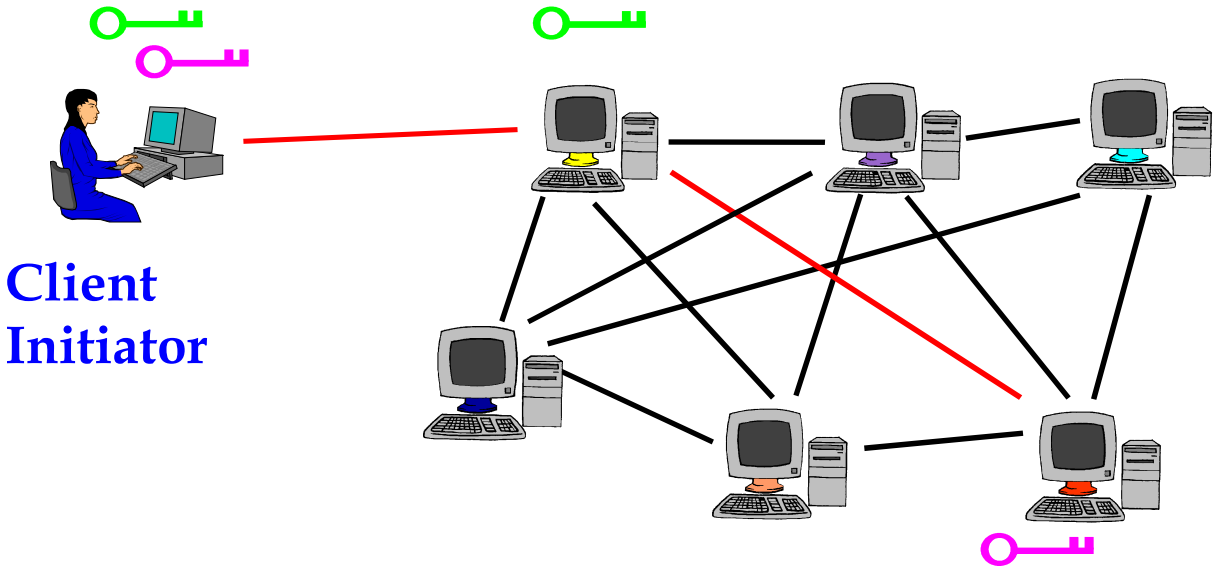
Tor Circuit Setup

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



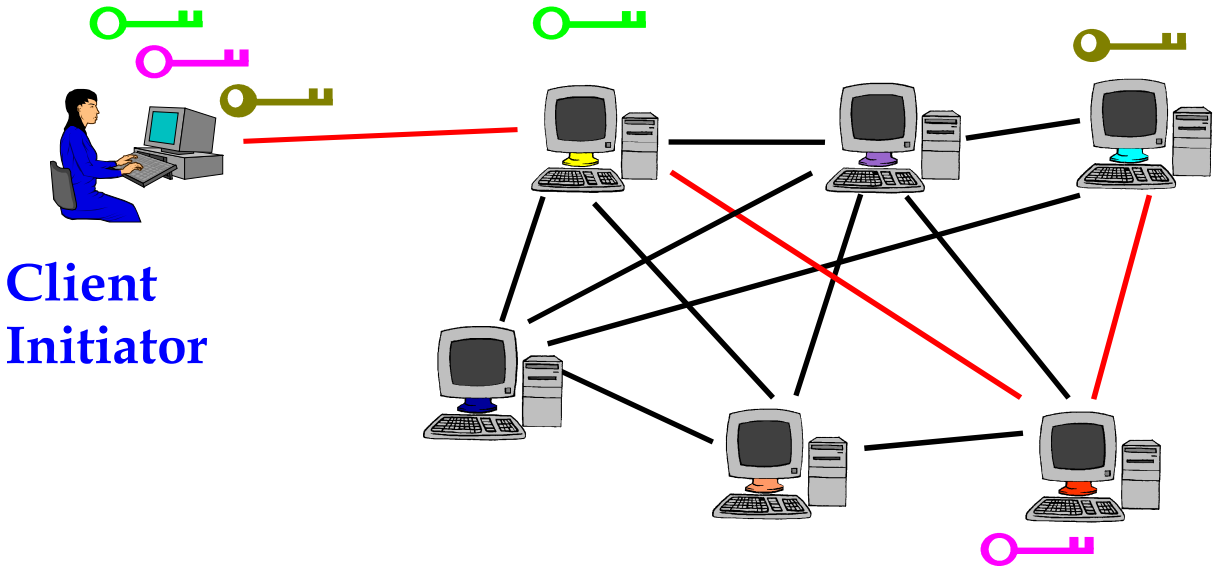
Tor Circuit Setup

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



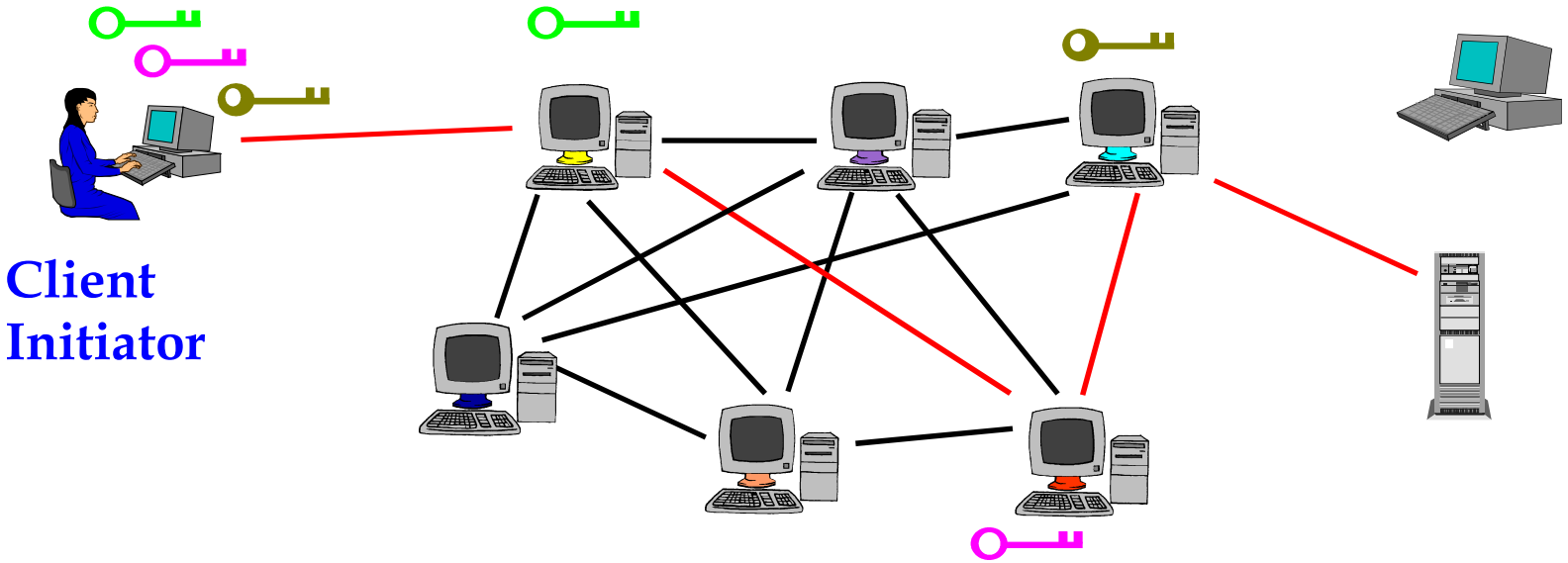
Tor Circuit Setup

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



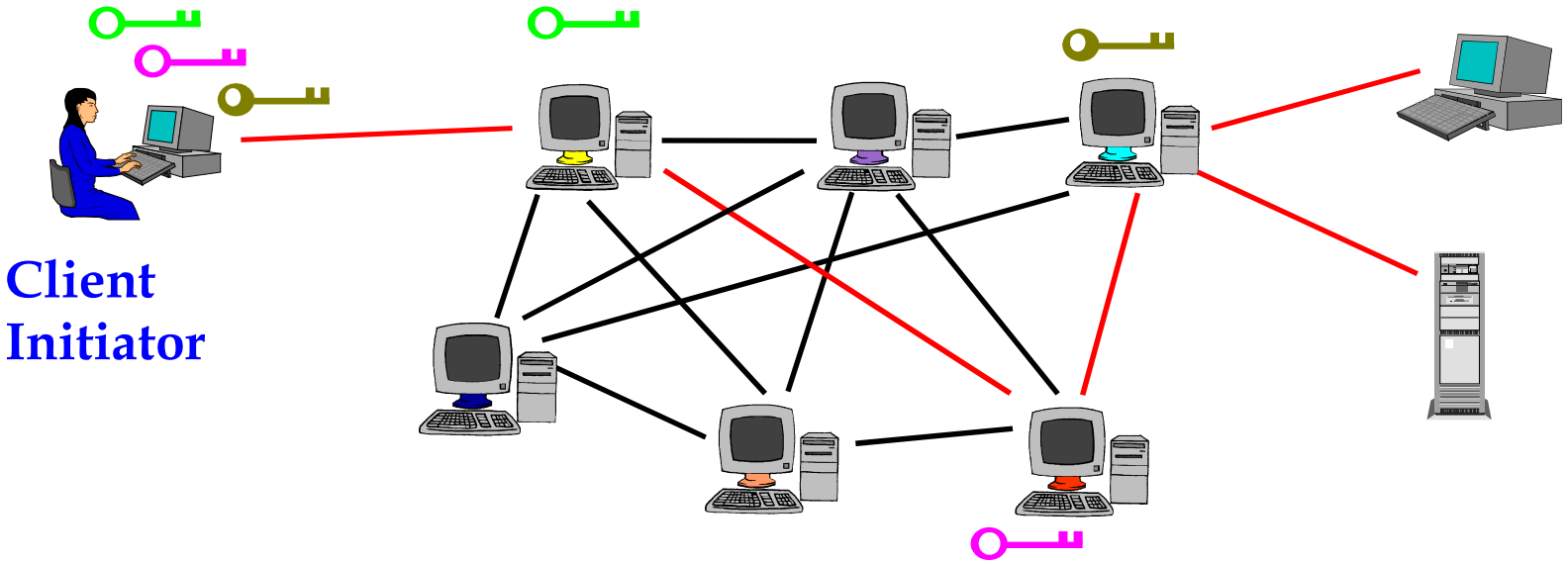
Tor Circuit Usage

- Client Proxy establishes session key + 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



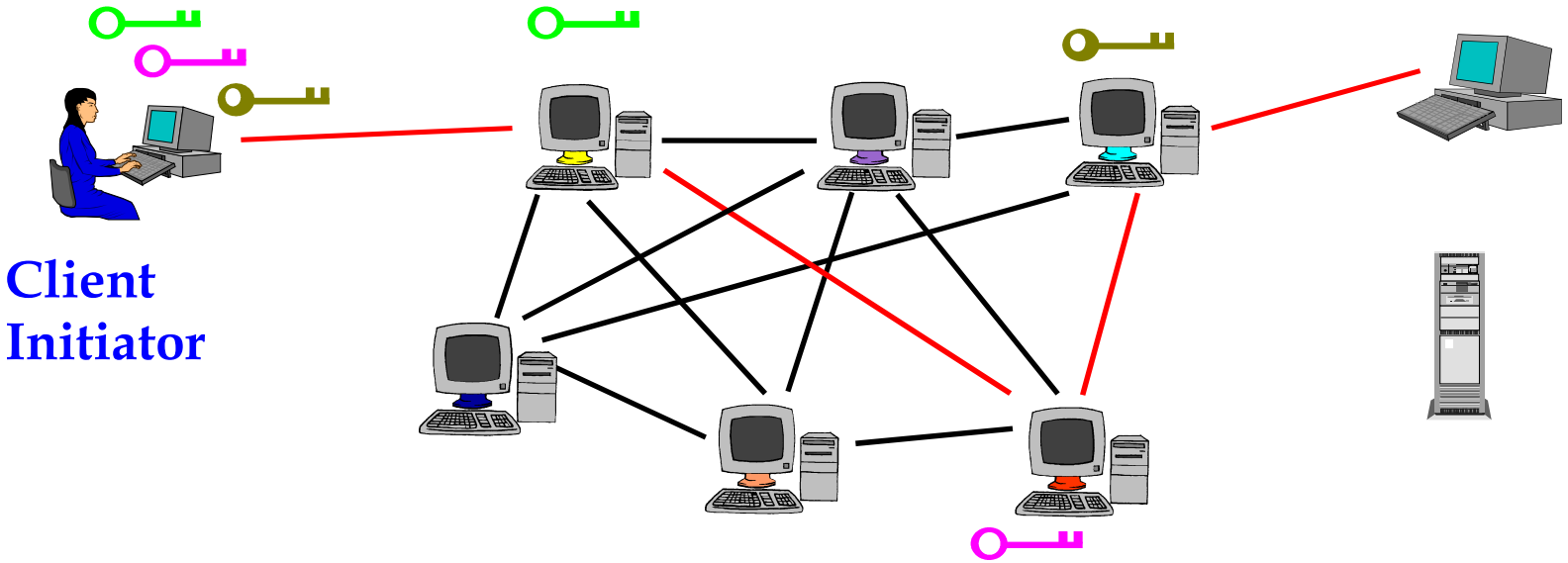
Tor Circuit Usage

- Client Proxy establishes session key + 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 + 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.
  - Directory server keys ship with the code
  - These directories are cached and served by other servers, to reduce bottlenecks

Some Tor Properties

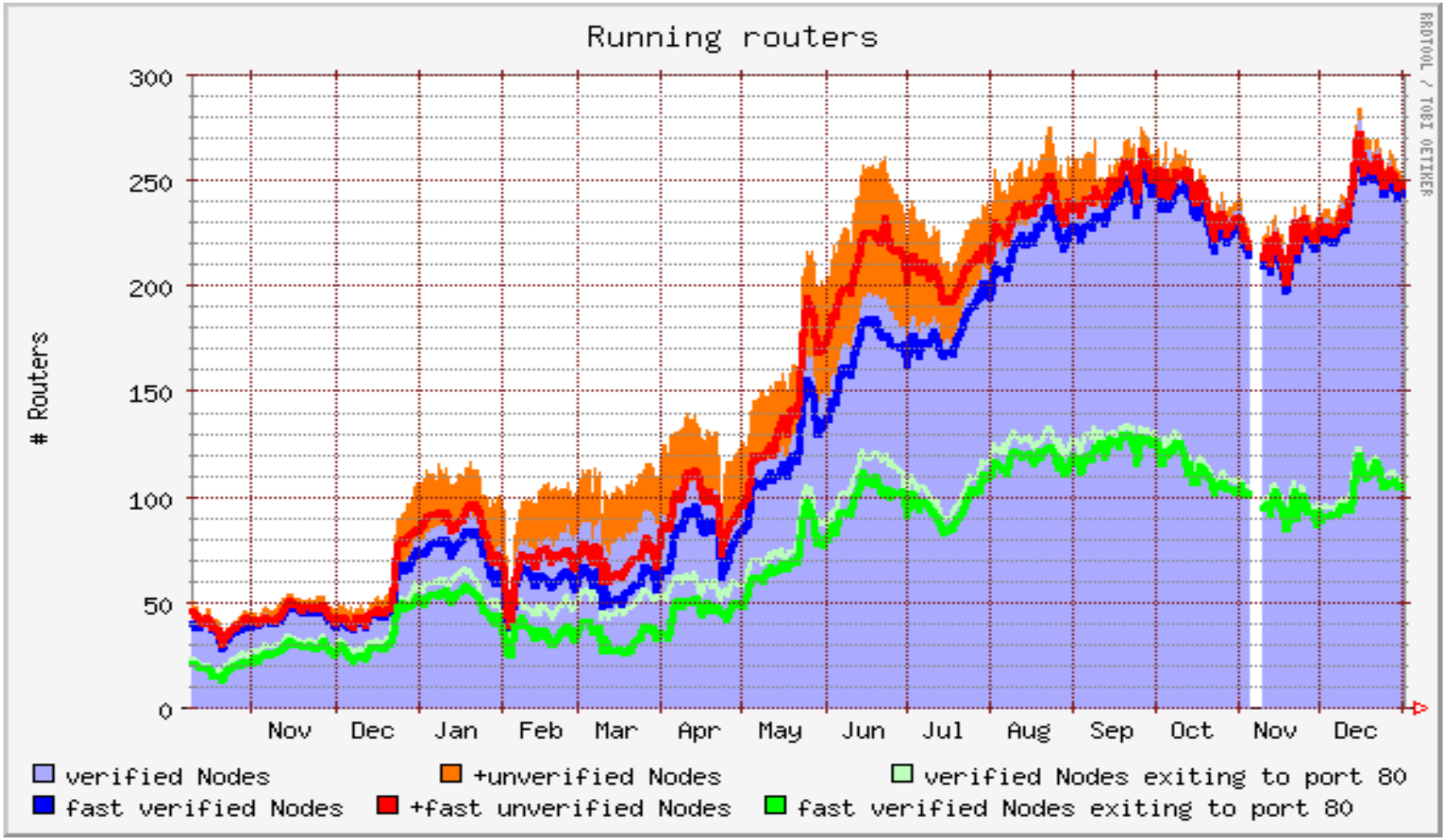
- Simple modular design, Restricted ambitions
  - 40-50K lines of C code
  - Well documented
  - Even servers run in user space, no need to be root
  - Just anonymize the pipe
    - Can use, e.g., privoxy as front end if desired to anonymize data
  - SOCKS compliant TCP: includes Web, remote login, mail, chat, more
    - No need to build proxies for every application
  - Flexible exit policies, each node chooses what applications/destinations can emerge from it
- Lots of supported platforms:

  Linux, BSD, MacOS X, Solaris, Windows

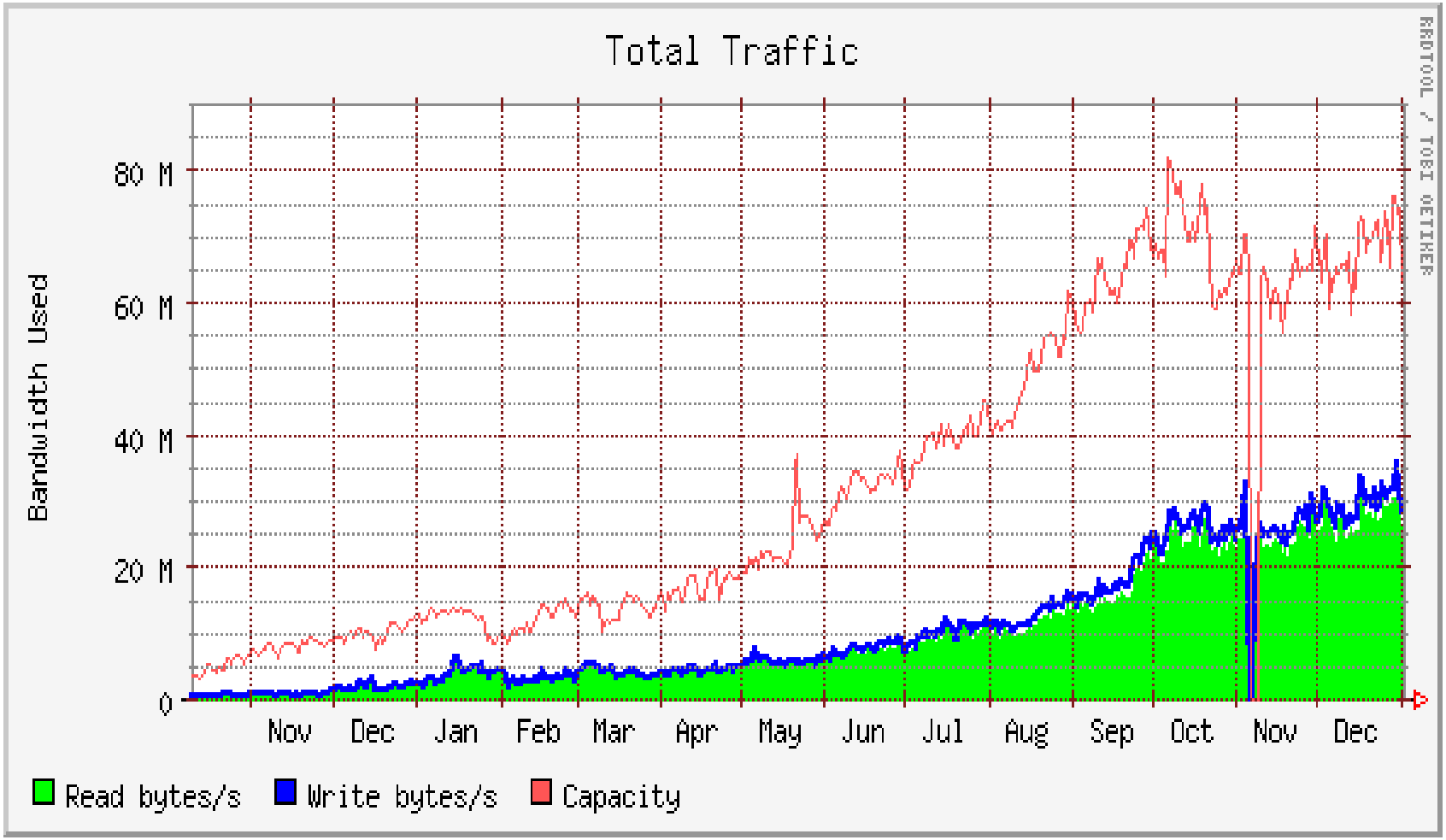
Numbers and Performance

- Running since October 2003
- About 250 nodes scattered through six continents
- Hundreds of thousands (?) of users
- Network processing c. 30 MB/sec application traffic
- Network has never been down since initial deployment

running Tor servers over last 15 months



Traffic rate over last 15 months



Location Hidden Servers

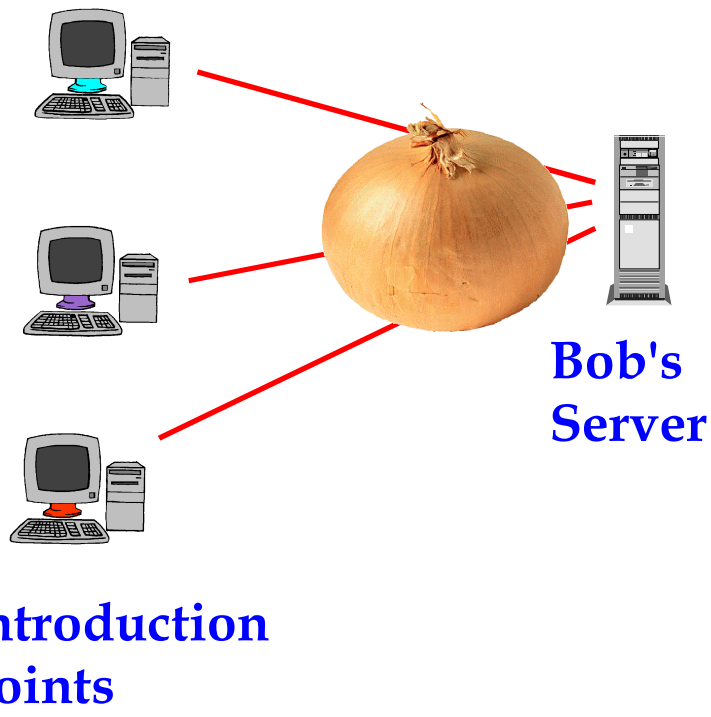
- Alice can connect to Bob's server without knowing where it is or possibly who he is
- Already told you why this is desirable, but...

- How is this possible?

Location Hidden Servers

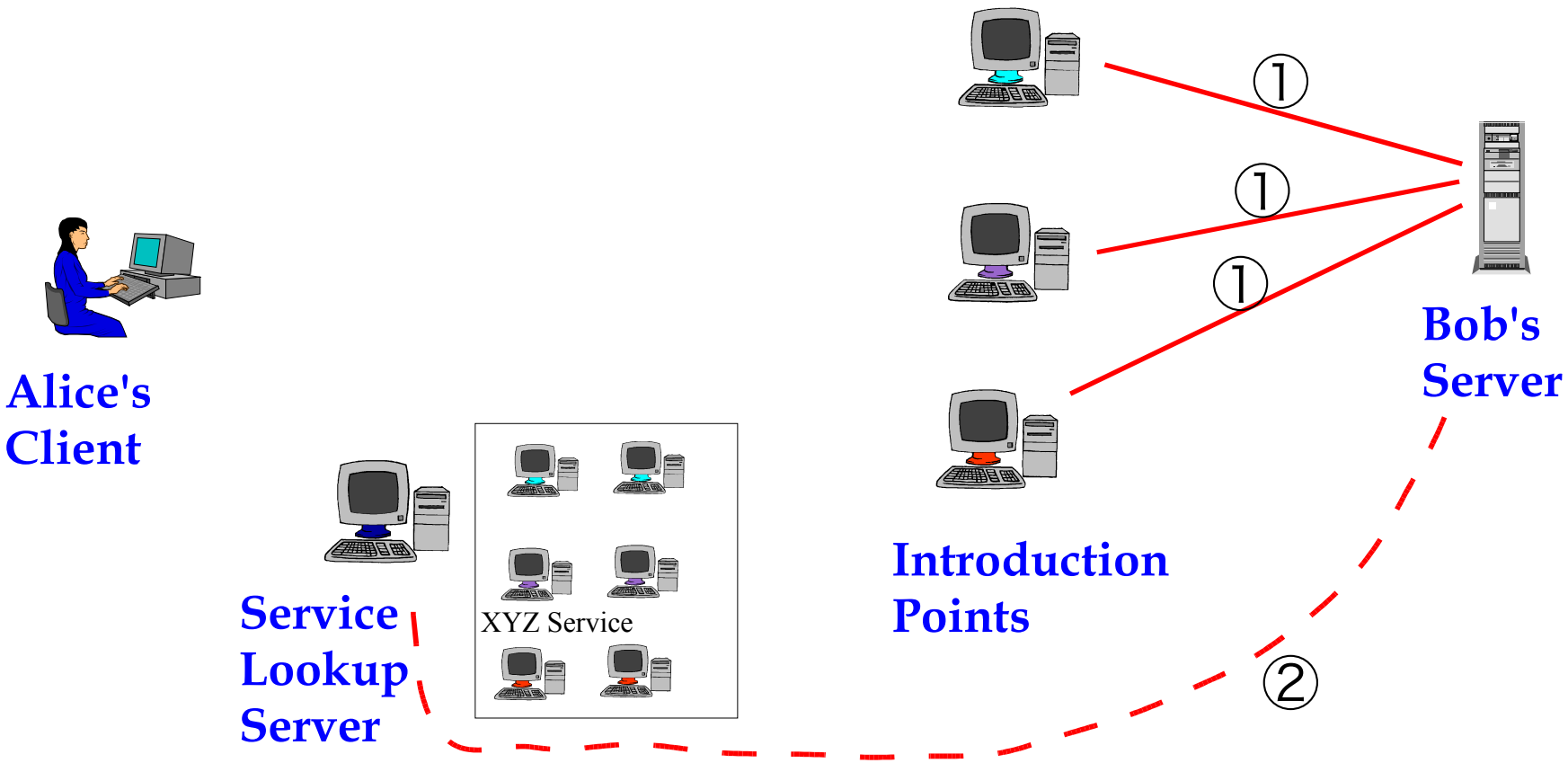
1. Server Bob creates onion routes to Introduction Points (IP)

   (All routes in these pictures are onion routed through Tor)



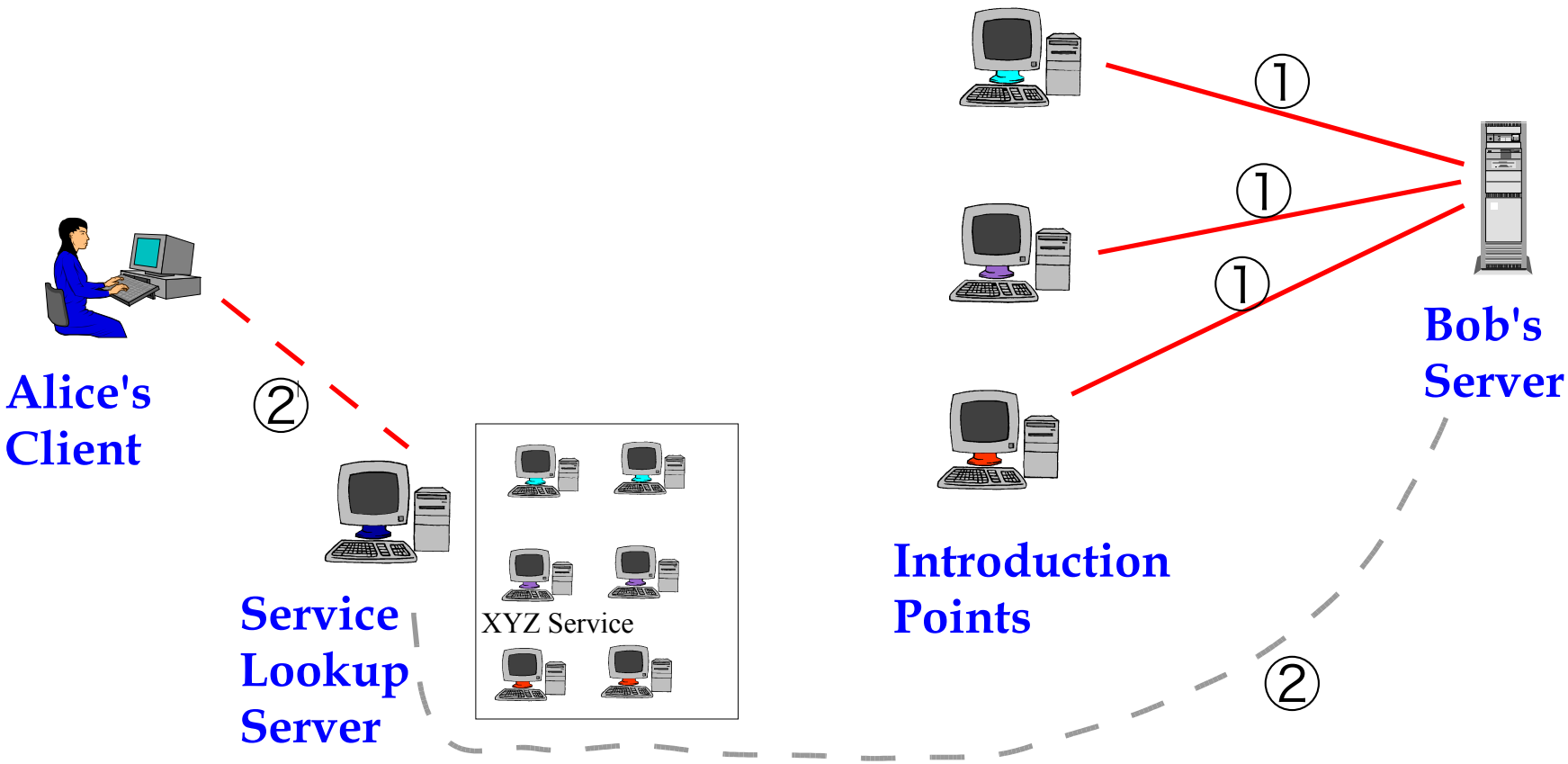
Location Hidden Servers

1. Server Bob creates onion routes to Introduction Points (IP)
2. Bob publishes his xyz.onion address and puts Service Descriptor incl. Intro Pt. listed under xyz.onion



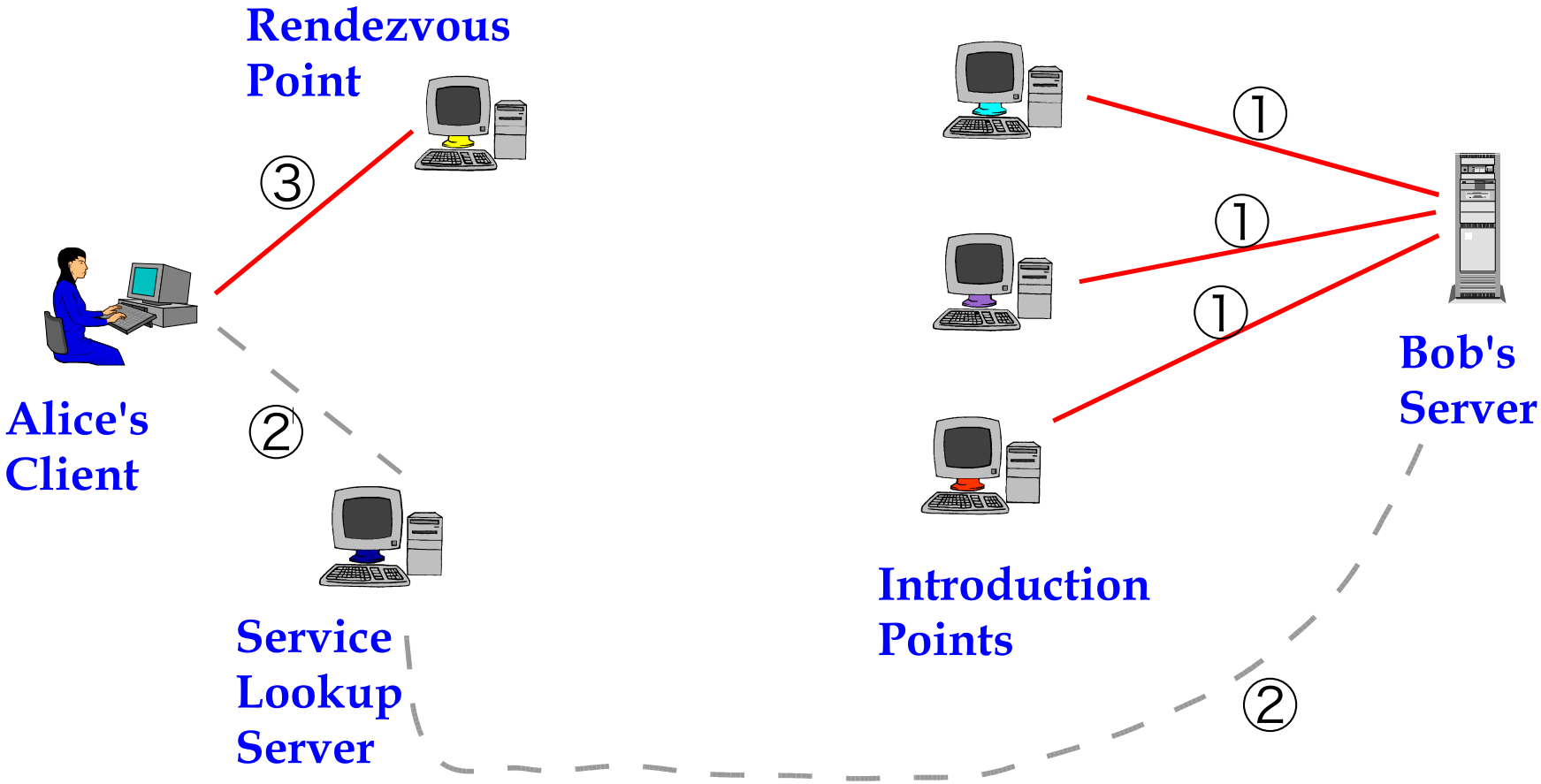
Location Hidden Servers

2'. Alice uses xyz.onion to get Service Descriptor (including Intro Pt. address) at Lookup Server



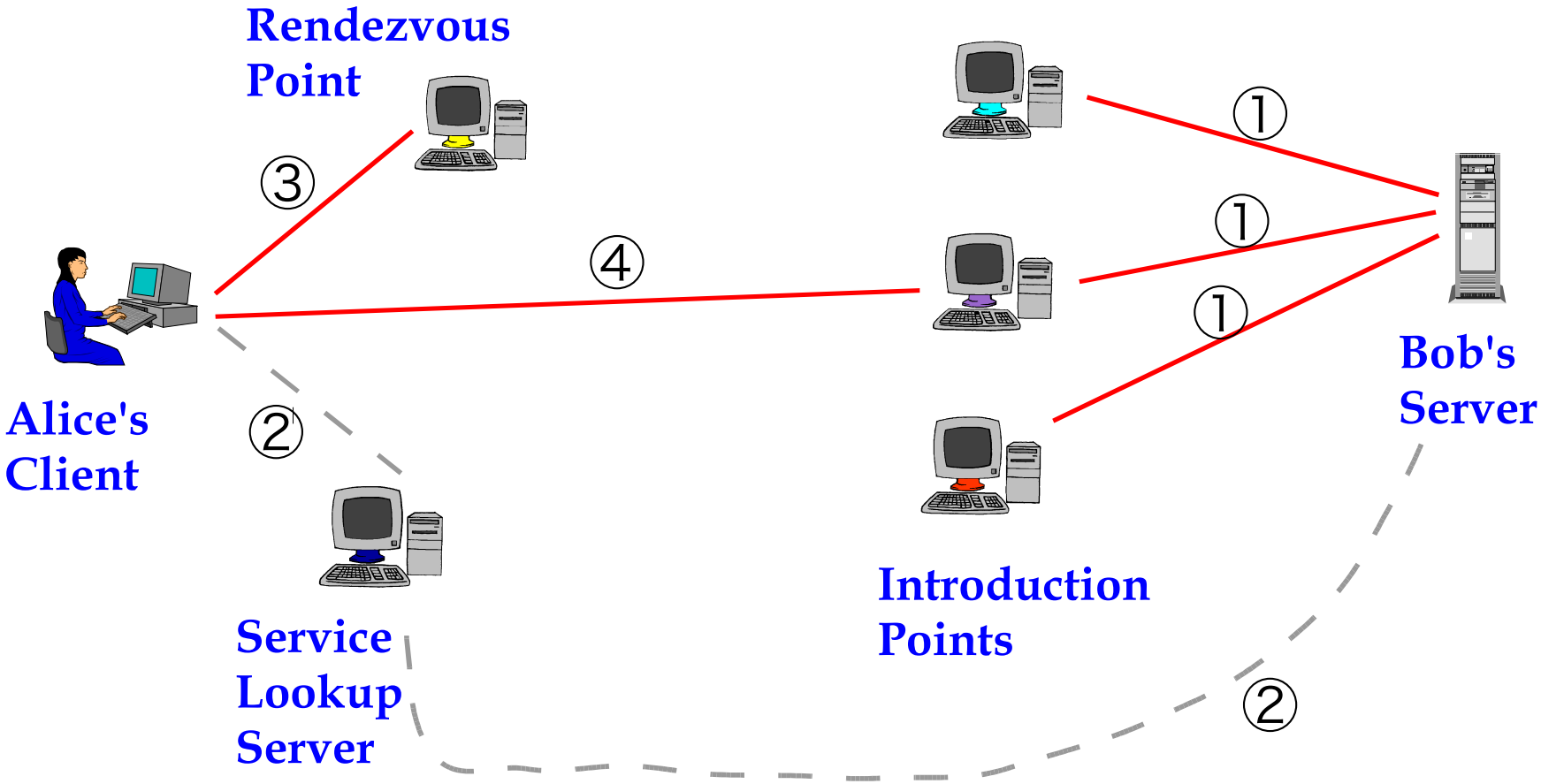
Location Hidden Servers

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



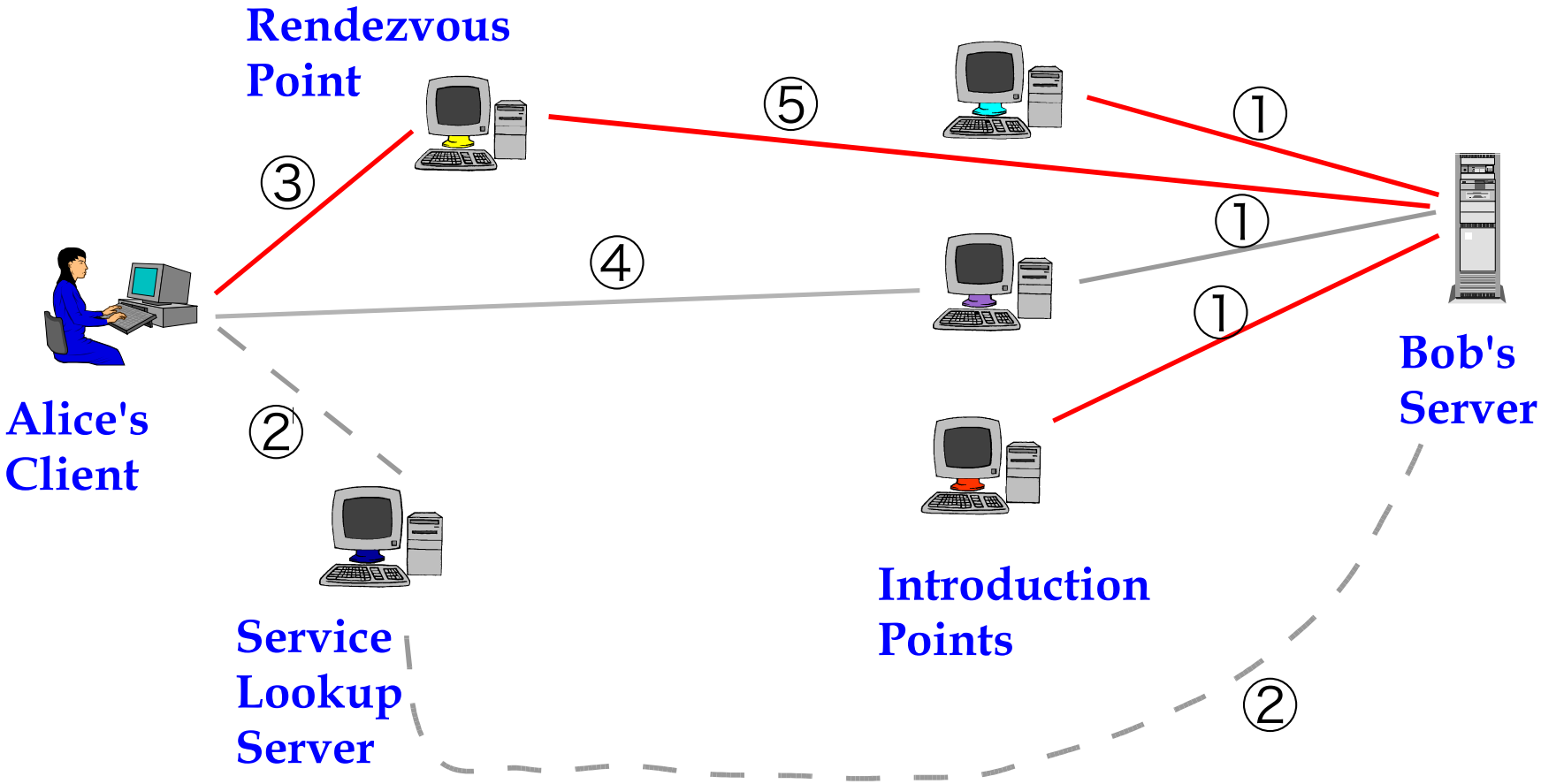
Location Hidden Servers

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



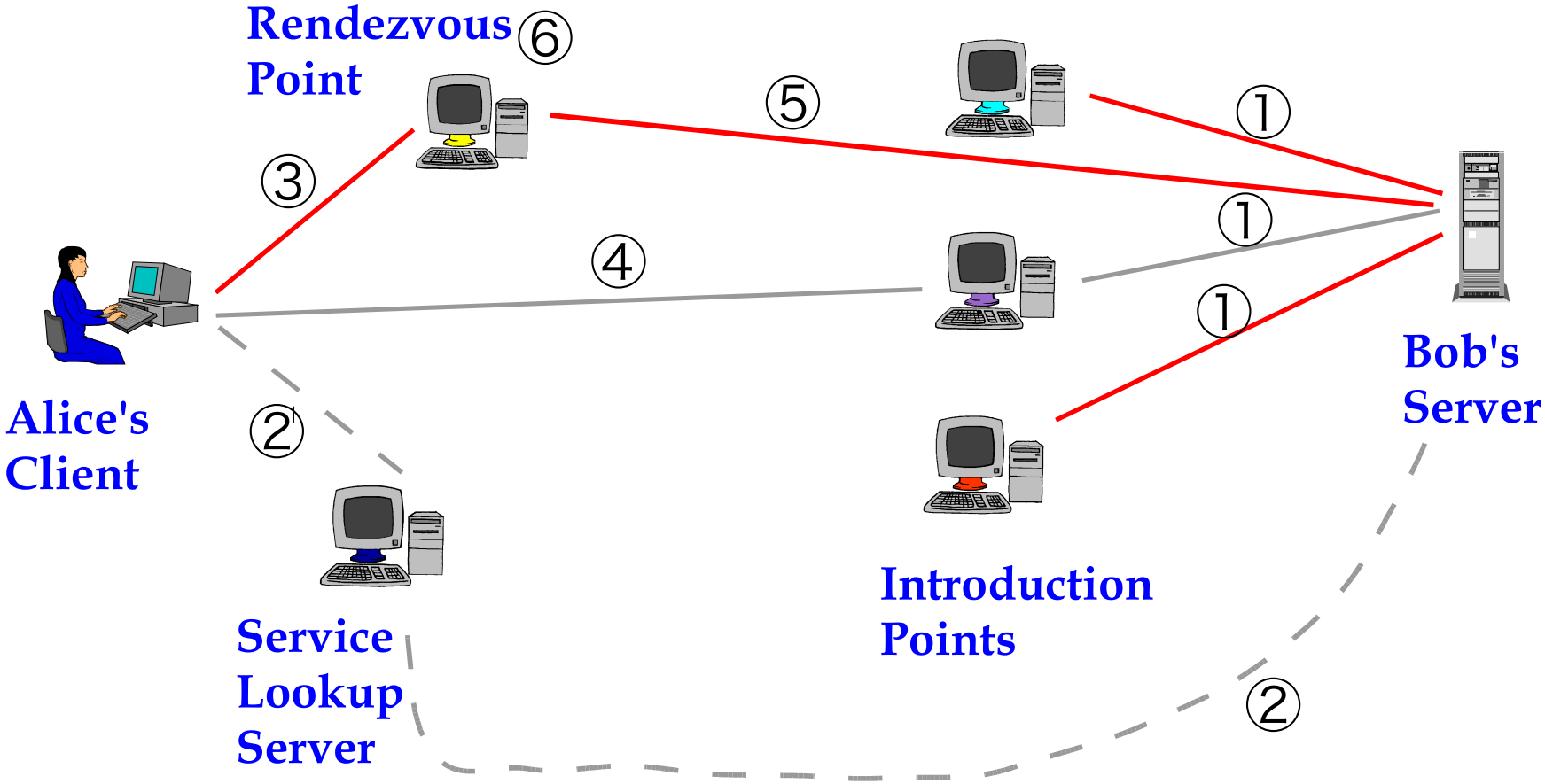
Location Hidden Servers

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



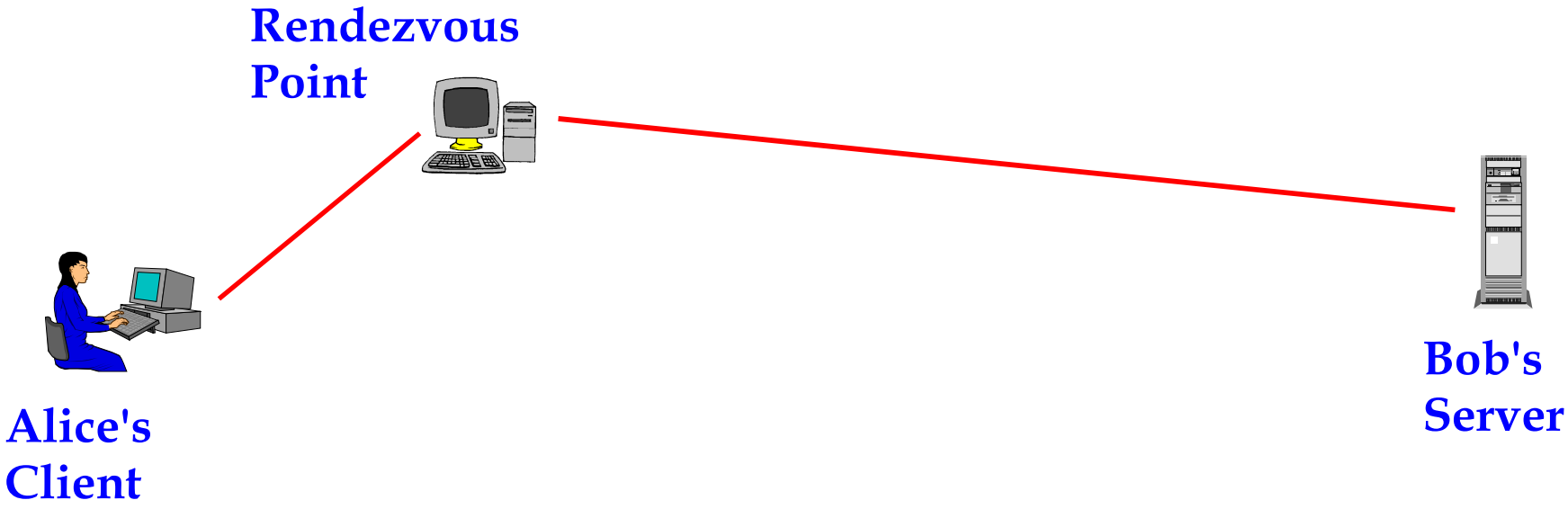
Location Hidden Servers

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



Location Hidden Servers

Final resulting communication channel



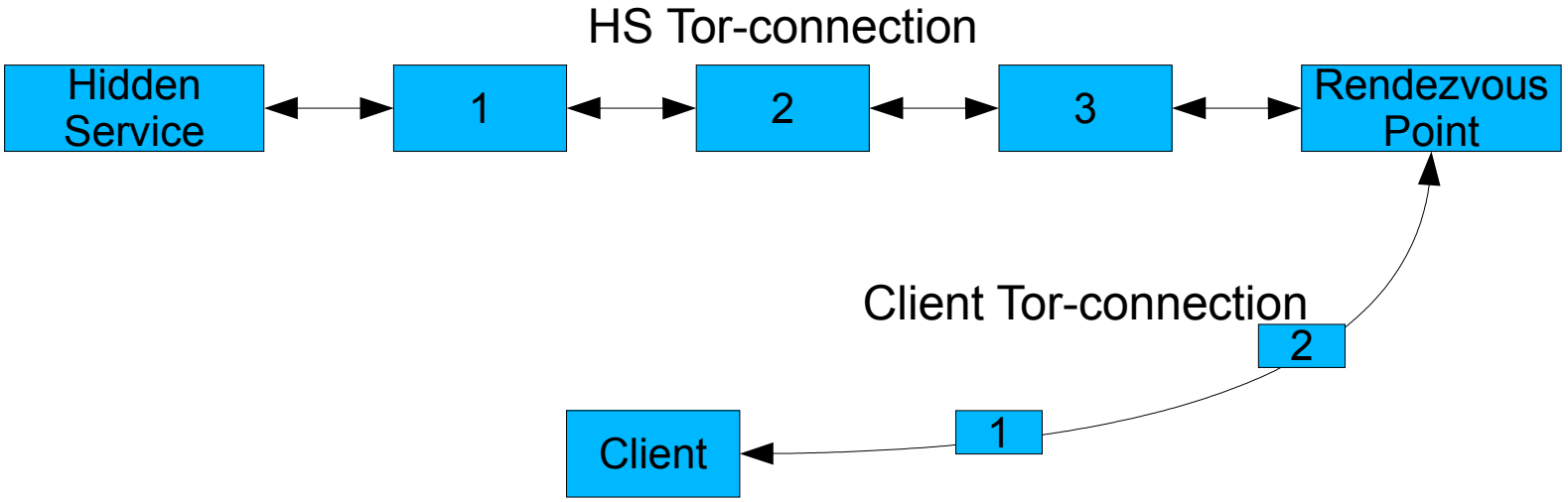
Location Attacks Outline

- Attack Types
- Attack Scenario
- Timing
- Round Trip Time
- Statistical Attack
- Client/Server challenge
- Two Node Attack
- Tor features

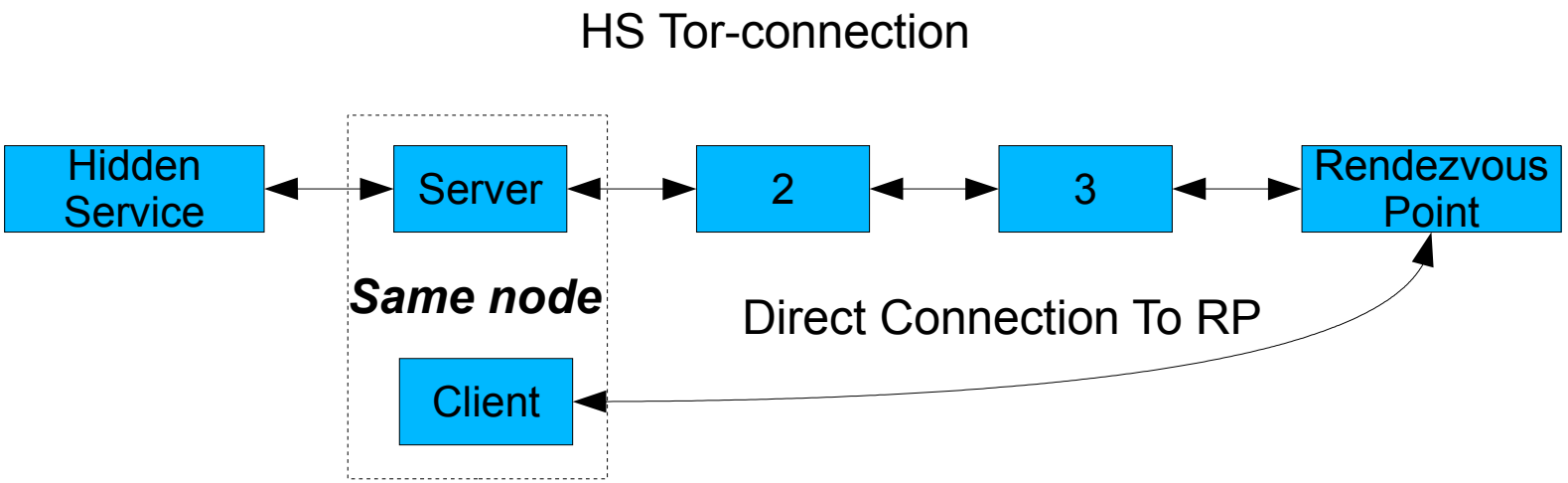
Hidden Services Attacks

- Locate server hosting a known location hidden service
- Attacks we will not discuss:
  - locate unknown location hidden services
  - locate user of our location hidden service
  - locate user of a publicly known location hidden service
  - locate user of an unknown location hidden service
  - identify list of publicly available hidden services
  - adding multiple nodes controlled by one entity
  - DoS - shutdown parts of the network

Normal Scenario Closeup



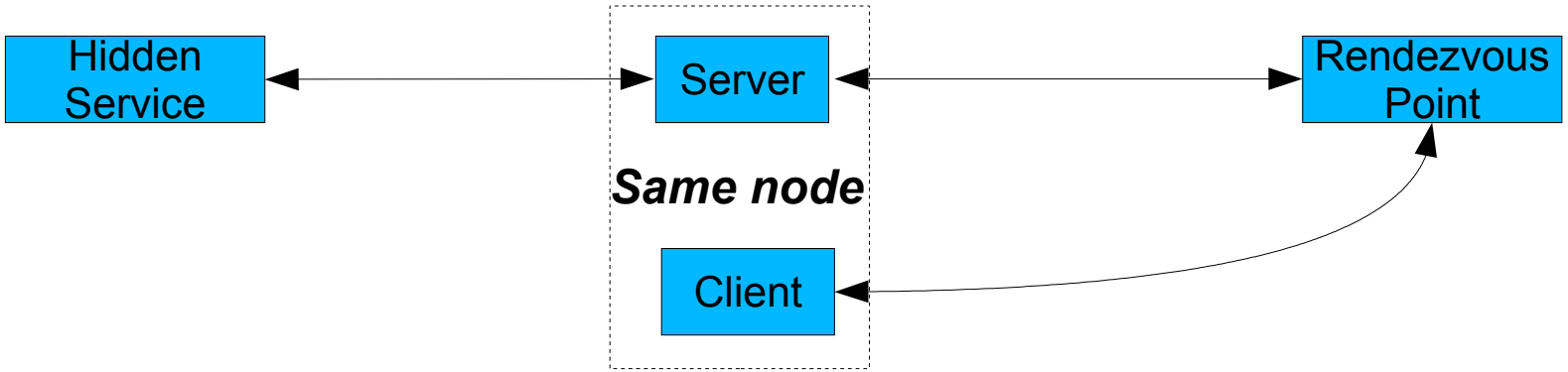
Attack Scenario Closeup



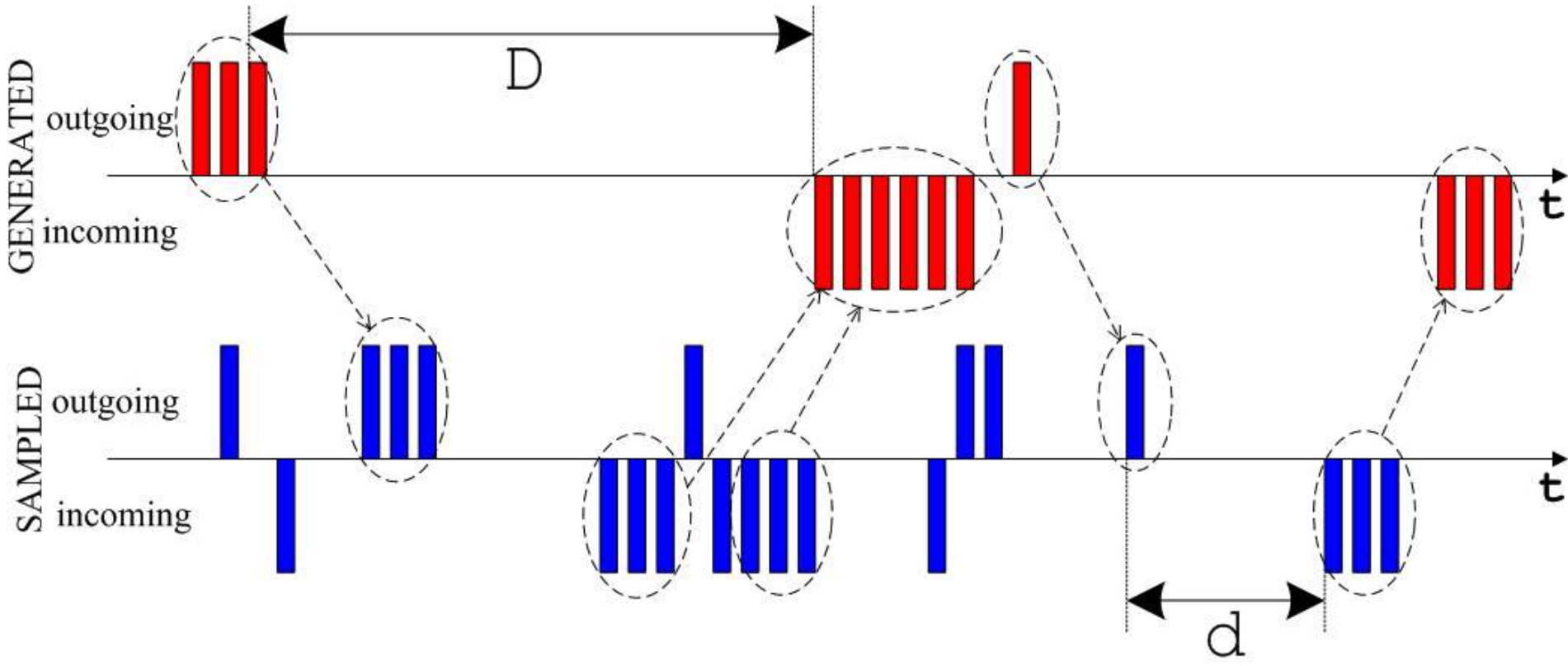
- We want to identify the situation shown above
  - Being used as first node by the location hidden service

Timing

- Client part can enforce any traffic pattern when sending data and the response is equally easy to tamper with at server part
- Combination makes circuit “easily” identifiable
- Know when we are on the path between HS and RP

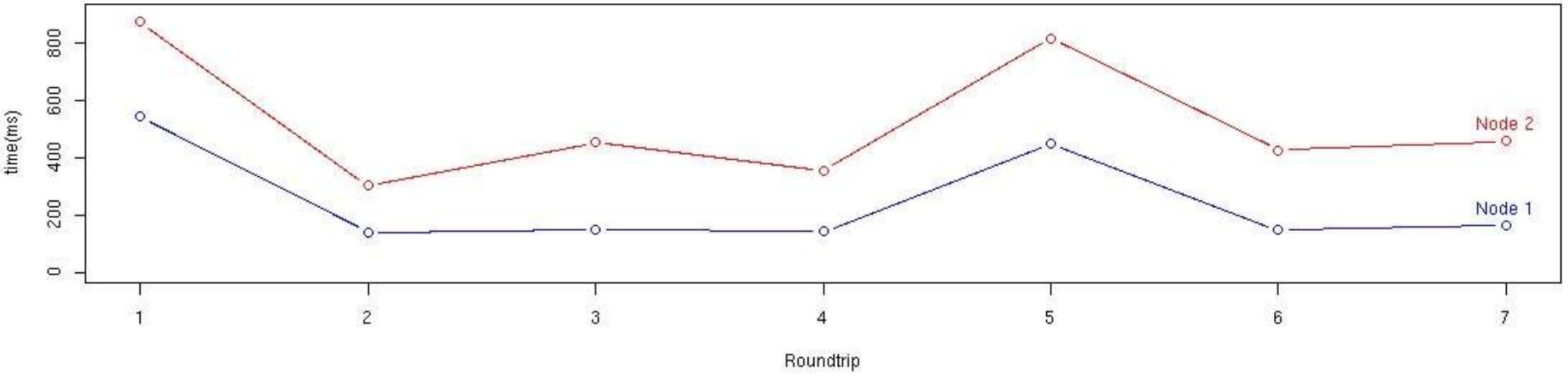


Round Trip Time



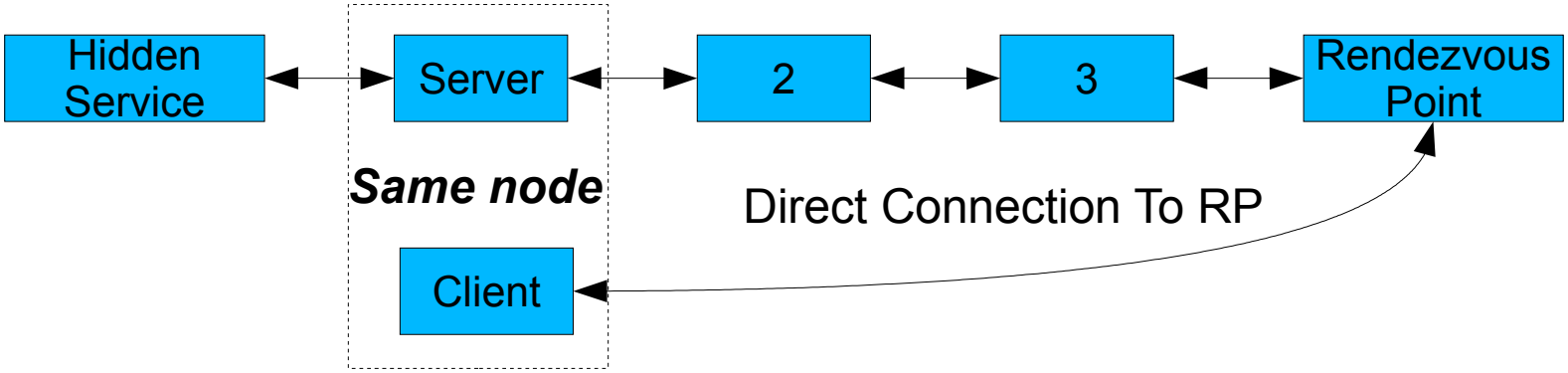
Round Trip Time (2)

- RTT may reveal distance to Hidden Service



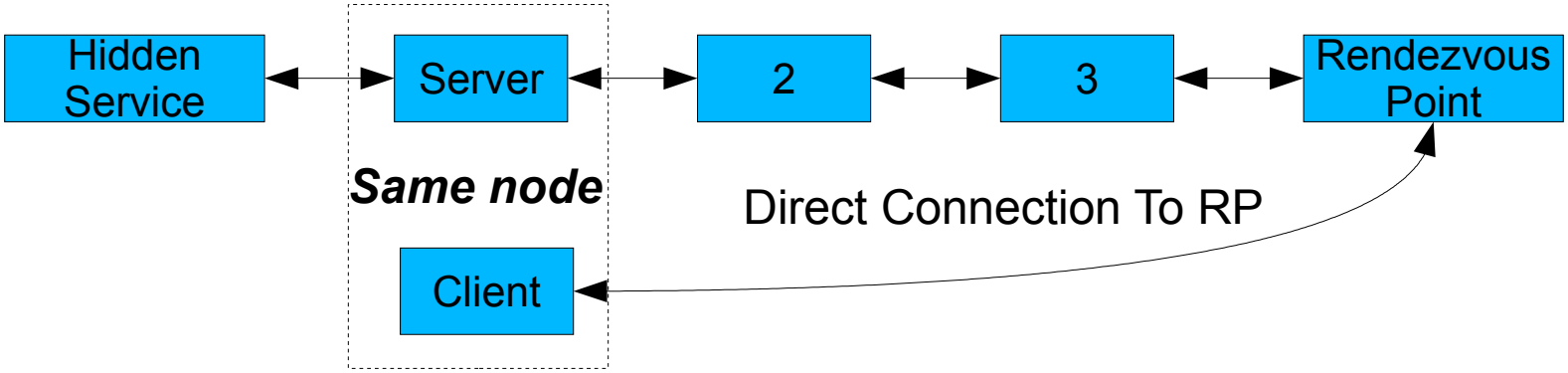
Statistical Attack

- If identified by Timing Attack as part of a specific circuit:
  - More likely to be contacted by originator than by any other node in circuit
  - One of three positions, 33% chance of either, BUT
    - in first position all connections are from same IP address
    - in second and third position the connections are coming from random nodes
  - Meaning more than 1/3 of all connections are coming from the Hidden Server



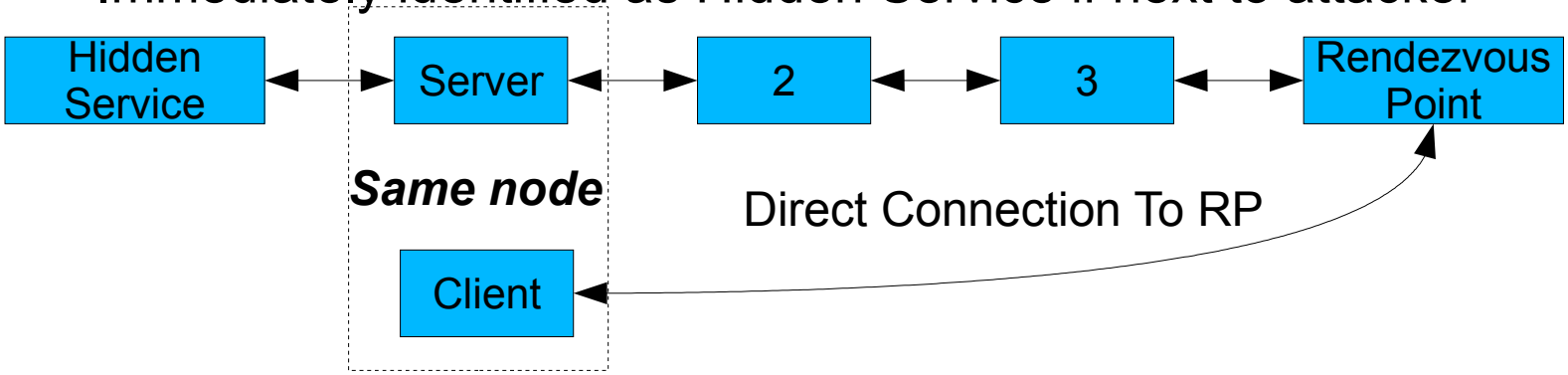
Two Node Attack

- In current design the Client selects RP
- If Client also controls RP, attacker will easily identify the last node in HS circuit towards RP
- After identified as part of circuit by Timing Attack, the attacker is able to confirm immediately when located as node 2 or 3



Client/Server Separation

- After confirming participation in circuit by Timing Attack
- Hidden Server as Tor Server
  - Listed. Identifiable as “one of the Tor nodes”
  - Requires access through public IP-address
  - Hides hidden service traffic in other Tor traffic
- Hidden Server as external Client
  - Not a part of the listing in the Directory Server
  - Can be used behind a NAT/firewall with ease
  - Immediately identified as Hidden Service if next to attacker



Tor Feature

- When selecting nodes towards Rendezvous Point, the last node was always an exit node
- Attacker using a middle-man node – never chosen as next to RP
- If identified in circuit – always as node 1 or 2
  - See Round Trip Time graph
- Confirmed when our Statistical Attack found >50% of connections to be directly from Hidden Service and none connecting to the Rendezvous Point
- This feature is now removed from Tor code base as a result of our analysis

Countermeasures

- Dummy Traffic
- Increased path length
- “Helper/Entry/Contact nodes”
  - Random
  - Friendly
  - Layered

Dummy Traffic

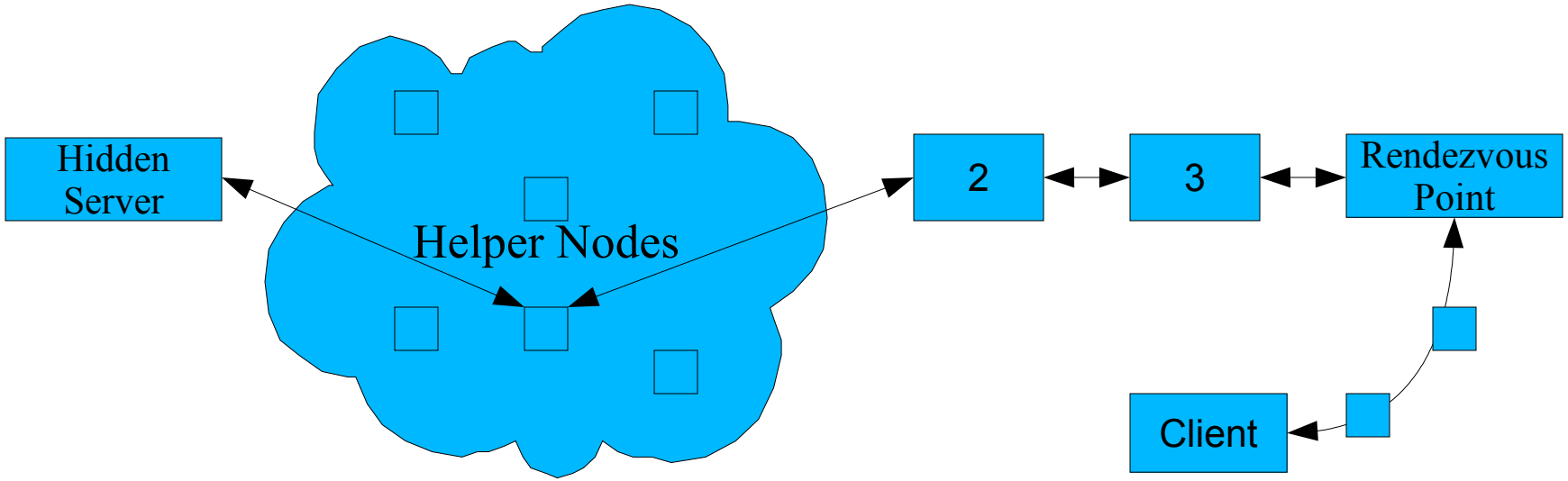
- Often suggested
- Expensive
- Does **not** resist active attacks in low latency systems
  - Easy to enforce a timing signature when inside a path

Increased path length

- Set default number of nodes between Hidden Server and Rendezvous Point to 4 (not 3)
- Will help on Two Node Attack (Rendezvous Point selection)
  - but not eliminate the problem
- Will not help on:
  - Timing and RTT attack
  - Predecessor Attack
  - Client/Server separation
- Increases latency and reduces available bandwidth in network

Helper/Entry/Contact Nodes

- All first connections from Hidden Service are done through the same set of nodes
- Attacker may be running “old trusted nodes”
- Will help against (but not eliminate!) the described attacks
- How to select nodes and determine size of set?
  - Random vs. Friendly vs. Layered Helper Nodes



Random Helper Nodes

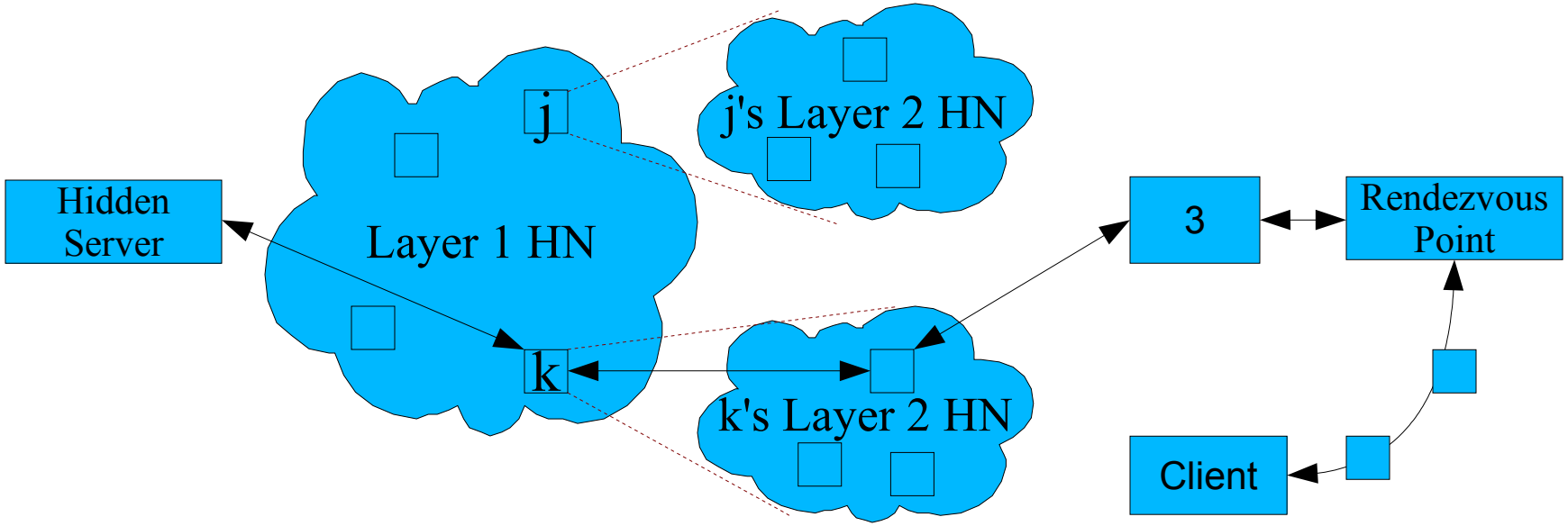
- Set up a selected set by random
- Must select from a commonly known set of nodes
- Semi-permanent set
  - same set every time
  - deleted from set only if inactive for a longer period
  - adding new members only when all existing are unavailable
- Simple and easy to set up
- Using our attacks we were able to identify Random Helper Nodes
  - Could identify Hidden Server if Helper Nodes themselves are attacked (root compromise, subpoena,...)
- If a compromised Helper Node is selected the Hidden Service gains no protection against our attacks

Friendly Helper Nodes

- Some nodes more trusted than others
- Problems if the nodes in the selected set are not operative
  - Be available – pick new node by random and will be back at original threat after DoS attack
- How to select? Do you know the operator? Why do you trust this node more than other nodes?
- Opens up for analysis of “who trusts who” and “what are their relation to each other”?
- Using our attacks we were able to identify Friendly Helper Nodes
  - Could identify Hidden Server if Helper Nodes themselves are attacked (root compromise, subpoena,...)

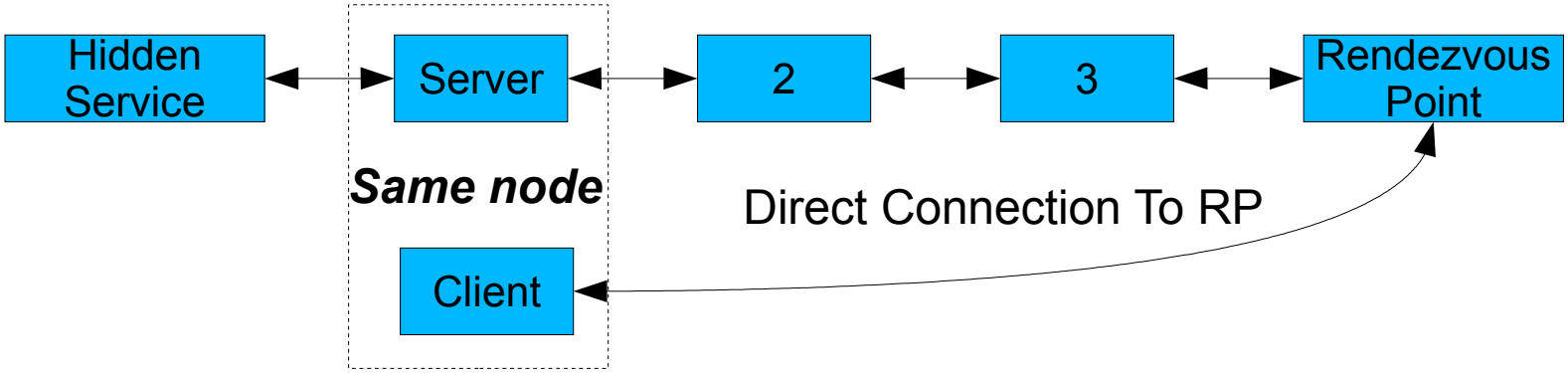
Layered Helper Nodes

- Pick also the second node from a separate set of nodes. Every first layer node has its own set of second layer nodes
- Harder to identify all first layer nodes
  - must be in all of the second layer groups
  - but being just one might identify one point of attack
- More problems with regards to uptime control



Experimental Setup

- Remembering previously described setup
  - Using only one active node
  - Node was part of Tor network as a middle-man server
  - The Node's Client part contacted Rendezvous Point directly
- Running attacks twice against two different Hidden Servers
  - not public hidden servers
  - located on two different continents



Experimental Results

	Sample time	*Circuits completed*	*Timing attack match*	*Client/Server attack*	*Largest IP group*	*2nd Largest IP group*
Server #1 - 1	7.8h	676	37	15 min	46 %	5 %
Server #1 - 2	6.8h	432	26	3 min	54 %	7 %
Server #2 - 1	4.9h	447	31	28 min	71 %	3 %
Server #2 - 2	10.6h	990	56	3 min	54 %	7 %

- More than 1 out of 20 circuits went through our node again
- Statistical Attack reveals that over 50% were from the same IP address (which was the Hidden Service)
- Client/Server Separation finds the IP address in a matter of minutes
- Two Node Attack was not tested due to lack of time and resources

Experimental Results (2)

	Circuits completed	*Timing attack match*	*Largest common IP*	*2nd Largest Common IP*	*3rd Largest Common IP*
Test 1	292	8	7	1	0
Test 2	106	6	5	1	0
Test 3	296	13	12	1	0
Test 4	292	10	4	3	3

- Using the attacks we were also able to locate the Helper Nodes
- In the first three experiments only two out of the three specified Helper Nodes were used (during our sampling)
- REMEMBER: This only reveals the location of the Helper Nodes and does NOT locate the Hidden Server

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

- Current Tor system code freely available (3-clause BSD license)
- Hidden Wiki of Hidden Services available at http://6sxoyfb3h2nvok2d.onion/
- Tor comes with a detailed specification – Dresden implemented an independent compatible Tor client in Java
- Design paper, system spec, code, see the list of current nodes, etc. at

http://tor.freehaven.net/

Conclusions and Future Work

- Tor and its Hidden Servers have wonderful existing records and future prospects for protecting online communication
- We have shown Hidden Servers deployed over the Tor network to be trivially vulnerable to rapid and complete exposure using only a single hostile Tor node
- We have also performed the first experimental demo of a predecessor attack on the live Tor network
- We proposed countermeasures to attacks we demonstrated
- Working with the Tor developers, have seen these implemented along with feature improvements we identified
- In the future, we will be presenting more attacks and network improvements
  - Attacks using multiple nodes, attacks on clients, authentication mechanisms, finding unknown hidden services,...

Thanks!

Questions?