


PURDUE UNIVERSITY

Computer Security

CS 426

Lecture 27

Anonymity and Traffic Analysis




Elisa Bertino
Purdue University
IN, USA
bertino@cs.purdue.edu
Presented by: Nabeel Mohamed


Center for Education and Research in Information Assurance and Security

PURDUE UNIVERSITY

Raising Hands



Are you aware that Lady Gaga is coming to Purdue in January (and the tickets are going on sale today!)?



PURDUE UNIVERSITY


Raising Hands



Now imagine that you had the **HIV testing** at PUSH. Are you willing to tell others?



PURDUE UNIVERSITY



Needs for Anonymity

- Hiding Identity
 - Sensitive issues, political reasons, secret operations
 - Freedom of speech
- Privacy
 - Human rights, Corporation benefits
 - Against surveillance, private information tracking and profiling
- Security
 - Hiding actual servers, existence of virtual private network
- Anonymity offers *certain degree of innocence or deniability to an action.*



Is more Anonymity Good?

- It's a double-edged sword!
- How can we make it one sided? Is it possible?
- Accountability vs. anonymity
 - Opposite goals
 - Can we achieve best of both
 - Some thoughts on this: <http://mohamednabel.blogspot.com/2009/10/accountability-or-anonymity-or-can-we.html>
- Other trade-offs?
 - Utility, computational cost, time, etc.
- We *focus only on good things* in this lecture!



Relevant Applications

- Anonymizing bulletin board and email
- Electronic voting
- Incident reporting
- Anonymous e-commerce
- Private information retrieval
- Anonymous Publishing
- Data Anonymization
- We focus only on **anonymity in communication!**



Anonymity

- 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 party sending a message to another party, encryption does not help.



Anonymity Metrics in Communication

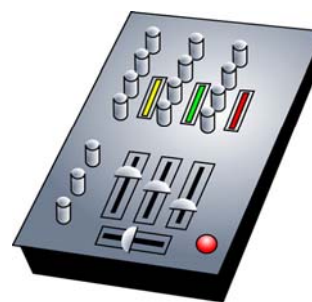
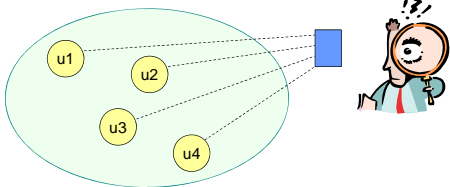
- Basic metrics:
 - **Sender anonymity** - who sends what
 - **Receiver anonymity** - who receives what
 - **Unlinkability** (relationship anonymity) - who talks to whom
- Providing *sender anonymity* and *unlinkability* are desirable enough for common Internet activities
- Goals:
 - The identities of 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

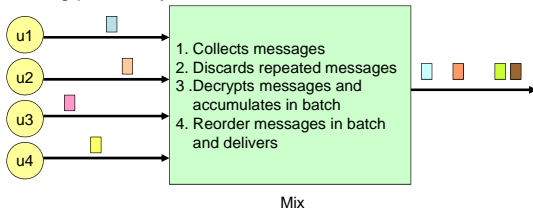


MIX - Basic Operations

- 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**
- Adversary knows all senders and receivers but cannot link a sent message with a received message

MIX - Example

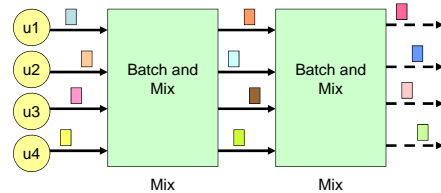
- Each mix has a public key
- Each sender encrypts its message (with randomness) using public key of mix



u1 sends a message M to P via MIX - steps
 $u1 \rightarrow \text{MIX} : K_{\text{MIX}} [R1, K_p (R0, M), P]$
 $\text{MIX} \rightarrow P : K_p (R0, M)$

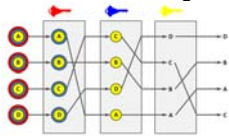
MIX - Variants

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



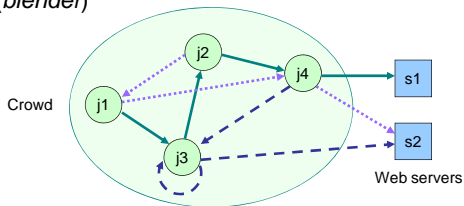
MIX (cont.)

- 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*)



Crowd (cont.)

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

Note: for query privacy in browsers; check TrackMeNot Firefox plug-in



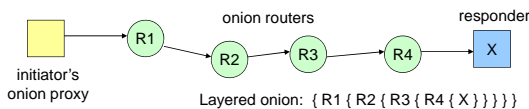
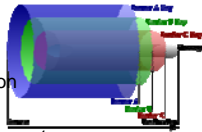
Onion Routing



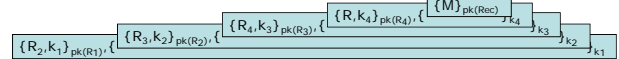
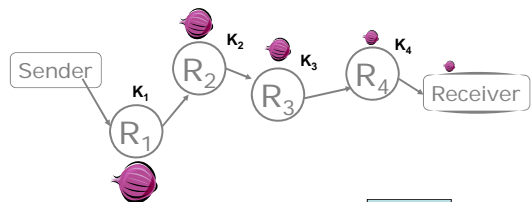
- The basic idea is to **randomize the routing**
- Routers don't know for sure if the apparent source of a message is the true sender or another router
- A (small) fixed core set of relays
 - **Core Onion Router (COR)**
- Designed to support low-latency service
- Initiator defines an anonymous path for 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



Route Establishment



- Routing info for each link encrypted with router's public key
- Each router learns only the identity of the next router

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

Tarzan: A Peer-to-Peer Anonymizing Network Layer, CCS 2002
 Introducing MorphMix: peer-to-peer based anonymous Internet usage with collusion detection, SIGSAC 2002

Tor



- www.torproject.org
- A popular implementation of onion routing
- An open/free anonymous network



Tor: The Second-Generation Onion Router, USENIX 2004

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 systems
 - 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 \Rightarrow NO sender anonymity
 - who's the responder \Rightarrow NO receiver anonymity
 - an initiator-responder mapping \Rightarrow 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/from 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

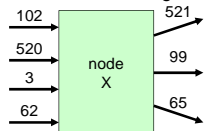
- If a message itself reveals one's identity or more, anonymity is defeated regardless of the strength of an anonymity system!
- Message features
 - size, format, writing style ..., etc
- Message size
 - Varieties of message sizes may help linking a message to some application or sender
 - Fixed by constant-size 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)

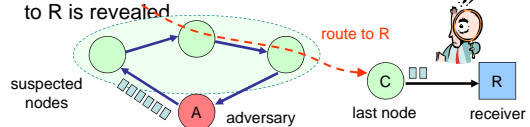
Packet Counting Attack

- 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 (of being on the path to R) 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

