Privacy-Preserving Data Dissemination in User Groups

Abhishek Singh
Context: Data Dissemination in User Groups

Support many groups. For e.g., Facebook has 620 million groups.

Support many users. For e.g., 1 billion users access the groups in Facebook and there 300 million monthly active users in Twitter.

Support user’s participation in multiple groups. For e.g., median number of users’ subscriptions in Twitter was 10 in 2009.

Support groups of varying sizes. For e.g., the largest group (followers of Barack Obama) in Twitter has ~112 M followers.

Provide fast dissemination of a member’s message in a group to the rest of the members.
Motivation: Preserve Users’ Privacy

How to design a robust and scalable platform that provides privacy-preserving data exchange in user groups?

Disclosure of user’s interests can be misused. For e.g., knowledge of a member’s participation in a group discussing chronic illness can lead to a denial of health insurance or a denial of equal job opportunity.

Disclosure of user’s contacts/communication partners can be misused. Such information can be used to learn if the user participates in a specific group.

Preventing such information disclosures have led to GDPR regulations.
Adversaries and Attacks

Identify users participating in many groups by observing the amount of resources (e.g., bandwidth) consumed by users.

Learn participation of a user in a group using the content of messages or the information stored in the platform.

Learn user’s communication partners by observing her network traffic.
Existing Approaches

1. Centralized vs. peer-to-peer (P2P) approach
   i. Data is decentralized with peer-to-peer (P2P) approach, but it lacks robustness.
   ii. Users’ communication partners are still disclosed.

2. Use an anonymity system to send messages in a P2P protocol to hide users’ communication partners.
   i. The combination must be done carefully to prevent information leakage. For example, user’s identity can be disclosed when Tor is integrated incorrectly with the BitTorrent protocol.

3. Anonymity systems typically sacrifice one of the following properties:
   i. Strong anonymity guarantee
   ii. Low-latency message dissemination
   iii. Robustness (availability)
   iv. Scalability (overhead)
Our Approach

Design new P2P algorithms and adapt the anonymity system of choice

<table>
<thead>
<tr>
<th>Chapter in the thesis</th>
<th>Problem</th>
<th>Anonymity System</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Robust data dissemination in a group</td>
<td>$k$-anonymity</td>
<td>Robust overlays for data dissemination within a group</td>
</tr>
<tr>
<td>3</td>
<td>Robust and scalable data dissemination in multiple groups</td>
<td>$k$-anonymity</td>
<td>Effectively maintaining overlays with multiple groups</td>
</tr>
<tr>
<td>4 and 5</td>
<td>Preventing information leaks in a Tor-based prototype</td>
<td>Tor</td>
<td>Reducing information leaks to resource providers in Tor’s infrastructure</td>
</tr>
</tbody>
</table>
Trust Graph

Trust graph is a connected friend-to-friend network formed by social interactions. Example, Facebook network.

Trust between nodes is symmetric and non-transitive.
Data Dissemination in a Group of Users

• Baseline approach
  • A node exchanges messages with only those nodes that it trusts
  • Overlay mimics trust graph
  • Example: Freenet in darknet mode [Clarke et al. 2010]

• Problem
  • Trust graph is not robust under churn
  • Social networks tend to get partitioned when a small number of high-degree nodes are removed [Mislove et al. IMC 2007]
Solution for a Group of Users

Node $A$ creates pseudonym $P_A$ at relay/mix $P$.

$P$ is unaware of the identity of node $A$.

Examples of pseudonyms: Tor hidden service, I2P eepsite.

Extend the overlay based on trust graph towards a random-like graph by adding edges in a privacy-preserving manner.
### Architecture

#### Layers and Components

**Application layer**
- Application-specific data-dissemination protocols

**Overlay layer**
- Pseudonym creation and removal
- Pseudonym distribution
- Overlay-link maintenance

**Privacy-preserving link layer**
- Anonymity and pseudonym service

---

**Node A**
- Own pseudonym
- Pseudonym cache (PC)
- Overlay links

**Node B**
- Own pseudonym
- Pseudonym cache (PC)
- Overlay links

---

**Edge in the overlay**
- Shuffle request/reply messages are exchanged periodically
- A shuffle exchange updates nodes' pseudonym caches and overlay links

**Pseudonyms are sampled using a protocol similar to Brahms [Bortnikov et al. CN 2009]**

---

A modified version of the shuffle protocol in [Staurov et al. ICNP 2002] is used to distribute nodes’ pseudonyms.
Augmented Trust Graph is Robust

- The trust graph is sampled from Facebook social network
- Baselines for the comparison:
  - Trust graph (sampled Facebook social graph)
  - Erdös-Rényi random graph with an average out-degree similar to that of our solution
Low Messaging Overhead

- Nodes with higher degree in trust graph send more messages.
- Maximum number of messages being sent is approximately twice the total average (= 2).
- The low differences between nodes’ messaging loads also result in low cover traffic with k-anonymity service.
Workload Characteristics with Multiple Groups

A group graph is a connected subgraph of the trust graph containing only the members of the group and their trust relationships with other members of the group.

A node is labeled as ID:#groups; for example, Node E participates in 3 groups.

Diversity in users’ subscriptions result in skewed distributions of subscription-sizes and group-sizes.
User’s Subscriptions is Her Fingerprint

Each group has thousands of members. But the intersection between the groups is small.

Identify by searching publicly available information for: “Indian” + “computer science research” + “living in Oslo”

Requirement: Hide the set of user’s subscriptions
Overhead with $k$-Anonymity

$k$-Anonymity prevents the disclosure of a node’s actual bandwidth consumption.

$k$-set: A set of $k$ users having the same bandwidth consumption.

A $k$-set is constructed by the anonymity service by randomly selecting nodes into the set.

$\rho$: Ratio of the users’ bandwidth consumed for providing cover traffic and the users’ bandwidth consumed in the data-dissemination protocol.

Requirement: Ensure nodes have similar bandwidth consumption in the data-dissemination protocol.
Architecture for Multiple Groups

Simple epidemics to measure avg. receiver delay

For a node, creates per-group pseudonyms

Provides k-anonymity for bandwidth consumption

A shuffle exchange includes information of only one group + load-balancing techniques

Node’s pseudonym neighbors in a group are selected independently of the selection of its pseudonym neighbors in another group

Maintain per-group mailbox to store member’s pseudonym. Necessary for small groups
Pseudonym Distribution for Multiple Groups

- Nodes select different strategies to ensure that max. no. of msg. sent per shuffle period is $N \times F$
- Strategies vary
  - the no. of msg. sent per shuffle period
  - the techniques used to restrict incoming messages
- In all strategies, the selection probability a node assigns to its trusted neighbor in the overlays is inversely proportional to the number of the neighbor’s subscriptions
- Node uses a distribution of nodes’ subscription-sizes to select its strategy

<table>
<thead>
<tr>
<th>Node’s strategy</th>
<th># req. initiated</th>
<th>Maintain the services for receiving requests sent to its long-term identity and its pseudonyms</th>
<th>Do not distribute its pseudonym in group $g$</th>
<th>Send back-off to its trusted neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not-overloaded</td>
<td>$N$</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Transiently-overloaded</td>
<td>$N$</td>
<td>Yes - The service corresponding to the long-term identity is maintained - $\text{pseudonym}_a^g$ for group $g$ exists only if $(D -</td>
<td>\text{trusted_neighbors}_a^g</td>
<td>) &gt; 0$</td>
</tr>
<tr>
<td>Persistently-overloaded</td>
<td>$N \times F$</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Robustness of Groups’ Overlays

1. A shuffle message contains information of only one group in all approaches. As a result, a node sends more than shuffle message per shuffle period to construct overlays.
2. Our solution provides better connected overlays and lower average receiver delay.
Lower $\rho$ with Our Load-Balancing Solution
Using Tor for Anonymity

- Provides weak anonymity guarantee
  - An adversary monitoring network traffic of two nodes learns of the communication between them
- An example adversary is node’s entry-guard
  - A node’s entry-guard is a guard in Tor that the node uses for the first hop of its onion circuits
  - Nodes’ entry-guards are selected independently
- The disclosures are possible with even low traffic [SEC’15, CCS’17]
  - A data cell is the unit of data transfer in Tor, and a data cell contains up to 512 bytes.
  - 4 data cells transmitted end-to-end for creating an onion circuit between the two nodes
  - Messages exchanged over an onion circuit results in transmission of more data cells
  - Guards can delay transmission of data cells

Steps for establishing an instance of hidden service and then accessing it.

A line represents an onion circuit.

All onion circuits constructed by a node goes over same entry guard.
Threat Model

- Adversaries monitor only a fraction of nodes, unlike contributions earlier in the thesis
- Adversaries control guards in Tor’s infrastructure
- An adversary does not control guards in more than one colluding guard sets
  - Similar assumption made in Tor
  - Nodes obtain the colluding guard sets from Tor’s consensus document

Problem: Disclosure of high number of communication links

- Distribution of bandwidth provided by colluding guard sets is skewed
- In Tor, probability of selecting a colluding guard set is proportional to the bandwidth provided by the guard
Our Approach

• Ensure two nodes communicating with each other select their entry guard from different colluding guard sets.
• We design the following guard selection algorithms: BGS, TAGS and SSAGS.
• Subscription-Size Aware Guard Selection (SSAGS)
  • Applies to communication links (pseudonym links) between nodes without trust relationship between them
  • Vulnerable communication link: A communication link between two nodes such that
    1. their entry guards are from the same colluding guard set, and
    2. there is a non-zero probability of creating a pseudonym link between them
• A node with more pseudonym links should select a guard from colluding guard set that provides smaller bandwidth
• Use node’s subscription-size as an estimate for the number of its pseudonym links

Colluding Guard Set \( S_1 \)

Node A

Node D

Colluding Guard Set \( S_2 \)

Node B

Node C

Tor

# vulnerable comm. links: 2

Colluding Guard Set \( S_1 \)

Node B

Colluding Guard Set \( S_2 \)

Node A

Node D

SSAGS

# vulnerable comm. links: 0
**SSAGS**

- Nodes generate different probability distributions for selecting their entry guards.

- A `guard_set`'s weight in SSAGS for a node is given by:
  \[ \text{bandwidth}_{guard\_set} \times (1 - \text{frac}) \times \text{skewness} \]

  where,
  - `frac` is the fraction of nodes with fewer number of subscriptions.
  - `skewness` is a global parameter that controls the bias of the weight towards `guard_set`'s bandwidth.

- Constraints for computing `skewness`:
  - Identifying a node’s entry guard should be non-trivial. That is, the entropy of the probability distribution should be high enough.
  - Total over-selection should be low.

* A colluding guard set contains 1 guard.
## Results

### Experiment settings:
- A colluding guard set contains 1 guard.
- Number of colluding guard sets: 2256
- Entropy with Tor: 10.4351 bits
- Minimum entropy with SSAGS > 8 bits

Measurements are avg. of 10 runs

<table>
<thead>
<tr>
<th>Workload</th>
<th>Vul. comm. links</th>
<th>Vulnerable groups</th>
<th>Entropy with SSAGS</th>
<th>Over-selection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># with Tor</td>
<td>Reduction with SSAGS</td>
<td># with Tor</td>
<td>Reduction with SSAGS</td>
</tr>
<tr>
<td>1</td>
<td>16147</td>
<td>25.6 %</td>
<td>5200</td>
<td>18.27 %</td>
</tr>
<tr>
<td>2</td>
<td>19060</td>
<td>25.3 %</td>
<td>5368</td>
<td>17.16 %</td>
</tr>
<tr>
<td>3</td>
<td>13874</td>
<td>18.8 %</td>
<td>4925</td>
<td>16.11 %</td>
</tr>
<tr>
<td>4</td>
<td>3874</td>
<td>27.9 %</td>
<td>4465</td>
<td>23.74 %</td>
</tr>
</tbody>
</table>

Distribution of probability of selecting guards in Tor with varying skewness (exp) in bw. distribution
Summary

How to design a robust and scalable platform that provides privacy-preserving data exchange in social communities?

Our approach: Design new P2P algorithms and adapt the anonymity system of choice

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Our Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robustness</td>
<td>Per-group random-graph like overlay</td>
</tr>
<tr>
<td>Scalability</td>
<td>• Small diameter of the group’s overlay</td>
</tr>
<tr>
<td></td>
<td>• No excessive read/writes to the group’s mailbox</td>
</tr>
<tr>
<td># group members</td>
<td>User’s maximum messaging load is restricted</td>
</tr>
<tr>
<td># user’s subscriptions</td>
<td>No group experiences starvation (blocked message dissemination)</td>
</tr>
<tr>
<td># of groups</td>
<td>Low cover traffic, as the load-balancing protocol ensures users’ messaging load is similar even when users have a diverse number of subscriptions</td>
</tr>
<tr>
<td># of users</td>
<td>Guard selection algorithms: BGS, TAGS and SSAGS</td>
</tr>
</tbody>
</table>

Privacy with Tor
Questions
Drawbacks of Anonymity Systems

Anonymity systems providing broadcast messaging primitive
1. High latency
   i. DC-nets, such as Dissent [SEC’13], in the presence malicious participants
   ii. Mix-networks providing broadcast such as Atom [SOSP’17]
   iii. Write-Private DB such as Riposte [SP’15]
2. High messaging overhead
   i. Not ideal for unicast communication or multicast for small groups

Anonymity systems providing unicast messaging primitive
1. High latency
   i. Mix-networks such as Vuvuzela [SOSP’15] and Stadium [SOSP’17]
2. Use of cover traffic (reduces scalability)
   i. Mix-networks such as Loopix [Security’17], Vuvuzela [SOSP’15] and Stadium [SOSP’17].
      a. The cover traffic is used for hiding access patterns to per-user mailbox or to ”dead-drops”.
   ii. k-anonymity providing anonymity system such as Aqua [SIGCOMM’13].
      a. The cover traffic is used for hiding the volume of bandwidth consumed by a user.
3. Weak anonymity guarantee
   i. Onion routing based anonymity systems, such as Tor and I2P.
Resource Consumption with Tor Prototype
Limitations

Passive adversary

Validation by simulations
- Prototype parts of the system.
- Perfect failure detector

Synthetic traces
- Social graph samples
- Synthetic churn traces
Future Work

Validation of privacy-preservation techniques
- Extend MATor [CCS’14] framework to analyze onion circuits created using hidden services.
- Analyze information leaks from the distribution of pseudonyms using differential privacy techniques.

Privacy-preservation techniques
- Adapt SSAGS to defend against stronger adversaries by improving construction of colluding guard sets and trustworthiness of relays.
- Explore heuristics other than the number of users’ subscriptions for selection of users’ entry guards in Tor.
- A robust technique for generating cover traffic and additional onion circuits.

Scalability
- Dynamically adapt user’s strategy for exchanging shuffle messages to speedup dissemination or decrease overhead.
- Decrease the number of users’ neighbors in the overlays without increasing the privacy risk.
- Reduce the load imbalances in Tor’s infrastructure when SSAGS is used by considering the shape of the distribution of guards’ bandwidth.

Churn in trust relationships and group memberships