Anonymous Communications: Point-to-Point

Giulia Fanti
Fall 2019
Based in part on slides by Anupam Datta, Piotr Mardziel
Administrative

• HW4 due Nov. 22 (<2 weeks from now)
  • “Fairness in Classification” problem updated on Canvas

• Recitation on Friday (Sruti)
  • New location tomorrow: Panther Hollow, CIC 4101
  • Anonymous communication

• If you want feedback on your project, please come to OH!
In-class Quiz

• On Canvas
In-class activity: DC Nets

• One of you has been selected from each class as the “rep”

• If you are the rep and you are present in class, your goal is to broadcast a “1”
# Shamir secret sharing vs. DC nets

<table>
<thead>
<tr>
<th></th>
<th>Shamir Secret Sharing</th>
<th>DC Nets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
<td>Hide message contents from any set of $&lt; k$ participants</td>
<td>Hide message author</td>
</tr>
<tr>
<td><strong>Approach</strong></td>
<td>Split data into shares that can only be reconstructed by $k+$ parties</td>
<td>Mask message through shared randomness that it not visible to all parties</td>
</tr>
<tr>
<td><strong>Strengths</strong></td>
<td>Information-theoretic secrecy</td>
<td>Information-theoretic anonymity</td>
</tr>
<tr>
<td><strong>Weaknesses</strong></td>
<td>Requires a lot of randomness</td>
<td>Requires a lot of randomness</td>
</tr>
<tr>
<td></td>
<td>High communication cost</td>
<td>High communication cost</td>
</tr>
<tr>
<td></td>
<td>Fragile to collisions</td>
<td>Fragile to collisions</td>
</tr>
</tbody>
</table>
Why are DC nets more fragile than secret sharing?

• In DC nets the signal is only being injected in one of the observed symbols
  • Corruptions in that symbol cannot be recovered
    • No redundancy
  • Problem is more difficult
    • Similar adversarial model
    • More general task

• State-of-the-art DC-nets can support ~1,000 nodes
Overview of the Unit

1. One-to-many communication

2. Point-to-point communication
Privacy on Public Networks

- Internet is designed as a public network
  - Nearby computers can see your traffic
  - Routers see all traffic that passes through them

- Routing information is public
  - Packet headers identify source and destination
  - Even a passive observer can easily figure out who is talking to whom

- Encryption does not hide identities (e.g. DNSSEC, HTTPS)
  - Encryption hides payload, but not routing information
Applications of Anonymity (I)

• Privacy
  • Hide online transactions, web browsing, etc. from intrusive governments, marketers and archivists

• Untraceable electronic mail
  • Corporate whistle-blowers
  • Political dissidents
  • Socially sensitive communications (online AA meeting)
  • Confidential business negotiations

• Law enforcement and intelligence
  • Sting operations and honeypots
  • Secret communications on a public network
Applications of Anonymity (II)

• Digital cash
  • Electronic currency with properties of paper money (online purchases unlinkable to buyer’s identity)

• Anonymous electronic voting

• Censorship-resistant publishing
What is Anonymity?

• Anonymity is the state of being not identifiable within a set of subjects
  • You cannot be anonymous by yourself!
  • Hide your activities among others’ similar activities

• Unlinkability of action and identity
  • For example, sender and his email are no more related after observing communication than they were before

• Unobservability (hard to achieve)
  • Any item of interest (message, event, action) is indistinguishable from any other item of interest
Attacks on Anonymity

- **Passive traffic analysis**
  - Infer from network traffic who is talking to whom
  - To hide your traffic, must carry other people’s traffic!

- **Active traffic analysis**
  - Inject packets or put a timing signature on packet flow

- **Compromise of network nodes**
  - Attacker may compromise some routers
  - It is not obvious which nodes have been compromised
    - Attacker may be passively logging traffic
  - Better not to trust any individual router
    - Assume that some fraction of routers is good, don’t know which
Outline: Point-to-Point Communication

• Symmetric vs Public key cryptography
• Protocols for anonymous communication
  • High-latency
    • Chaum Mixes as a building block, onion routing
  • Low-latency
    • Optimized Onion Routing and Tor
Symmetric vs Public key crypto

• **Symmetric**
  - (Symmetric) key $k$
  - Message $M$
  - Encrypt: $\text{Enc}_k(M)$ or $\{M\}_k$
  - Decrypt: $\text{Dec}_k(M)$
  - $\text{Dec}_k(\text{Enc}_k(M)) = M$
  - **Fast**

• **Public-private**
  - Bob’s public key: $\text{pk}(B)$
  - Bob’s private key: $\text{pr}(B)$
  - $\text{Dec}_{\text{pr}(B)}(\text{Enc}_{\text{pk}(B)}(M)) = M$
  - $\text{Dec}_{\text{pk}(B)}(\text{Enc}_{\text{pk}(B)}(M)) \neq M$
  - **Slow**
Outline

• Symmetric vs Public-private key encryption
• Protocols for anonymous communication
  • High-latency
    • Chaum Mixes as a building block, onion routing
  • Low-latency
    • Optimized Onion Routing and Tor
Chaum’s Mix

• Early proposal for anonymous email

• Public key crypto + trusted re-mailer (Mix)
  • Untrusted communication medium
  • Public keys used as persistent pseudonyms

• Modern anonymity systems use Mix as the basic building block

Before spam, people thought anonymous email was a good idea 😊
Basic Mix Design

Adversary knows all senders and all receivers, but cannot link a sent message with a received message.
Anonymous Return Addresses

M includes \(\{K_1, A\}_{pk(mix)}, K_2\) where \(K_2\) is a fresh public key

\(\{r_1, \{r_0, M\}_{pk(B)}, B\}_{pk(mix)}\)

Response MIX

\(\{K_1, A\}_{pk(mix)}, \{r_2, M'\}_{K_2}\)

Secrecy without authentication (good for an online confession service 😊)
Mix Cascade

- Messages are sent through a **sequence of mixes**
  - Can also form an arbitrary network of mixes (“mixnet”)
- Some of the mixes may be controlled by attacker, but even a single good mix guarantees anonymity
- Pad and buffer traffic to foil correlation attacks
What are some downsides of mix networks?

• Susceptible to timing attacks

• Latency can be high
  • Must wait for “enough” inputs before the mix relays traffic

• These challenges led to 2\textsuperscript{nd} commonly-used anonymous routing technique
Idea: Randomized Routing

• Hide message source by routing it randomly
  • Popular technique: Crowds, Freenet, Onion routing
• Routers don’t know for sure if the apparent source of a message is the true sender or another router
Onion Routing

Sender chooses a random sequence of routers
  
  Some routers are honest, some controlled by attacker
  
  Sender controls the length of the path

[Reed, Syverson, Goldschlag ‘97]
Route Establishment

- Routing info for each link encrypted with router’s public key
- Each router learns only the identity of the next router
Disadvantages of Basic Mixnets/Onion Routing

• Public-key encryption and decryption at each mix/router are computationally expensive

• Basic mixnets have high latency
  • Ok for email, not Ok for anonymous Web browsing

• Challenge: low-latency anonymity network
Outline

• Symmetric vs Public-private key encryption
• Protocols for anonymous communication
  • High-latency
    • Chaum Mixes as a building block, onion routing
  • Low-latency
    • Optimized Onion Routing and Tor
Tor

- Second-generation onion routing network
  - [http://tor.eff.org](http://tor.eff.org)
  - Developed by Roger Dingledine, Nick Mathewson and Paul Syverson
  - Specifically designed for low-latency anonymous Internet communications
- Running since October 2003
- Thousands of nodes, 2MM+ users
- “Easy-to-use” client proxy
  - Freely available, can use it for anonymous browsing
Tor Circuit Setup

• Client proxy establishes symmetric session keys with onion routers
Tor Circuit Setup (details)

- Routing info for each link encrypted with router’s public key
- Each router learns only the identity of the next router and symmetric key with source
Using a Tor Circuit

• Client applications connect and communicate over the established Tor circuit
  • Note onion now uses only symmetric keys for routers
Using a Tor Circuit (details)

Note onion now uses only symmetric keys for routers
Tor Management Issues

• 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 onion routers, their locations, current public keys, etc.
  • Control how new routers join the network
    • “Sybil attack”: attacker creates a large number of routers
  • Directory servers’ keys ship with Tor code
Common misconception:
Onion routing = Mix networks

Nikita Borisov
@nikitab

Me, an anonymity researcher, hiding under the bed:

Armed robber:

Me:

Armed robber:

Me:

Armed robber: I think TOR is the best MIX network

Me: first of all I can hear you incorrectly spelling those with all caps, and also Tor is not a mi..shit

twitter.com/jessamyn/status...
Deployed Anonymity Systems

• Free Haven project has an excellent bibliography on anonymity

• Tor (http://tor.eff.org)
  • Overlay circuit-based anonymity network
  • Best for low-latency applications such as anonymous Web browsing

• Mixminion (http://www.mixminion.net)
  • Network of mixes
  • Best for high-latency applications such as anonymous email
Outline

• Symmetric vs Public-private key encryption
• Protocols for anonymous communication
  • High-latency
    • Chaum Mixes as a building block, onion routing
  • Low-latency
    • Optimized Onion Routing and Tor
A simple idea: Basic Anonymizing Proxy

• Channels appear to come from proxy, not true originator
• Appropriate for Web connections etc.: SSL, TLS (Lower cost symmetric encryption)
• Example: Anonymizer (developed at CMU!)
• Simple, focuses lots of traffic for more anonymity
• Main disadvantage: Single point of failure, compromise, attack
Extension: Location Hidden Servers

• Goal: deploy a server on the Internet that anyone can connect to without knowing where it is or who runs it
• Accessible from anywhere
• Resistant to censorship
• Can survive full-blown DoS attack
• Resistant to physical attack
  • Can’t find the physical server!

• Commonly referred to as... the dark web
Creating a Location Hidden Server

Client obtains service descriptor and intro point address from directory

Server creates onion routes to "introduction points"

Server gives intro points' descriptors and addresses to service lookup directory
Using a Location Hidden Server

Client creates onion route to a “rendezvous point”

Client sends address of the rendezvous point and any authorization, if needed, to server through intro point

Rendezvous point mates the circuits from client & server

If server chooses to talk to client, connect to rendezvous point

Rendezvous Point

Client Alice

Introduction Points

Server Bob
In-Class Demo: Accessing Hidden Services

• Many news organizations have started offering hidden services
• New York Times
  • https://www.nytimes3xbfgragh.onion/
How would you attack this?

• Try to DoS the rendezvous point?
  • Client can switch rendezvous

• DoS an introduction point?
  • Server can switch introduction points

• DoS the lookup server?
  • This is probably the weakest link