Detecting Social Cliques for Automated Privacy Control in Online Social Networks

Hakan Yıldız

(Joint work with Chris Kruegel)
Outline

- Introduction
- Related Work
- Algorithm
- Evaluation
Privacy in Social Networks

- A “BIG” Issue.
- Not only related to adversaries.
- Often, share target is a specific group of friends.
- Requires more than a simple privacy model.
Friend Lists

• User manually specifies the people that the data exposed to.

• Ideal solution?

• Zuckerberg: “Nobody wants to make lists.” [Siegler, TechCrunch Article '10]

• Facebook recently introduced pre-defined lists.
  • Depends on profile data.
  • Friend list manually assigned.
Automated Privacy Control

• Automatically generate and assign the friend list to the data.
• Network site figures out which people should have access.
• No user involvement.
• User might modify the generated list.
Social Cliques

• Group of people sharing significant interaction, possibly due to a common cause.

• Example: Families, classmates, colleagues ...

• In this study, focused on friendship relationships only.
Detecting Social Cliques from Participating Group

- Data is often associated to a group of people directly contributing. (Participating Group)
- Example: Photo tags, wall feed authorship, etc.
- Expose data to the social clique containing the participating group.
- Limit social clique to the friends of the host user.
Related Work

- **Local Community Detection in SNs:**
  - Community: Densely connected component
  - Detect a single community containing given nodes with local visibility
    [Clauset PR E'05] [Luo et al. WIAS'08] [Chen et al. ASNAM'09]
  - Our problem is a variant of LCD.
  - We introduce novel algorithms.
  - We introduce a new evaluation idea using photo tags.
Related Work

- Privacy in Social Networks
  - Mostly related to privacy against adversaries.
  - Privacy Wizards:
    [Fang and LeFevre, WWW'10]
    - Similar approach
    - Combines machine learning with existing community detection techniques.
    - Actively learns by user input.
Algorithm

- **Objective:** Given a host user $s$ and a subset of her friends $P$ as a participating group, determine a larger subset $C$ of $s$'s friends so that $C$ forms a social clique.
High Level Algorithm

- Assign $C$ to $P$.
- Repeatedly add a node $v$ from $n(s) - C$ to $C$.
  - $v$ maximizes the heuristic measure.
  - $v$ satisfies a boolean function $f(C, v)$. (Clique expansion function)
- Stop when all nodes in $n(s) - C$ fail to satisfy $f$.
- Output $C$. 
Heuristic Measure

- Choose the node $v$ maximizing the number of friends in $C$:
  $$|C \cap n(v)|$$

- In case of equality, choose $v$ that maximizes average number of common friends in $C$:
  $$\frac{\sum_{c \in C} |n(c) \cap n(v)|}{|C|}$$
Number of Common Friends

• Real Example
Clique Expansion Scheme 1: CLQ

- \( v \) has friendship links to all nodes in \( C \).

\[
f(C, v) = \begin{cases} 
  \text{true} & \text{if } n(v) \cap C = C \\
  \text{false} & \text{otherwise}
\end{cases}
\]

- Too strict. Produces small cliques.
Clique Expansion Scheme 2: \textbf{BAND}_K

- \nu \text{ has at least } K \text{ common friends with each node in } C.

\[ f(C, \nu) = \begin{cases} 
  \text{true} & \text{if } \forall c \in C. |n(c) \cap n(\nu)| \geq K \\
  \text{false} & \text{otherwise}
\end{cases} \]
Clique Expansion Scheme 3: $\text{IN}_K$

- The idea is to adapt the expansion to the tightness of the clique.
- Initial clique loose $\rightarrow$ Final clique loose.
  Initial clique tight $\rightarrow$ Final clique tight.
- Let $r(v, C)$ be the percentage of $v$'s friends in $C$.

$$r(v, C) = \frac{|n(v) \cap C|}{|C|}$$
Clique Expansion Scheme 3: IN$_K$

- Average the percentage values for each $c$ in $C$ to get a measure of the tightness of $C$.

\[
t(C) = \sum_{c \in C} \frac{r(c, C - \{c\})}{|C|}
\]

- $\nu$'s percentage of friends is at least $K$ times the tightness of $C$.

\[
f(C, \nu) = \begin{cases} 
true & \text{if } r(\nu) > K \times t(C) \\
false & \text{otherwise}
\end{cases}
\]
Experimental Evaluation

- “AutoClique” Facebook application.
- Evaluated the following schemes on photos by 22 volunteers.
  - CLQ, $\text{BAND}_2$, $\text{BAND}_3$, $\text{BAND}_4$, $\text{IN}_{0.3}$, $\text{IN}_{0.5}$, $\text{IN}_{0.7}$
- For comparison,
  - CLA [Clauset, PR E '05]
  - CZG [Chen, Zaïane, Goebel, ASNAM'09]
Methodology

• **Assumption:** People tagged in a photo form a social clique.

• A test on a photo shared by host $s$:
  • $T =$ Users tagged in the photo.
  • Pick a random subset $P$ of $T$.
  • Run the algorithm with $P$ as the participating set and $s$ as the host.
  • Ideally, detected clique $C$ covers all of $T$. 
Recall and Coverage

- **Recall**: Fraction of tagged people covered by the clique.
  \[
  \frac{|T \cap C| - |P|}{|T| - |P|}
  \]

- **Coverage**: Ratio of host user's friends covered by the clique.
  \[
  \frac{|C|}{|n(s)|}
  \]

- **Goal**: High Recall, Low Coverage.
Results

- 1416 photos, 10 tests for each possible $|P|$ between 1 and 8
- 33860 tests total, $\text{Avg}(|T|) = 5.4$, $\text{Avg}(|P|) = 2.5$

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Av. Recall</th>
<th>Av. Coverage</th>
<th>Av. Clique Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLQ</td>
<td>0.49</td>
<td>0.05</td>
<td>11</td>
</tr>
<tr>
<td><strong>BAND\textsubscript{2}</strong></td>
<td><strong>0.90</strong></td>
<td><strong>0.28</strong></td>
<td><strong>65</strong></td>
</tr>
<tr>
<td>BAND\textsubscript{3}</td>
<td>0.88</td>
<td>0.25</td>
<td>54</td>
</tr>
<tr>
<td>BAND\textsubscript{4}</td>
<td>0.85</td>
<td>0.23</td>
<td>48</td>
</tr>
<tr>
<td><strong>IN\textsubscript{0.3}</strong></td>
<td><strong>0.92</strong></td>
<td><strong>0.36</strong></td>
<td><strong>97</strong></td>
</tr>
<tr>
<td>IN\textsubscript{0.5}</td>
<td>0.86</td>
<td>0.23</td>
<td>53</td>
</tr>
<tr>
<td>IN\textsubscript{0.7}</td>
<td>0.74</td>
<td>0.14</td>
<td>30</td>
</tr>
<tr>
<td>CLA</td>
<td>0.75</td>
<td>0.28</td>
<td>71</td>
</tr>
<tr>
<td>CZG</td>
<td>0.57</td>
<td>0.17</td>
<td>36</td>
</tr>
</tbody>
</table>
## Recall vs $|P|$ 

| $|P|$ | CLQ  | BAND$_2$ | IN$_{0.3}$ | CLA  | CZG  |
|-----|------|----------|------------|------|------|
| 1   | 0.29 | 0.74     | 0.79       | 0.53 | 0.39 |
| 2   | 0.31 | 0.76     | 0.81       | 0.74 | 0.56 |
| 3   | 0.31 | 0.74     | 0.81       | 0.79 | 0.60 |
| 4   | 0.28 | 0.74     | 0.81       | 0.79 | 0.61 |
| 5   | 0.28 | 0.75     | 0.81       | 0.81 | 0.62 |
| 6   | 0.25 | 0.74     | 0.83       | 0.82 | 0.61 |
| 7   | 0.26 | 0.74     | 0.84       | 0.81 | 0.61 |
| 8   | 0.24 | 0.74     | 0.84       | 0.82 | 0.63 |

Statistics from 55 photos with >8 tags.
Conclusion

- Automated privacy control can be done by detecting social cliques.
- Proposed methods with ~%90 accuracy and only ~%30 coverage.
- Outperformed existing LCD methods in all evaluation scenarios.
Thank you for your attention.

- Questions?
The number of photos containing specific number of tags.

<table>
<thead>
<tr>
<th>Number of Tags</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Photos</td>
<td>634</td>
<td>337</td>
<td>157</td>
<td>105</td>
<td>63</td>
<td>43</td>
<td>24</td>
<td>55</td>
</tr>
</tbody>
</table>