Anonymity Systems and Traffic Analysis

Presented by Chi Bun Chan
on April 15, 2004

Outline

- Brief description of some anonymity system designs
- Summary of several traffic analysis techniques

Needs for Anonymity

- Hiding Identity
  - Sensitive issues, political reasons, secret operations
  - Freedom of speech
- Privacy
  - Human right, Corporation benefits
  - Against surveillance, private information tracking and profiling
- Security
  - Hiding actual servers, existence of virtual private network
  - Transfer or "load-balance" attacks to some other relays (?)
- Anonymity offers certain degree of innocence or deniability to an action. Is it good or bad?

Relevant Applications

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

We do have …

- Data Confidentiality
  - Encryption schemes (symmetric, public-key)
- Data Integrity
  - Secure Hashing, HMAC
- Authentication
  - Digital signature, certificate, Kerberos

- Data confidentiality + data integrity + authentication → **not enough** to guarantee anonymity
- Trivial example: If there is only one guy sending a message to another guy, encryption doesn't help.

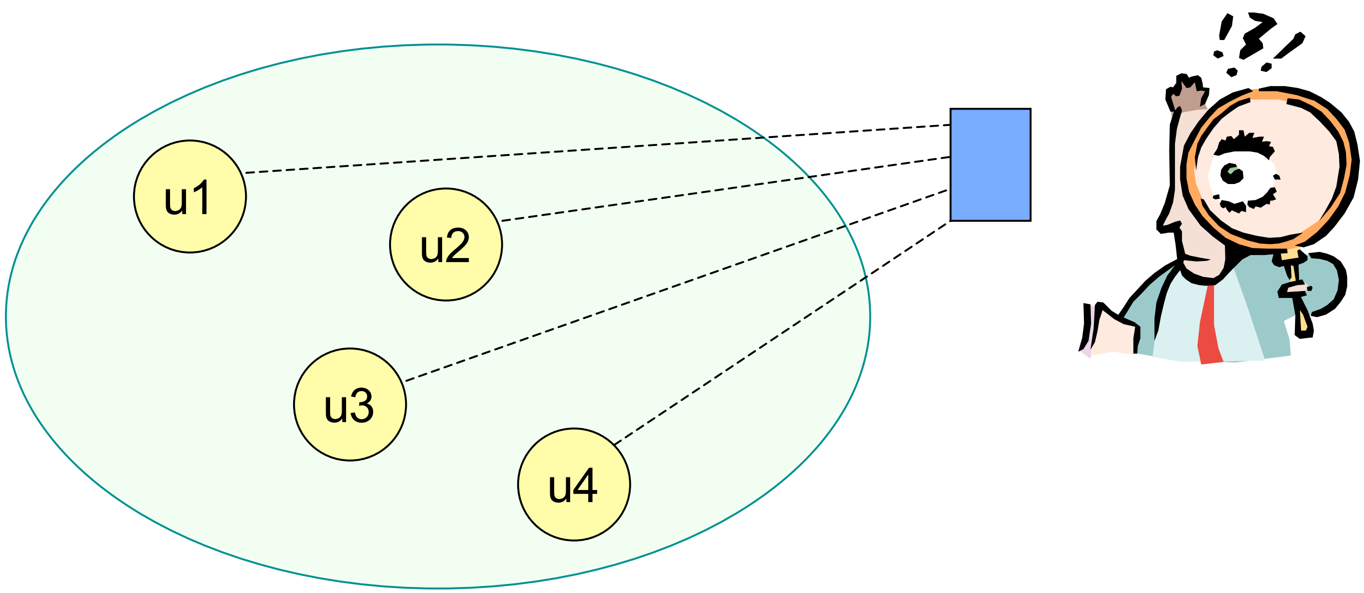
Anonymity Metrics in Communication

- 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

Anonymity Systems

Anonymity Set

- Hiding one's action in many others' actions
- **Anonymity set** - a group of users in which every one is equal-probable to be associated with a given action
   every one has certain degree of innocence or deniability to an action



MIX-based Systems

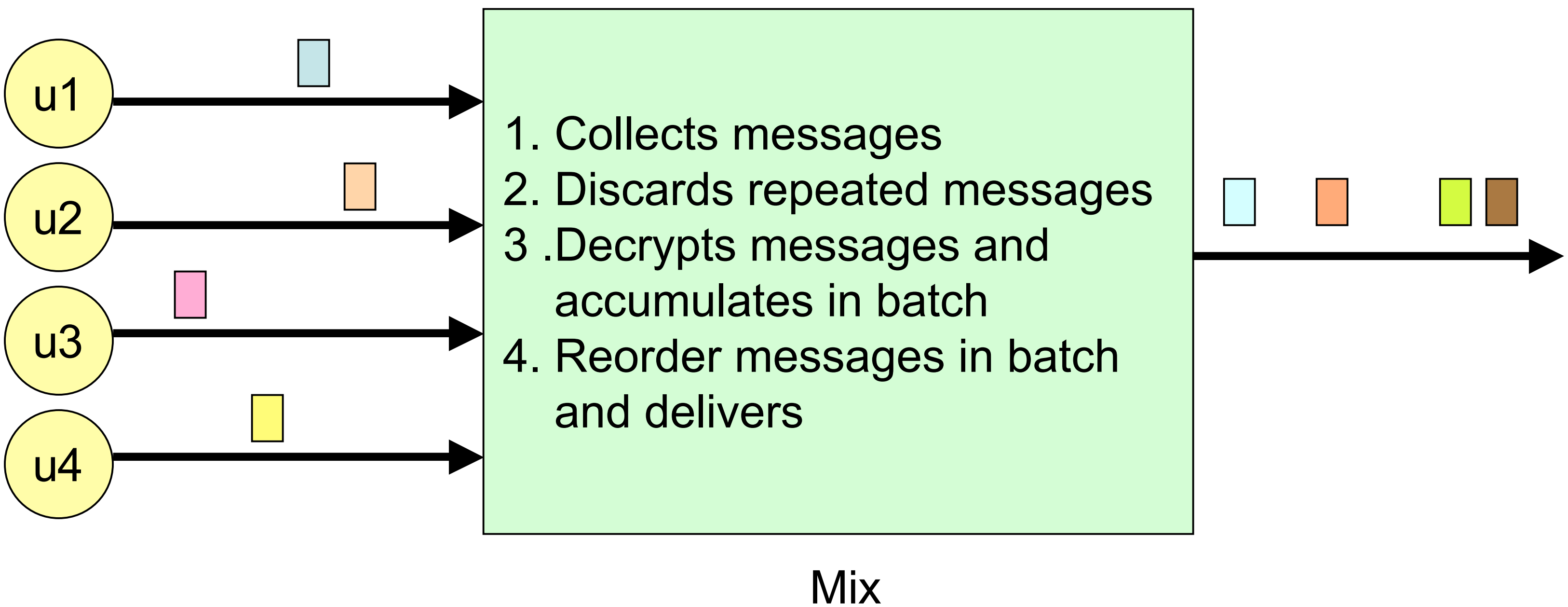
- Concept of using relay servers (MIXes) for anonymous communication
- Introduced by David Chaum (1981)
- Goals
  - Sender anonymity
  - Unlinkability against global eavesdroppers
- Idea: Messages from sender **“look”** (contents, time) differently than messages to recipient

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**

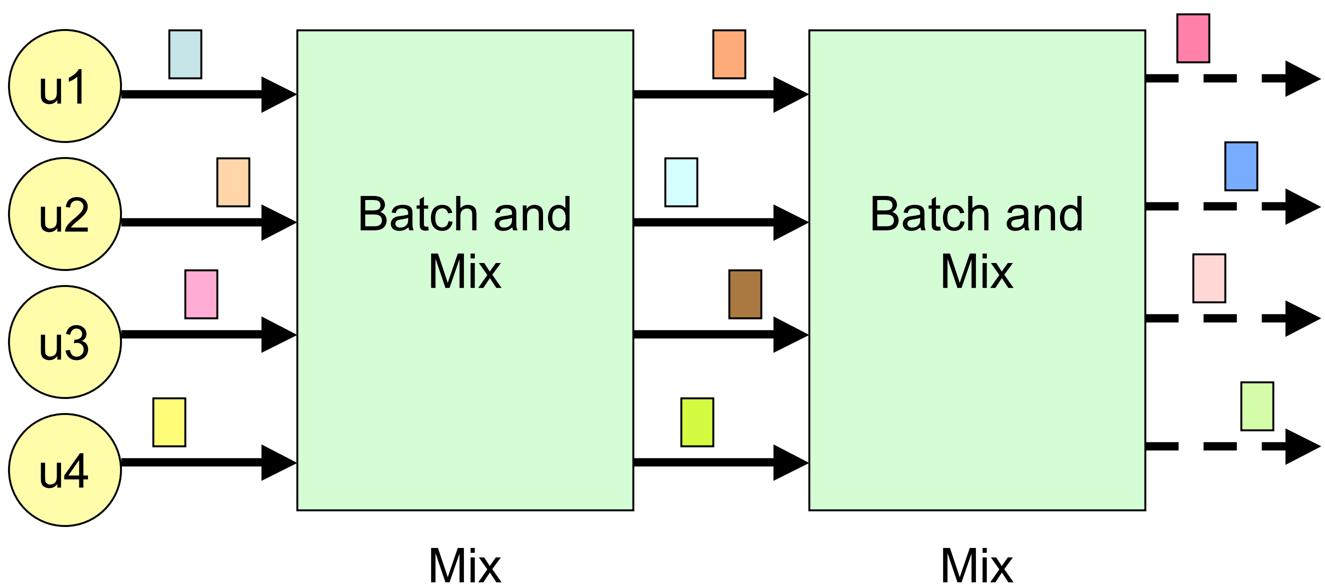
MIX - Example

- 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

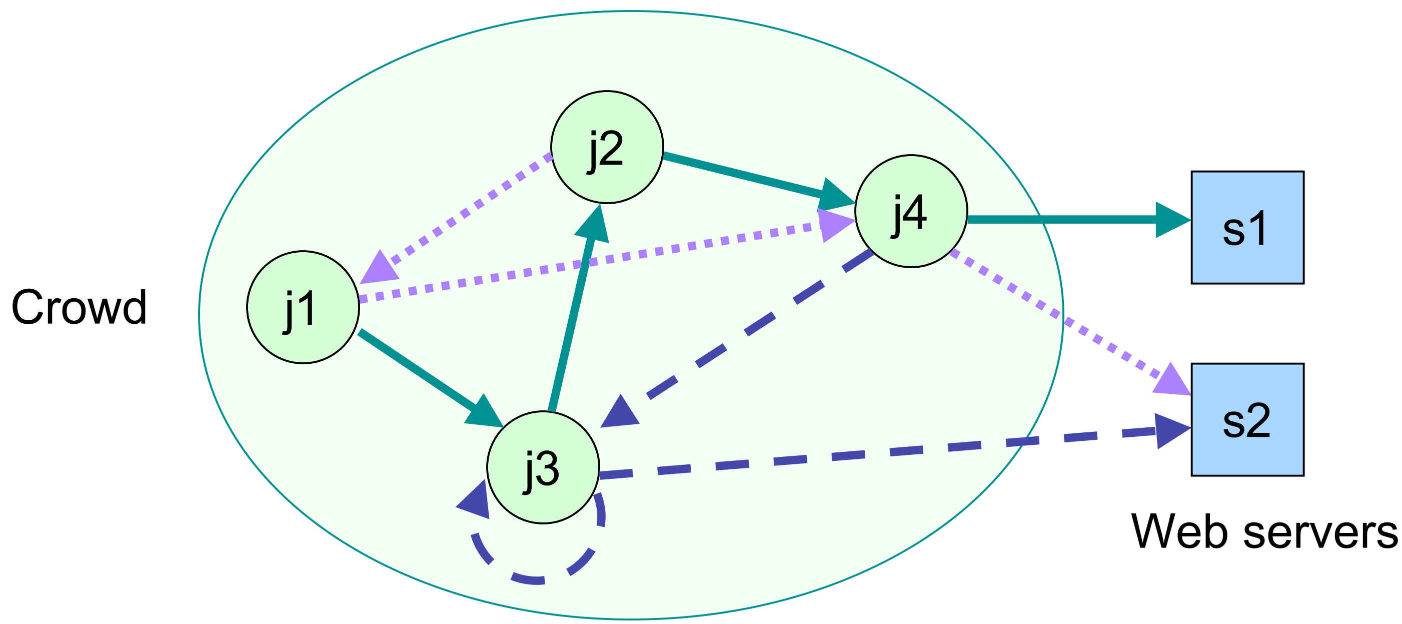


MIX (cont.)

- 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?



Crowd (cont.)

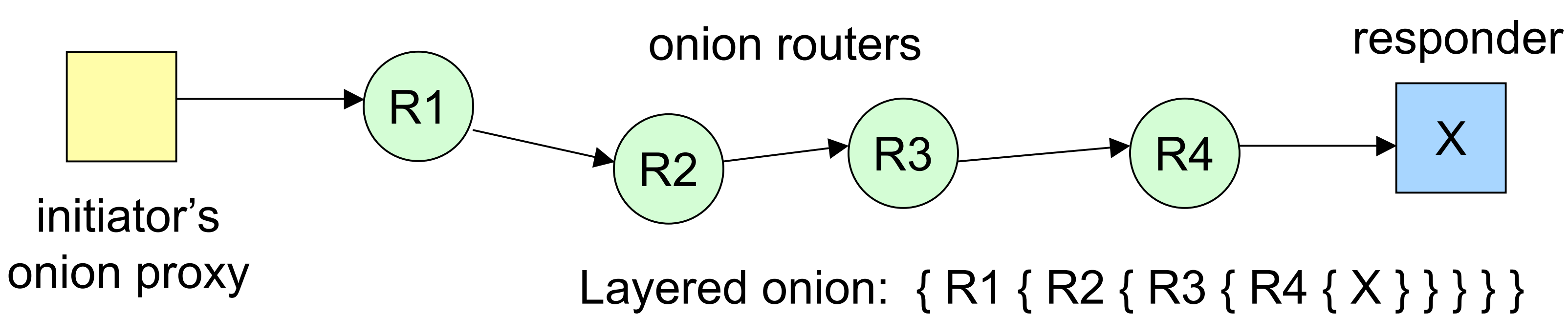
- Initiator submits request to a **random member**
- Upon receiving a request, a member either:
  - forwards to another random member **($p = p_f$)**
  - submits to end server ($p = 1 - p_f$)
- a random path is created during the first request, subsequent requests use the same path; server replies using 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



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

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’s interested in breaking the anonymity …
- Based on features in communication traffic, one may infer
  - who’s the initiator NO sender anonymity
  - who’s the responder NO receiver anonymity
  - an initiator-responder mapping NO unlinkability

Types of Adversary

- **Passive**: eavesdrop traffic
- **Active**: able to observe, delay, alter and drop messages in the system
- **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**: 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

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

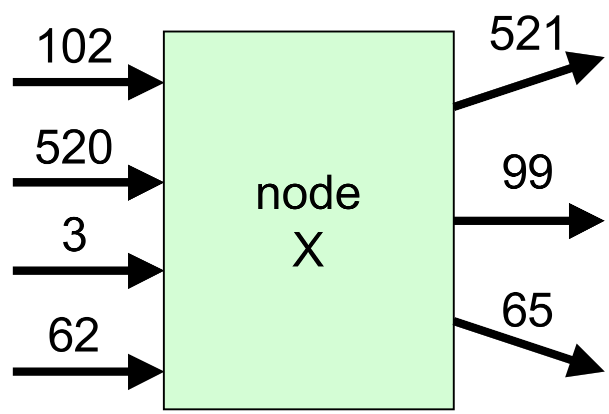
- Message size
  - Varieties of message sizes may help linking a message to some application or sender
  - Fixed by constant-size message padding

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)

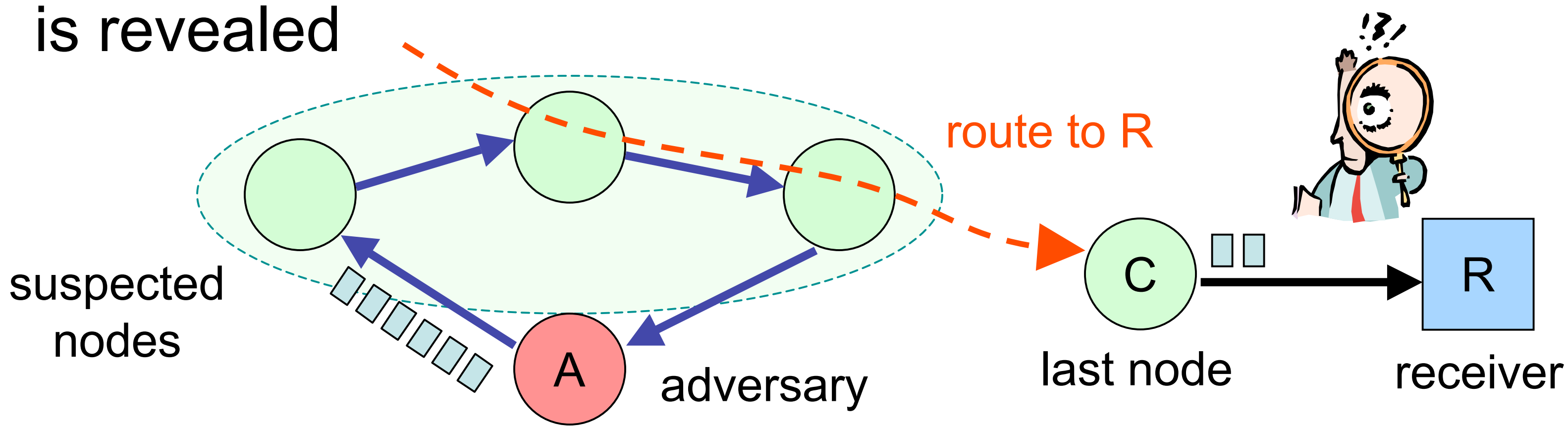
Packet Counting Attack

- 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



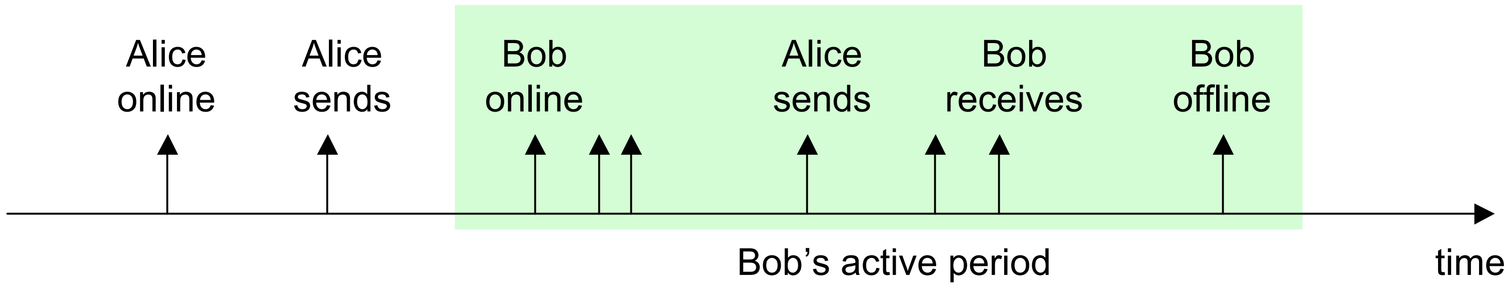
Clogging Attack

- 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



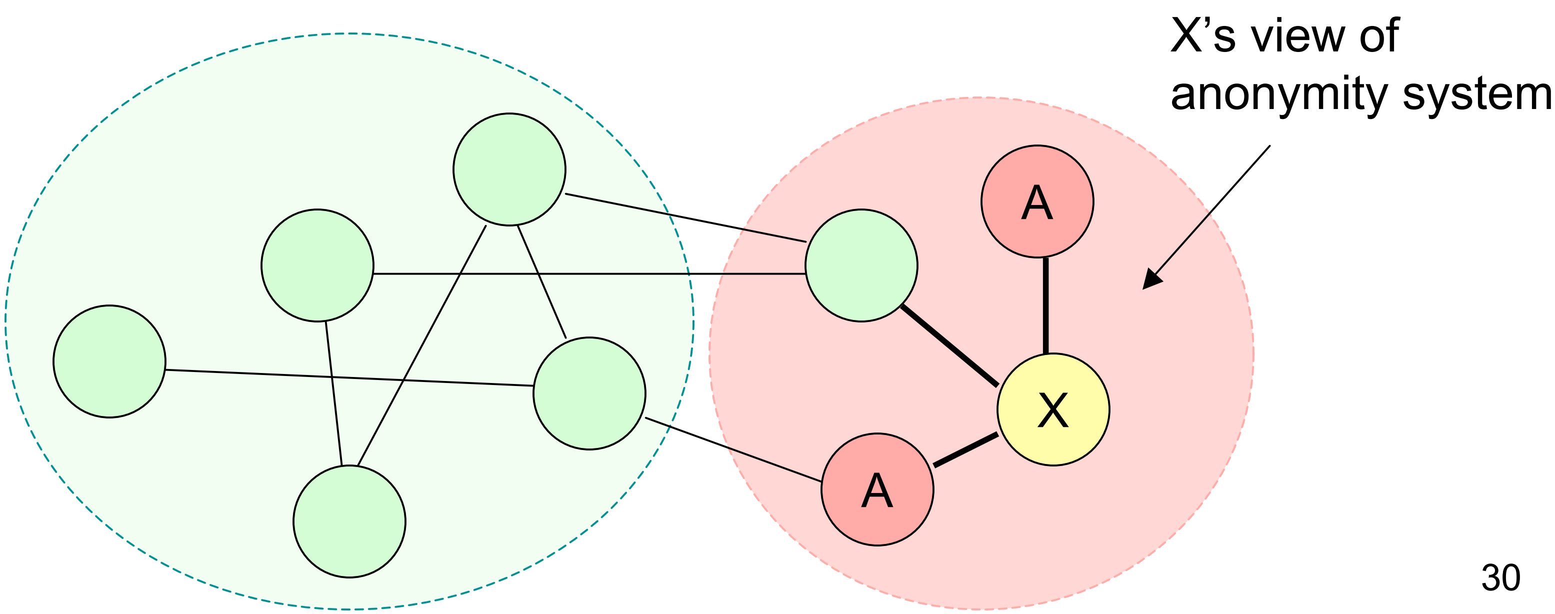
Intersection Attacks

- 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



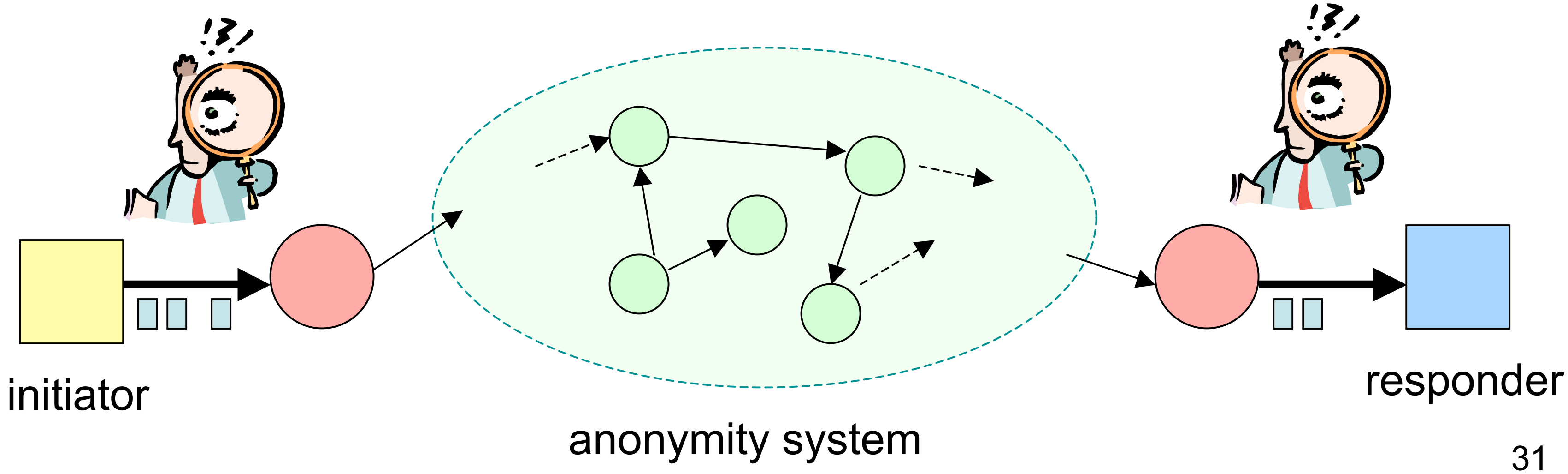
Partition Attack on Client Knowledge

- 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

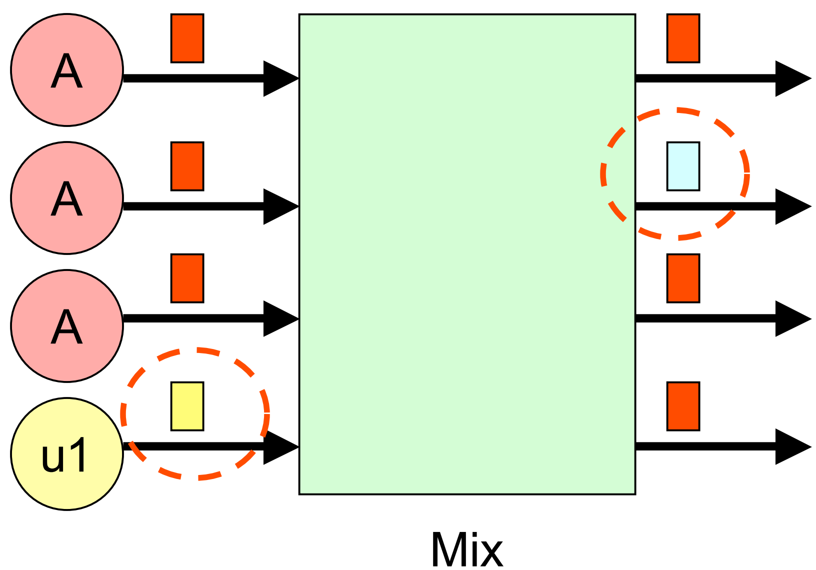


Attacks on Endpoints (cont.)

- 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

Open Questions

- More users (relays) means better?
  - P2P approaches - more scalable?
  - high dynamicity can be good or bad
  - prevent adversaries from signing up many colluding nodes

- Every traffic should look the same?
  - cover traffic? Constant link padding?
  - effectiveness and performance

- It's a matter of tradeoff!

References

- Jean-François Raymond. **Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems**. *In the Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability,* July 2000, pages 10-29.

- Adam Back, Ulf Möller, and Anton Stiglic. **Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems**. *In the Proceedings of Information Hiding Workshop (IH 2001)*, April 2001, pages 245-257.

- Michael Reiter and Aviel Rubin. **Crowds: Anonymity for Web Transactions**. *In ACM Transactions on Information and System Security*, June 1998

References

- Michael J. Freedman and Robert Morris. **Tarzan: A Peer-to-Peer Anonymizing Network Layer**. *In the Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002),* Washington, DC, November 2002.

- Marc Rennhard and Bernhard Plattner. **Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection**. *In the Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002)*, Washington, DC, USA, November 2002.

- **Onion Routing**. http://www.onion-router.net/