Seed and Grow: An Attack Against Anonymized Social Networks

Wei Peng\textsuperscript{1}  Feng Li\textsuperscript{1}  Xukai Zou\textsuperscript{1}  Jie Wu\textsuperscript{2}

\textsuperscript{1}Indiana University-Purdue University Indianapolis (IUPUI)
\textsuperscript{2}Temple University

21 June 2012
Online social networking services are everywhere.
User connections become the new assets.
Naive anonymization: Conceal who but retain utility.
Who?

We allow advertisers to choose the characteristics of users who will see their advertisements and we may use any of the non-personally identifiable attributes we have collected (including information you may have decided not to show to other users, such as your birth year or other sensitive personal information or preferences) to select the appropriate audience for those advertisements.

Facebook Privacy Policy, 22 December 2010
The question.

Q: Can naive anonymization alone preserve user privacy?
Q: Can naive anonymization alone preserve user privacy?

A: Yes!
Q: Can naive anonymization alone preserve user privacy?
A: Yes! This is what the industry wishes us to believe.
Q: Can naive anonymization alone preserve user privacy?

A: Yes?
Q: Can naive anonymization alone preserve user privacy?

A: Yes? This is what researchers, including ourselves, are asking.
Q: Can naive anonymization alone preserve user privacy?

A: Yes, only if the attacker knows nothing but the graph.
Q: Can naive anonymization alone preserve user privacy?

A: Yes, only if the attacker knows nothing but the graph. Given the increasing overlap in user-bases, the answer is becoming NO.
Seed and Grow.
The idea.

Exploit the similarity of user connections across sites to de-anonymize (naively) anonymized social network.
Seed and Grow.
The motto.

Plant a seed, then grow it.
Meet Bob.

- Bob obtains a naively anonymized target graph $G_T$ (with user IDs removed) from the F company.
- He crawls a background graph $G_B$ (with user IDs retained) from the site of the T company.
- $G_T$ and $G_B$ are partially overlapped in vertices and have similar (but not necessarily identical) connections among the overlapped vertices.
- The goal: to identify vertices on $G_T$ with the help of $G_B$. 
Meet Bob.

- Bob obtains a naively anonymized target graph $G_T$ (with user IDs removed) from the F company.
- He crawls a background graph $G_B$ (with user IDs retained) from the site of the T company.
- $G_T$ and $G_B$ are partially overlapped in vertices and have similar (but not necessarily identical) connections among the overlapped vertices.
- The goal: to identify vertices on $G_T$ with the help of $G_B$. 
Meet Bob.

- Bob obtains a naively anonymized **target graph** $G_T$ (with user IDs removed) from the F company.
- He crawls a **background graph** $G_B$ (with user IDs retained) from the site of the T company.
- $G_T$ and $G_B$ are **partially overlapped in vertices** and have **similar** (but not necessarily identical) connections among the overlapped vertices.
- The goal: to **identify** vertices on $G_T$ with the help of $G_B$. 
Meet Bob.

- Bob obtains a naively anonymized **target graph** $G_T$ (with user IDs removed) from the F company.
- He crawls a **background graph** $G_B$ (with user IDs retained) from the site of the T company.
- $G_T$ and $G_B$ are **partially overlapped in vertices** and have **similar** (but not necessarily identical) connections among the overlapped vertices.
- The goal: to **identify** vertices on $G_T$ with the help of $G_B$. 
Seed.

Plant a specially constructed fingerprint $G_F$ into $G_T$ before $G_T$’s anonymization and release.

Recover $G_F$ from $G_T$ after $G_T$’s anonymization and release.

Identify the neighbors $V_S$ of $G_F$ as the initial seed.
Seed.

Plant a specially constructed fingerprint $G_F$ into $G_T$ before $G_T$’s anonymization and release.

Recover $G_F$ from $G_T$ after $G_T$’s anonymization and release.

Identify the neighbors $V_S$ of $G_F$ as the initial seed.
Plant a specially constructed fingerprint $G_F$ into $G_T$ before $G_T$’s anonymization and release.

Retrieve $G_F$ from $G_T$ after $G_T$’s anonymization and release.

Identify the neighbors $V_S$ of $G_F$ as the initial seed.
The symbols.

$G_T$  
$G_B$  
$G_F \subseteq G_T$  
$V_*$  
$E_*$  
$V_S$  
$V_F(u)$  
$u$’s neighboring vertices in $V_F$
A first try in planting a fingerprint.

Generate a random fingerprint $G_F$ and connect it with some vertices in the target $G_T$. 
A twist.

A randomly generated graph $G$ may be symmetric.
The fingerprint: **ideal** vs. reality.

- **Uniquely identifiable**  
  No subgraph $H \subseteq G_T$ except $G_F$ is isomorphic to $G_F$.

- **Asymmetric**  
  $G_F$ does not have any non-trivial automorphism.
The fingerprint: ideal vs. reality.

- **Uniquely identifiable**: Not guaranteed but very likely with a large enough $G_F$.
- **Asymmetric**: Can be relaxed.
The insights.

- The goal is to identify the initial seed $V_S$ rather than the fingerprint $G_F$.
- For each pair of vertices, say $u$ and $v$, in $V_S$, as long as $V_F(u)$ and $V_F(v)$ are distinguishable in $G_F$, once $G_F$ is recovered from $G_T$, $V_S$ can be identified uniquely.
- “$V_F(u)$ and $V_F(v)$ are distinguishable in $G_F$” means no automorphism of $G_F$ exists which maps $V_F(u)$ to $V_F(v)$, e.g., $|V_F(u)| \neq |V_F(v)|$ or the degree sequences are different.
The insights.

- The goal is to identify the initial seed $V_S$ rather than the fingerprint $G_F$.
- For each pair of vertices, say $u$ and $v$, in $V_S$, as long as $V_F(u)$ and $V_F(v)$ are distinguishable in $G_F$, once $G_F$ is recovered from $G_T$, $V_S$ can be identified uniquely.
- “$V_F(u)$ and $V_F(v)$ are distinguishable in $G_F$” means no automorphism of $G_F$ exists which maps $V_F(u)$ to $V_F(v)$, e.g., $|V_F(u)| \neq |V_F(v)|$ or the degree sequences are different.
The insights.

- The goal is to identify the initial seed $V_S$ rather than the fingerprint $G_F$.
- For each pair of vertices, say $u$ and $v$, in $V_S$, as long as $V_F(u)$ and $V_F(v)$ are distinguishable in $G_F$, once $G_F$ is recovered from $G_T$, $V_S$ can be identified uniquely.
- “$V_F(u)$ and $V_F(v)$ are distinguishable in $G_F$” means no automorphism of $G_F$ exists which maps $V_F(u)$ to $V_F(v)$, e.g., $|V_F(u)| \neq |V_F(v)|$ or the degree sequences are different.
Initially, Bob creates 7 accounts $V_F = \{v_h, v_1, \ldots, v_6\}$.
Plant a fingerprint.

He first connects $v_h$ with $v_1, \ldots, v_6$. 
Plant a fingerprint.

After awhile, users $V_S = \{v_7, \ldots, v_{10}\}$ are connected with $V_F - \{v_h\}$. 
Plant a fingerprint.

He then *randomly* connects $v_1, \ldots, v_6$ and get the resulting graph $G_F$. 
Plant a fingerprint.

The **ordered internal degree sequence**

\[ S_D = \langle 2, 2, 2, 3, 3, 4 \rangle. \]
Bob finds $S_D(v_7) = \langle 2 \rangle$, $S_D(v_8) = \langle 2, 2 \rangle$, $S_D(v_9) = \langle 3, 3, 4 \rangle$, and $S_D(v_{10}) = \langle 2, 3 \rangle$. 
Plant a fingerprint.

Since they are **mutually distinct**, Bob is sure that he can identify the **initial seeds** $V_S = \{v_7, \ldots, v_{10}\}$ once the **fingerprint** $V_F$ is found in the published anonymized graph $G_T$. 
Plant a fingerprint.
The details.

1: Create $V_F = \{v_h, v_1, v_2, \ldots\}$.
2: Given connectivity between $V_F$ and $V_S$.
3: Connect $v_h$ with $v$ for all $v \in V_F - \{v_h\}$.
4: loop
5: \textbf{for all pairs} $v_a \neq v_b$ in $V_F - \{v_h\}$ \textbf{do}
6: \hspace{1em} Randomly connect $v_a$ to $v_b$.
7: \textbf{for all} $u \in V_S$ \textbf{do}
8: \hspace{1em} Find $S_D(u)$.
9: \textbf{if} $S_D(u)$ are \textbf{mutually distinct} for all $u \in V_S$ \textbf{then}
10: \hspace{1em} return
Recover the fingerprint.
Match the fingerprint secrets.

- Degree of $v_h$.
- The ordered internal degree sequence.
Recover the fingerprint.

Bob examines all the vertices in $G_T$ for one with degree 6 (the degree of $v_h$).
Recover the fingerprint.

When Bob actually reaches $v_h$, he isolates it along with its 1-hop neighbors $G_C$ (candidate) and records, for each of the neighbors, the number of connections in $G_C$ (internal degrees).
Recover the fingerprint.

$G_C$ has an ordered internal degree sequence $\langle 2, 2, 2, 3, 3, 4 \rangle$, which matches with that of $V_F$.
Recover the fingerprint.

He then isolates $v_h$’s **exact 2-hop neighbors** and checks their **ordered internal degree subsequences**, which again matches with those of $V_S$. 
Identify the initial seeds.

Bob identifies the initial seeds $V_S = \{v_7, \ldots, v_{10}\}$ by matching ordered internal degree subsequences.
Recover and identify.
The details.

1: for all $u \in G_T$ do
2:   if $\deg(u) = |V_F| - 1$ then
3:     $U \leftarrow$ 1-hop neighborhood of $u$
4:   for all $v \in U$ do
5:     $d(v) \leftarrow$ number of $v$'s neighbors in $U \cup \{u\}$
6:     $s(u) \leftarrow$ sort($d(v)|v \in U$)
7:   if $s(u) = S_D$ then
8:     $V \leftarrow$ exact 2-hop neighborhood of $u$
9:   for all $w \in V$ do
10:    $U(w) \leftarrow w$'s neighbors in $U$
11:    $s(w) \leftarrow$ sort($d(v)|v \in U(w)$)
12:   if $\langle s(w)|w \in V \rangle = \langle S_D(v)|v \in V_S \rangle$ then
13:    $\{w \in V \text{ is identified with } v \in V_S \text{ if } s(w) = S_D(v)\}$
From *Seed* to Grow.

Bob has identified the initial seeds $V_S = \{v_7, \ldots, v_{10}\}$. 
From Seed to Grow.

How can he identify other users in the target graph with the help of the background?
Grow the seeds.

Measuring structural similarity, or equivalently, dissimilarity.

![Network diagram showing the target and background structures. The target network on the left consists of nodes labeled 7, 8, 9, 10, *1, *2, *3, *4, *5, *6, and *7, interconnected. The background network on the right consists of nodes labeled 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16, also interconnected.]
Grow the seeds.

- Define $\mathcal{N}^T_m(u): u \in V_T$'s **mapped** neighbors.
- Example: $\mathcal{N}^T_m(u_{*1}) = \{u_7, u_8, u_9\}$.
- Similar definition $\mathcal{N}^B_m(v)$ for $v \in V_B$. 

![Diagram showing target and background nodes](image)
Grow the seeds.

For \( u \in V_T \) and \( v \in V_B \), define the **dissimilarity** of \( u \) and \( v \):
\[
\Delta(u, v) = (\Delta_T(u, v), \Delta_B(u, v)).
\]

\[
\Delta_T(u, v) = \frac{|\mathcal{N}^T_m(u) - \mathcal{N}^B_m(v)|}{|\mathcal{N}^T_m(u)|}, \quad \Delta_B(u, v) = \frac{|\mathcal{N}^B_m(v) - \mathcal{N}^T_m(u)|}{|\mathcal{N}^B_m(v)|}.
\]
Grow the seeds.

Bob does the maths...

<table>
<thead>
<tr>
<th>$\Delta$</th>
<th>$u_{*1}$</th>
<th>$u_{*2}$</th>
<th>$u_{*3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_{11}$</td>
<td>(0.00, 0.00)</td>
<td>(0.00, 0.33)</td>
<td>(0.50, 0.67)</td>
</tr>
<tr>
<td>$v_{12}$</td>
<td>(0.67, 0.50)</td>
<td>(0.50, 0.50)</td>
<td>(0.00, 0.00)</td>
</tr>
</tbody>
</table>
Grow the seeds.

... and find most similar matches.

<table>
<thead>
<tr>
<th>$\Delta$</th>
<th>$u_{*1}$</th>
<th>$u_{*2}$</th>
<th>$u_{*3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_{11}$</td>
<td>(0.00, 0.00)</td>
<td>(0.00, 0.33)</td>
<td>(0.50, 0.67)</td>
</tr>
<tr>
<td>$v_{12}$</td>
<td>(0.67, 0.50)</td>
<td>(0.50, 0.50)</td>
<td>(0.00, 0.00)</td>
</tr>
</tbody>
</table>
A twist: What if there are conflicts?
Choose the ones that stand out.

Conservativeness pays off: Early mismatches have an avalanche effect.

Eccentricity: how does a number $x$ stand out among its peers in a multiset $X$?

$$\mathcal{E}_X(x) = \begin{cases} \frac{\Delta_x(x)}{\sigma(X) \#_x(x)} & \text{if } \sigma(X) \neq 0 \\ 0 & \text{if } \sigma(X) = 0 \end{cases}.$$
A twist: What if there are *conflicts*?
Choose the ones that *stand out*.

**Conservativeness** pays off: Early mismatches have an *avalanche* effect.

**Eccentricity**: how does a number $x$ stand out among its peers in a multiset $X$?

$$
\mathcal{E}_X(x) = \begin{cases} 
\frac{\Delta_X(x)}{\sigma(X)\#_X(x)} & \text{if } \sigma(X) \neq 0 \\
0 & \text{if } \sigma(X) = 0 
\end{cases}.
$$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta_X(x)$</td>
<td>Absolute difference between $x$ and its <em>closest different</em> value in $X$.</td>
</tr>
<tr>
<td>$#_X(x)$</td>
<td>Multitude of $x$ in $X$.</td>
</tr>
<tr>
<td>$\sigma(X)$</td>
<td>Standard deviation of $X$.</td>
</tr>
</tbody>
</table>
A twist: What if there are conflicts?
Choose the ones that stand out.

**Conservativeness** pays off: Early mismatches have an avalanche effect.

**Eccentricity**: how does a number $x$ stand out among its peers in a multiset $X$?

$$
E_X(x) = \begin{cases} 
\frac{\Delta_X(x)}{\sigma(X)\#_X(x)} & \text{if } \sigma(X) \neq 0 \\
0 & \text{if } \sigma(X) = 0 
\end{cases}
$$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta_X(x)$</td>
<td>Absolute difference between $x$ and its closest different value in $X$.</td>
</tr>
<tr>
<td>$#_X(x)$</td>
<td>Multitude of $x$ in $X$.</td>
</tr>
<tr>
<td>$\sigma(X)$</td>
<td>Standard deviation of $X$.</td>
</tr>
</tbody>
</table>
A twist: What if there are conflicts? Choose the ones that stand out.

**Conservativeness** pays off: Early mismatches have an avalanche effect.

**Eccentricity**: how does a number $x$ stand out among its peers in a multiset $X$?

$$
\mathcal{E}_X(x) = \begin{cases} 
\frac{\Delta_X(x)}{\sigma(X)\#_X(x)} & \text{if } \sigma(X) \neq 0 \\
0 & \text{if } \sigma(X) = 0 
\end{cases}.
$$

<table>
<thead>
<tr>
<th>$\Delta_X(x)$</th>
<th>Absolute difference between $x$ and its closest different value in $X$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$#_X(x)$</td>
<td>Multitude of $x$ in $X$.</td>
</tr>
<tr>
<td>$\sigma(X)$</td>
<td>Standard deviation of $X$.</td>
</tr>
</tbody>
</table>
Grow

The details.

1: Given the initial seed $V_S$.
2: $C = \emptyset$
3: loop
4: $C_T \leftarrow \{u \in V_T | u \text{ connects to } V_S\}$
5: $C_B \leftarrow \{v \in V_B | v \text{ connects to } V_S\}$
6: if $(C_T, C_B) \in C$ then
7: return $V_S$
8: $C \leftarrow C \cup \{(C_T, C_B)\}$
9: for all $(u, v) \in (C_T, C_B)$ do
10: Compute $\Delta_T(u, v)$ and $\Delta_B(u, v)$.
11: $S \leftarrow \{(u, v) | \Delta_T(u, v) \text{ and } \Delta_B(u, v) \text{ are smallest among conflicts}\}$
12: for all $(u, v) \in S$ do
13: if $(u, v)$ has no conflict in $S$ or $(u, v)$ has the uniquely largest eccentricity among conflicts in $S$ then
14: $V_S \leftarrow V_S \cup \{(u, v)\}$
Inspirations.

L. Backstrom, C. Dwork, and J. Kleinberg.
Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography.

Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee.
Measurement and analysis of online social networks.

A. Narayanan and V. Shmatikov.
De-anonymizing social networks.
Questions?
Thank you for your attention!
Datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Vertex</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Livejournal [MMG07]</td>
<td>5.2 million</td>
<td>72 million</td>
</tr>
<tr>
<td>emailWeek(^1)</td>
<td>200</td>
<td>1,676</td>
</tr>
</tbody>
</table>

| Dataset   | \(|V_T|\) | \(|V_B|\) | \(|V_T \cap V_B|\) |
|-----------|--------|--------|----------------|
| Livejournal | 600    | 600    | 400             |
| emailWeek  | 125    | 125    | 100             |

\(^1\)The dataset and its visualization are publicly available at http://www.infovis-wiki.net/index.php/Social_Network_Generation.
Estimation of essentially different fingerprint constructions.

For a fingerprint graph $G_F$ with $n$ vertices, there are at least

$$\frac{2^{(n-1)(n-2)/2}}{(n-1)!}$$

essentially different seed constructions.

<table>
<thead>
<tr>
<th>$n$</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>estimate</td>
<td>$1.89 \times 10^6$</td>
<td>$9.70 \times 10^7$</td>
<td>$9.03 \times 10^8$</td>
<td>$1.54 \times 10^{11}$</td>
</tr>
</tbody>
</table>
A comparative study.

- We were inspired by Narayanan and Shmatikov [NS09].
- So we compare the Grow algorithm with theirs.
- Narayanan and Shmatikov algorithm [NS09] (Narayanan for short) has a mandatory parameter for adjusting matching aggressiveness.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Parameter</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conservative</td>
<td>1</td>
<td>nar c</td>
</tr>
<tr>
<td>Aggressive</td>
<td>0.0001</td>
<td>nar a</td>
</tr>
</tbody>
</table>
Different initial seed sizes.

$|V_S| = 5$, Livejournal

$|V_S| = 10$, Livejournal

$|V_S| = 15$, Livejournal

$|V_S| = 5$, emailWeek

$|V_S| = 10$, emailWeek

$|V_S| = 15$, emailWeek
Edge perturbation.

0.5%, Livejournal

1%, Livejournal

1.5%, Livejournal

0.5%, emailWeek

1%, emailWeek

1.5%, emailWeek
Summary.

- Seed and Grow does not rely on arbitrary parameter.
- Seed and Grow finds a good balance between effectiveness (i.e., number of correct identification) and accuracy (i.e., number of incorrect identification).
- Seed-and-Grow favors high accuracy, which is more important than effectiveness in connection with confidence on the result.
- Conservative in Grow pays off with high accuracy!
Summary.

- Seed and Grow does not rely on arbitrary parameter.
- Seed and Grow finds a good balance between **effectiveness** (i.e., number of correct identification) and **accuracy** (i.e., number of incorrect identification).
- Seed-and-Grow **favors high accuracy**, which is more important than effectiveness in connection with confidence on the result.
- **Conservative in Grow** pays off with **high accuracy**!
Summary.

- Seed and Grow does not rely on arbitrary parameter.
- Seed and Grow finds a good balance between **effectiveness** (i.e., number of correct identification) and **accuracy** (i.e., number of incorrect identification).
- Seed-and-Grow **favors high accuracy**, which is more important than effectiveness in connection with confidence on the result.
- **Conservative in Grow pays off with high accuracy!**
Summary.

- Seed and Grow does not rely on arbitrary parameter.
- Seed and Grow finds a good balance between **effectiveness** (i.e., number of correct identification) and **accuracy** (i.e., number of incorrect identification).
- Seed-and-Grow **favors high accuracy**, which is more important than effectiveness in connection with confidence on the result.
- **Conservative in Grow** pays off with **high accuracy**!