Basic Course on Onion Routing

Paul Syverson

U.S. Naval Research Laboratory

paul.syverson@nrl.navy.mil

SAC Summer School

Mount Allison University Aug 10, 2015

Course Outline

- Lecture 1: Basics and Formalization
  - Usage examples, basic notions of traffic-secure communications, mixes and onion routers
  - Onion routing design basics: circuit construction protocols, network discovery
  - Formalization and analysis, possibilistic and probabilistic definitions of anonymity
- Lecture 2: Security for the real world
  - Simple demo of obtaining/using Tor
  - Security of obtain/using Tor
  - Adding network link awareness
  - Importance of modeling users
  - Importance of realistic and practical
    - Adversary models
    - Security definitions

"Our motivation here is not to provide anonymous communication, but to separate identification from routing."

- "Proxies for anonymous routing". Reed, Syverson, and Goldschlag. ACSAC 1996

A Motivational Use Case Example

- Navy Petty Officer Alice is on temporary duty out of the U.S.

https://atlevel1.dtic.mil/at/atl1/CONUS/AT/LevelB/at_4/index.html

Level I Training System

ANTITERRORISM

AT FUNDAMENTALS | Surveillance Detection | Government Facility | Active Shooter | Residential | Off-Duty Activities | Air Travel | Ground Travel | Hotel | Hostage Survival | CBRNE

Don't be a Target

Items that display your DOD affiliation may also help identify you as a potential target.

Not all threats are predictable or can be recognized in advance. As a result, you should concentrate on not being an easy target for attack.

Reduce your exposure by being anonymous and blending in with your surroundings.

- Do not wear clothing or carry items that might attract criminal attention
- Remain low key and do not draw attention to yourself
- Avoid places of high criminal activity

In addition to blending in, try to reduce your vulnerability and exposure:

- Select places with security measures appropriate for the local threat
- Be unpredictable and vary your routes and times of travel
- Travel with a friend or in a small group
- Use automobiles and residences with adequate security features

You can greatly increase your personal protection posture by remaining anonymous and reducing your exposure.

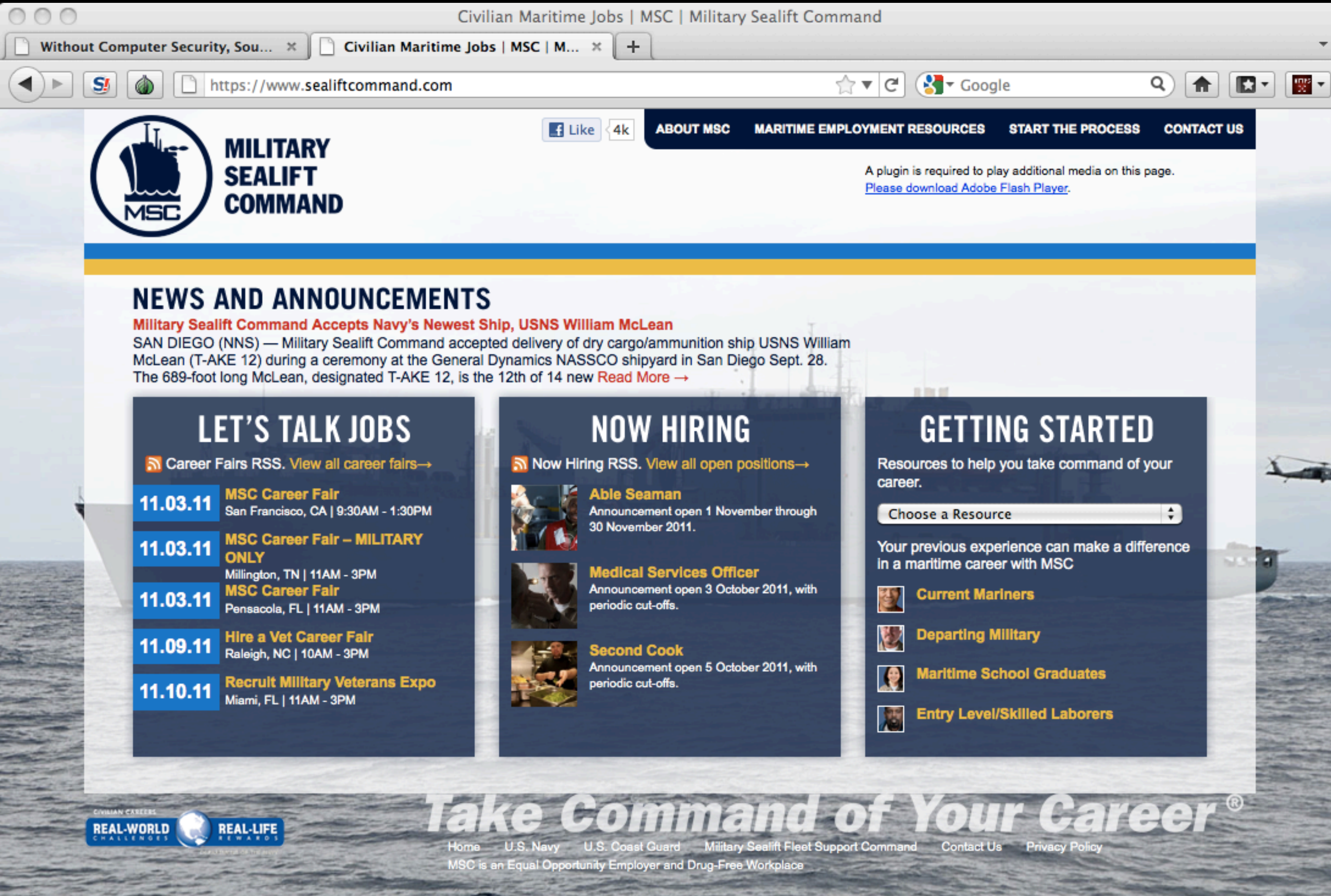
Select Next to continue.

Anticipate | Be Vigilant | Don't be a Target | Respond & Report

Motivational Use Case Example

- Safe back in her hotel, PO Alice wants to read and/or post to sealiftcommand.com

1. The site is blocked where she is deployed
2. The Internet is monitored where she is deployed

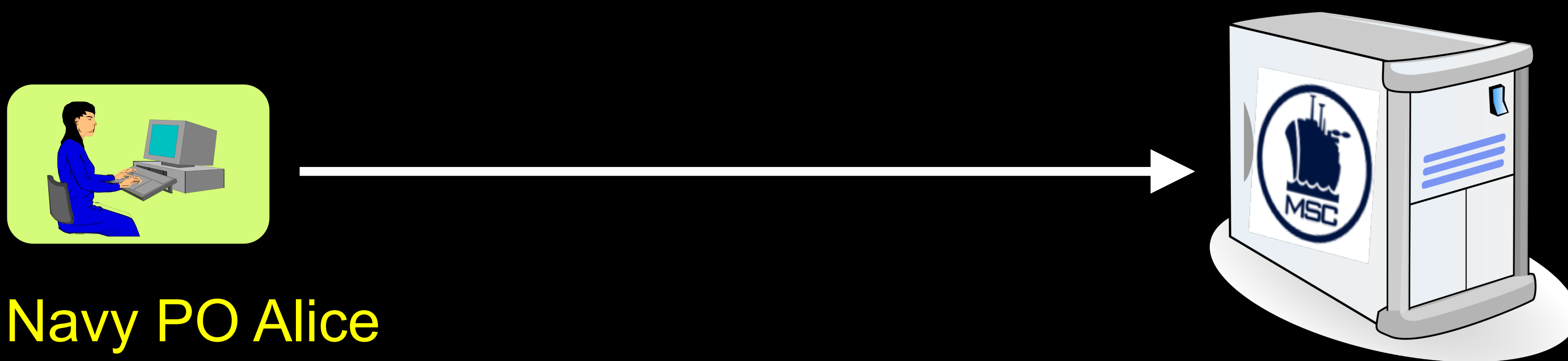

Civilian Maritime Jobs | MSC | Military Sealift Command
Without Computer Security, Sou...
Civilian Maritime Jobs | MSC | M...
https://www.sealiftcommand.com
Google
MILITARY SEALIFT COMMAND
MSC
Like 4k
ABOUT MSC
MARITIME EMPLOYMENT RESOURCES
START THE PROCESS
CONTACT US
A plugin is required to play additional media on this page.
Please download Adobe Flash Player.
NEWS AND ANNOUNCEMENTS
Military Sealift Command Accepts Navy's Newest Ship, USNS William McLean
SAN DIEGO (NNS) — Military Sealift Command accepted delivery of dry cargo/ammunition ship USNS William McLean (T-AKE 12) during a ceremony at the General Dynamics NASSCO shipyard in San Diego Sept. 28. The 689-foot long McLean, designated T-AKE 12, is the 12th of 14 new Read More →
LET'S TALK JOBS
Career Fairs RSS. View all career fairs→
11.03.11 MSC Career Fair
San Francisco, CA | 9:30AM - 1:30PM
11.03.11 MSC Career Fair – MILITARY ONLY
Millington, TN | 11AM - 3PM
11.03.11 MSC Career Fair
Pensacola, FL | 11AM - 3PM
11.09.11 Hire a Vet Career Fair
Raleigh, NC | 10AM - 3PM
11.10.11 Recruit Military Veterans Expo
Miami, FL | 11AM - 3PM
NOW HIRING
Now Hiring RSS. View all open positions→
Able Seaman
Announcement open 1 November through 30 November 2011.
Medical Services Officer
Announcement open 3 October 2011, with periodic cut-offs.
Second Cook
Announcement open 5 October 2011, with periodic cut-offs.
GETTING STARTED
Resources to help you take command of your career.
Choose a Resource
Your previous experience can make a difference in a maritime career with MSC
Current Mariners
Departing Military
Maritime School Graduates
Entry Level/Skilled Laborers
REAL-WORLD CHALLENGES REAL-LIFE REWARDS
Take Command of Your Career®
Home U.S. Navy U.S. Coast Guard Military Sealift Fleet Support Command Contact Us Privacy Policy
MSC is an Equal Opportunity Employer and Drug-Free Workplace

Use Case Example

- Safe back in her hotel, PO Alice wants to read and/or post to sealiftcommand.com

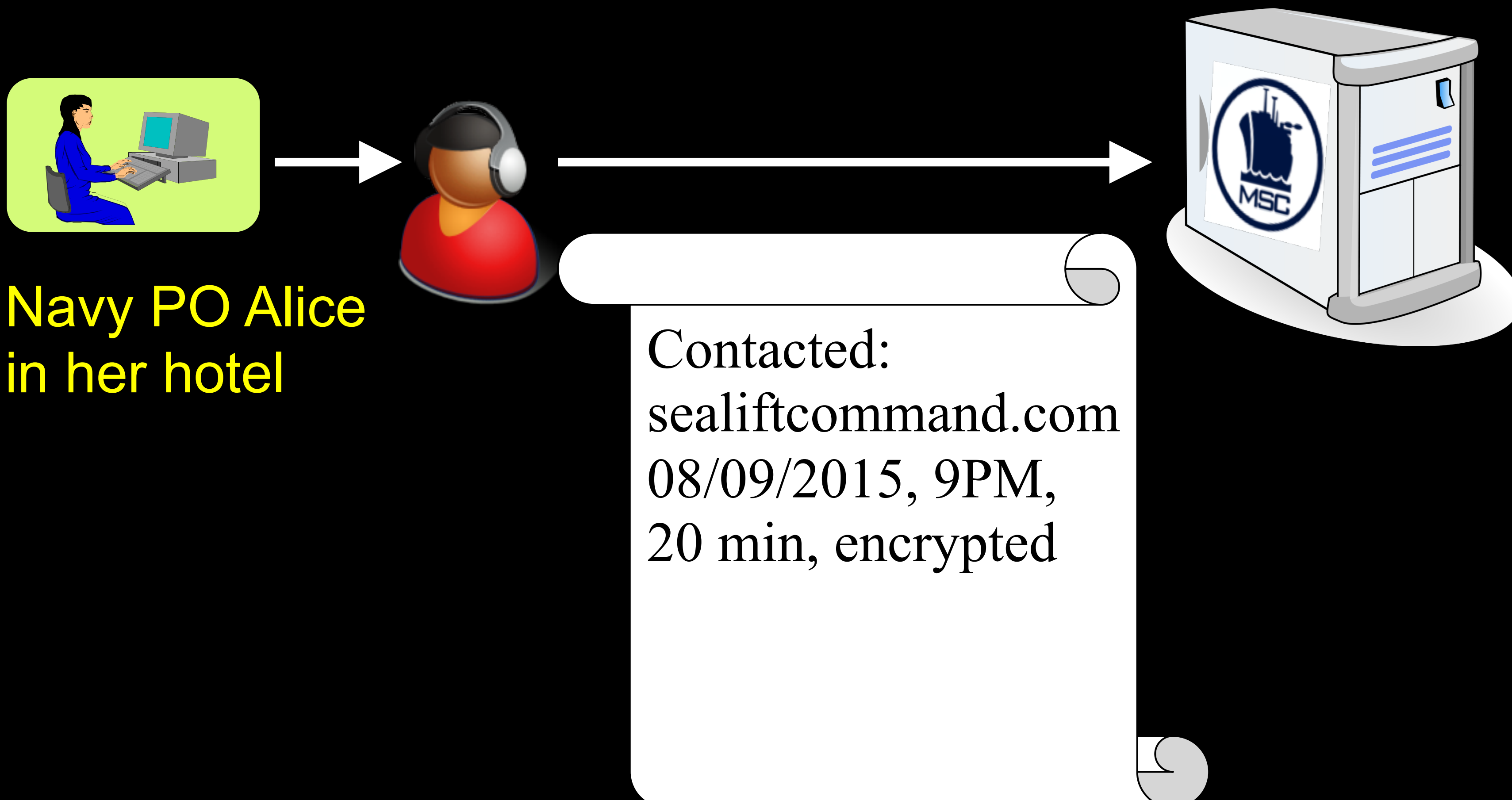
1. The site is blocked where she is deployed
2. The Internet is monitored where she is deployed

Connecting when overseas

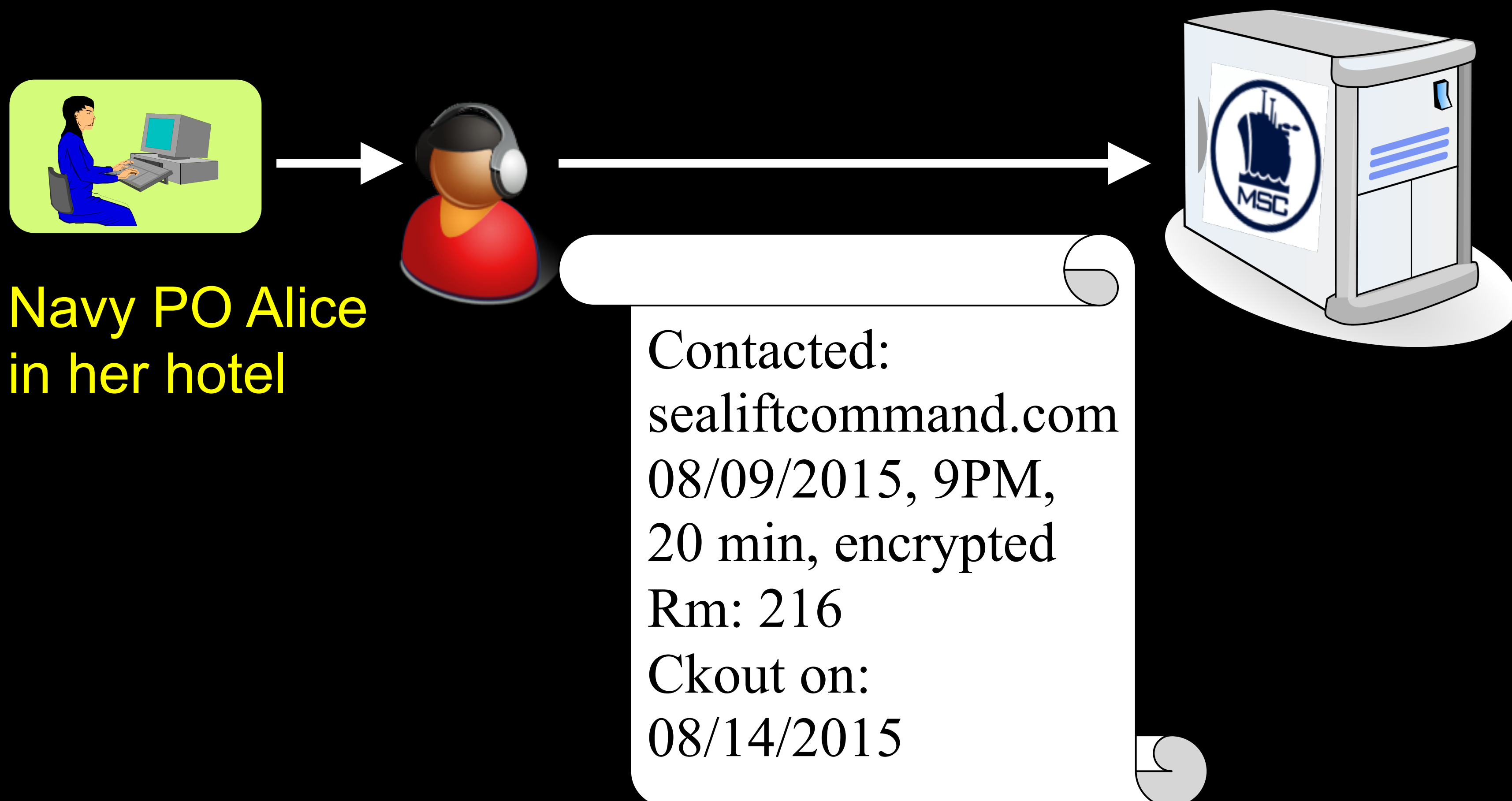


Navy PO Alice
in her hotel

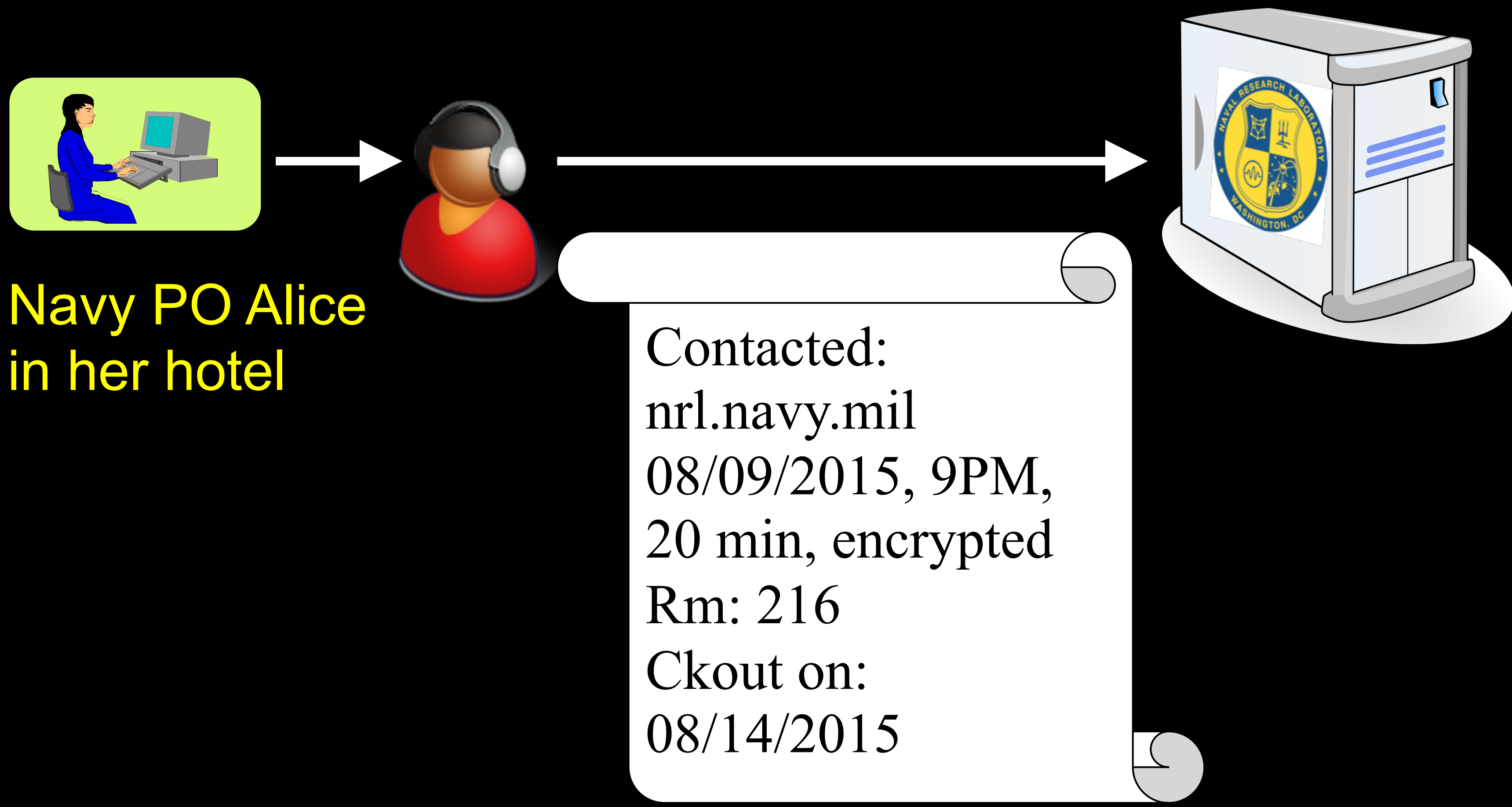
Connecting when overseas



Connecting when overseas



Security of operations concern as well as personnel security concern



Some more government uses

- Open source intelligence gathering
- Sensitive communications with untrusted/ untrusting parties
- Encouraging open communications with citizens
- Reduce risk/liability from data breaches, DNS hijacks,…
- Location protected servers for defense in depth
- Protecting the public infrastructure
  - Interacting with network sensors

Officer Alice

- Setting up a sting operation:
  - as a collaborator
  - as a service provider
- Monitoring criminal activity online
- Encouraging anonymous tips

Corporation Alice

- Checking out the competition
- Exploration of collaborations and partnerships
- Patent searches
- Protecting her customers

Researcher/Reporter/Rights Worker Alice

- Gathering information while protecting sources
- Accessing information that is locally censored or monitored
- Reporting information that is locally censored or monitored

Ordinary citizen Alice

- Protecting her behavior from:
- Cyberstalking abusive ex-spouse
- Behavior tracking, DNS shenanigans by her ISP
- Misunderstanding from her employer when she investigates disease info for an ailing friend
- Harassment for blogging her views
- Spear phishers watching her log into her bank

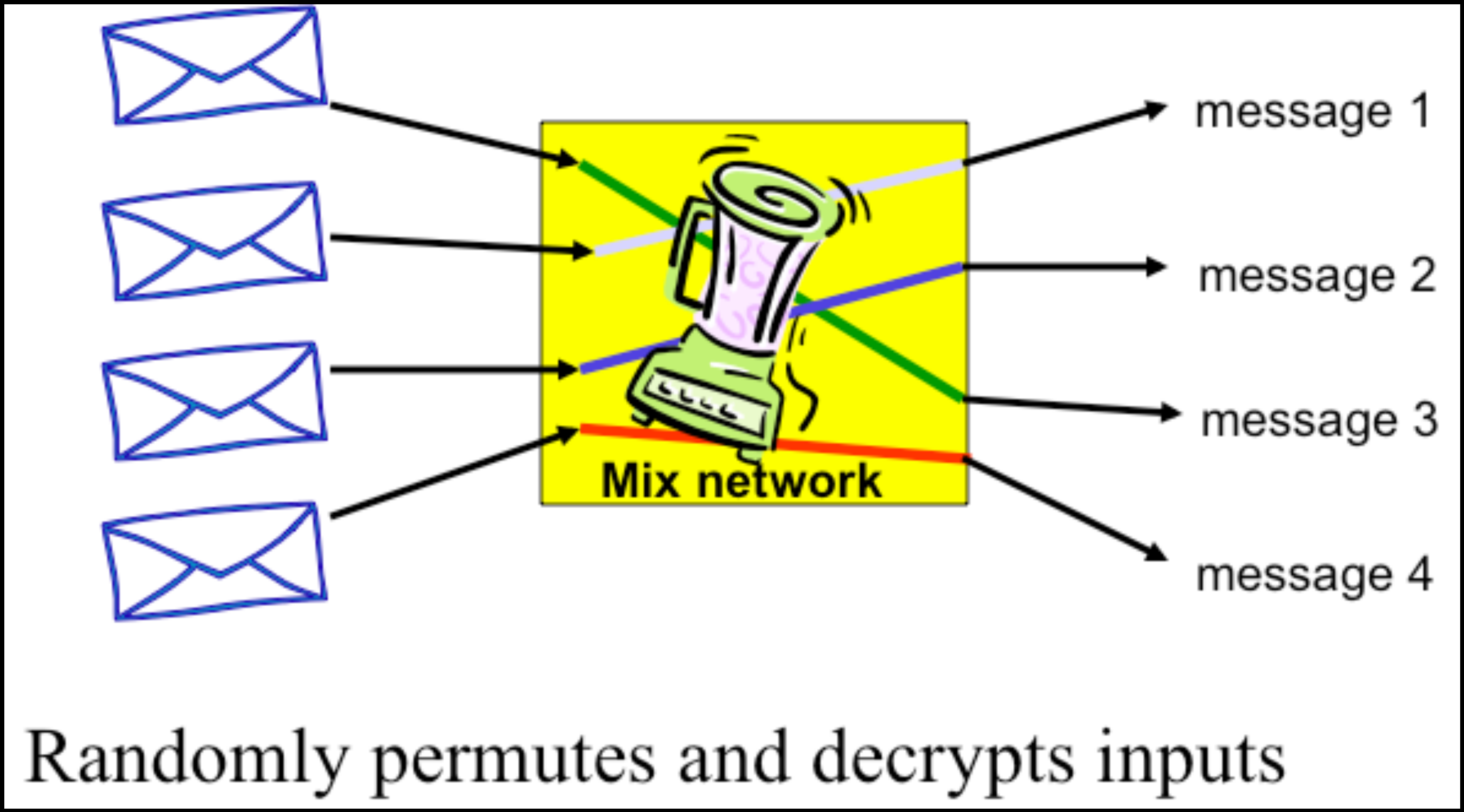
Recent U.N. Human Rights Commission Report Conclusion

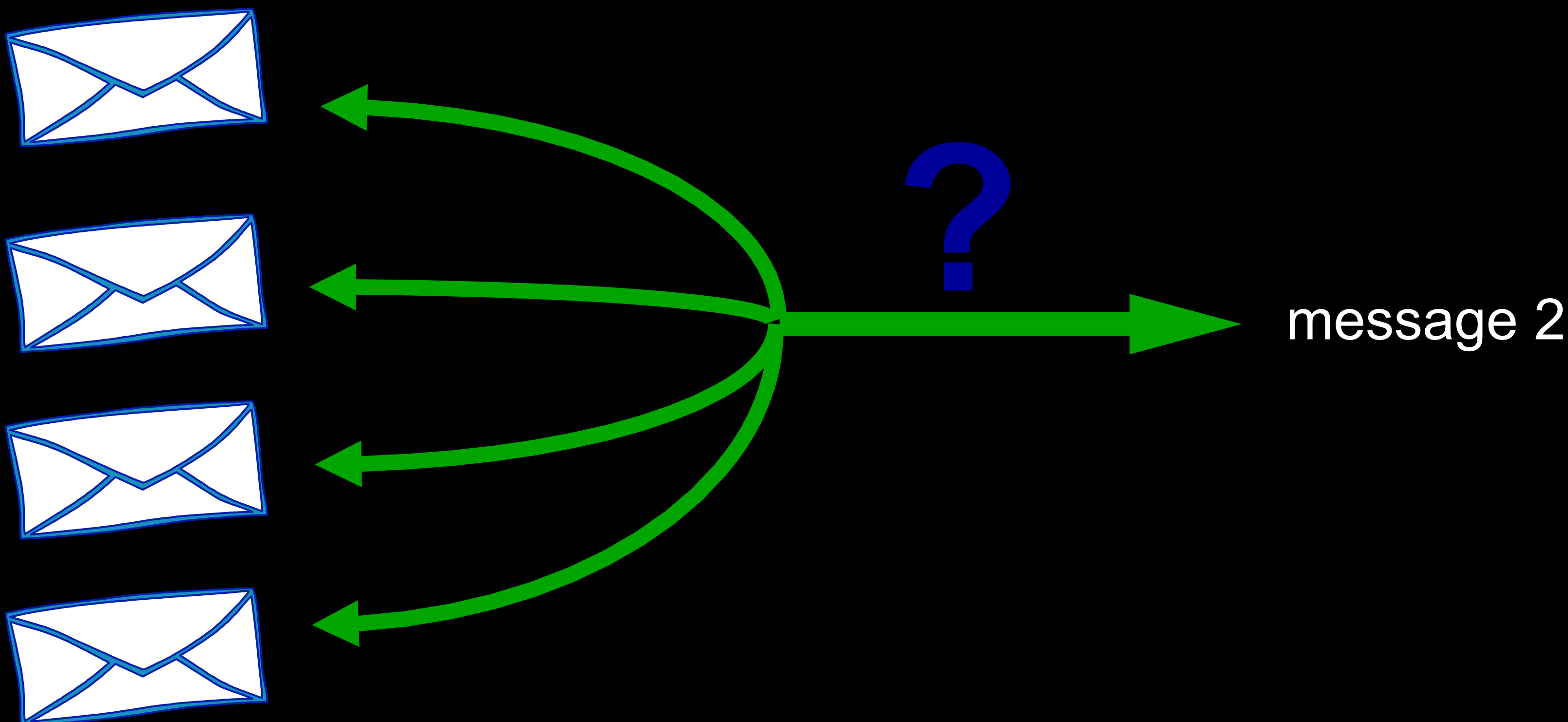
"States should promote strong encryption and anonymity."

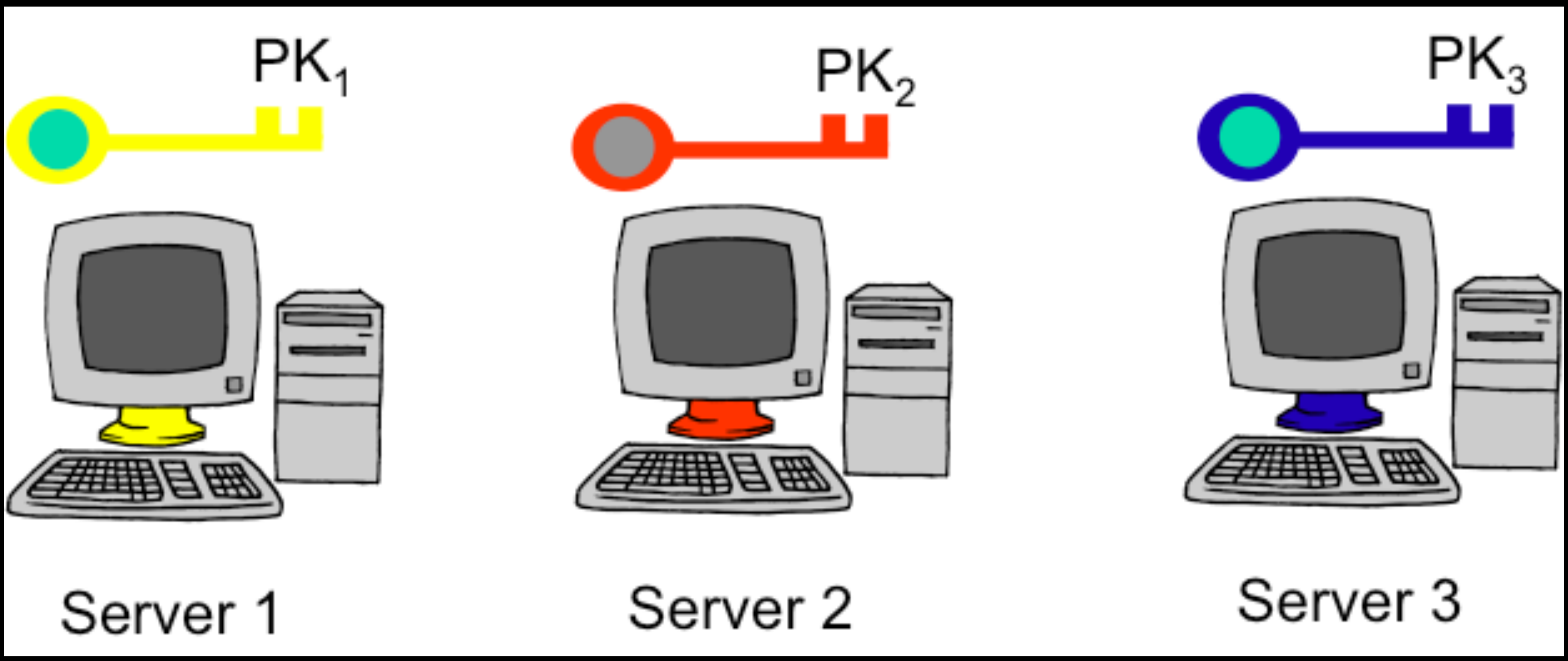
- Also specifically mentions the importance of protecting IP address, and Tor as an important technology protecting freedom to hold and express opinions.

First Anonymous Comms Design (Chaum Mix)

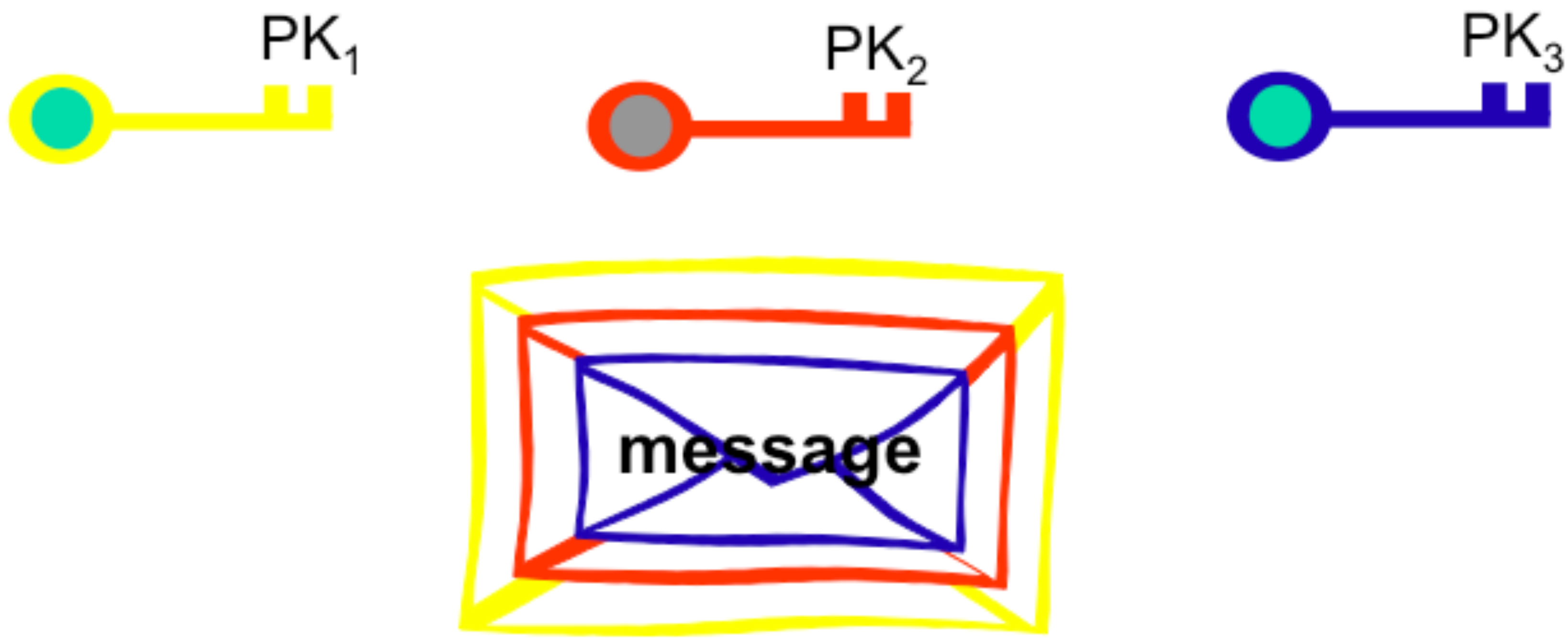
- Untraceable Electronic Mail, Return addresses, and Digital Pseudonyms – David Chaum, CACM 1981



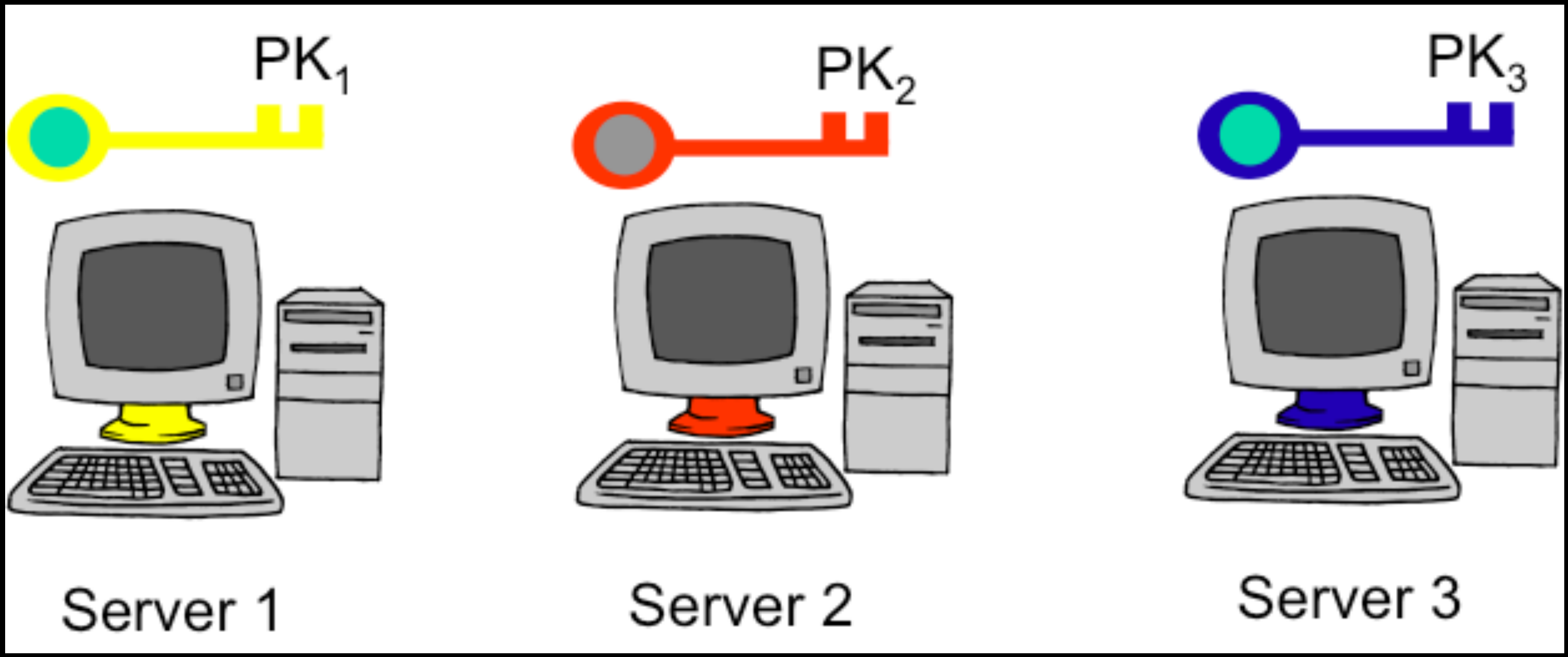

?
message 2



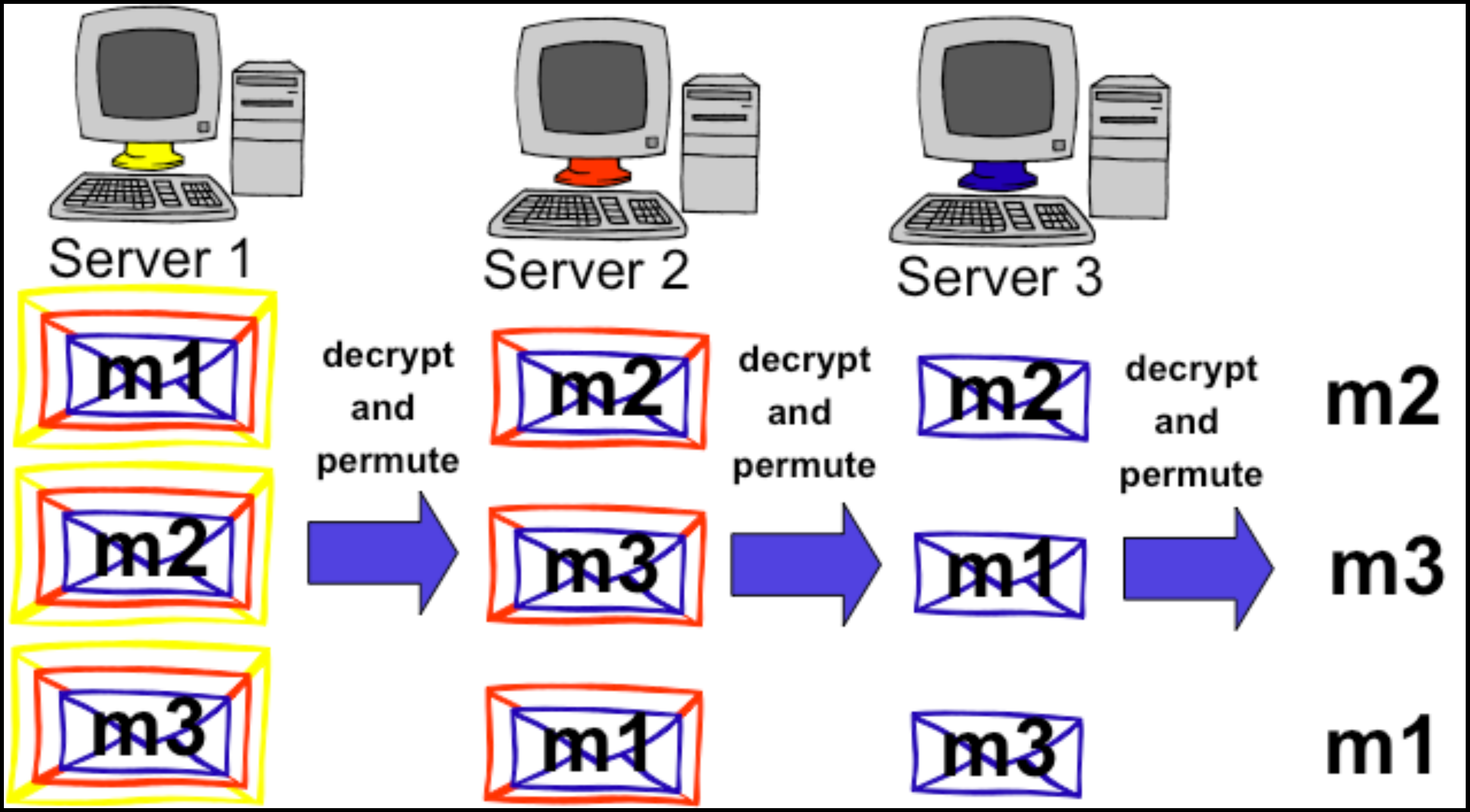
PK1
PK2
PK3
Server 1
Server 2
Server 3



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



PK_1
PK_2
PK_3
Server 1
Server 2
Server 3


Server 1
Server 2
Server 3
m1
m2
m3
decrypt and permute
m2
m3
m1
decrypt and permute
m2
m1
m3
decrypt and permute
m2
m3
m1

Mixes

- Invented by Chaum 1981 (not counting ancient Athens)
- As long as one mix is honest, network hides anonymity up to capacity of the mix
- Sort of
  - Flooding
  - Trickling
- Many variants
  - Timed
  - Pool
  - ...

Athenian Jury Ballots (4^{th} C BCE)

Also had cool randomized jury selection mechanism (kleroterion)

Mixes

- Invented by Chaum 1981 (not counting ancient Athens)
- As long as one mix is honest, network hides anonymity up to capacity of the mix
- Sort of
  - Flooding
  - Trickling
- Many variants
  - Timed
  - Pool
  - ...

Chaum '81: First Adversary Model

1. "No one can determine anything about the correspondences between a set of sealed items and the corresponding set of unsealed items, or create forgeries without the appropriate random string or private key."

Crypto is black-box secure (Dolev-Yao Model)

2. "Anyone may learn the origin, destination(s), and representation of all messages in the underlying telecommunication system and anyone may inject, remove, or modify messages."

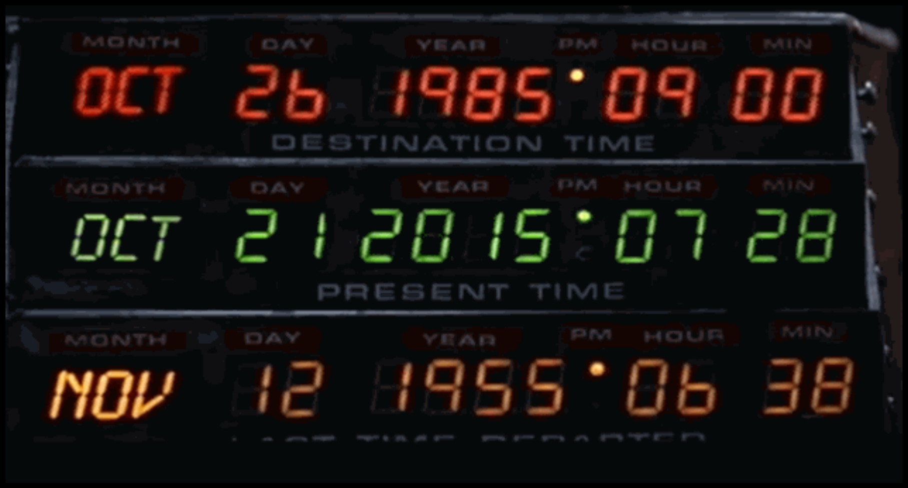
Active and Global Adversary

David Chaum: Way Ahead of His Time

In 1981

- SMTP was one year in the future
- IRC was seven years in the future
- The Web (Mosaic) was twelve years in the future
- (Dolev-Yao Model was two years in the future)

- Explicitly recognized and countered replay attacks



1993: The Web takes off

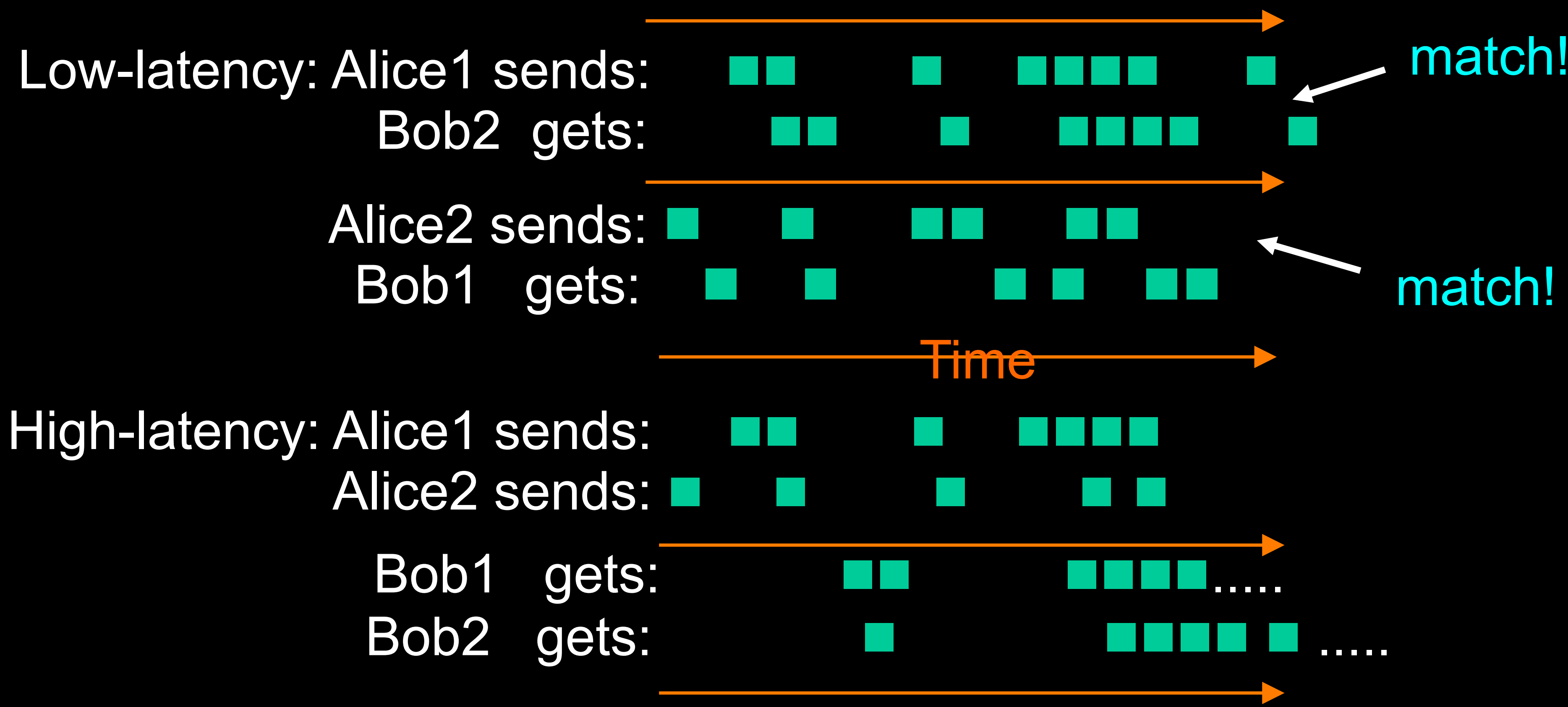
A useful adversary model must fit usage environment

- Application protocols must function
- Usability is a security property

Both interactivity and low-latency break Chaum's assumptions

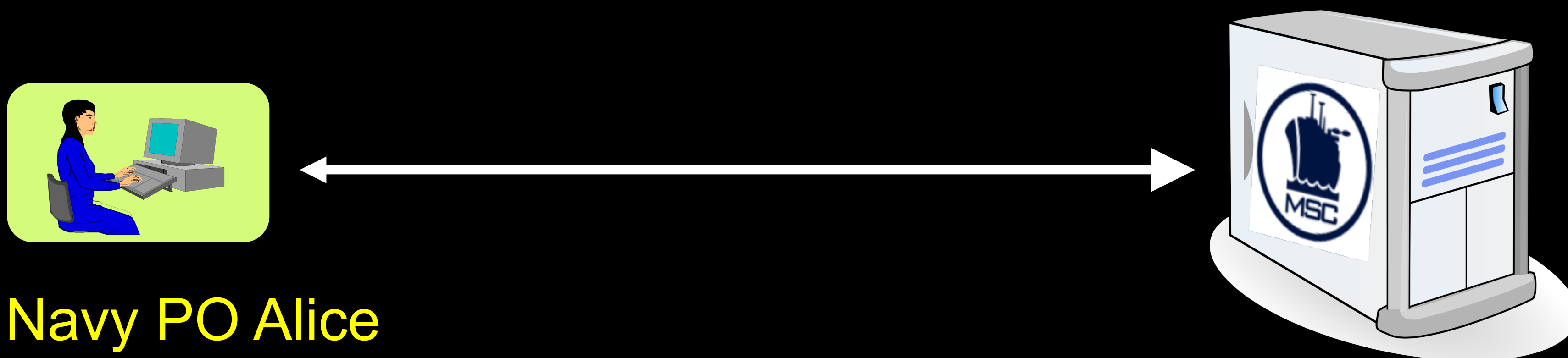
- Web comms mostly based on bidirection TCP connections
- Web comms are low latency

Low-latency systems are vulnerable to correlation by a global adversary



These attacks work in practice. The obvious defenses are expensive (like high-latency), useless, or both.

Connecting when overseas



Navy PO Alice
in her hotel

Practical Secure Solutions

Solution must

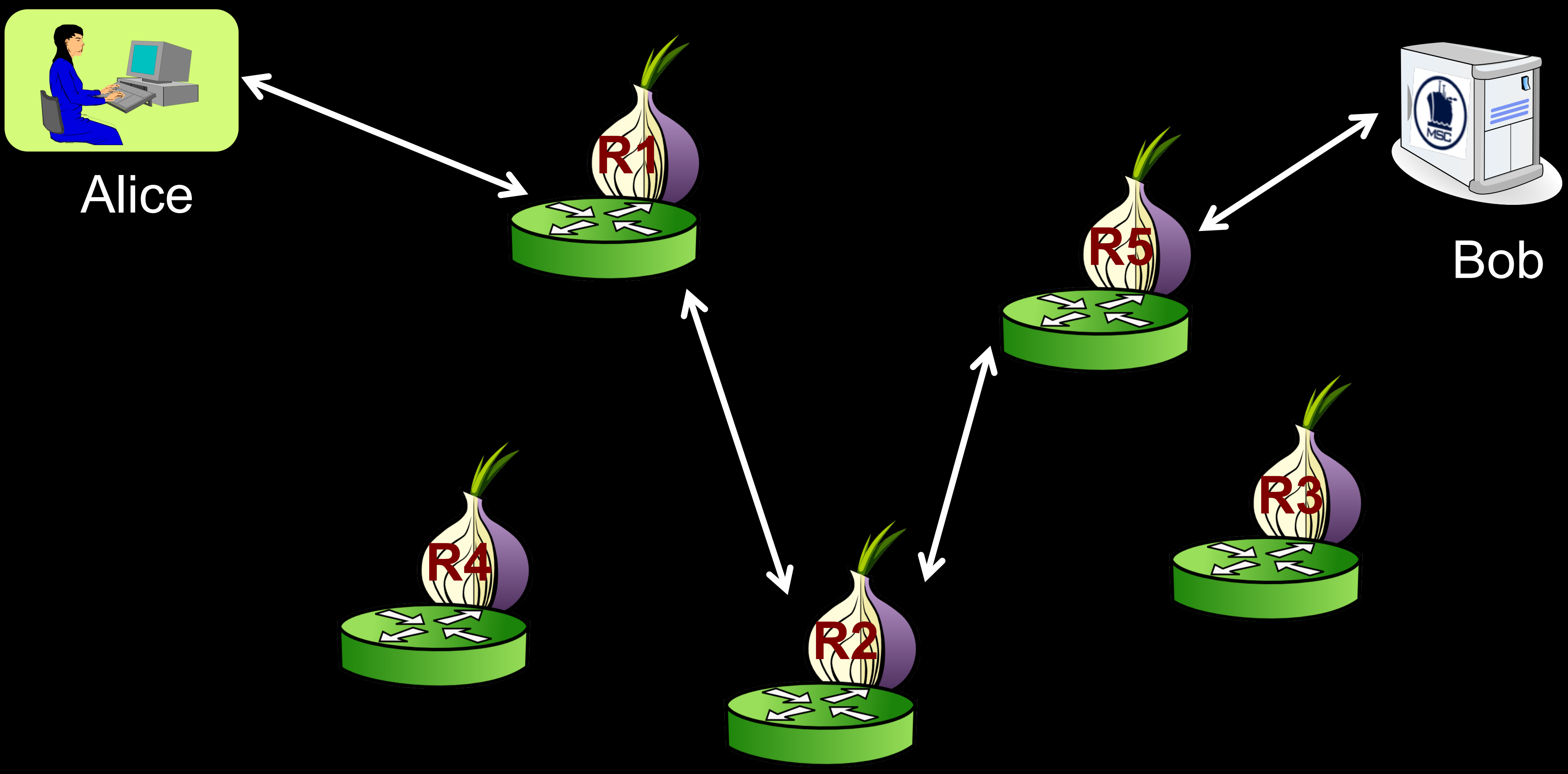
- Carry traffic bidirectionally with low latency
- But that is broken against our adversary model?!

Practical Secure Solutions

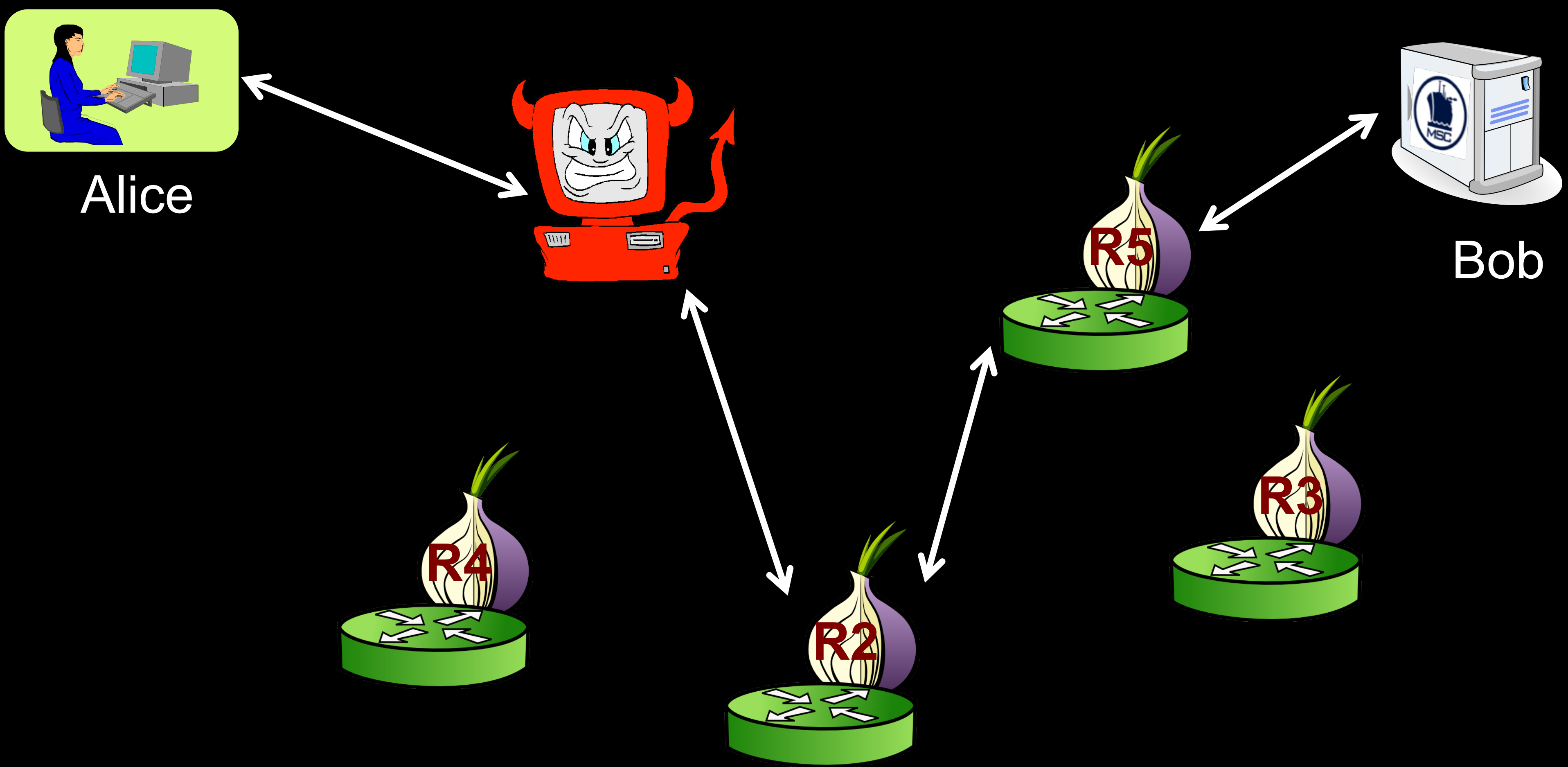
Solution must

- Carry traffic bidirectionally with low latency
- But that is broken against our adversary model?!
- So need a design making global adversary unlikely
  - very large and diversely managed network
- Carry traffic for a diverse user population
  - not just Navy or U.S. govt.
  - cannot have single point of failure/trust for any type of user
    - Diversely managed infrastructure
      - Open source

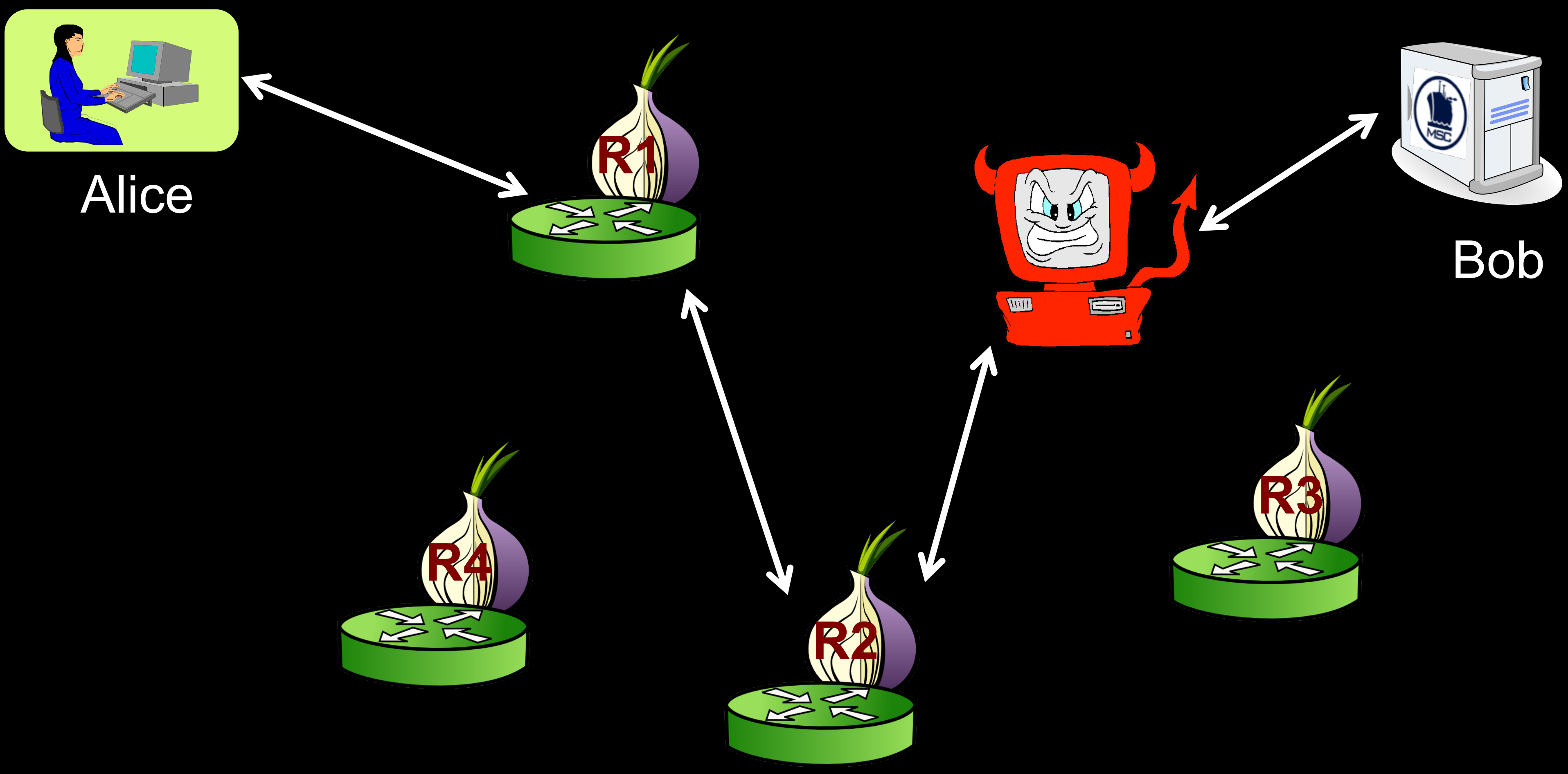
Network of diversely managed relays so that no single one can betray Alice.



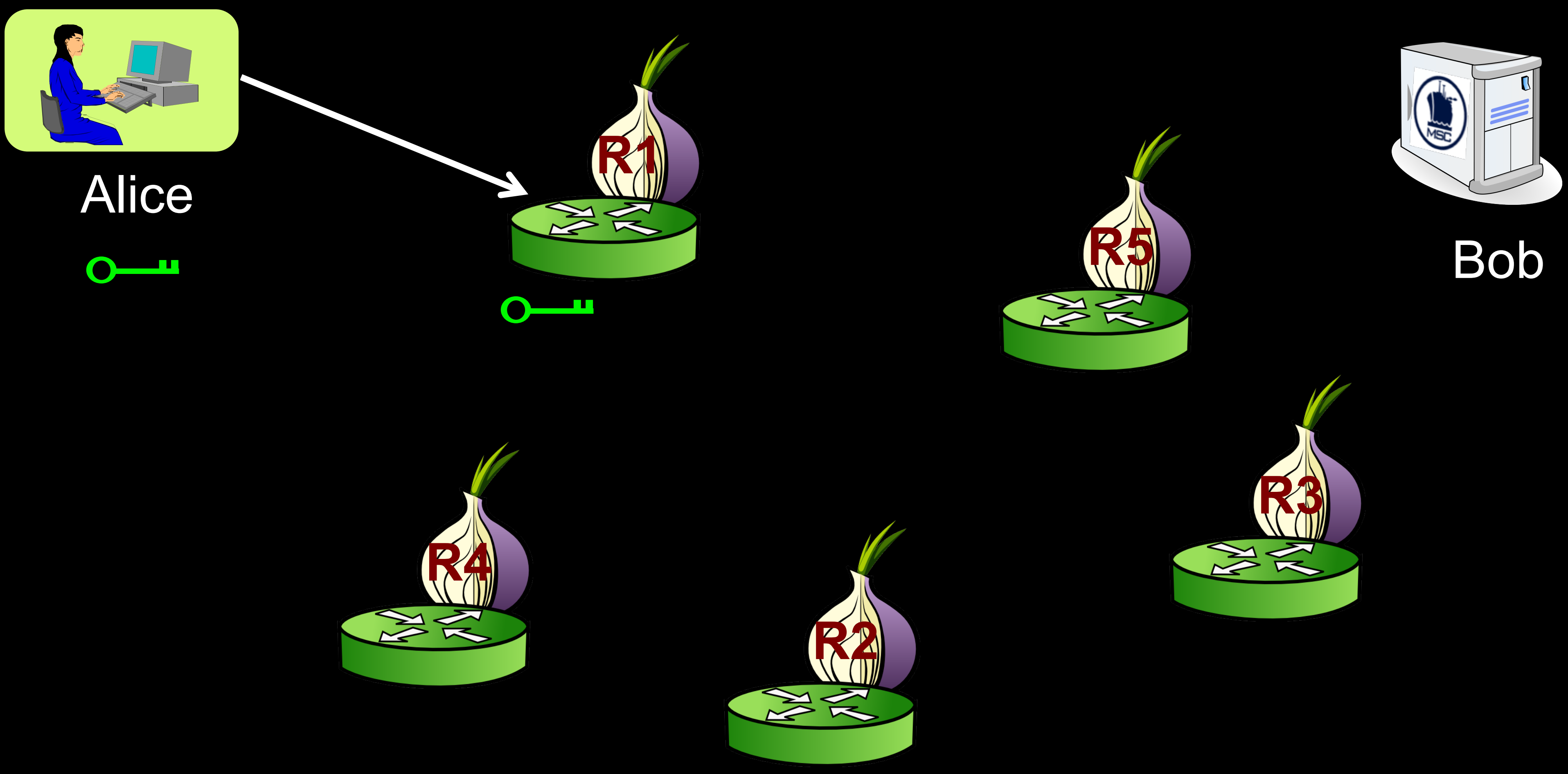
A corrupt first hop can tell that Alice is talking, but not to whom.



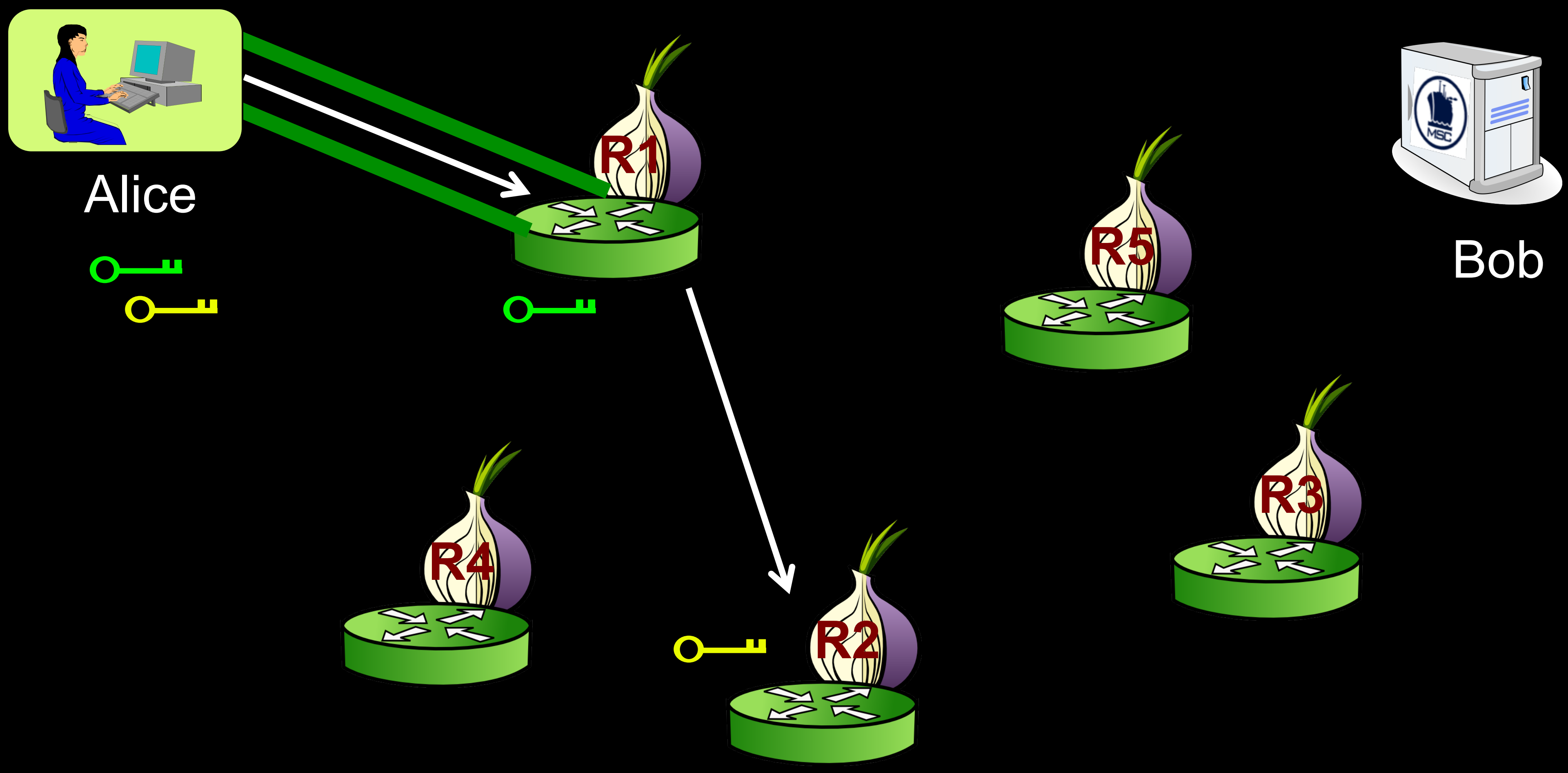
A corrupt last hop can tell someone is talking to Bob, but not who.



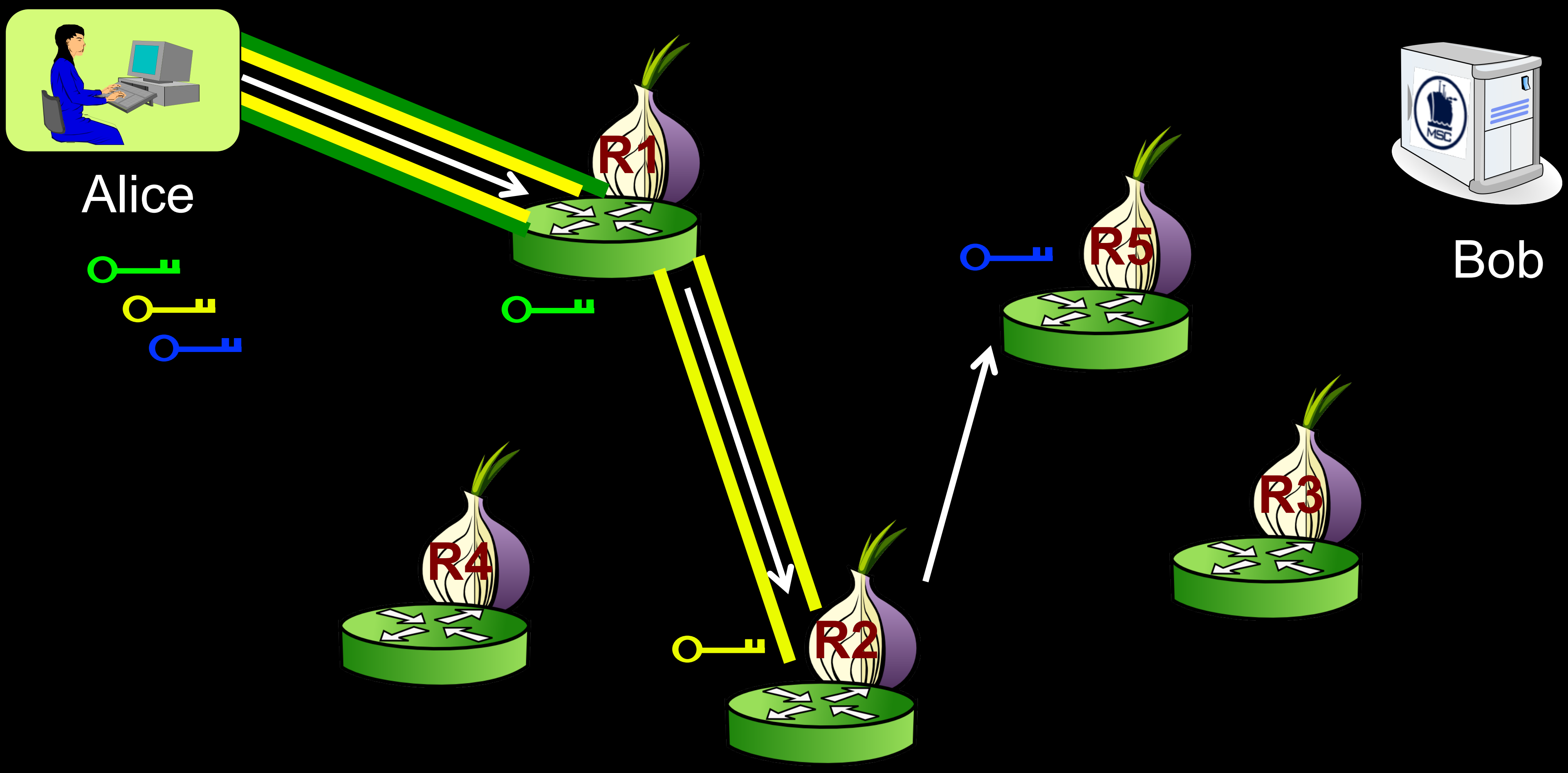
Onion Routing: Circuit construction



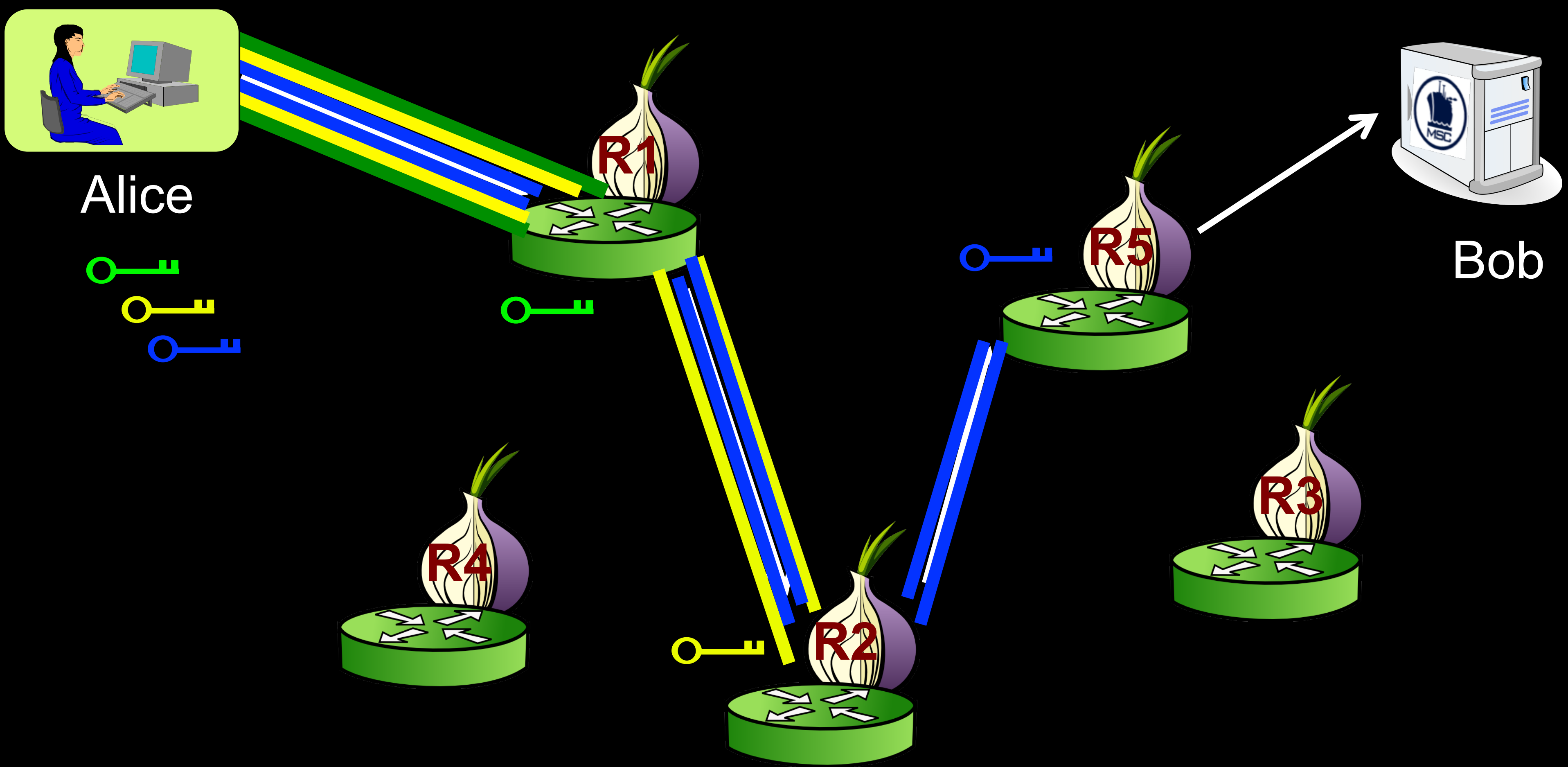
Onion Routing: Circuit construction



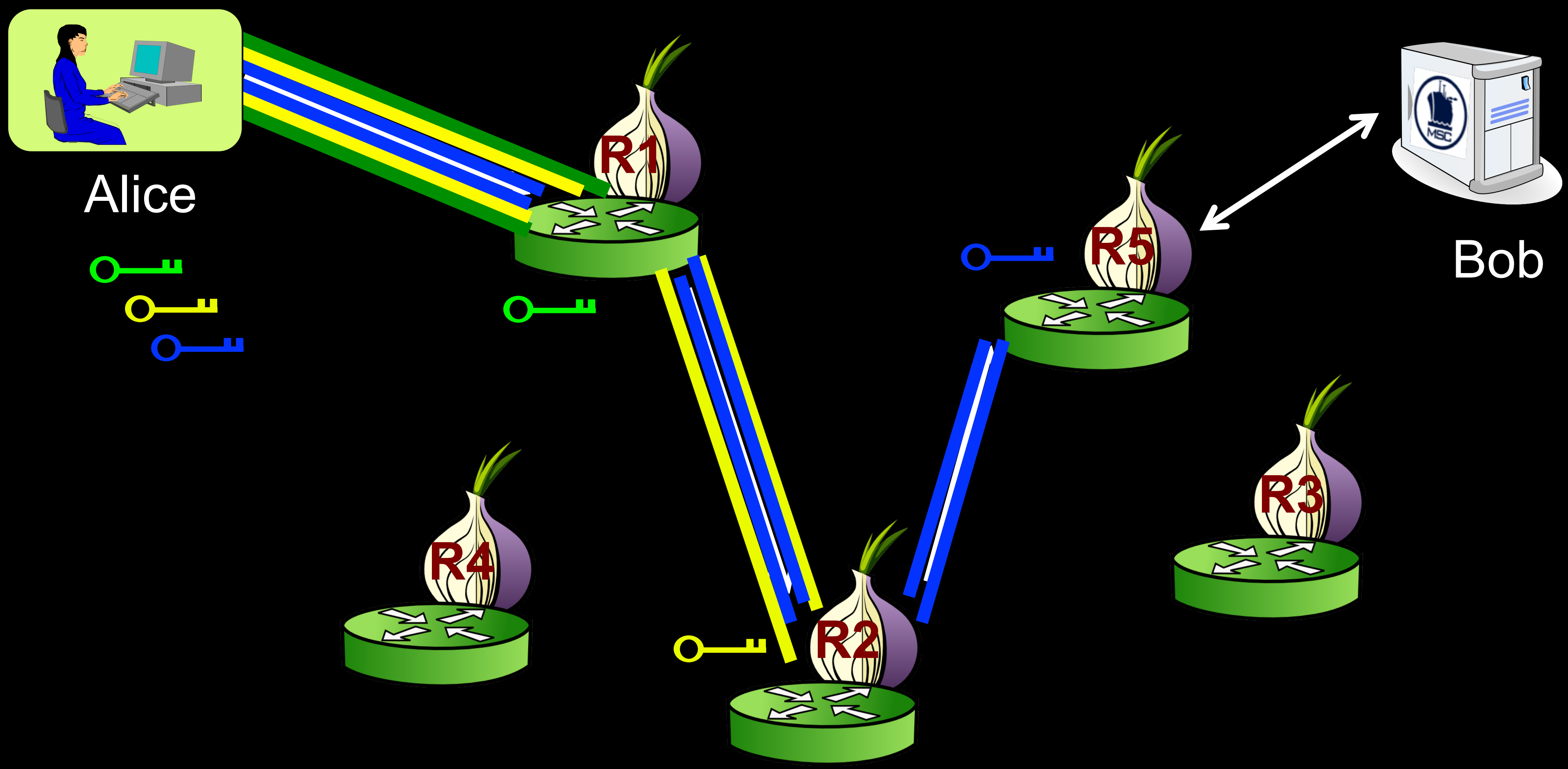
Onion Routing: Circuit construction



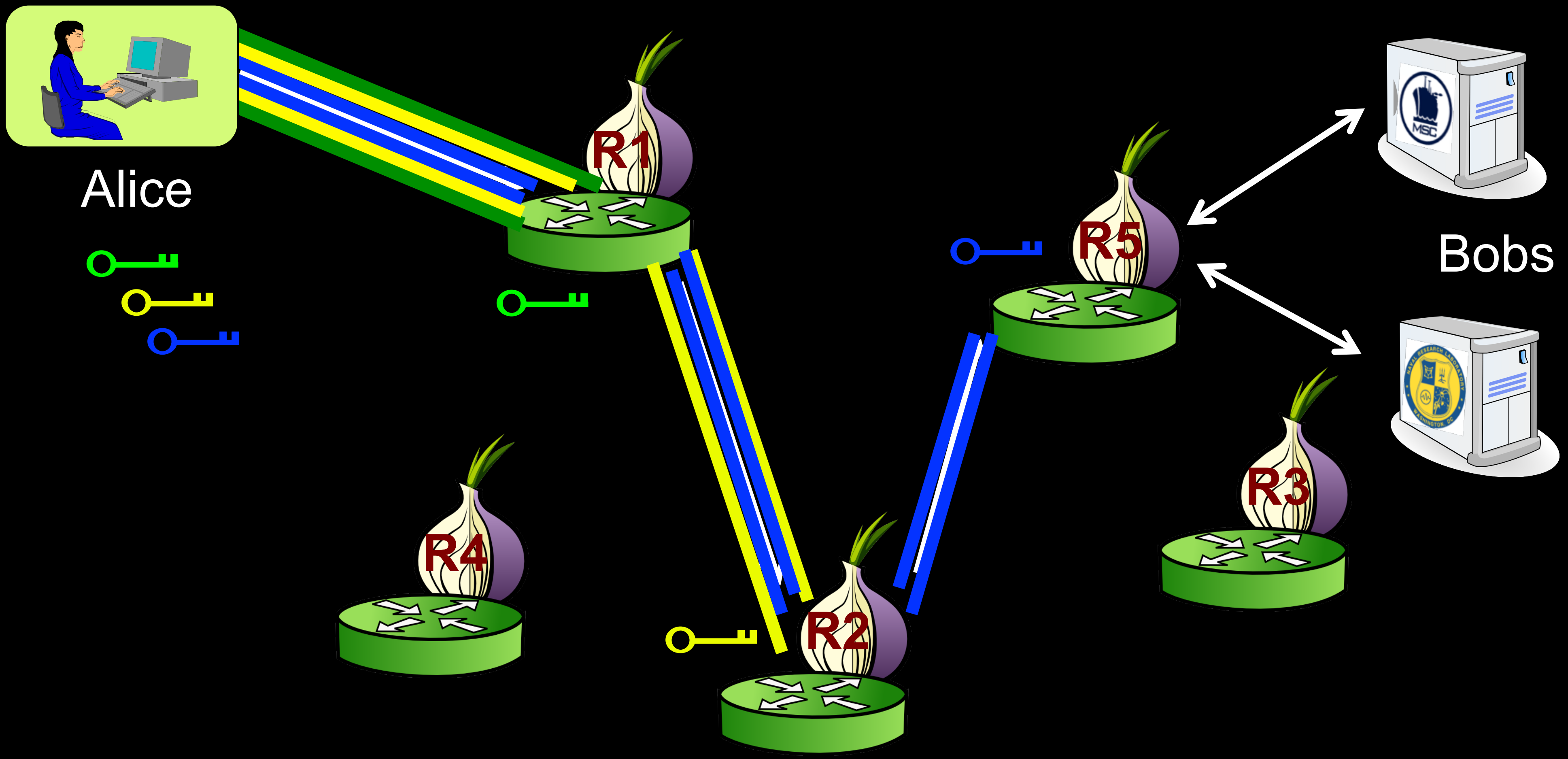
Onion Routing: Connection creation



Onion Routing: Data Exchange



Onion Routing: Data Exchange



That's onion routing in a nutshell

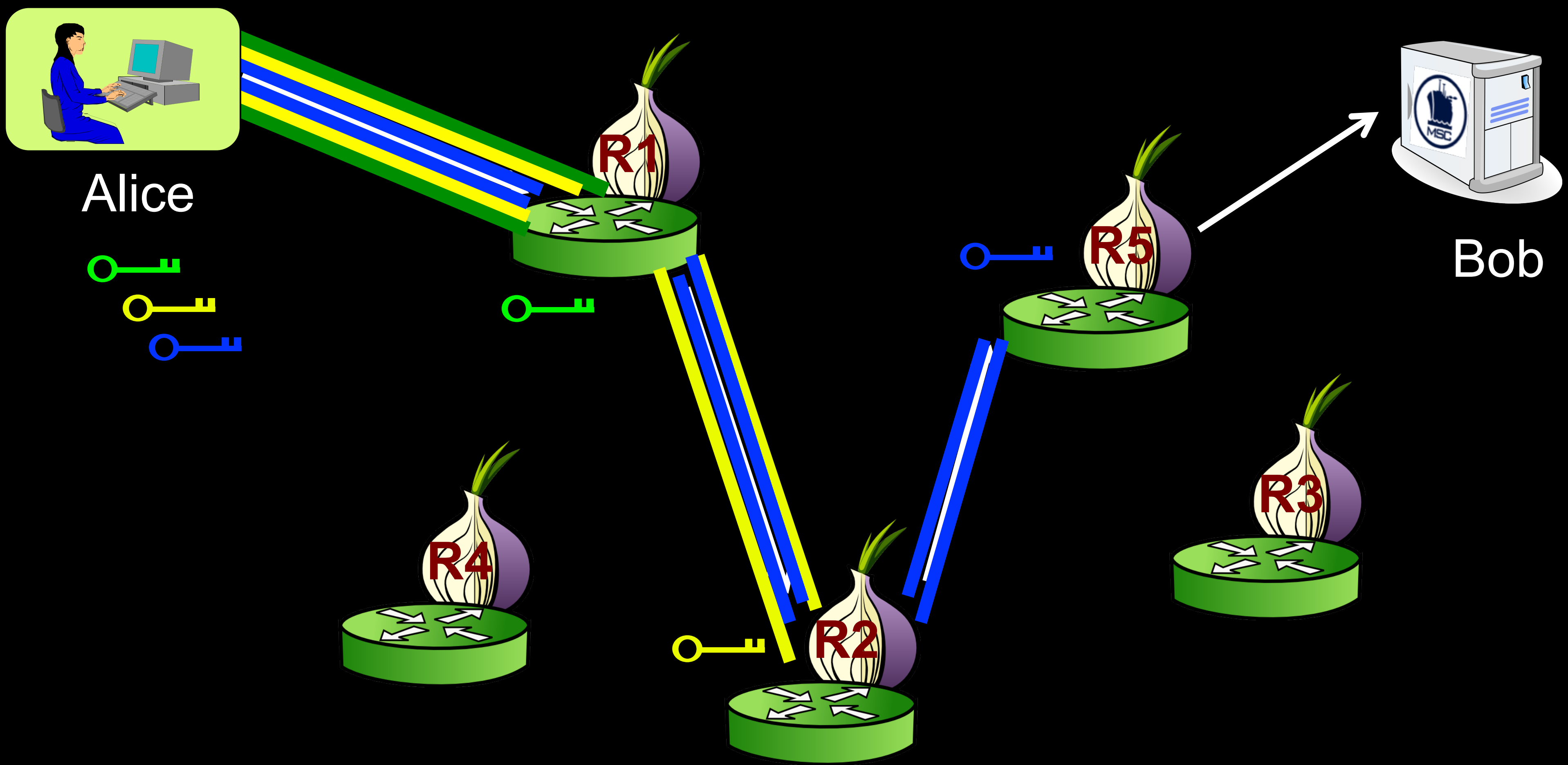
What onion routing is NOT: Mixes

- Entirely different threat model
  - mixes are based on an adversary not being able to correlate inputs and outputs he sees
  - onion routing is based on an adversary not being able to see both inputs and outputs to correlate
- Entirely different communications paradigm: Circuit based encryption vs. per message
  - onion routing supports bidirectional communication
  - onion routing supports low-latency communication
- Can be combined to make mixing onion routers, but not typically done or desired

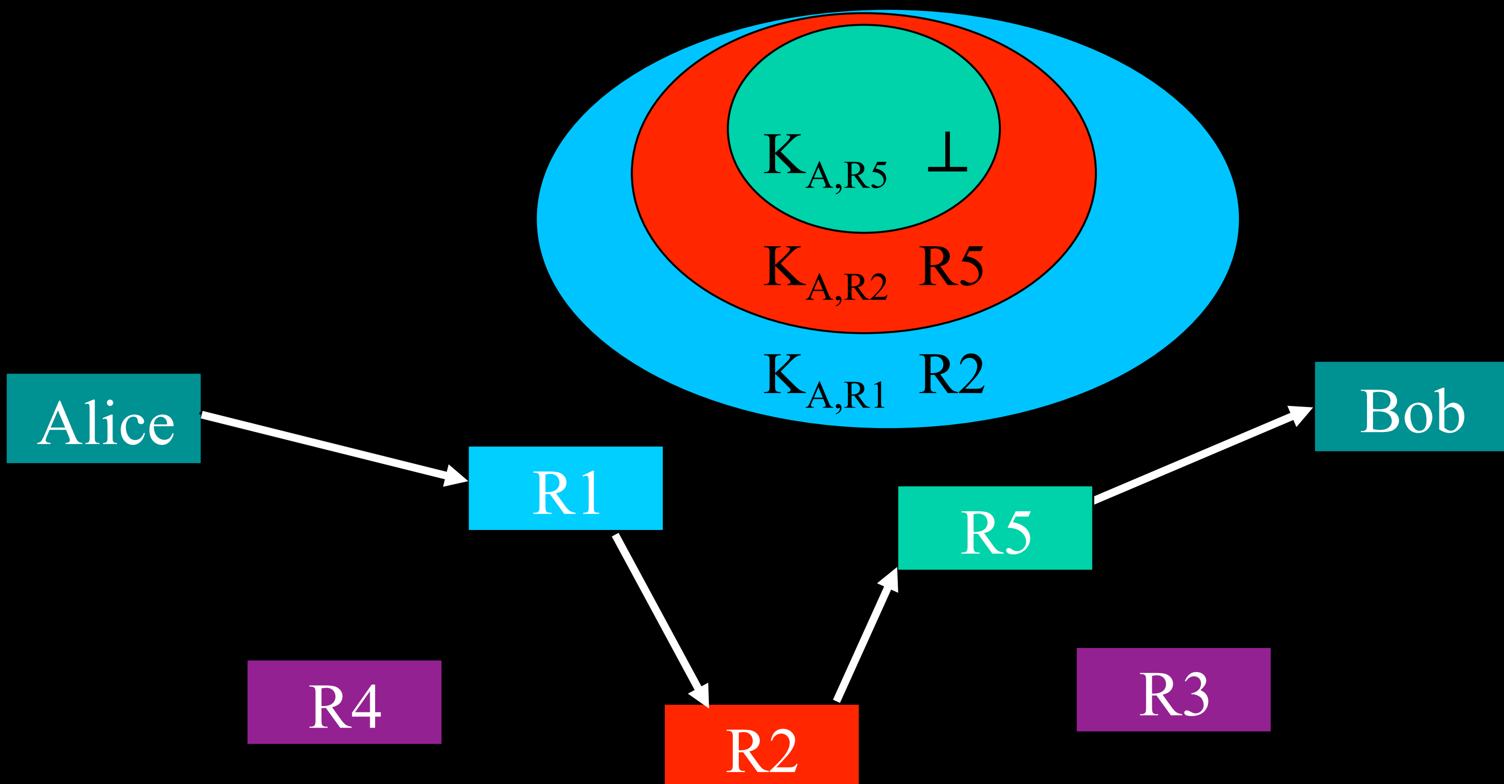
What onion routing is

- Uses expensive crypto (public-key) to lay a cryptographic circuit over which data is passed
- Typically uses free-route circuit building to make location of circuit endpoints unpredictable

Why call it “onion routing”? Answer: Because of the original key distribution data structure

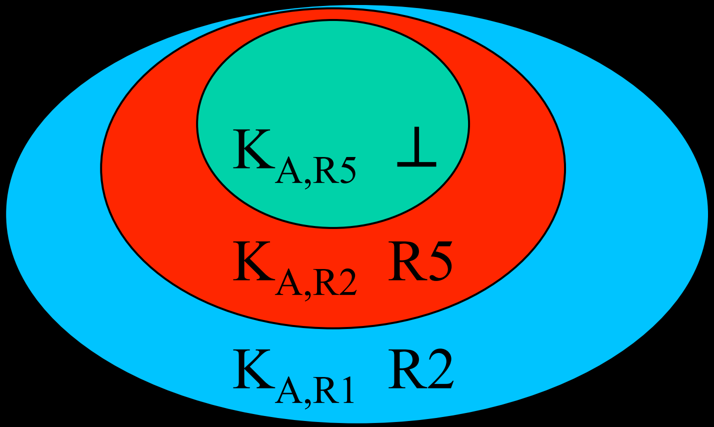


Why is it called onion routing?



- Onion: Just layers of public-key crypto
  - Nothing in the center, just another layer

Circuit setup



- NRL v0 and v1 onion routing and also ZKS Freedom network used onions to build circuits
  - Lacked Forward Secrecy
  - Required storing record of onions against replay
- Tor (NRL v2) uses one layer “onion skins”
  - ephemeral Diffie-Hellman yields forward secrecy
  - No need to record processed onions against replay
  - From suggestion out of Zack Brown’s Cebolla

Aside: Why is it called 'Tor' and what does 'Tor' mean?

- Frequent question to Roger c. 2001-2: Oh you're working on onion routing... which one?
- Roger: *THE* onion routing. The original onion routing project from NRL.
- Rachel: That's a good acronym.
- Roger: And it's a good recursive acronym.
- Plus, as a word, it has a good meaning in German (door/gate/portal) and Turkish (fine-meshed net)

Aside: Why is it called ‘Tor’ and what does ‘Tor’ mean?

- We foolishly called the first Tor paper “Tor: the second generation onion router”
- But this was very confusing
  - ‘Tor’ stands for “The onion routing” or “Tor’s onion routing”. It does not stand for “the onion router”
  - The paper is about the whole system, not just the onion routers
  - Tor is not the second generation

Onion routing origins: Generation 0

- Fixed-length five-node circuits
- Integrated configuration
- Static topology
- Loose-source routing
- ⋆ Partial active adversary
- Rendezvous servers and reply onions

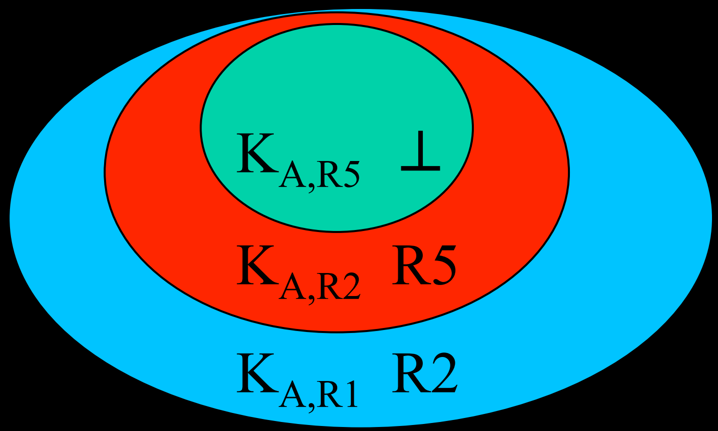
Onion routing, the next generation

- ⋆ Running a client separated from running an OR
- Variable length circuits (up to 11 hops per onion---or tunnel for more)
- Application independent proxies (SOCKS) plus redirector
- ⋆ Entry policies and exit policies
- Dynamic network state, flat distribution of state info
- Multiplexing of multiple application connections in single onion routing circuit
- Mixing of cells from different circuits
- Padding and bandwidth limiting

Third-generation onion routing (Tor)

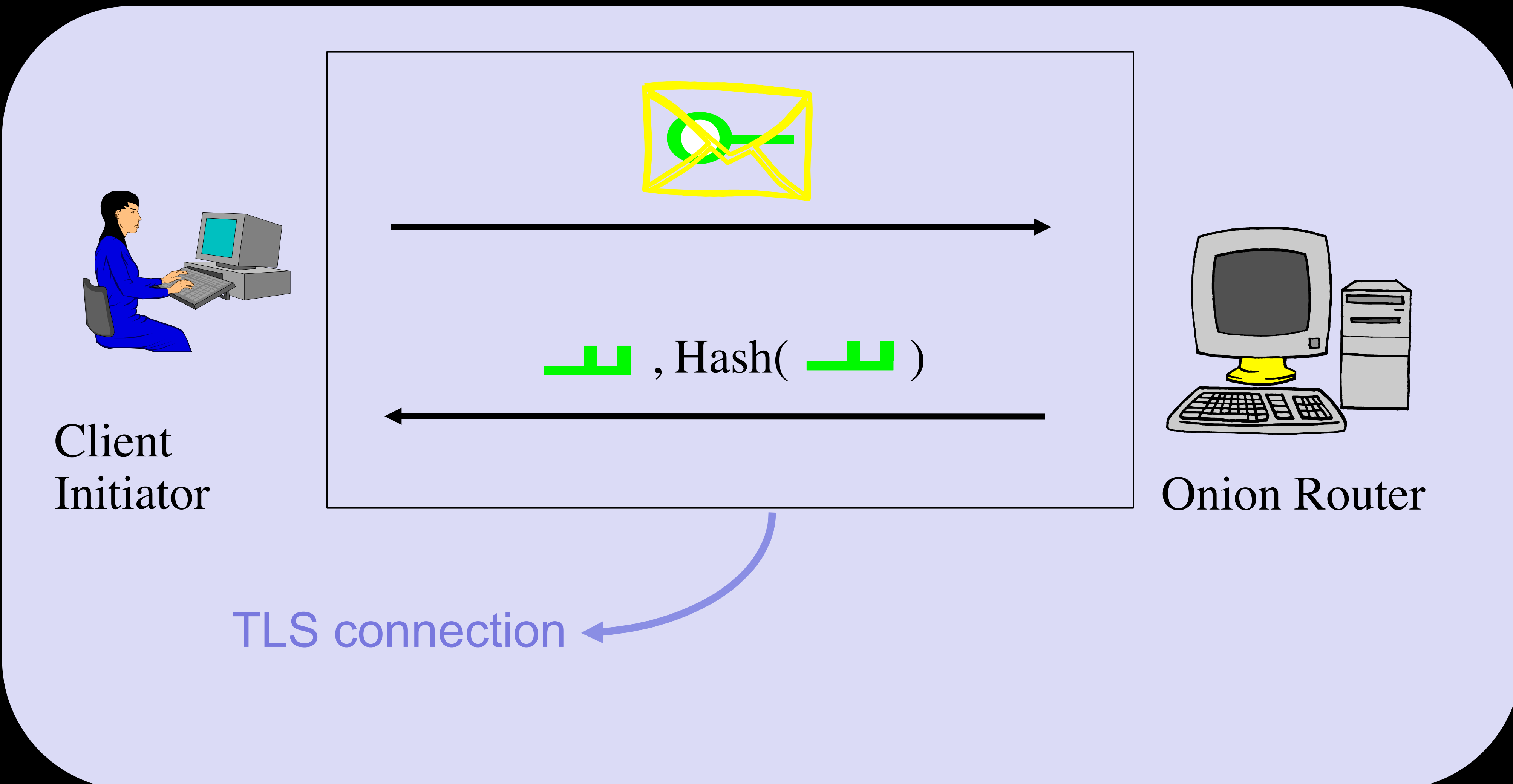
- ⋆ Onion skins, not onions: Diffie-Hellman based circuit building
- Fixed-length three-hop circuits
- Rendezvous circuits and hidden servers
- Directory servers, caching (evolved w/in Tor)
- Most application specific proxies no longer needed (still need e.g. for DNS)
- Congestion control
- End-to-end integrity checking
- No mixing and no padding

Circuit setup

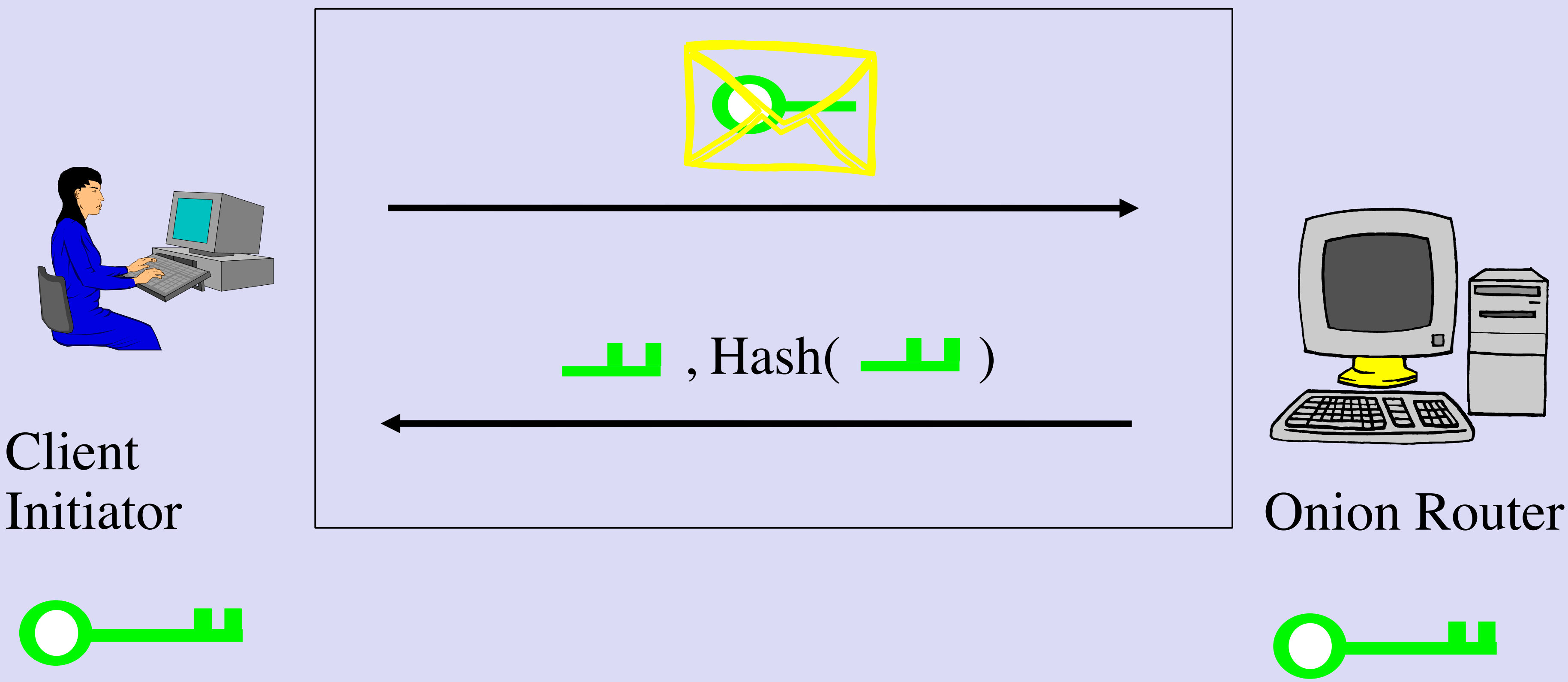


- NRL v0 and v1 onion routing and also ZKS Freedom network used onions to build circuits
  - Lacked Forward Secrecy
  - Required storing record of onions against replay
- Tor (NRL v2) uses one layer “onion skins”
  - ephemeral Diffie-Hellman yields forward secrecy
  - No need to record processed onions against replay
  - From suggestion out of Zack Brown’s Cebolla

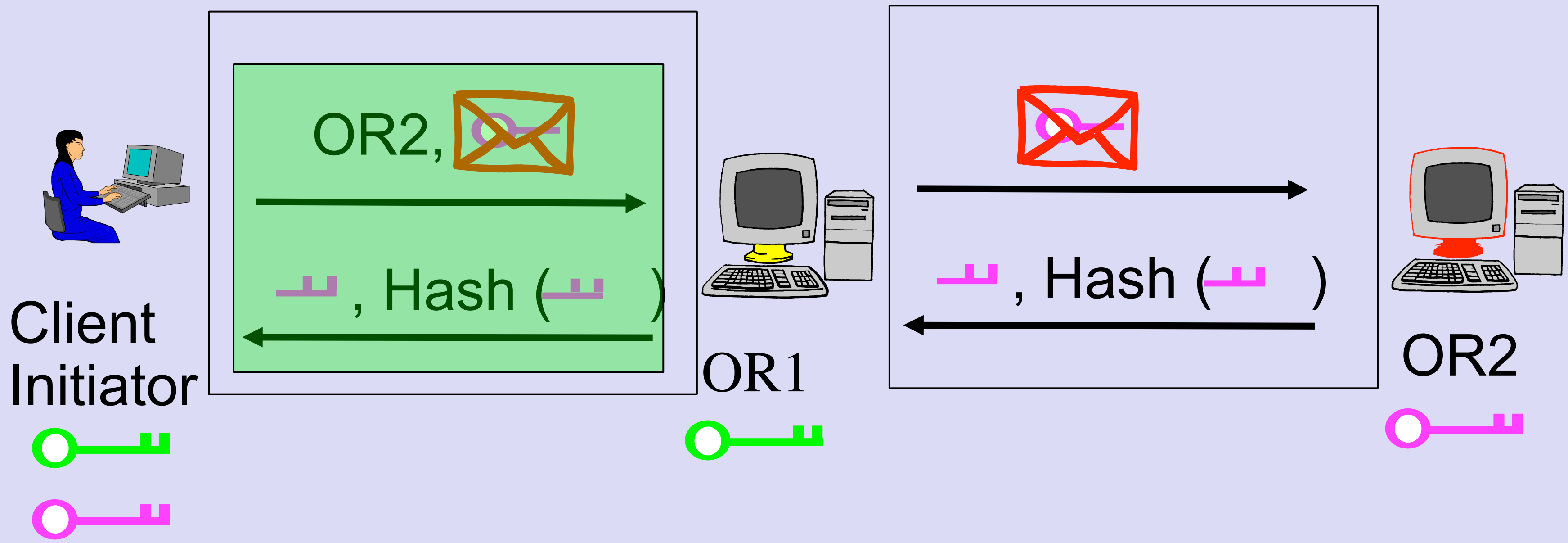
Original Tor Circuit Setup (Create)



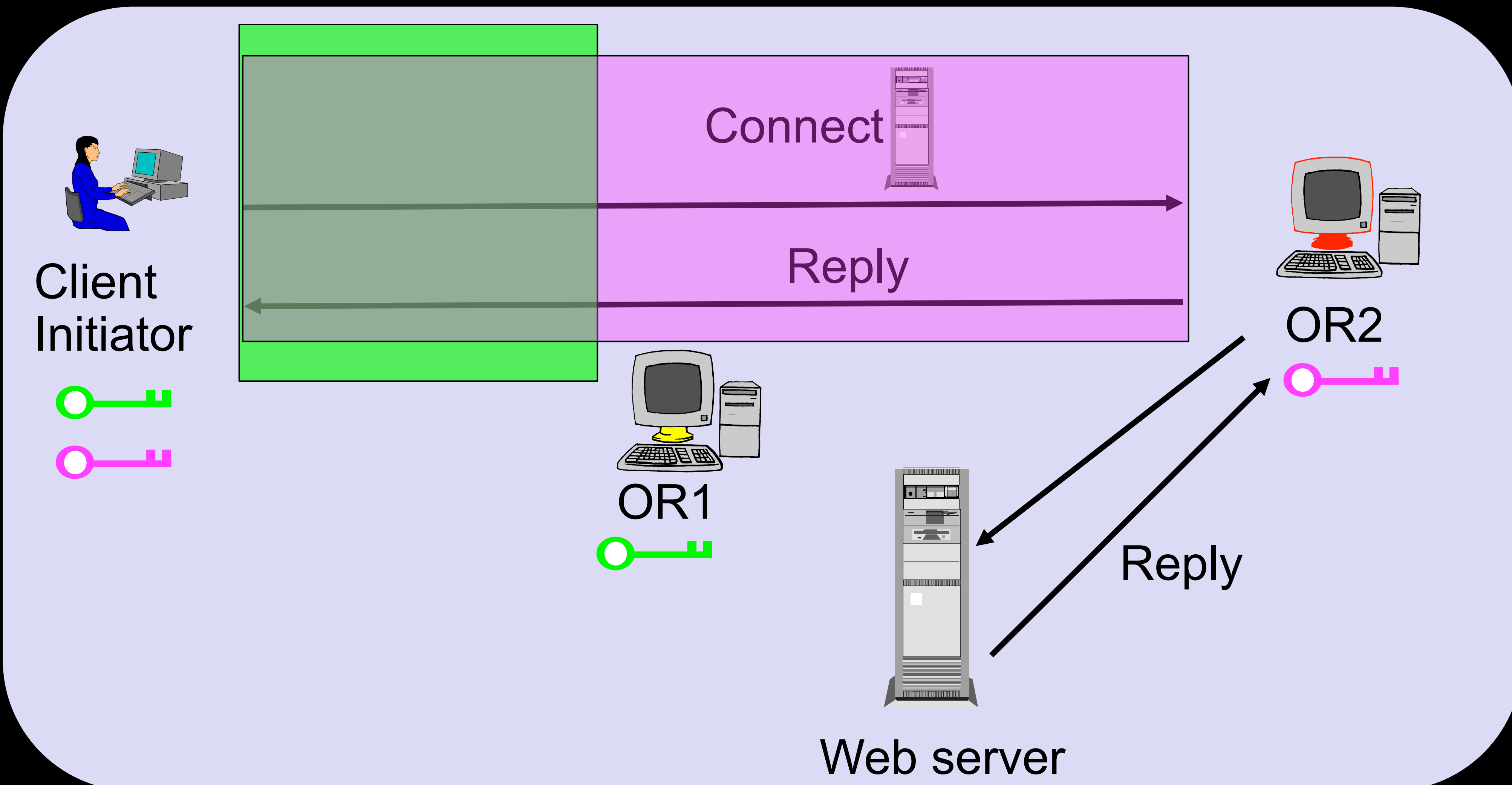
Original Tor Circuit Setup (Create)



Original Tor Circuit Setup (Extend)



Original Tor Circuit Setup (Begin) and Data Flow

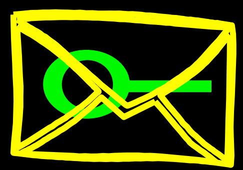


Why a Tor authenticated key establishment protocol

- Designing your own authentication protocol is error prone. Why not use an established protocol in the first place?
- Answer 1: We require only one-way authentication. Two way wastes expensive computation.
- Answer 2: To fit whole messages inside Tor cells. A public key and a signature don't both fit in one 512-byte cell.
- Protocol was verified using the NRL protocol analyzer in the Dolev-Yao model.
- In 2005 Ian Goldberg found flaw in the way Tor implemented this protocol (checking that a public value was not based on a weak key).
- In 2006 Ian proved the (properly implemented) protocol secure in the random oracle model.

Circuit establishment efficiency

- Original Tor Authentication Protocol (TAP) uses RSA to encrypt the Diffie-Hellman public key



- New/Old idea (1st considered in 1996): Let the nodes use published public DH keys
- Clients create ephemeral DH keys to combine with published node DH keys (ElGamal key exchange aka half-certified Diffie-Hellman exchange)
- Saves one exponentiation at client and one at node for each node in circuit (about half current load)
- Significantly reduces Tor overhead for running a volunteer node

Brief history of using Diffie-Hellman in Tor Circuit Establishment

1996: We (Goldschlag, Reed, me) considered including DH keys in layers of circuit-establishment onion

- For computational efficiency (not considering Forward Secrecy then)

2004: TAP replaces onions, includes DH for Forward Sec. (Dingledine, Mathewson, me)

- Verified using NRL Protocol Analyzer in Dolev-Yao Model

2005: Goldberg verifies TAP in Random Oracle Model

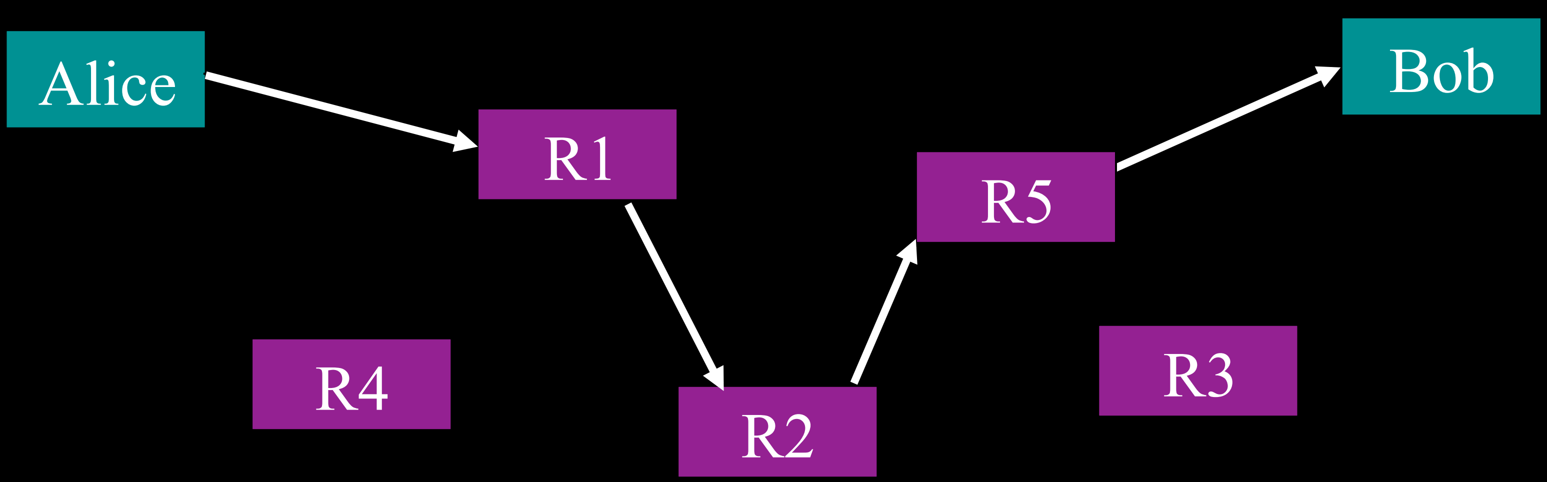
2007: We (Øverlier and me) propose “fourth protocol”, DH-based authentication and key-establisment (with informal security argument)

2012: Goldberg, Stebila, and Ostaoglu break fourth protocol, introduce ntor

2013: ECDH (curve 25519) version of ntor included in Tor stable release

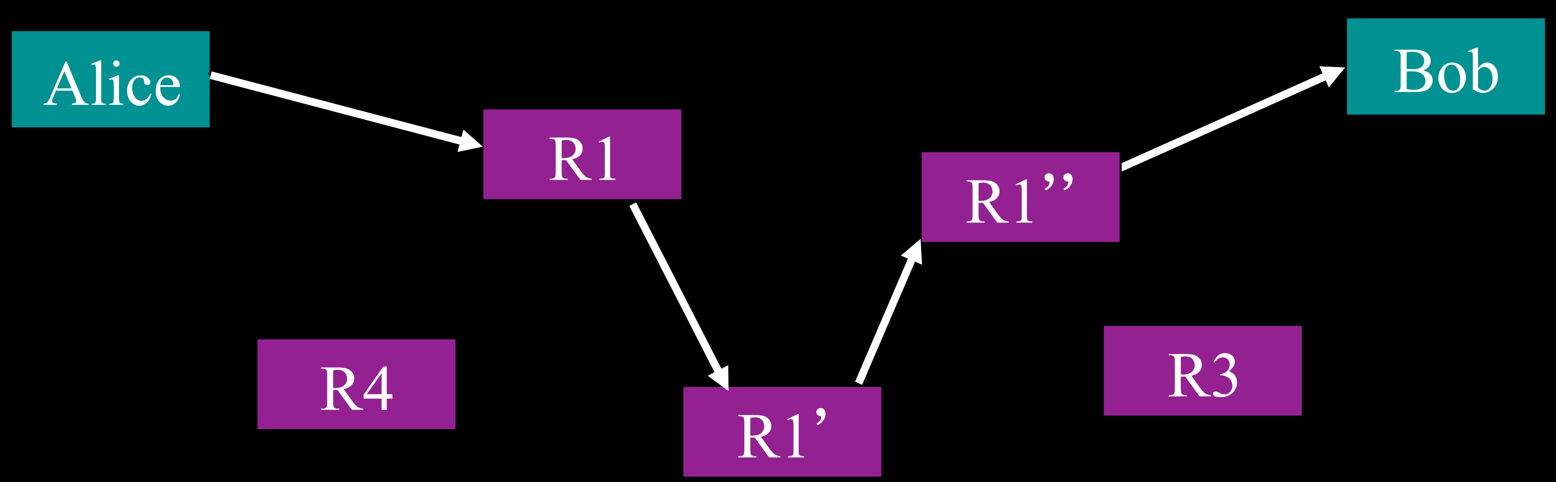
Network and Route Discovery

- Alice has to know a set of nodes and pick a route from them
  - Must know how to find R1
  - Must learn more network nodes to pick a route
  - Cannot trust R1 to tell about the rest of the network



Network and Route Discovery

- Alice has to know a set of nodes and pick a route from them
  - Must know how to find R1
  - Must learn more network nodes to pick a route
  - Cannot trust R1 to tell about the rest of the network



How do we know where to build a circuit? Network discovery.

- Important for all clients to get same picture of network to avoid epistemic partitioning
- Initial onion routing design had trivial/brittle solution
  - network and identity keys simply hard coded in prototypes
- Next generation had a design for flat flooding of network state to all relays
  - complex, tricky, scales in principal but ?
  - not ever deployed in practice
- Tor has a directory system: See next slides
- Bridge distribution: Should have gone to FOCI.

Tor Directory Version 1

- Directory authorities serve
  - Relay descriptors containing e.g. identity keys & IP addresses
  - Network status: whether relays are up or down
- Network caches added to prevent DirAuths from being comms bottleneck

Tor Directory Version 2

Issue 1:

- As Tor network grew so did size of directory
  - large downloads a problem
- Most descriptor info relatively persistent

Solution:

- Directory authorities serve
  - Network status summary: Hashes of each relay’s current descriptor
  - Clients retrieve full descriptors only for relays they don’t know or that have changed

Tor Directory Version 2

Issue 2:

- Version 1 DirAuths function independently
  - trust bottleneck
  - Risk grows with number of DirAuths

Solution:

- Clients trust statements about relays made by majority of DirAuths

Tor Directory Version 3

Issue 1: Descriptors contain much useful info not needed for basic contact

Solution: Microdescriptors go into consensus

- Identity key, address, exit ports
- Good for about a week typically

Tor Directory Version 3

Issue 2: Update and synch. Issues can lead to clients having different network views

Solution: Collective DirAuth vote on network consensus

Issue 3: Identity keys for DirAuths most sensitive data

Solution: Create DirAuth network info signing keys and keep identity keys offline

Onion routing started from a practical motivation

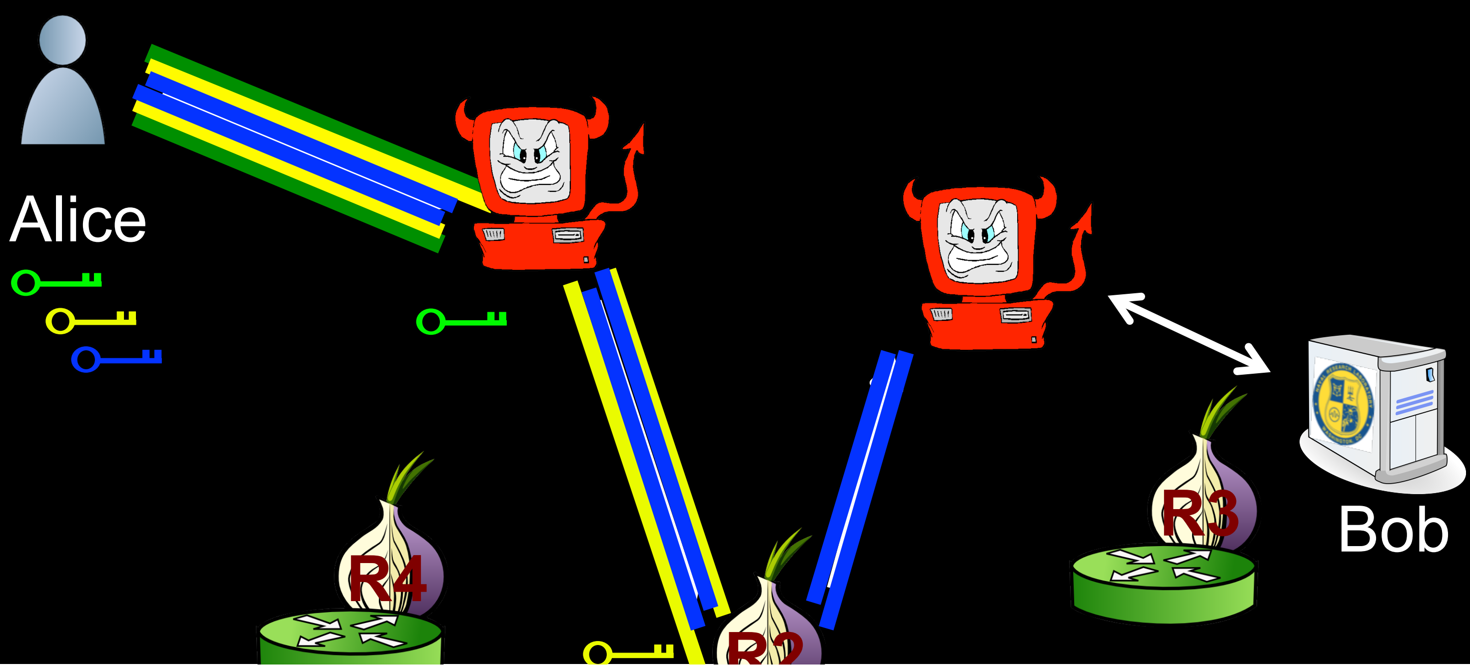
- How do we know the whole enterprise is not fundamentally broken on an abstract level?
- Where's the underlying theory to back this up?

- Note: OK not to have a rigorous notion of security at first
  - Analyses based on state of the art would have been misleading
  - Global Passive Adversary both too strong and too weak
    - Cf. "Why I'm not an Entropist" and "Sleeping Dogs Lie in a Bed of Onions but Wake when Mixed"

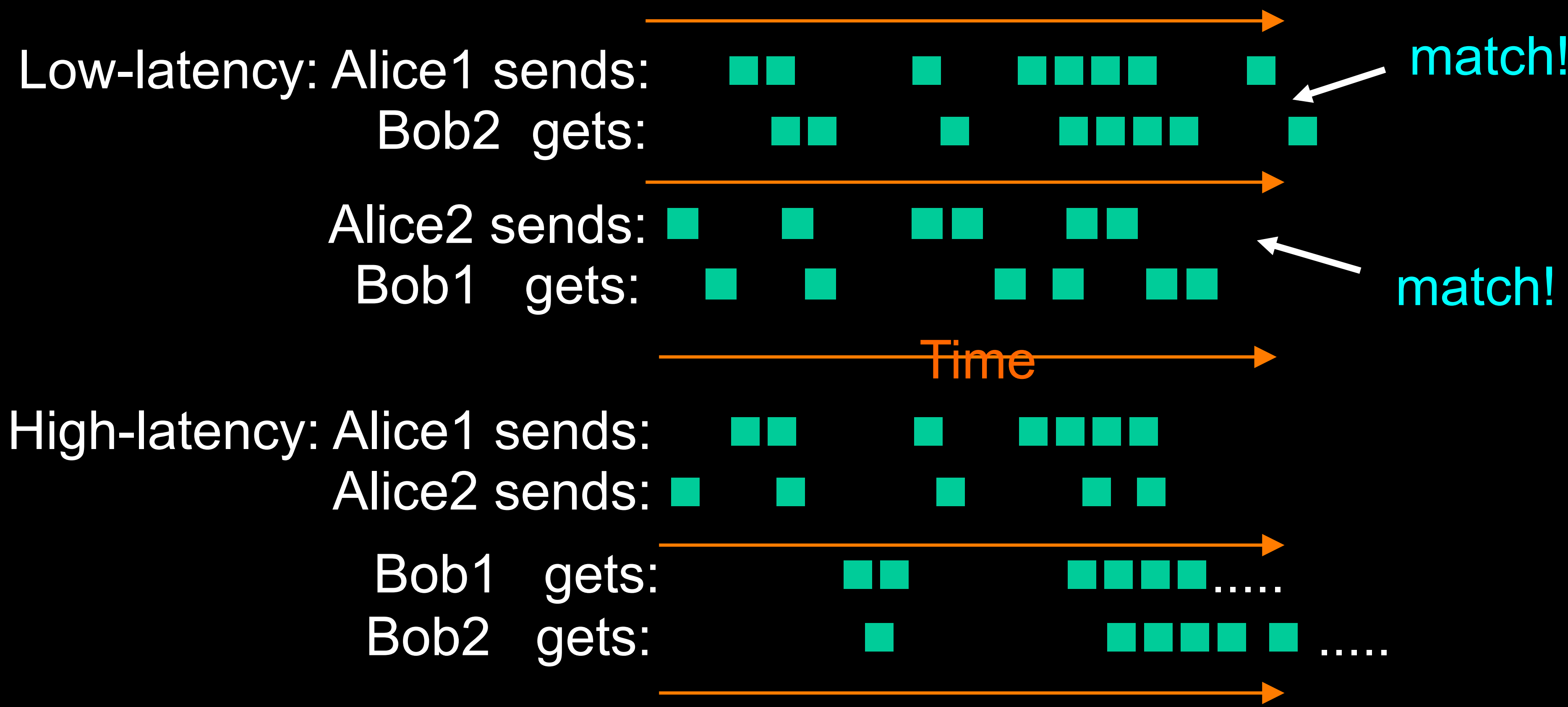
Anonymous Communication c. 2000

	Deployed	Analyzed
Mix Networks	+	+
Dining cryptographers	−	+
Onion routing	+	−
Crowds	−	+

Adversary observing all traffic entering and leaving network breaks onion routing



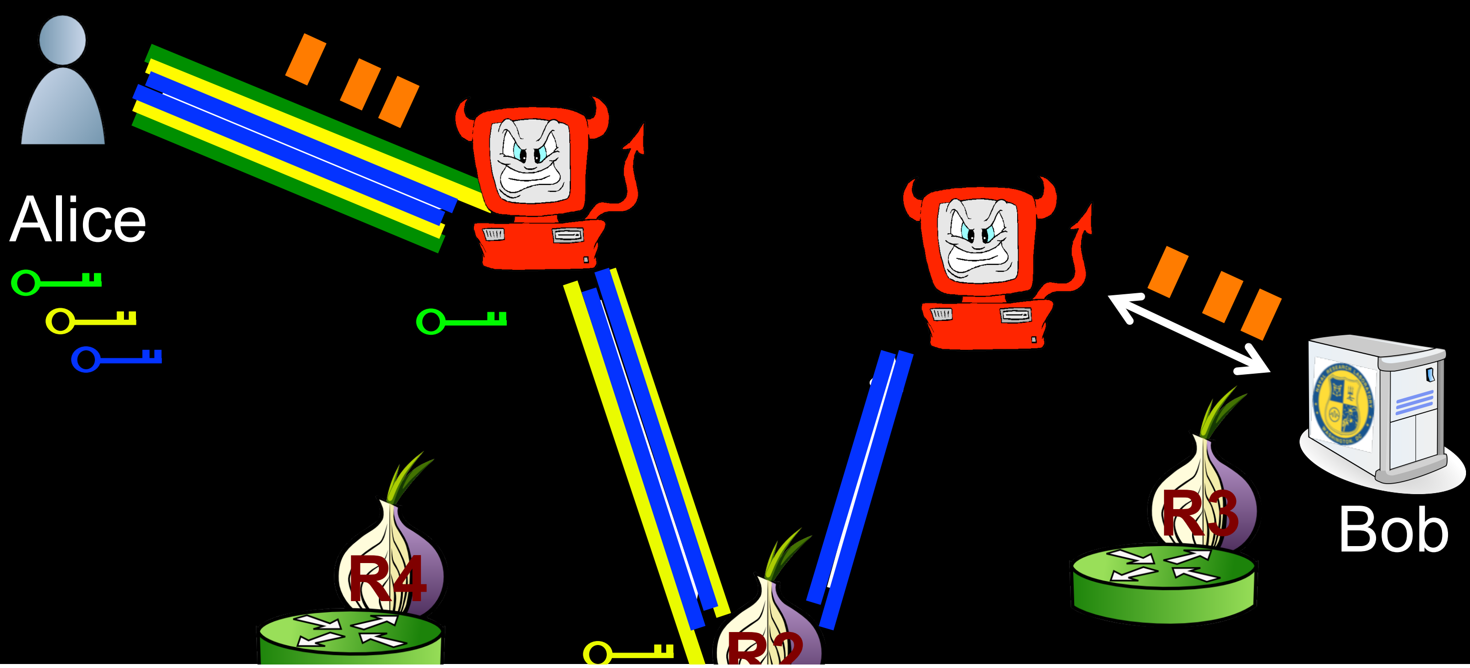
Low-latency systems are vulnerable to correlation by a global adversary



These attacks work in practice. The obvious defenses are expensive (like high-latency), useless, or both.

Adversary observing all traffic entering and leaving network breaks onion routing

- "Towards an Analysis of Onion Routing Security" Syverson et al. PETS 2000
- Presented and analyzed adversary model assumed in prior onion routing work
  - Network of n onion routers, c compromised onion routers
  - Security approx. c^2 / n^2



Formal analysis of onion routing

1. Possibilistic characterization using I/O automata
2. Probabilistic analysis abstracting I/O automata characterization to a black box
3. Representing black-box results in standard cryptographic models

Possibilistic Analysis Overview

- Formally model onion routing using input/output automata
- Characterize the situations that provide anonymity

Possibilistic Analysis Overview

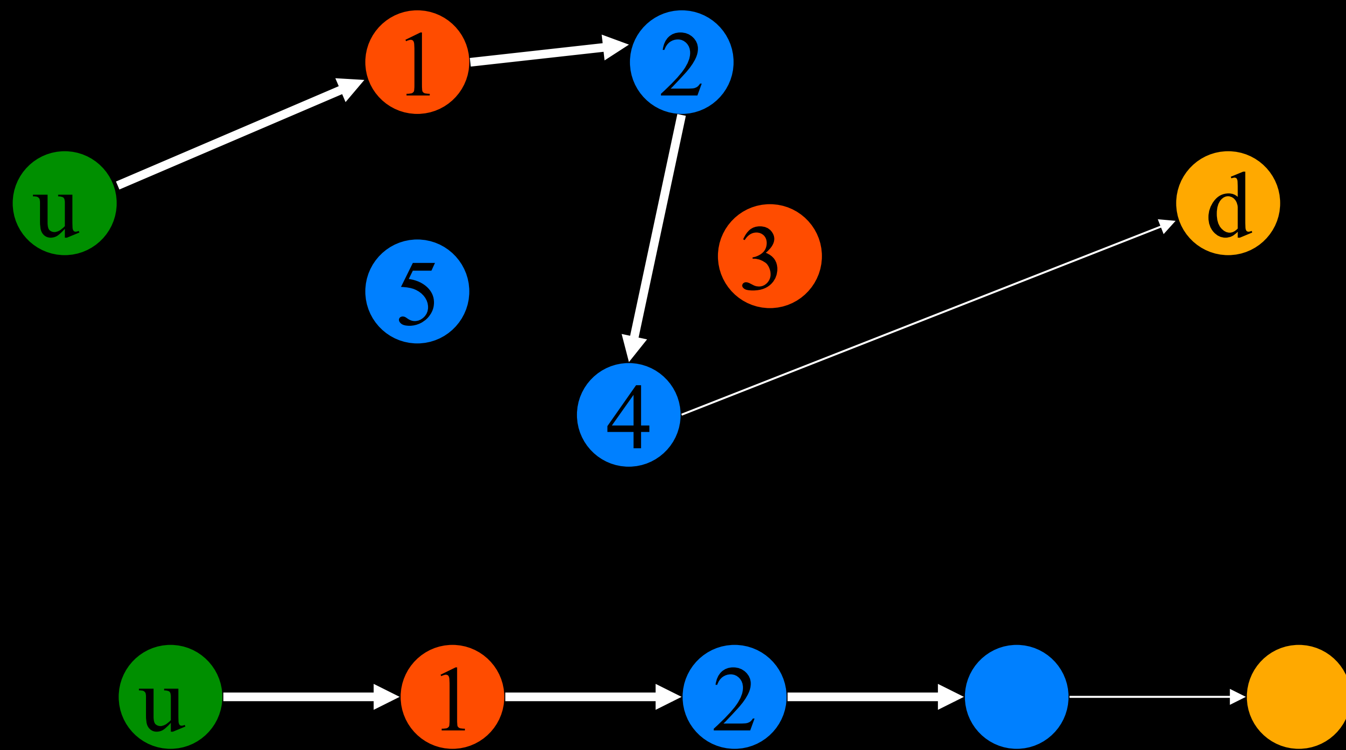
- Formally model onion routing using input/ output automata
  - Simplified onion-routing protocol
  - Non-cryptographic analysis
- Characterize the situations that provide anonymity

Possibilistic Analysis Overview

- Formally model onion routing using input/ output automata
  - Simplified onion-routing protocol
  - Non-cryptographic analysis
- Characterize the situations that provide anonymity
  - Send a message, receive a message, communicate with a destination
  - Possibilistic anonymity

Main Theorem

[“*A Model of Onion Routing with Provable Anonymity*”, Feigenbaum, Johnson, and Syverson, in FC07



Main theorem: Adversary can only determine the parts of a circuit it controls or is next to.

Anonymous Communication

- *Sender anonymity*: Adversary can’t determine the sender of a given message
- *Receiver anonymity*: Adversary can’t determine the receiver of a given message
- *Unlinkability*: Adversary can’t determine who talks to whom

Model

- Constructed with I/O automata
  - Models asynchrony
  - Relies on abstract properties of cryptosystem
- Simplified onion-routing protocol
  - No key distribution
  - No circuit teardowns
  - No separate destinations
  - No streams
  - No stream cipher
  - Each user constructs a circuit to one destination
  - Circuit identifiers

Input/Ouput Automata

- States
- Actions
  - Input, ouput, internal
  - Actions transition between states
- Every state has *enabled* actions
- Input actions are always enabled
- Alternating state/action sequence is an *execution*
- In *fair* executions actions enabled infinitely often occur infinitely often
- In *cryptographic* executions no encrypted control messages are sent before they are received unless the sender possesses the key

I/O Automata Model

- Automata
  - User
  - Server
  - Fully-connected network of FIFO Channels
  - Adversary replaces some servers with arbitrary automata

- Notation
  - U is the set of users
  - R is the set of routers
  - $N = U \cup R$ is the set of all agents
  - $A \subseteq N$ is the adversary
  - K is the keyspace
  - l is the (fixed) circuit length
  - $k(u,c,i)$ denotes the ith key used by user u on circuit c

User automaton

$c \in \{(r_1, \ldots, r_l) \in R^l \mid \forall_i r_i \neq r_{i+1}\}$; init: arbitrary ▷ User's circuit
$i \in \mathbf{N}$; init: random ▷ Circuit identifier
$b \in \mathbf{N}$; init: 0 ▷ Next hop to build

---

procedure START
SEND$(c_1, [i, 0, \{\text{CREATE}\}_{k(u,c,1)}])$
$b = 1$
end procedure
procedure MESSAGE(msg, j) ▷ $msg \in M$ received from $j \in N$
if $j = c_1$ **then**
if $b = 1$ **then**
if $msg = [i, 0, \text{CREATED}]$ **then**
$b{+}{+}$
SEND$(c_1, [i, 0, \{[\text{EXTEND}, c_b, \{\text{CREATE}\}_{k(u,c,b)}]\}_{k(u,c,b-1),\ldots,k(u,c,1)}])$
end if
else if $b < l$ **then**
if $msg = [i, 0, \{\text{EXTENDED}\}_{k(u,c,b-1),\ldots,k(u,c,1)}]$ **then**
$b{+}{+}$
SEND$(c_1, [i, 0, \{[\text{EXTEND}, c_b, \{\text{CREATE}\}_{k(u,c,b)}]\}_{k(u,c,b-1),\ldots,k(u,c,1)}])$
end if
else if $b = l$ **then**
if $msg = [i, 0, \{\text{EXTENDED}\}_{k(u,c,b-1),\ldots,k(u,c,1)}]$ **then**
$b{+}{+}$
end if
end if
end if
end procedure

Server automaton

$keys \subseteq K$, where $|keys| \geq |U| \cdot \lceil \frac{l}{2} \rceil$; init: arbitrary $\triangleright$ Private keys

$T \subset N \times \mathbf{N} \times R \times \mathbf{Z} \times keys$; init: $\emptyset$ $\triangleright$ Routing table

procedure $\textsc{Message}([i,n,p],q)$ $\triangleright$ $[i,n,p] \in M$ received from $q \in N$

  if $[q,n,\emptyset,-1,k] \in T$ **then** $\triangleright$ In link created, out link absent

    if $\exists_{s \in R-r, b \in P} p = \{[\text{EXTEND}, s, b]\}_k$ **then**

      $\textsc{send}(s, [minid(T,s), b])$

      $T = T - [q,n,\emptyset,-1,k] + [q,n,s,-minid(T,s),k]$

    end if

  else if $[s,m,q,-n,k] \in T$ **then** $\triangleright$ In link created, out link initiated

    if $p = \text{CREATED}$ **then**

      $T = T - [s,m,q,-n,k] + [s,m,q,n,k]$

      $\textsc{send}(s, [i,m,\{\text{EXTENDED}\}_k])$

    end if

  else if $\exists_{m>0}[q,n,s,m,k] \in T$ **then** $\triangleright$ In and out links created

    $\textsc{send}(s, [i,m,\{p\}_{-k}])$ $\triangleright$ Forward message down the circuit

  else if $[s,m,q,n,k] \in T)$ **then** $\triangleright$ In and out links created

    $\textsc{send}(s, [i,m,\{p\}_k])$ $\triangleright$ Forward message up the circuit

  else

    if $\exists_{k \in keys} p = \{\text{CREATE}\}_k$ **then** $\triangleright$ New link

      $T = T + [q,n,\emptyset,-1,k]$

      $\textsc{send}(q, [i,n,\text{CREATED}])$

    end if

  end if

end procedure

Anonymity

Definition (configuration):

A ***configuration*** is a function $U \rightarrow R^l$ mapping each user to his circuit.

Anonymity

Definition (configuration):

A ***configuration*** is a function $U \rightarrow R^l$ mapping each user to his circuit.

Definition (indistinguishability):

Executions α and β are ***indistinguishable*** to adversary *A* when his actions in β are the same as in α after possibly applying the following:

ξ: A permutation on the keys not held by *A*.

π: A permutation on the messages encrypted by
a key not held by *A*.

Anonymity

Definition (anonymity):

User u performs action α ***anonymously*** in *configuration C* with respect to adversary A if, for every execution of C in which u performs α, there exists an execution that is *indistinguishable* to A in which u does not perform α.

Anonymity

Definition (anonymity):

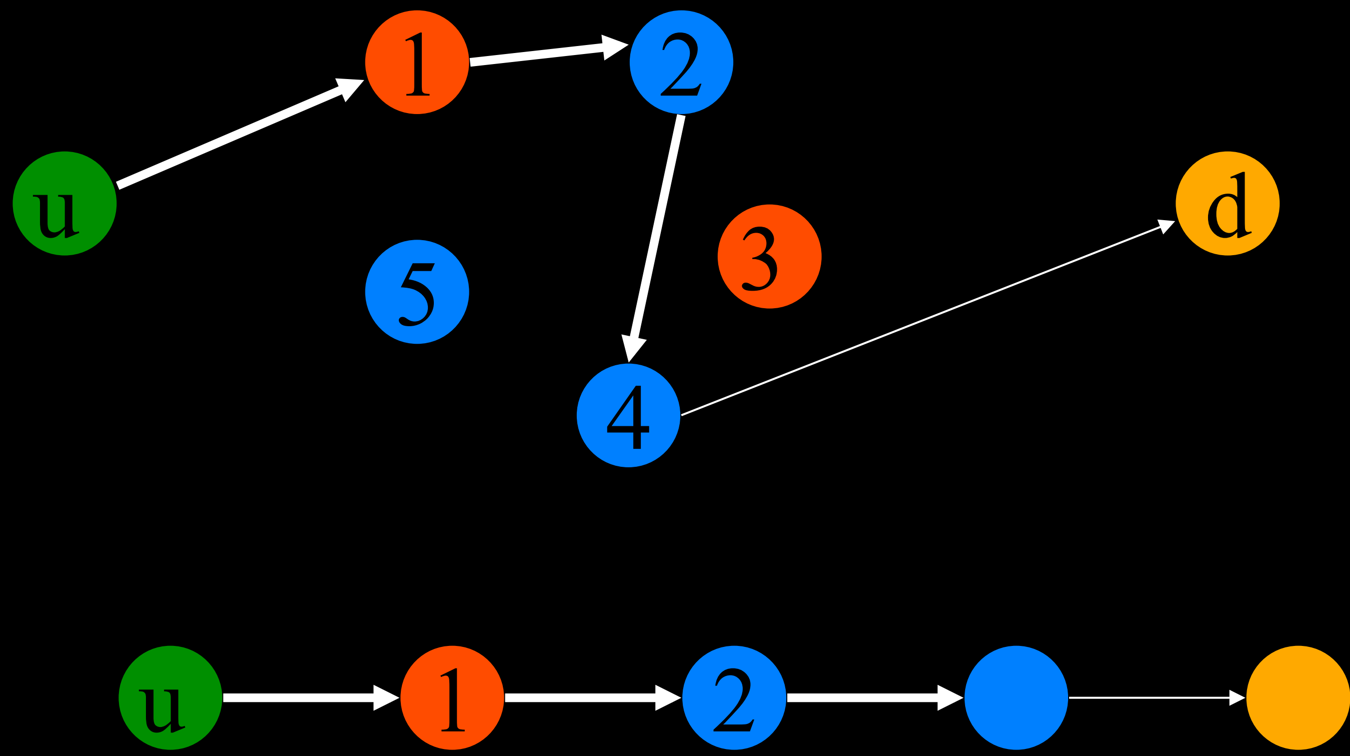
User *u* performs action α ***anonymously*** in *configuration C* with respect to adversary *A* if, for every execution of *C* in which *u* performs α, there exists an execution that is *indistinguishable* to *A* in which *u* does not perform α.

Definition (unlinkability):

User *u* is ***unlinkable*** to *d* in configuration *C* with respect to adversary *A* if, for every fair, cryptographic execution of *C* in which *u* talks to *d*, there exists a fair, cryptographic execution that is indistinguishable to *A* in which *u* does not talk to *d*.

Theorem: Let C and D be configurations for which there exists a permutation ρ: $U \rightarrow U$ such that $C_i(u) = D_i(\rho(u))$ if $C_i(u)$ or $D_i(\rho(u))$ is compromised or is adjacent to a compromised router. Then for every fair, cryptographic execution α of C there exists an indistinguishable, fair, cryptographic execution β of D. The converse also holds.

Main Theorem



Main theorem: Adversary can only determine the parts of a circuit it controls or is next to.

Unlinkability

Corollary: A user is unlinkable to its destination when:

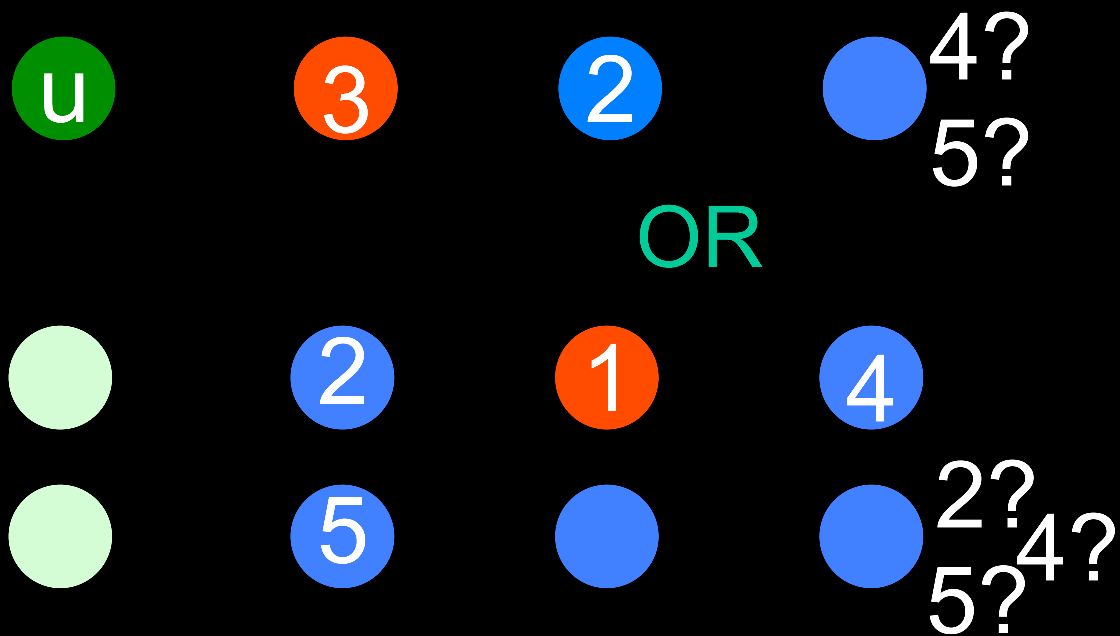
Unlinkability

Corollary: A user is unlinkable to its destination when:

The last router is unknown.

Unlinkability

Corollary: A user is unlinkable to its destination when:

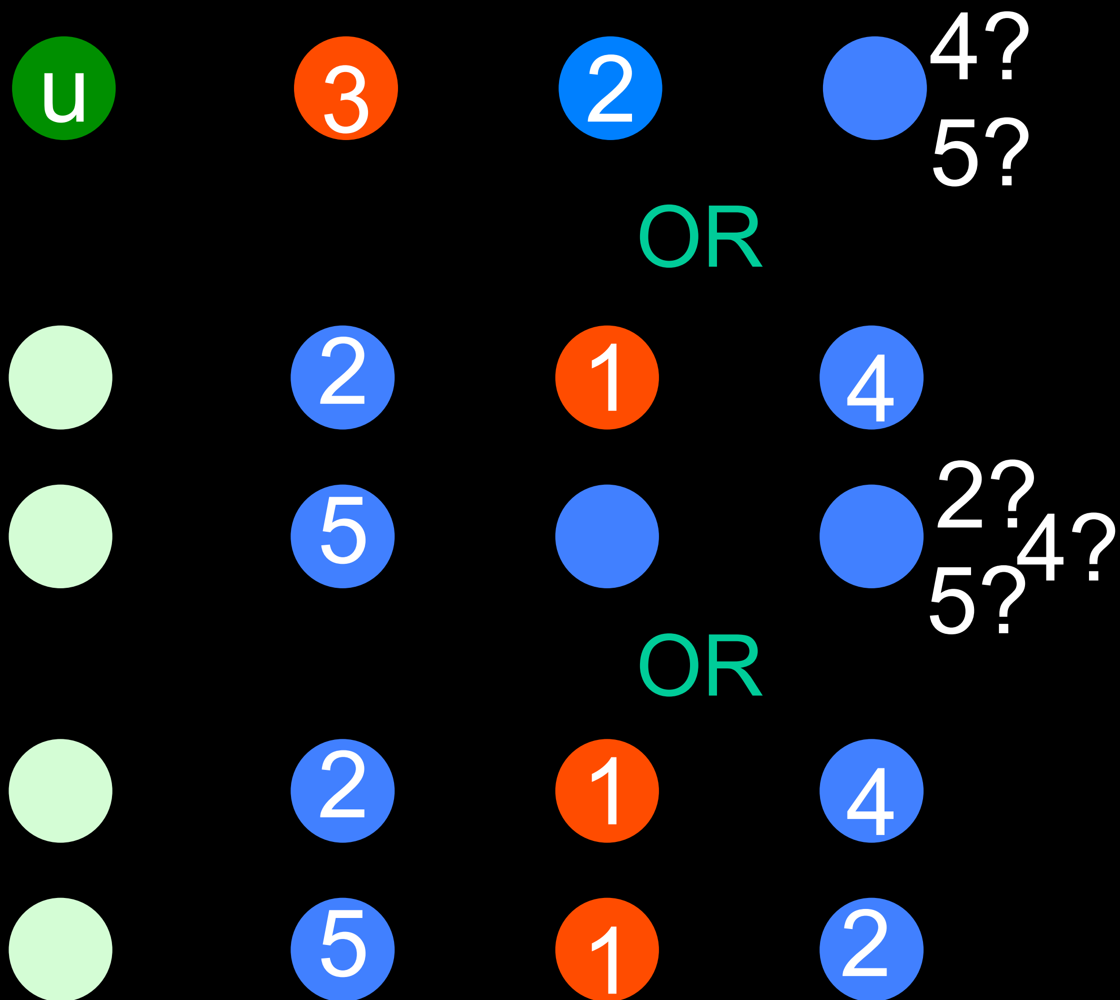


The last router is unknown.

The user is unknown and another unknown user has an unknown destination.

Unlinkability

Corollary: A user is unlinkable to its destination when:



The last router is unknown.

The user is unknown and another unknown user has an unknown destination.

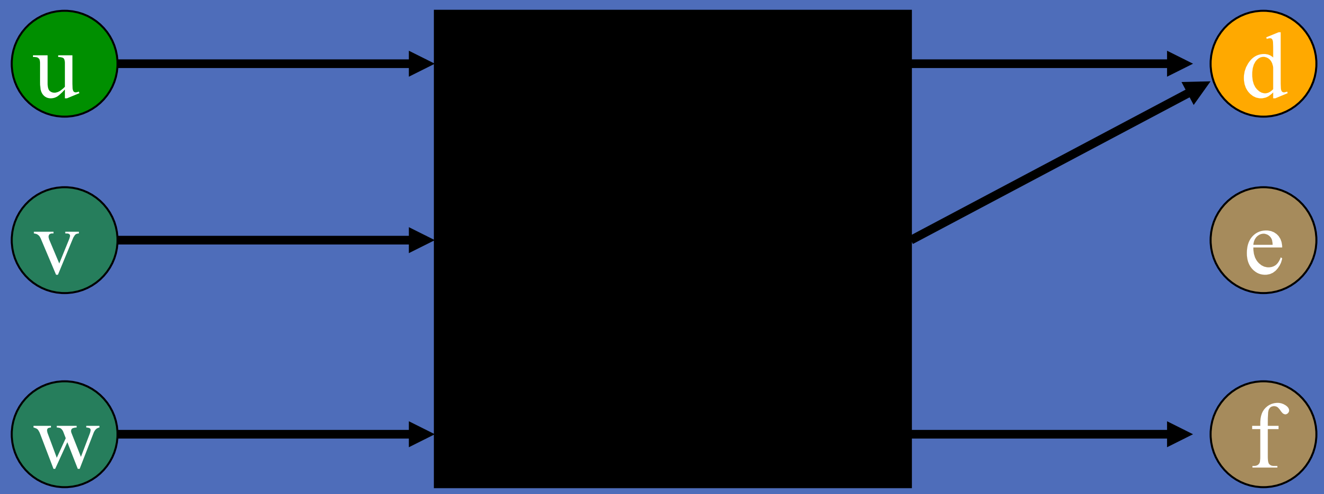
The user is unknown and another unknown user has a different destination.

Probabilistic anonymity

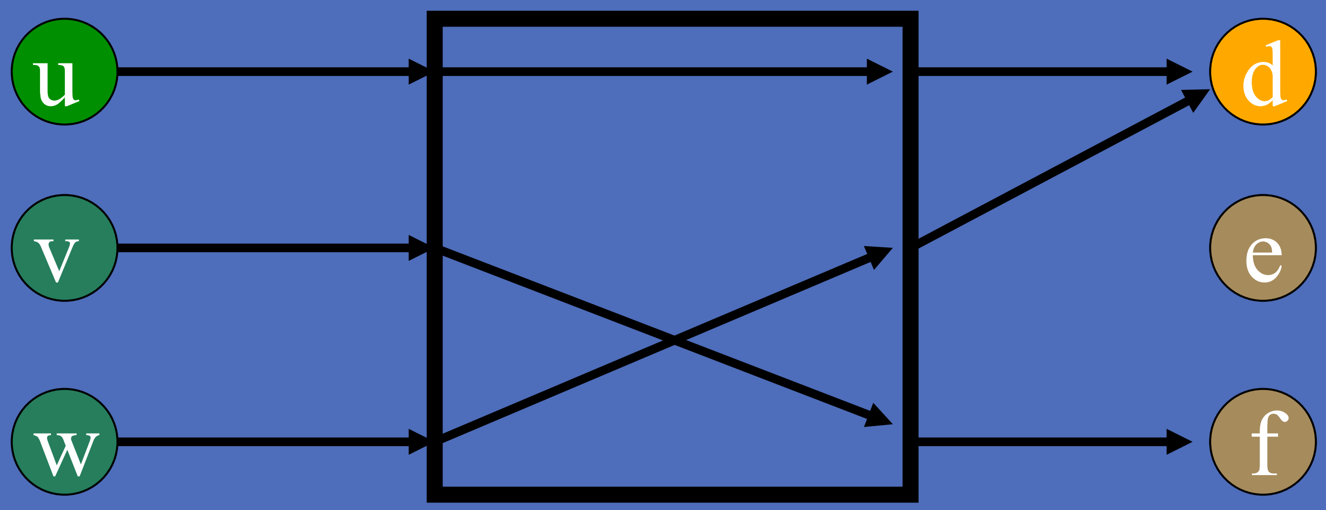
- Possibilistic result is nice, but we would like to quantify the anonymity provided by a system

- And we want to use a black box model, like this

Black-box Abstraction

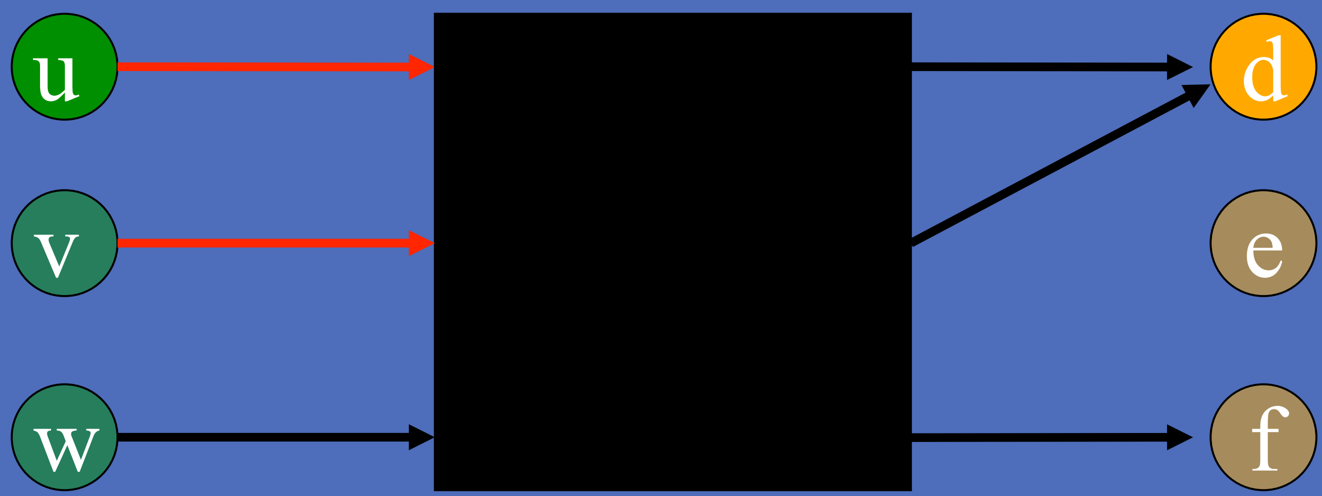


Black-box Abstraction



1. Users choose a destination

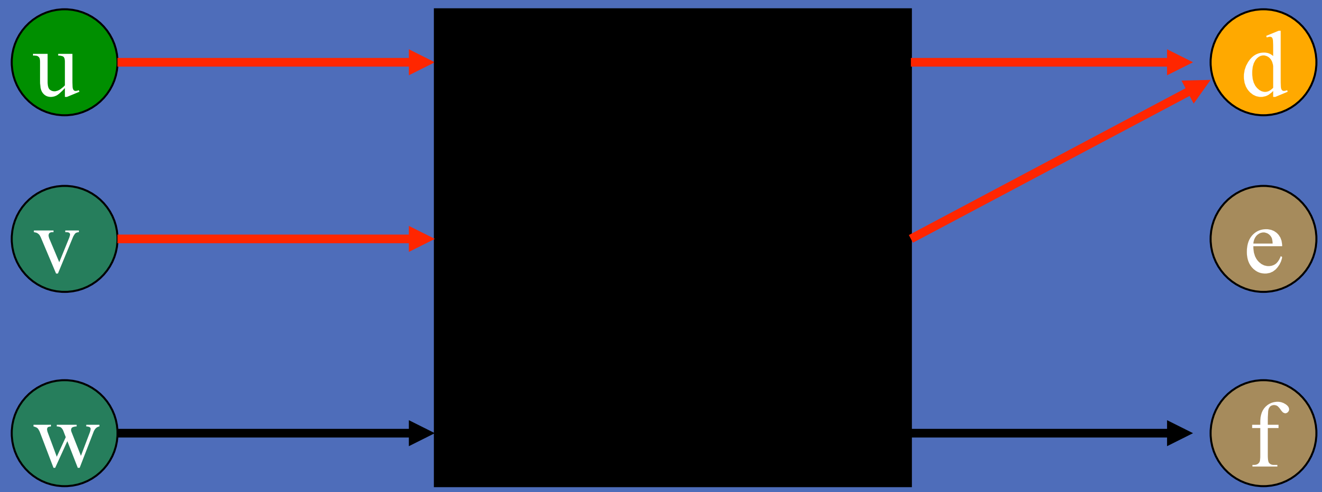
Black-box Abstraction



1. Users choose a destination

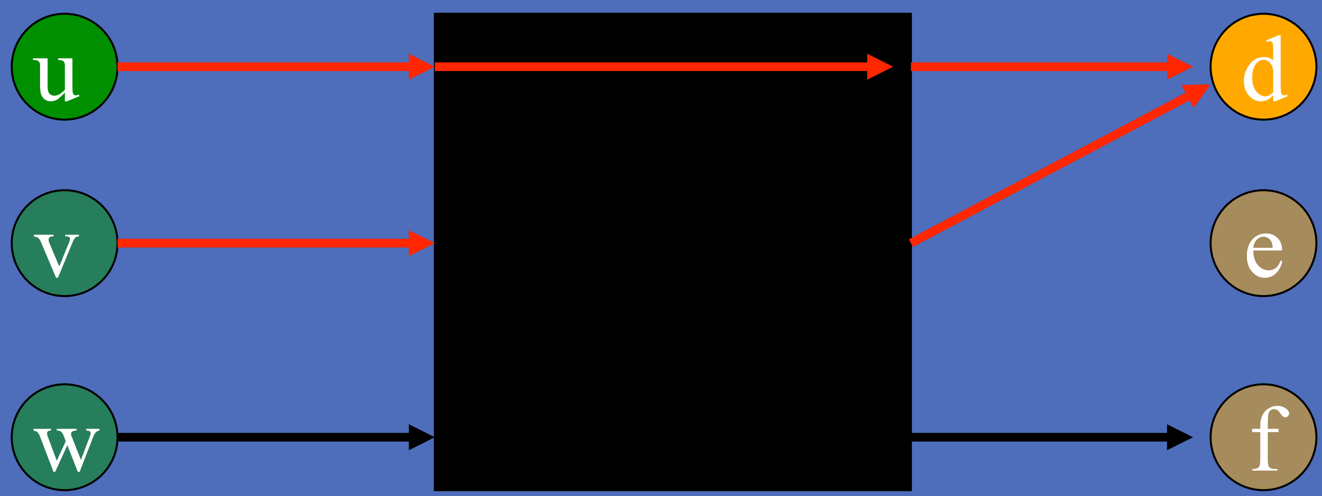
2. Some inputs are observed

Black-box Abstraction



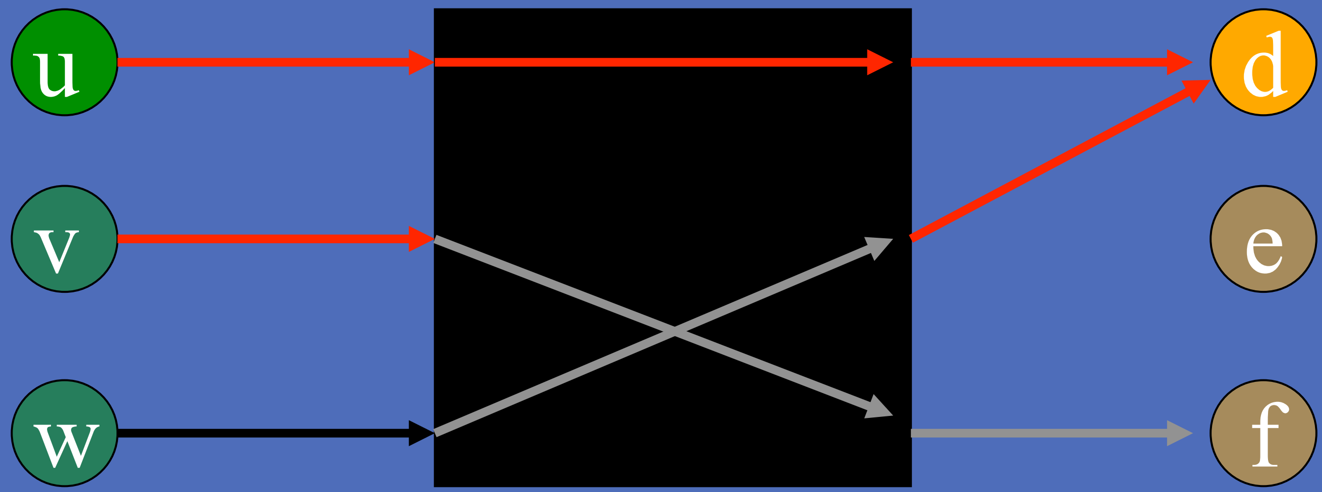
1. Users choose a destination
2. Some inputs are observed
3. Some outputs are observed

Black-box Anonymity



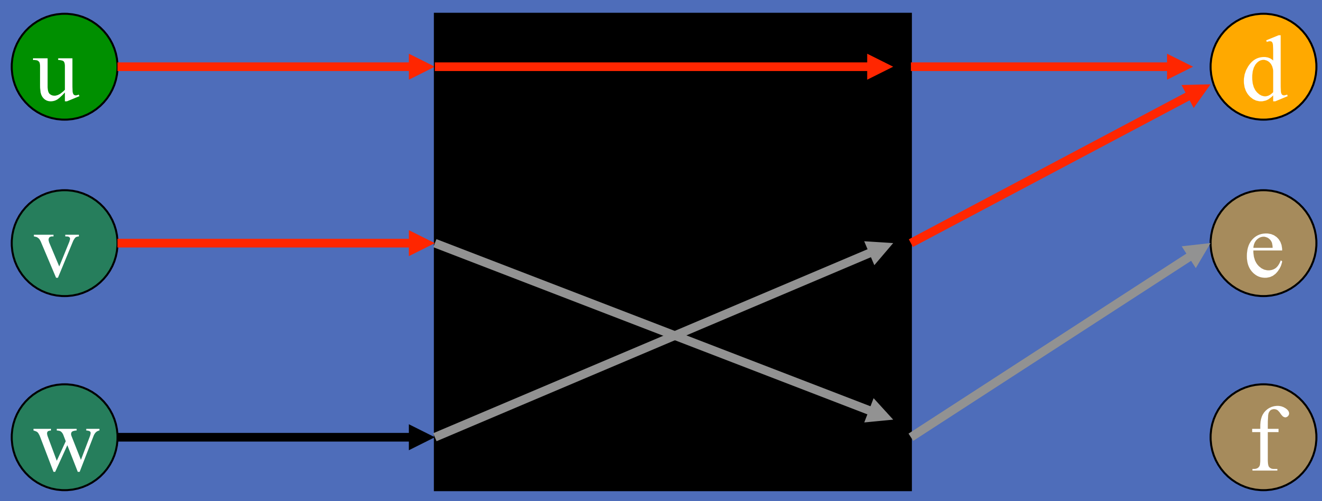
- The adversary can link observed inputs and outputs of the same user.

Black-box Anonymity



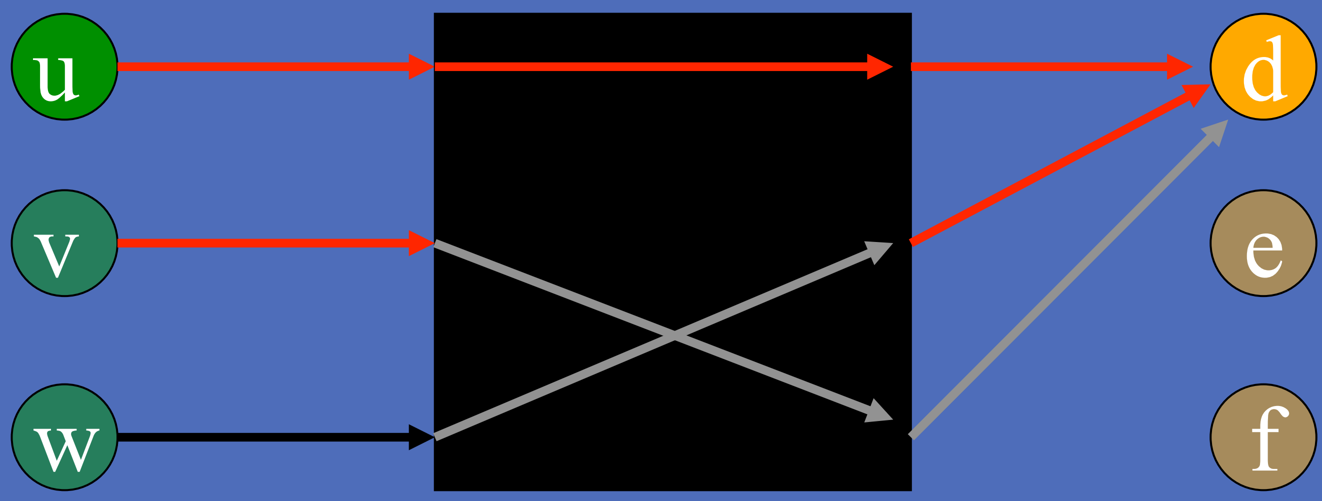
- The adversary can link observed inputs and outputs of the same user.
- Any configuration consistent with these observations is indistinguishable to the adversary.

Black-box Anonymity



- The adversary can link observed inputs and outputs of the same user.
- Any configuration consistent with these observations is indistinguishable to the adversary.

Black-box Anonymity



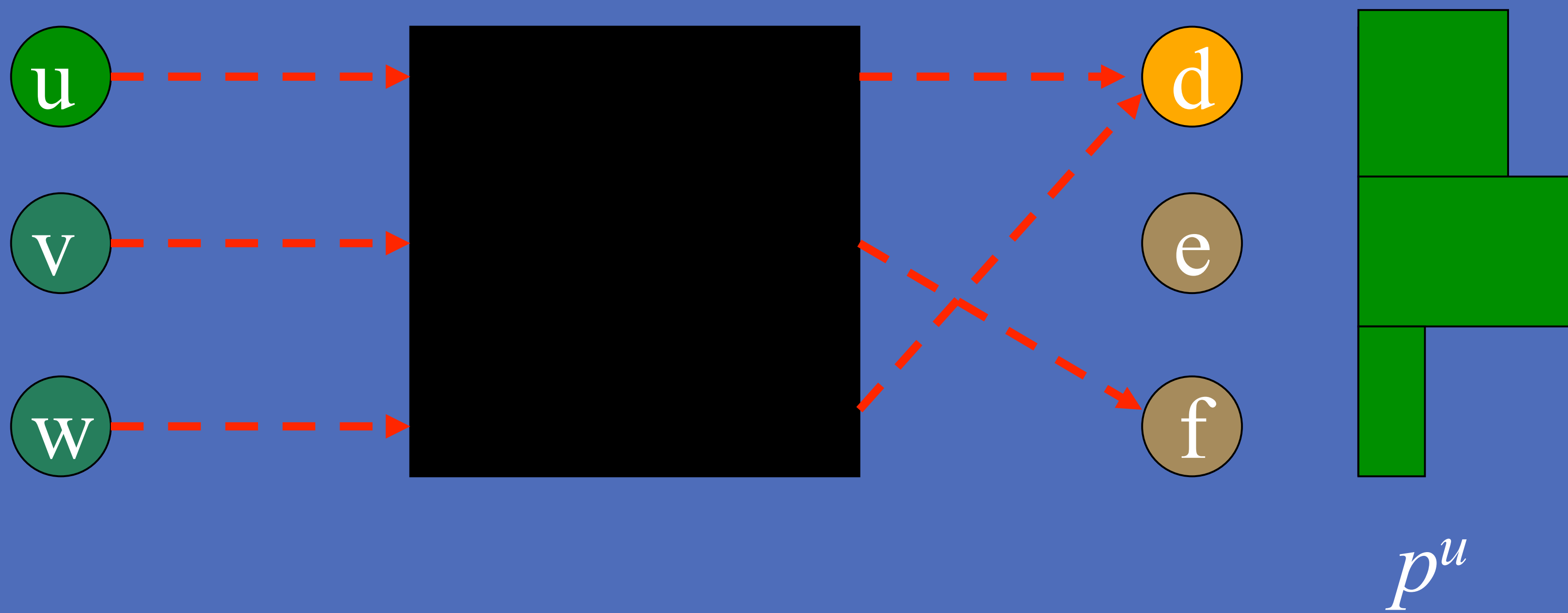
- The adversary can link observed inputs and outputs of the same user.
- Any configuration consistent with these observations is indistinguishable to the adversary.

Probabilistic Black-box

Probabilistic Black-box

- Each user *v* selects a destination from distribution p^v

Probabilistic Black-box



- Each user *v* selects a destination from distribution p^v
- Inputs and outputs are observed independently with probability *b*

Anonymity Analysis Results of Black Box

- *“Probabilistic Analysis of Onion Routing in a Black-box Model” Feigenbaum, Johnson and Syverson* WPES07
- Can lower bound expected anonymity with standard approximation: $b^2 + (1-b^2)p^u{}_d$
- Worst case for anonymity is when user acts exactly unlike or exactly like others
- Worst-case anonymity is typically as if $\sqrt{b}$ routers compromised: $b + (1-b)p^u{}_d$
- Anonymity in typical situations approaches lower bound

Formal analysis of onion routing

- [FJS07a] - Onion-routing I/O-automata model
  - Possibilistic anonymity analysis
- [FJS07b] - Onion-routing abstract model
  - Probabilistic anonymity analysis

Problem

- [FJS07a] - Onion-routing I/O-automata model
  - Possibilistic anonymity analysis
- [FJS07b] - Onion-routing abstract model
  - Probabilistic anonymity analysis
- […] - How do we apply results in standard cryptographic models?

Problem

- [FJS07a] - Onion-routing I/O-automata model
  - Possibilistic anonymity analysis
- [FJS07b] - Onion-routing abstract model
  - Probabilistic anonymity analysis
- […] - How do we apply results in standard cryptographic models?
- [CL05] - "Onion routing" formalized with Universal Composability (UC)
- "*A Formal Treatment of Onion Routing*" Camenisch & Lysyanskaya, CRYPTO 05

  - No anonymity analysis, not onion routing

Problem

- [FJS07a] - Onion-routing I/O-automata model
  - Possibilistic anonymity analysis
- [FJS07b] - Onion-routing abstract model
  - Probabilistic anonymity analysis
- […] - How do we apply results in standard cryptographic models?
- [CL05] - "Onion routing" formalized with Universal Composability (UC)
- "*A Formal Treatment of Onion Routing*" Camenisch & Lysyanskaya, CRYPTO 05

  - No anonymity analysis, not onion routing
- [FJS12] – Onion-routing UC formalization
  - "Free" probabilistic anonymity analysis

Onion-Routing UC Ideal Functionality

- *"Probabilistic Analysis of Onion Routing in a Black-box Model" Feigenbaum, Johnson and Syverson* ACM TISSEC 2012

Upon receiving destination *d* from user *U*

$$x \leftarrow \begin{cases} u \text{ with probability } b \\ \text{ø with probability } 1\text{-}b \end{cases}$$

$$y \leftarrow \begin{cases} d \text{ with probability } b \\ \text{ø with probability } 1\text{-}b \end{cases}$$

Send (x,y) to the adversary.

$\mathcal{F}_{\mathrm{OR}}$

Black-box Model

- Ideal functionality $\mathcal{F}_{\text{OR}}$
- Environment assumptions
  - Each user gets a destination
  - Destination for user u chosen from distribution p^u
- Adversary compromises a fraction b of routers before execution

UC Formalization

- Captures necessary properties of any crytographic implementation
- Easy to analyze resulting information leaks
- Functionality is a composable primitive
- Anonymity results are valid in probabilistic version of I/O-automata model

Problem

- [FJS07a] - Onion-routing I/O-automata model
  - Possibilistic anonymity analysis
- [FJS07b] - Onion-routing abstract model
  - Probabilistic anonymity analysis
- […] - How do we apply results in standard cryptographic models?
- [CL05] - "Onion routing" formalized with Universal Composability (UC)
- "*A Formal Treatment of Onion Routing*" Camenisch & Lysyanskaya, CRYPTO 05
- [FJS12] – Onion-routing UC formalization
  - "Free" probabilistic anonymity analysis
- [BGKM12] - Onion routing formalized with UC
  - Our work will provide anonymity

Ideal Functionality modeling more of reality

- "Provably Secure and Practical Onion Routing" Backes, Goldberg, Kate, and Mohammadi, IEEE CSF12
- Functionality can actually send messages
- Also presented ideal functionality covering key exchange, circuit building
  - Needs wrapper to hide irrelevant circuit-building options
- Shown to UC-emulate $\mathcal{F}_{\mathrm{OR}}$

Course Outline

- Lecture 1: Basics and Formalization
  - Usage examples, basic notions of traffic-secure communications, mixes and onion routers
  - Onion routing design basics: circuit construction protocols, network discovery
  - Formalization and analysis, possibilistic and probabilistic definitions of anonymity
- Lecture 2: Security for the real world
  - Simple demo of obtaining/using Tor
  - Security of obtain/using Tor
  - Adding network link awareness
  - Importance of modeling users
  - Importance of realistic and practical
    - Adversary models
    - Security definitions