Seed and Grow: An Attack Against Anonymized Social Networks

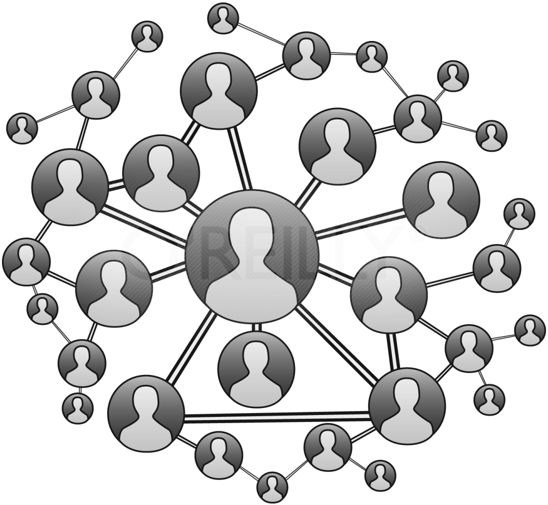
Wei Peng[1] Feng Li[1] Xukai Zou[1] Jie Wu[2]

[1]Indiana University-Purdue University Indianapolis (IUPUI)
[2]Temple University

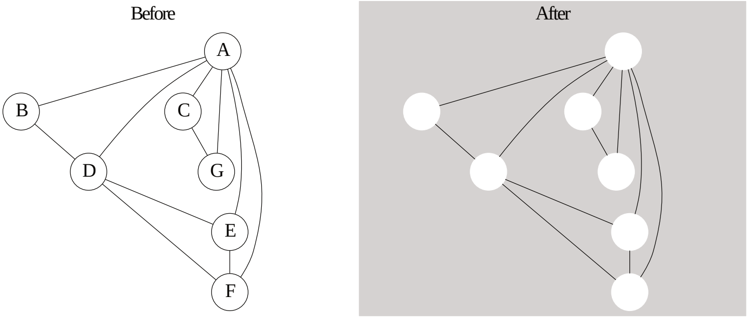
21 June 2012

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**?

The question.

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

A: Yes!

The question.

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

A: Yes! This is what the **industry wishes** us to **believe**.

The question.

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

A: Yes?

The question.

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

A: Yes? This is what **researchers**, including ourselves, are **asking**.

The question.

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

A: Yes, **only if** the **attacker knows nothing but the graph**.

The question.

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 **Plant** a specially constructed **fingerprint** G_F into G_T **before** G_T's anonymization and release.

Recover **Retrieve** G_F from G_T after G_T's anonymization and release.

Identify **Identify** the neighbors V_S of G_F as the **initial** seed.

Seed.

Plant	**Plant** a specially constructed **fingerprint** G_F into G_T **before** G_T's anonymization and release.
Recover	**Retrieve** G_F from G_T after G_T's anonymization and release.
Identify	**Identify** the neighbors V_S of G_F as the **initial** seed.

Seed.

Plant	**Plant** a specially constructed **fingerprint** G_F into G_T **before** G_T's anonymization and release.
Recover	**Retrieve** G_F from G_T after G_T's anonymization and release.
Identify	**Identify** the neighbors V_S of G_F as the **initial** seed.

The symbols.

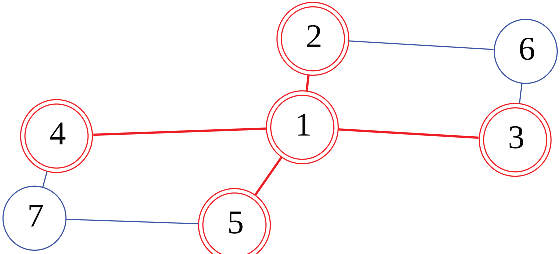
G_T	**T**arget **g**raph
G_B	**B**ackground **g**raph
$G_F \subseteq G_T$	**F**ingerprint **g**raph
V_*	**V**ertices
E_*	**E**dges
V_S	**S**eeds
$V_F(u)$	u's neighboring vertices in V_F

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.

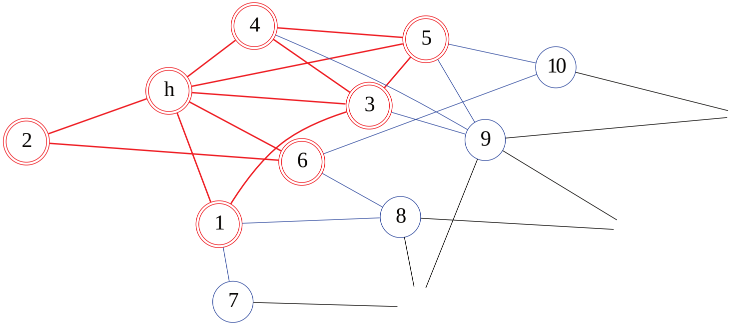
- The goal is to identify the **initial seed** V_S rather than the fingerprint G_F.
- For each pair of vertices, say u and v, in V_S, as long as $V_F(u)$ and $V_F(v)$ are **distinguishable in** G_F, once G_F is recovered from G_T, V_S can be identified **uniquely**.
- "$V_F(u)$ and $V_F(v)$ are distinguishable 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.

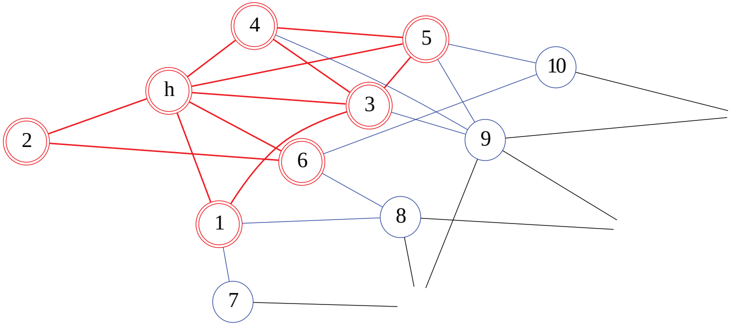
Plant a fingerprint.

Initially, Bob creates 7 accounts $V_F = \{v_h, v_1, \ldots, v_6\}$.



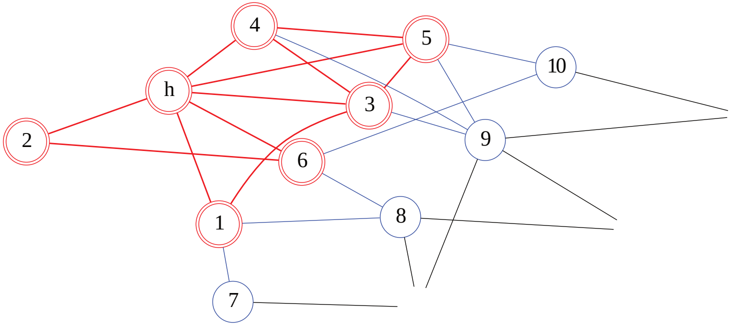
Plant a fingerprint.

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



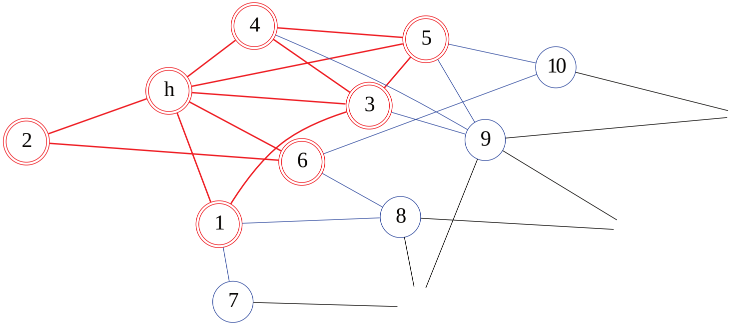
Plant a fingerprint.

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



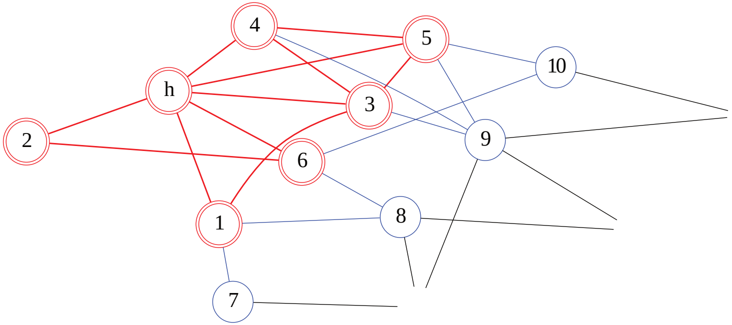
Plant a fingerprint.

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



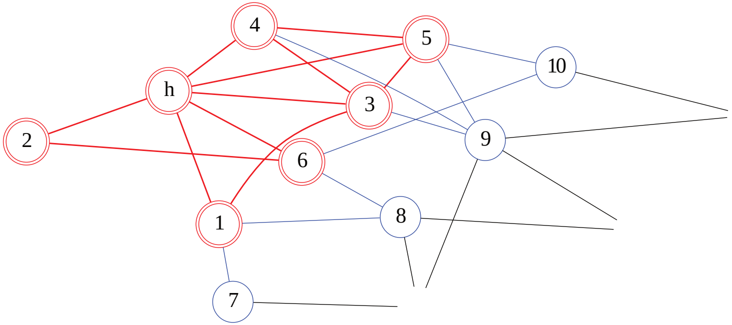
Plant a fingerprint.

The **ordered internal degree sequence**
$\mathcal{S}_D = \langle 2, 2, 2, 3, 3, 4 \rangle$.



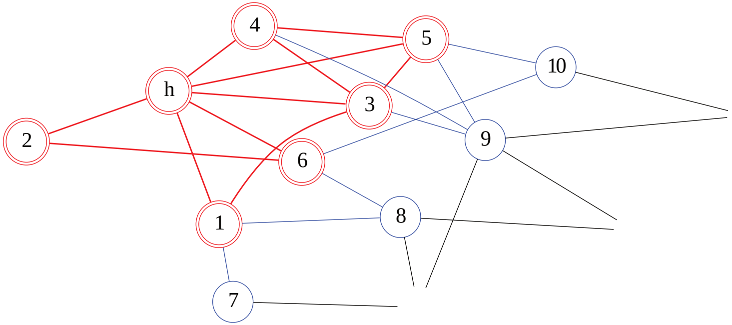
Plant a fingerprint.

Bob finds $\mathcal{S}_D(v_7) = \langle 2 \rangle$, $\mathcal{S}_D(v_8) = \langle 2, 2 \rangle$, $\mathcal{S}_D(v_9) = \langle 3, 3, 4 \rangle$, and $\mathcal{S}_D(v_{10}) = \langle 2, 3 \rangle$.



Plant a fingerprint.

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



Plant a fingerprint.

The details.

Create $V_F = \{v_h, v_1, v_2, \ldots\}$.
Given connectivity between V_F and V_S.
Connect v_h with v for all $v \in V_F - \{v_h\}$.
loop
  for all pairs $v_a \neq v_b$ in $V_F - \{v_h\}$ **do**
    Randomly connect v_a to v_b.
  for all $u \in V_S$ **do**
    Find $\mathcal{S}_D(u)$.
  if $\mathcal{S}_D(u)$ are mutually distinct for all $u \in V_S$ **then**
    return

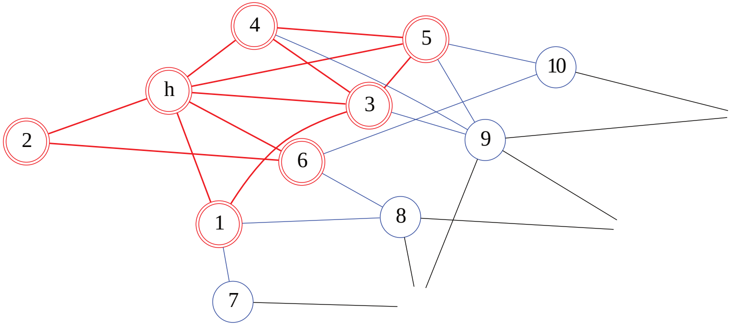
Recover the fingerprint.

Match the **fingerprint secrets**.

- Degree of v_h.
- The ordered internal degree sequence.

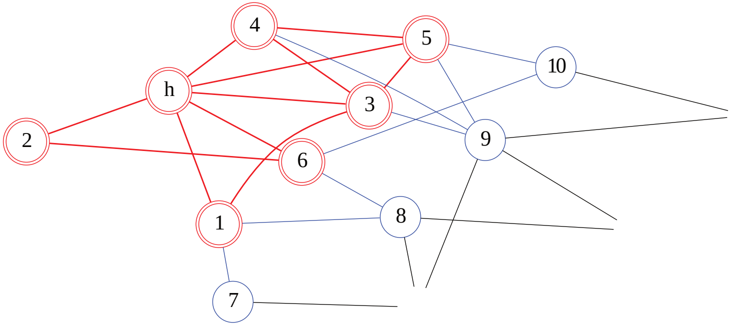
Recover the fingerprint.

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



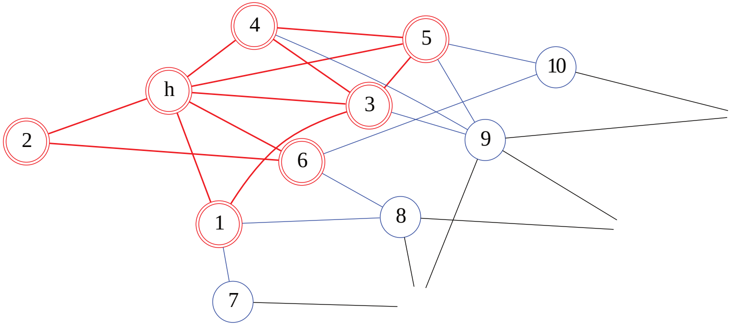
Recover the fingerprint.

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).



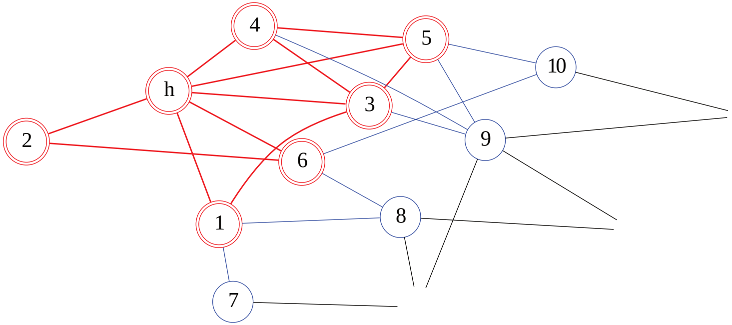
Recover the fingerprint.

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



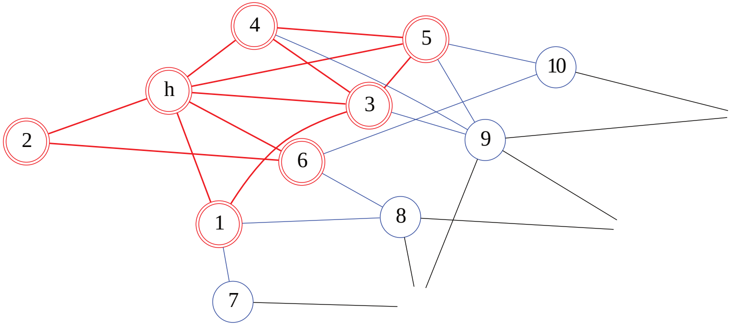
Recover the fingerprint.

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.



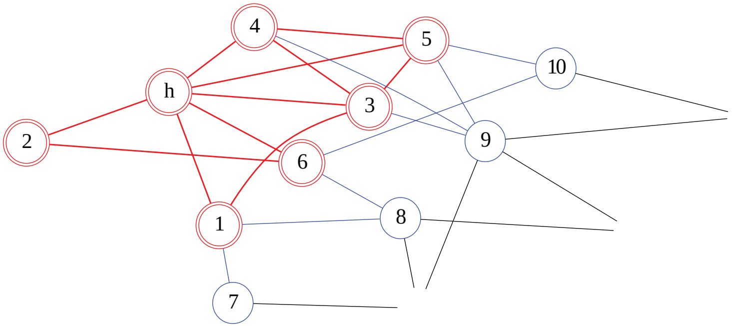
Recover and identify.

The details.

for all $u \in G_T$ **do**
    if $\deg(u) = |V_F| - 1$ **then**
        $U \leftarrow$ 1-hop neighborhood of u
        for all $v \in U$ **do**
            $d(v) \leftarrow$ number of v's neighbors in $U \cup \{u\}$
        $s(u) \leftarrow \mathrm{sort}(d(v)|v \in U)$
        if $s(u) = \mathcal{S}_D$ **then**
            $V \leftarrow$ exact 2-hop neighborhood of u
            for all $w \in V$ **do**
                $U(w) \leftarrow w$'s neighbors in U
                $s(w) \leftarrow \mathrm{sort}(d(v)|v \in U(w))$
            if $\langle s(w)|w \in V\rangle = \langle \mathcal{S}_D(v)|v \in V_S\rangle$ **then**
                {$w \in V$ is identified with $v \in V_S$ if
                $s(w) = \mathcal{S}_D(v)$}

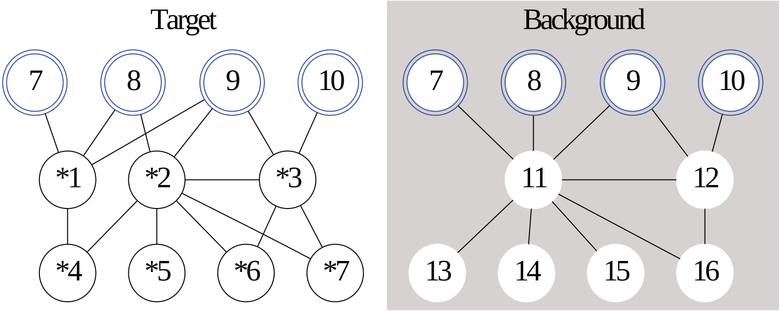
From **Seed** to Grow.

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



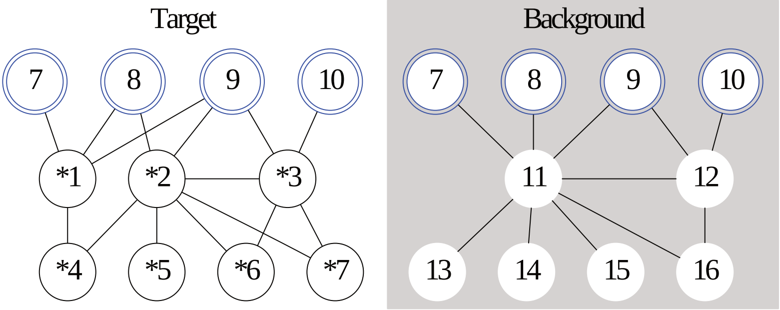
From Seed to **Grow**.

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



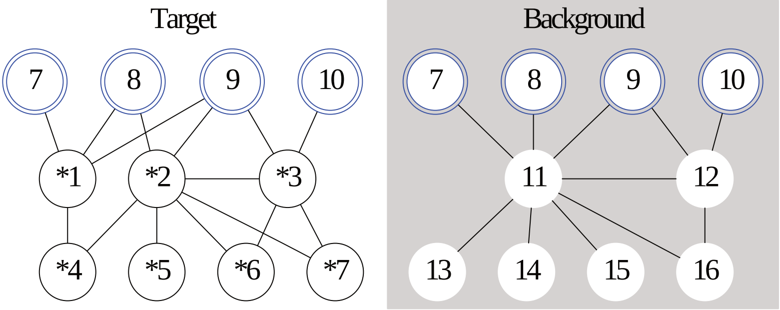
Grow the seeds.

Measuring structural **similarity**, or **equivalently**, **dissimilarity**.



Grow the seeds.

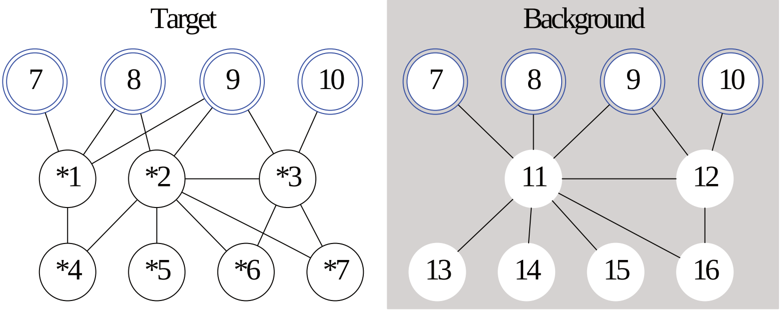
- Define $\mathcal{N}_m^T(u)$: $u \in V_T$'s **mapped** neighbors.
- Example: $\mathcal{N}_m^T(u_{*1}) = \{u_7, u_8, u_9\}$.
- Similar definition $\mathcal{N}_m^B(v)$ for $v \in V_B$.



Grow the seeds.

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

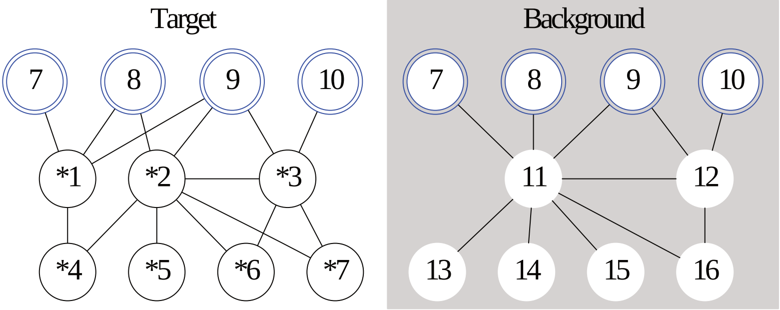
$$\Delta_T(u,v) = \frac{|\mathcal{N}_m^T(u) - \mathcal{N}_m^B(v)|}{|\mathcal{N}_m^T(u)|}, \Delta_B(u,v) = \frac{|\mathcal{N}_m^B(v) - \mathcal{N}_m^T(u)|}{|\mathcal{N}_m^B(v)|}.$$



Grow the seeds.

Bob does the maths...

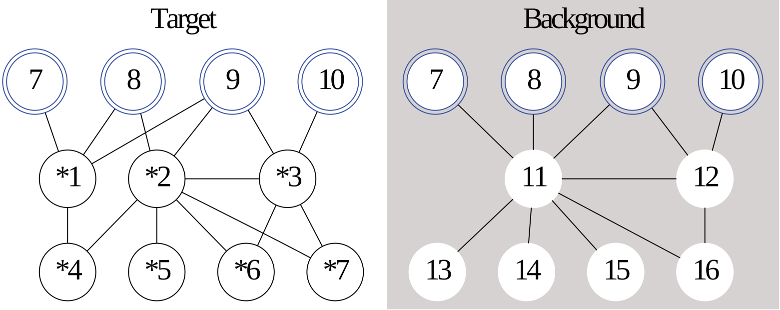
Δ	u_{*1}	u_{*2}	u_{*3}
v_{11}	(0.00, 0.00)	(0.00, 0.33)	(0.50, 0.67)
v_{12}	(0.67, 0.50)	(0.50, 0.50)	(0.00, 0.00)



Grow the seeds.

. . . and find most **similar** matches.

Δ	u_{*1}	u_{*2}	u_{*3}
v_{11}	**(0.00, 0.00)**	(0.00, 0.33)	(0.50, 0.67)
v_{12}	(0.67, 0.50)	(0.50, 0.50)	**(0.00, 0.00)**



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?

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

$\Delta_X(x)$	Absolute difference between x and its **closest different** value in X.
$\#_X(x)$	Multitude of x in X.
$\sigma(X)$	Standard deviation of X.

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?

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

$\Delta_X(x)$	Absolute difference between x and its **closest different** value in X.
$\#_X(x)$	Multitude of x in X.
$\sigma(X)$	Standard deviation of X.

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?

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

$\Delta_X(x)$	Absolute difference between x and its **closest different** value in X.
$\#_X(x)$	Multitude of x in X.
$\sigma(X)$	Standard deviation of X.

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?

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

$\Delta_X(x)$	Absolute difference between x and its **closest different** value in X.
$\#_X(x)$	Multitude of x in X.
$\sigma(X)$	Standard deviation of X.

Grow

The details.

Given the initial seed V_S.
$C = \emptyset$
loop
  $C_T \leftarrow \{u \in V_T | u$ connects to $V_S\}$
  $C_B \leftarrow \{v \in V_B | v$ connects to $V_S\}$
  if $(C_T, C_B) \in C$ **then**
    return V_S
  $C \leftarrow C \cup \{(C_T, C_B)\}$
  for all $(u, v) \in (C_T, C_B)$ **do**
    Compute $\Delta_T(u, v)$ and $\Delta_B(u, v)$.
  $S \leftarrow \{(u, v) | \Delta_T(u, v)$ and $\Delta_B(u, v)$ are smallest among conflicts$\}$
  for all $(u, v) \in S$ **do**
    if (u, v) has no conflict in S *or* (u, v) has the uniquely largest eccentricity among conflicts in S **then**
      $V_S \leftarrow V_S \cup \{(u, v)\}$

Inspirations.

L. Backstrom, C. Dwork, and J. Kleinberg.

Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography.

In *Proc. of ACM International Conference on World Wide Web (WWW)*, 2007.

Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee.

Measurement and analysis of online social networks.

In *Proc. of ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2007.

A. Narayanan and V. Shmatikov.

De-anonymizing social networks.

In *Proc. of IEEE Symposium on Security and Privacy*, 2009.

Questions?

Thank you for your attention!

Datasets.

Dataset	Vertex	Edges
Livejournal [MMG07]	5.2 million	72 million
emailWeek[1]	200	$1,676$

Dataset	$\lvert V_T \rvert$	$\lvert V_B \rvert$	$\lvert V_T \cap V_B \rvert$
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.

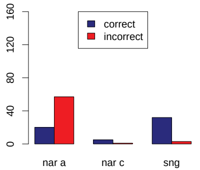
n	10	11	12	13
estimate	1.89×10^6	9.70×10^7	9.03×10^8	1.54×10^{11}

A comparative study.

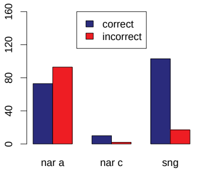
- We were inspired by Narayanan and Shmatikov [NS09].
- So we compare the Grow algorithm with theirs.
- Narayanan and Shmatikov algorithm [NS09] (Narayanan for short) has a manadatory parameter for adjusting matching aggressiveness.

Variant	Parameter	Abbreviation
Conservative	1	nar c
Aggressive	0.0001	nar a

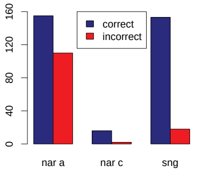
Different initial seed sizes.



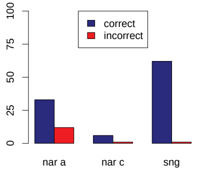
$|V_S| = 5$, Livejournal



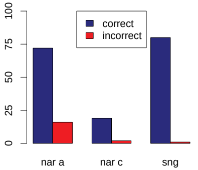
$|V_S| = 10$, Livejournal



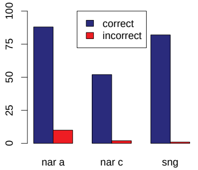
$|V_S| = 15$, Livejournal



$|V_S| = 5$, emailWeek

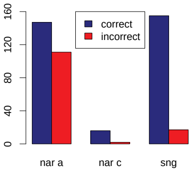


$|V_S| = 10$, emailWeek

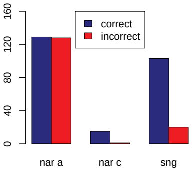


$|V_S| = 15$, emailWeek

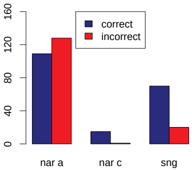
Edge perturbation.



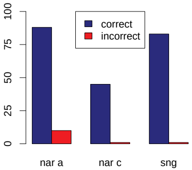
0.5%, Livejournal



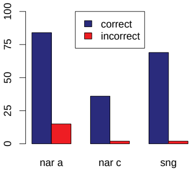
1%, Livejournal



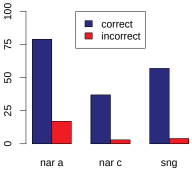
1.5%, Livejournal



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**!