Cristina Nita-Rotaru

# CS526: Information security

## Anonymity systems.

Based on slides by Chi Bun Chan

# 1: Terminology.

# Anonymity

> Anonymity (``without name'') means that a person is not identifiable within a set of subjects

- Unlinkability of action and identity
  - For example, sender and his email are no more related after adversary' s observations than they were before
  - Who talks to whom
- Unobservability
  - Adversary cannot tell whether someone is using a particular system and/or protocol

# Needs for anonymity

- Hiding identity
- Privacy
- Security
- Degree of innocence or deniability

# Relevant applications

- Anonymizing bulletin board and email
- Electronic voting
- Incident reporting
- Anonymous e-commerce
- Private information retrieval
- Anonymous communication

# Privacy on public networks

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

# Anonymity metrics in communication

- Basic metrics:
  - Sender anonymity - who sends what
  - Receiver anonymity - who receives what
  - Unlinkability (relationship anonymity) - who talks to whom
- Providing sender anonymity and unlinkability are desirable enough for common Internet activities
- Goals:
  - The identities of the communicating parties should stay anonymous to the outside community
  - Even the parties in communication may not know each other’s real identity

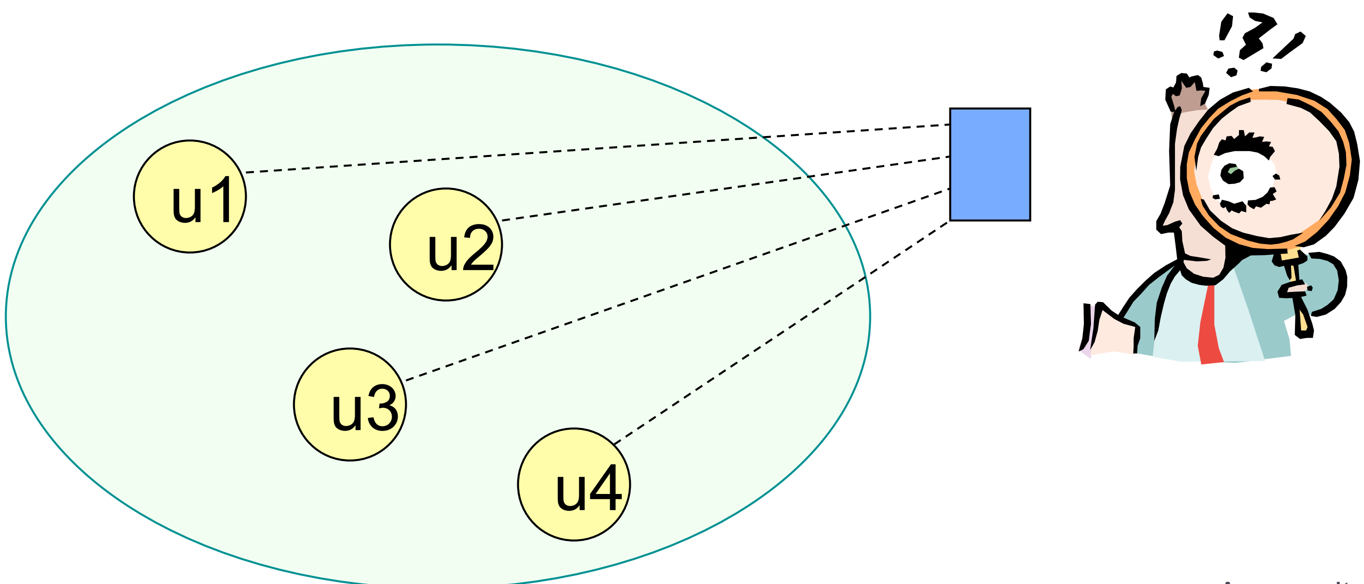
# Types of adversary

- **Passive/Active**
  - **Passive**: eavesdrop traffic
  - **Active**: able to observe, delay, alter and drop messages in the system
- **Local/Global**
  - **Local**: able to observe traffic to/form user's network link, within LAN
  - **Global**: able to observe effectively large amount or all network links, across LAN boundaries
- **Internal/External**
  - **Internal**: participants in the anonymity system, adversary-operated nodes
  - **External**: not participate in the protocol but may be able to observe, inject or modify traffic in the system

# 2: Anonymity systems.

# Anonymity set

- Hiding ones action in many others' actions
- Anonymity set: a group of users in which every one is equally-probable to be associated with a given action $\Rightarrow$ every one has certain degree of innocence or deniability to an action



# MIX-based systems

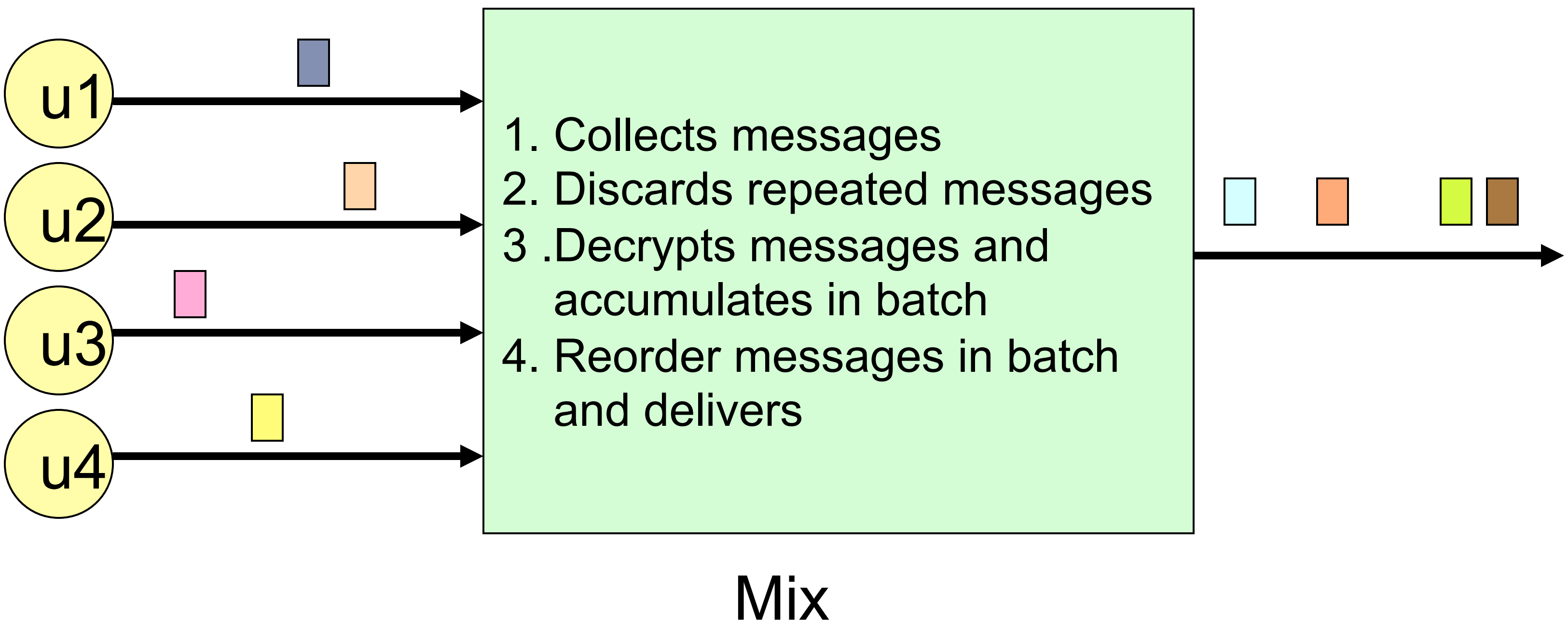
- Concept of using relay servers (MIXes) for anonymous communication
- Introduced by David Chaum (1981)
- Goals
  - Sender anonymity
  - Unlinkability against global eavesdroppers
- Idea: Messages from sender **"look"** (contents, time) differently than messages to recipient

# MIX – basic operations

- A mix is a store-and-forward relay
- Batching
  - collect fixed-length messages from different sources
  - accumulate a batch of n messages
- Mixing
  - cryptographically transform collected messages
  - forwarding messages to their recipients in random order

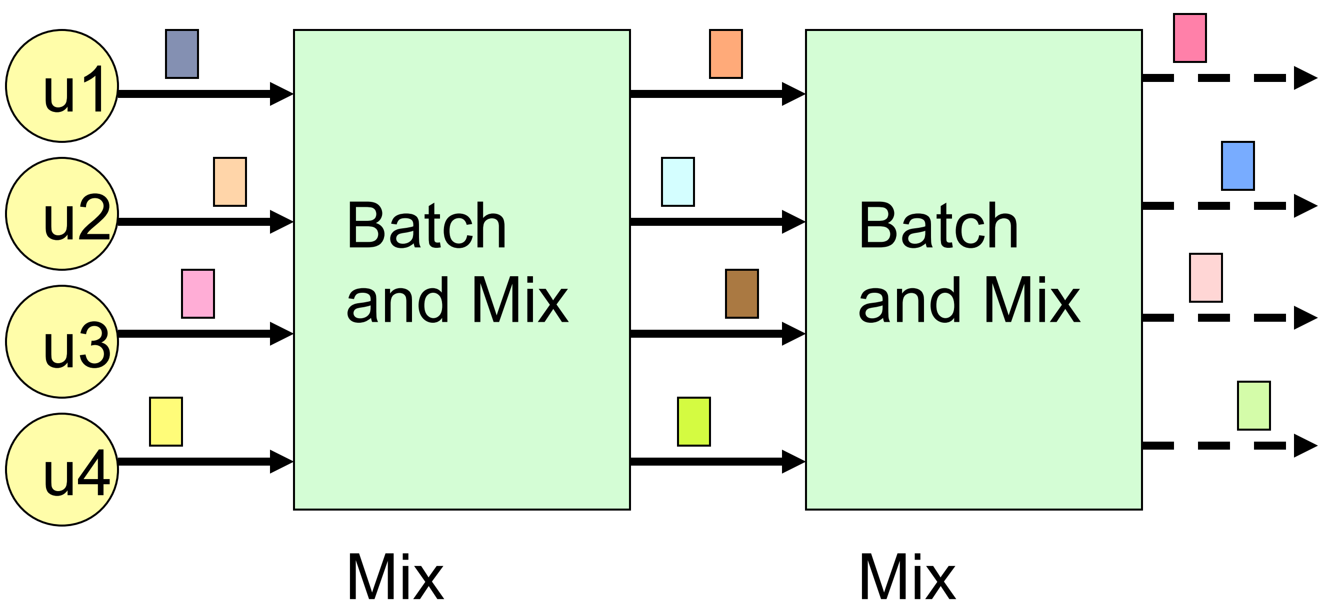
# MIX - example

- Each mix has a public key
- Each sender encrypts its message (with randomness) using public key of mix



# MIX - variants

- Single mix (also single point of trust, attack and failure)
- Mix cascade
- Mix network
- Different ways of batch and mix operations

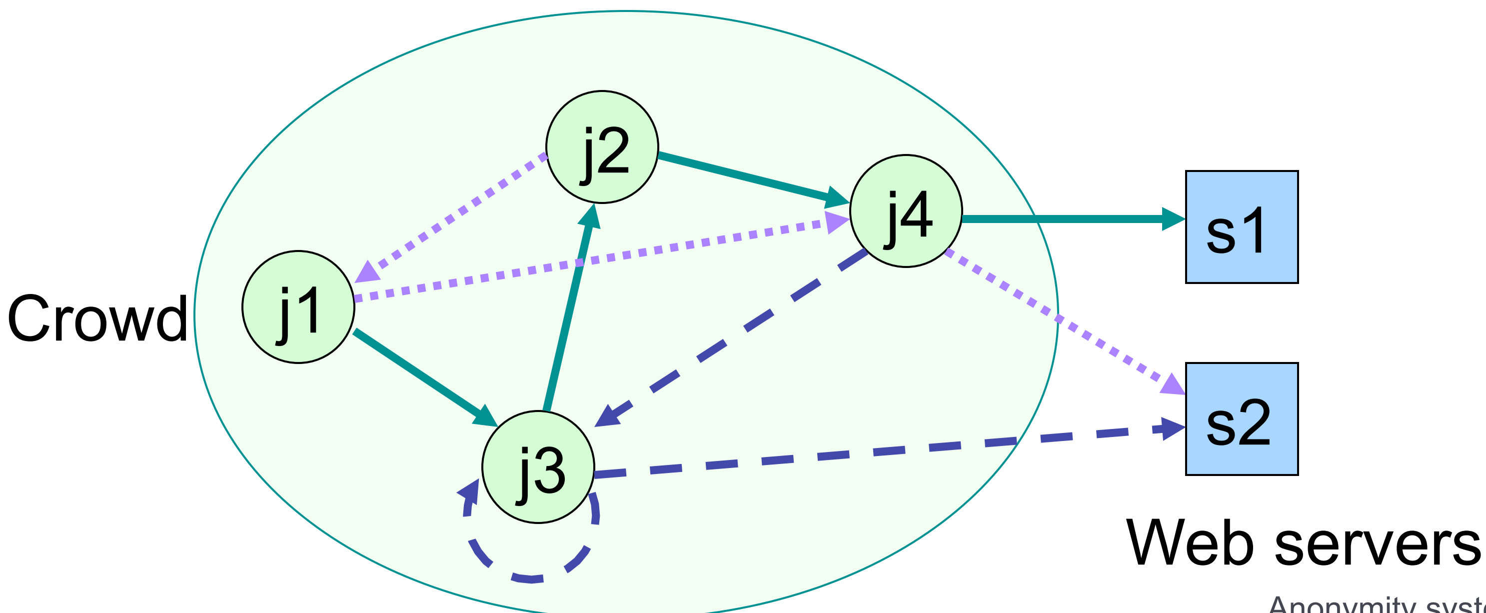


# MIX (cont.)

- Traditional designs are message-based
- Usually high latency and asynchronous due to batch and mix operations
  - may be acceptable for applications like email
  - frustrating user experience in low latency or interactive applications: web browsing, instant messaging, SSH
- Alternatives: circuit-based designs

# Crowds

- Anonymous web browsing
- Dynamic collecting users (jondo) in a group (crowd)
- Member list maintained in a central server (blender)
- Idea: Who is the initiator?



# Crowds (cont.)

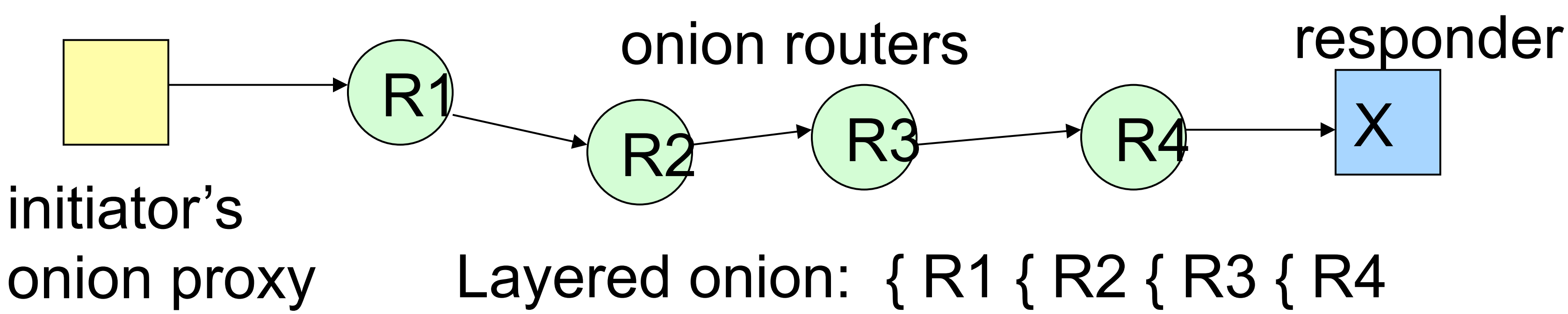
- Initiator submits request to a random member
- Upon receiving a request, a member either:
  - forwards to another random member (p = pf)
  - submits to end server (p = 1 - pf)
- A random path is created during the first request, subsequent requests use the same path; server replies using the same path but in reserve order
- Link encryption of messages with a shared key known to all members

# Onion routing

- A (small) fixed core set of relays
  - Core Onion Router (COR)
- Designed to support low-latency service
- Initiator defines an anonymous path for a connection through an "onion"
- An onion is a layered structure (recursively encrypted using public keys of CORs) that defines:
  - path of a connection through CORs
  - properties of the connection at each point, e.g. cryptographic algorithms, symmetric keys

# Onion routing (cont.)

- Initiator’s onion proxy (OP)
  - connects to COR
  - initiates a random circuit using an onion
  - converts data to fixed size cells
  - performs layered encryption, one per router
- Circuit-based multi-hop forward
  - Each COR decrypts and removes a layer of received cells, then forwards to next COR




Layered onion: { R1 { R2 { R3 { R4 { X } } } } }

# Tarzan & MorphMix

- Similar to Onion routing, Mix-net approach but extended to peer-to-peer environment
  - Again, layered/nested encryption with multi-hop forwarding
- All peers are potential message originators and relays
  - More potential relays than a small fixed core set
  - More scalable
  - Hide one's action in a large dynamic set of users
- Tarzan targets at network layer while MorphMix runs at application layer

# Tarzan & MorphMix (cont.)

- Larger dynamic set of unreliable nodes
- More efforts to defense against colluding nodes (dishonest or adversary controlled)
  - Restricted peer-selection in Tarzan
  - Collusion detection mechanism in MorphMix

# 3: Traffic analysis.

# Attacks on anonymity systems

- Degrading the quality of anonymity service
  - Break sender/receiver anonymity, unlinkability
  - Control anonymity to certain level
  - Traffic analysis, traffic confirmation

- Degrading the utilization of anonymity system
  - Decrease the performance, reliability and availability of system, so as to drive users not using the service
  - Denial-of-Service attacks

# Traffic analysis

- If one is interested in breaking the anonymity …
- Based on features in communication traffic, one may infer
  - who's the initiator $\Rightarrow$ NO sender anonymity
  - who's the responder $\Rightarrow$ NO receiver anonymity
  - an initiator-responder mapping $\Rightarrow$ NO unlinkability

# Common vulnerabilities

- Message features
  - distinguishable contents, size
- Communication patterns
  - user online/offline period
  - send-receive sequence
  - message frequencies, e.g. burst stream
- Properties/constraints in anonymity systems
  - low-latency requirement
  - link capacity and traffic shaping

# Attacks on message features

- If a message itself reveals one's identity or more, anonymity is defeated regardless of the strength of an anonymity system!

- Message features
  - size, format, writing style ..., etc

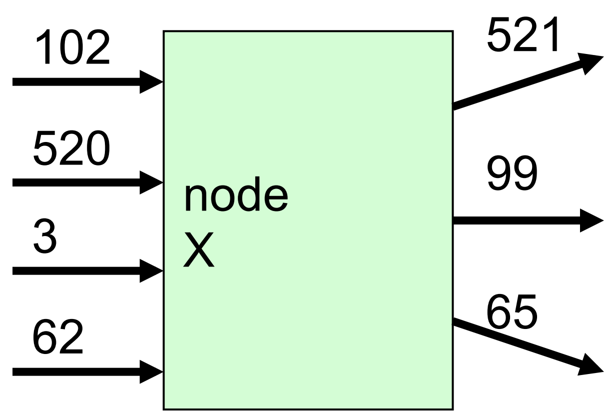
- Message size
  - Varieties of message sizes may help linking a message to some application or sender
  - Fixed by constant-size message padding

# Distinguishable message contents

- Message contents
  - may expose user information or the route of a message
  - e.g. host information, Referer, User-Agent fields in HTTP header
- Active adversary can perform message tagging attack
  - Alter bits in message header/payload
  - Recognize altered messages to exploit the route
- Solutions
  - Proper message transformation: e.g. encryption
  - Removal of distinguishable information: e.g. Privoxy (privacy enhancing proxy)

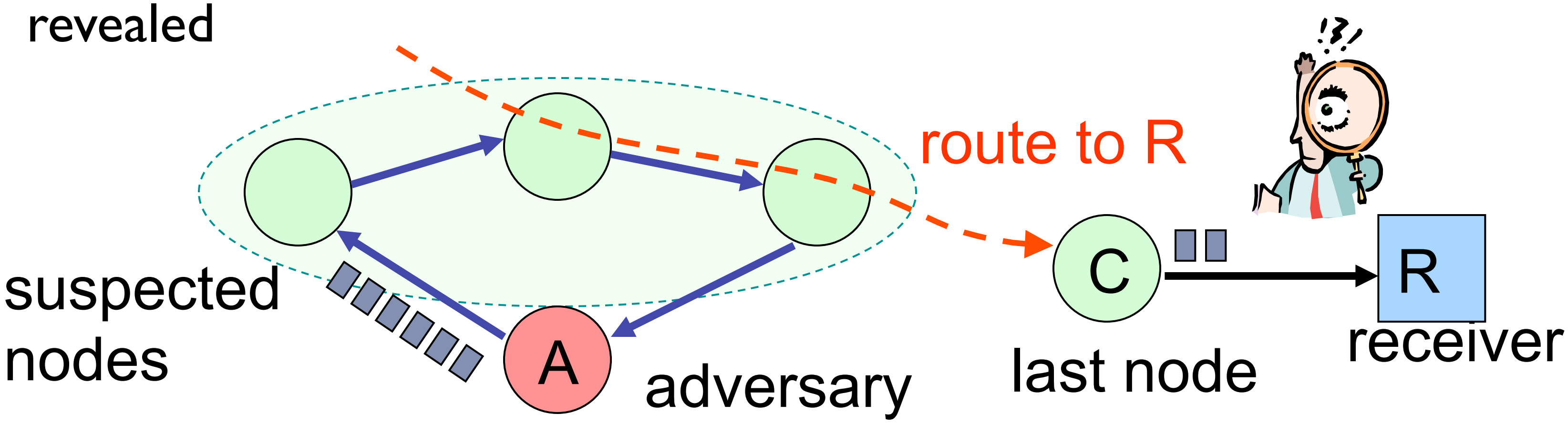
# Packet counting attack

- Count the number of messages entering a node and leaving an anonymous tunnel
- Constant link padding may help:
  - Two nodes exchange a constant number of same-sized packets per time unit
  - Generate dummy traffic on idle or lightly loaded links
  - Costly
  - Still vulnerable to other attacks, e.g. latency attacks



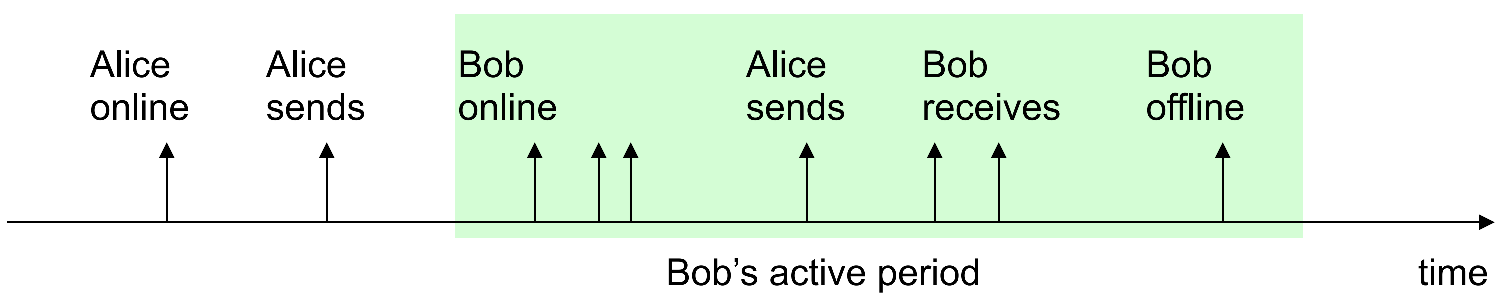
# Clogging attack

- Observe traffic between a certain last node C and end receiver R
- Create a route through a set of suspected nodes
- Clog the route with high volume of traffic
- Decrease in throughput from C to R may indicate at least one node in the suspected route belongs to a route containing C
- Continue with different sets of nodes until a route is to R is revealed



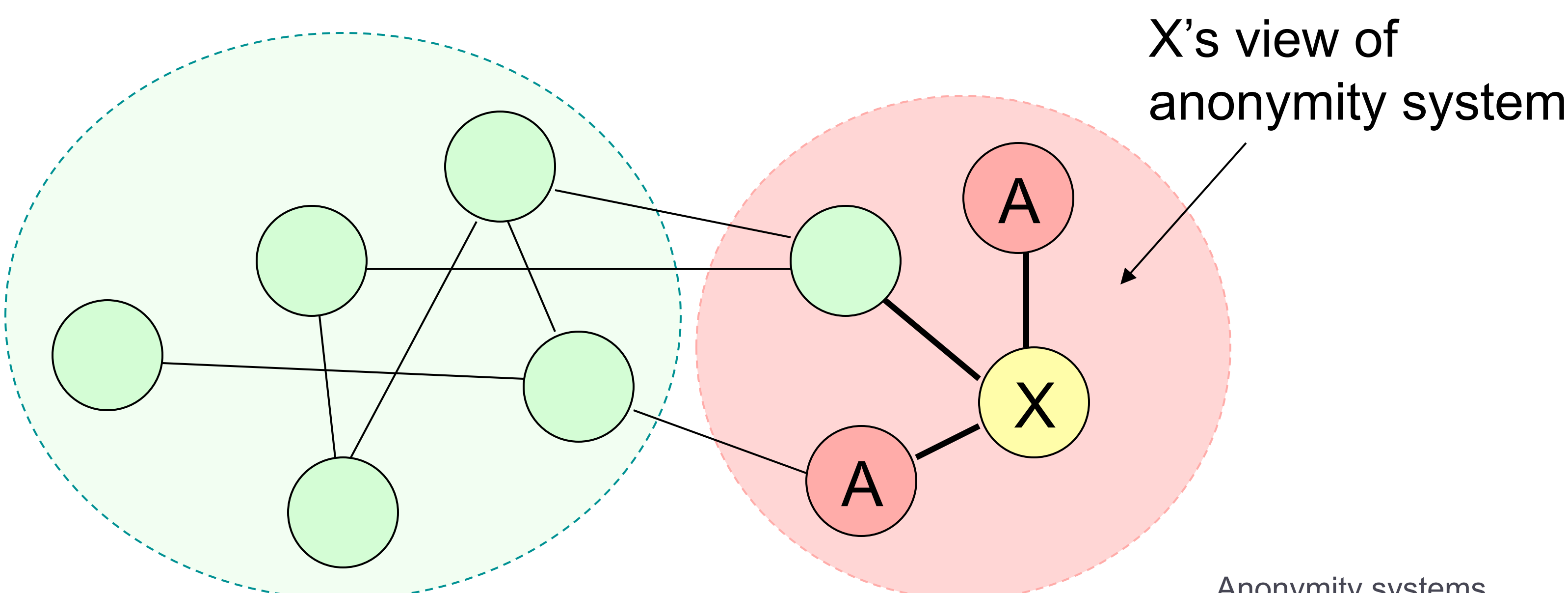
# Intersection attacks

- Communication pattern
  - Users join and leave the system from time to time
  - Users are not active in communication all the time
  - Some receivers receive messages after some senders transmit messages
- Intersecting sets of possible senders over different time periods → anonymity set shrinks
- Short term vs Long term



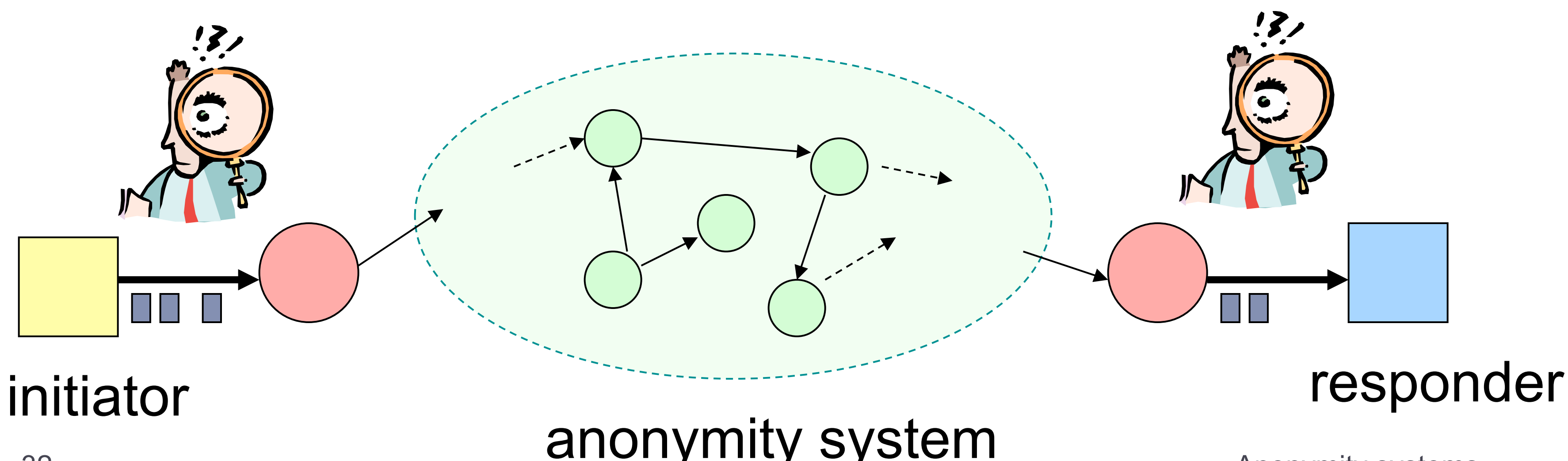
# Partition attack on client knowledge

- Render inconsistent views of anonymity system on clients
  - e.g. member list on directory server
- Identify clients who always choose a particular subset of neighbors



# Attacks on endpoints

- Sometimes referred as traffic confirmation rather than traffic analysis
- Suppose an adversary controls the first and the last node of a route
- Observe the traffic entering the first node and leaving the last node

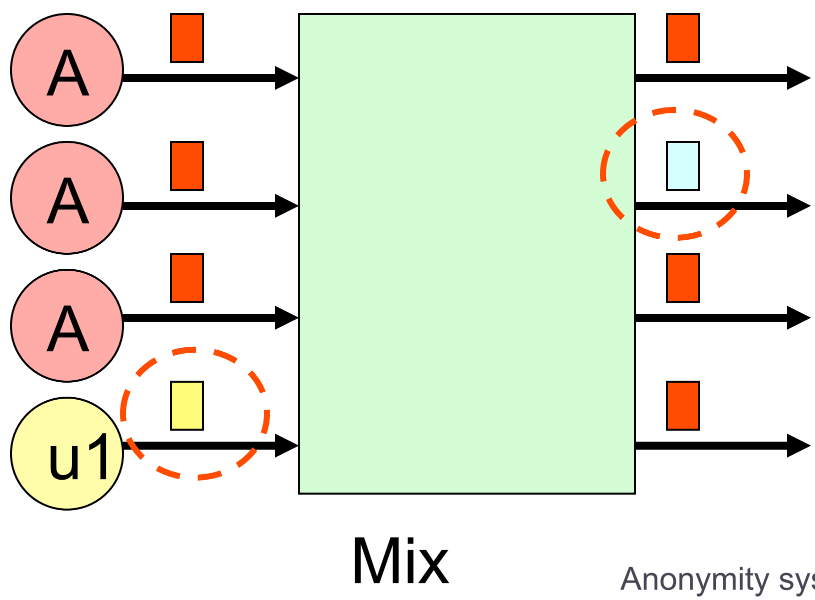


# Attacks on endpoints (cont.)

- Correlate the timings of a message entering the first node with those coming out of the last node
  - Packet counting attack, Timing attacks, Message frequency attack
- An adversary may be able to:
  - figure out some input message to output message mappings
  - rule out some potential senders or receivers from the anonymity sets
  - link a particular pair of sender and receiver
- An active adversary may increase the chance of success and speedup the analysis by delaying and dropping messages, flooding several nodes and links

# Node flushing attack

- Intended to defeat MIX-based systems
- Flooding attack, (n-1) attack
- Flood a node with identifiable fake messages but leave a room for a single message to be traced
- Link user's input message with messages leaving the node



# Trickle attack

- Trickle, flushing attack - referred as blending attack
- Suppose a MIX accumulates and emits messages in rounds
- An active attacker holds a target message until the mix emits a batch of messages
- He then submits target message to mix while blocking other incoming messages
- Only the target message is emitted in the next round

- Requires control over traffic flow - practical to launch?

# More attacks …

- The "Sting" Attack
- The "Send n' Seek" Attack
- Active Attacks Exploiting User Reactions
- Denial of Service Attack
- Social Engineering

- Alternative attack goal:
  - Drive users to less secure anonymity systems or not using anonymity service at all

# 4: Tor:The Second-Generation Onion Router

R. Dingledine, N. Mathewson, P. Syverson

Slides by Vitaly Shmatikov

# Disadvantages of Basic Mixnets

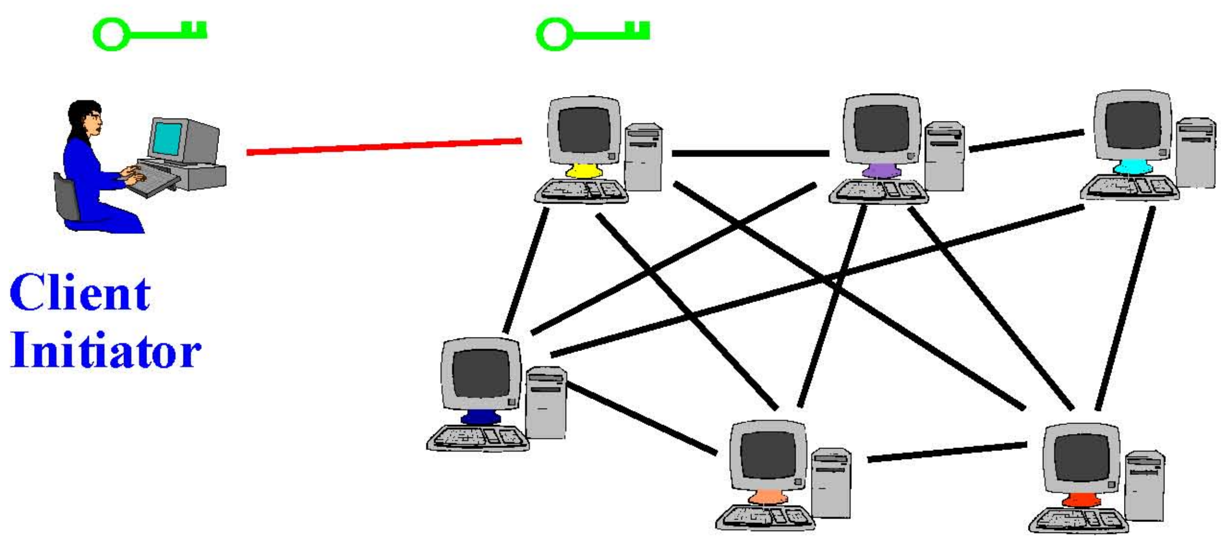
- Public-key encryption and decryption at each mix are computationally expensive
- Basic mixnets have high latency
  - Ok for email, not Ok for anonymous Web browsing
- Challenge: low-latency anonymity network
  - Use public-key cryptography to establish a "circuit" with pairwise symmetric keys between hops on the circuit
  - Then use symmetric decryption and re-encryption to move data messages along the established circuits
  - Each node behaves like a mix; anonymity is preserved even if some nodes are compromised

# Tor

- Deployed onion routing network
  - http://torproject.org
  - Specifically designed for low-latency anonymous Internet communications
- Running since October 2003
  - Thousands of relay nodes, 100K-500K? of users
- Easy-to-use client proxy,
- integrated Web browser

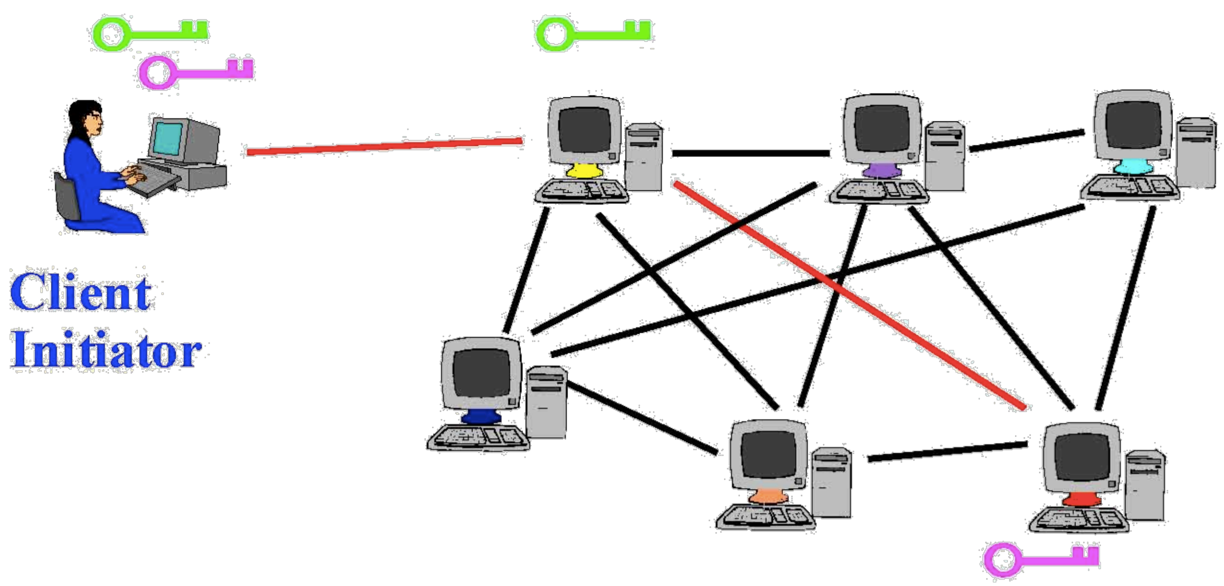
# Tor circuit setup (1)

- Client proxy establish a symmetric session key and circuit with relay node #1



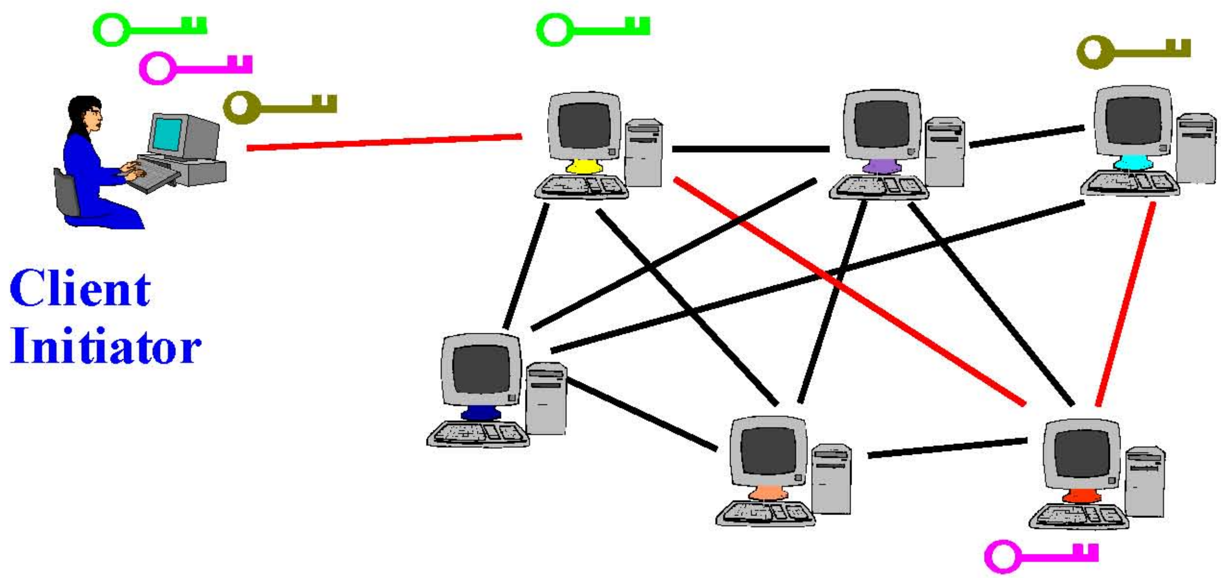
# Tor circuit setup (2)

- Client proxy extends the circuit by establishing a symmetric session key with relay node #2
  - Tunnel through relay node #1 - don't need !



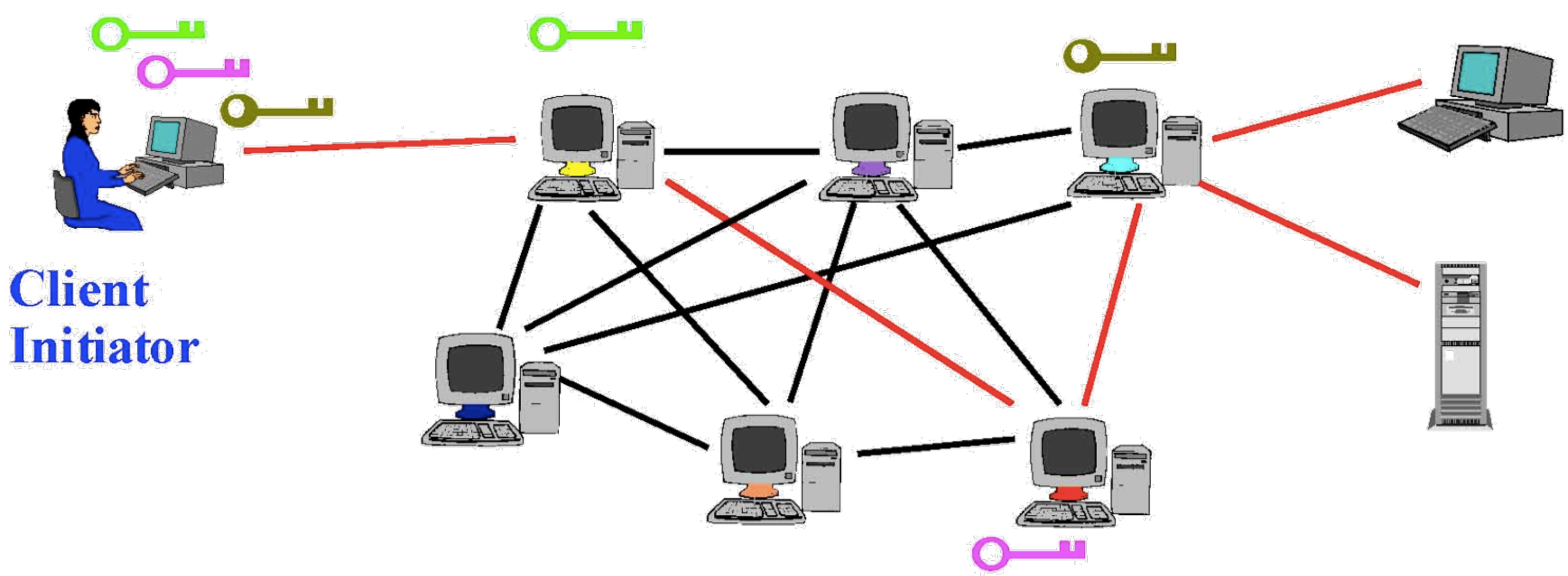
# Tor circuit setup (3)

- Client proxy extends the circuit by establishing a symmetric session key with relay node #3
  - Tunnel through relay nodes #1 and #2



# Using a Tor circuit

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



# Using Tor

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

# Passive attacks

- Observe Traffic Patterns
  - Multiplexing minimizes damage
- Observe User Content
  - Use of Privoxy
- Option Distinguishability
  - Leads to tracing due to distinct pattern behavior
- End-to-end Timing Correlation
  - Tor does not hide timing (low-latency requirement)
- End-to-end Size Correlation
  - Leaky-Pipe Topology
- Website Fingerprinting
  - New attack as of 2004, semi-defended by mitigation

# Active attacks

- Compromise Keys
  - Mitigated by key rotation and redundant multiple layer encryption. Replacing a node via identity key could theoretically avoid this defense.
- Iterated Compromise
  - Short lifetimes for circuits
- Run Recipient
  - Adversary controls end server, which allows him to use Tor to attack the other end. Privoxy would help minimize chance of revealing initiator
- Run Onion Proxy
  - Compromised OPs compromise all information sent through OP
- DoS non-observed nodes
  - Only real defense is robustness
- Run hostile OR
  - Requires nodes at both ends of a circuit to obtain information
- Introduce Timing
  - Similar to timing discussed in passive version

# Active attacks (cont.)

- Tag Attacks
  - Integrity check mitigates this
- Replay Attacks
  - Session key changes if replay used
- Replace End Server
  - No real solution, verify that server is actually server with authentication. Similar to Recipient attack
- Smear Attacks
  - Good press and exit policies
- Hostile Code Distribution
  - All Tor releases signed

# Directory subversion

- Destroy Servers
  - Directories require majority rule, or human intervention if more than half destroyed.
- Subvert Server
  - At worst, cast tie-breaker vote
- Subvert Majority of Servers
  - Ensure Directories are independent and resistant to attacks
- Encourage Dissent in Directory Operators
  - People problem, not Tor problem.
- Trick Directories
  - Server Operators should be able to filter out hostile nodes.
- Convince Directories that OR is Functional
  - Directory servers should test by building circuit and streams to OR.

# Rendezvous point attacks

- Many Introduction Point Requests
  - IP can block requests with authorization tokens, or require certain amounts of computation per request.
- Attack Introduction Point
  - Server re-advertises on different IP, or advertise secretly. Attacker must disable all IPs.
- Compromise Introduction Point
  - Servers should occasionally verify their IPs, and close circuits that flood them.
- Compromise Rendezvous Point
  - Similar to active attacks against ORs