An Information Theoretic Framework for Active De-anonymization in Social Networks Based on Group Memberships

Farhad Shirani, Siddharth Garg and Elza Erkip
New York University
Allerton, Oct. 2017
Active Attacks

- A website acts as an attacker.
Active Attacks

- A website acts as an attacker.
- The visitor is the victim.
Active Attacks

- A website acts as an attacker.
- The visitor is the victim.
- Attacker wants to de-anonymize the victim.

An Information Theoretic Framework for Active De-anonymization in Social Networks Based on Group Memberships

Farhad Shirani, Siddharth Garg and Elza Erkip

New York University
Active Attacks

- A website acts as an attacker.
- The visitor is the victim.
- Attacker wants to de-anonymize the victim.
- Attacker has access to victim’s IP.
- Attacker wants user’s real-world identity.
Active Attacks

- A website acts as an attacker.
- The visitor is the victim.
- Attacker wants to de-anonymize the victim.
- Attacker has access to victim’s IP.
- Attacker wants user’s real-world identity.
- Objective:
  1. Find defensive strategies against attacks.
  2. Find malicious users in the network.
Active Attacks

- Attacker uses browsing history sniffing.
Active Attacks

- **Attacker** uses browsing history sniffing.
- **Victim**’s social media activity is used.
Active Attacks

- **Attacker** uses browsing history sniffing.
- **Victim’s** social media activity is used.
- **Attacker** sends queries to de-anonymize the **victim**.
Active Attacks

- **Attacker** uses browsing history sniffing.
- **Victim’s** social media activity is used.
- **Attacker** sends queries to de-anonymize the **victim**.
- **Attacker’s** goal:
  1. Maximize reliability.
  2. Minimize number of queries.
1. Condition for successful attacks?
1. Condition for successful attacks?
2. Probability of success?
1. Condition for successful attacks?
2. Probability of success?
3. Fundamental limits of active attack strategies?
1. Condition for successful attacks?
2. Probability of success?
3. Fundamental limits of active attack strategies?
4. Effective defensive strategies?
Prior Work

(1) http://www.facebook.com/home.php?ref=home
(2) http://www.facebook.com/ajax/profile/picture/upload.php?id=[userID]
(3) http://www.facebook.com/group.php?gid=[groupID]&v=info&ref=nf
(4) https://www.xing.com/net/[groupId]/forums
(5) http://www.amazon.com/tag/[groupId]/
(6) http://community.ebay.de/clubstart.htm?clubId=[groupId]
Prior Work

(1) http://www.facebook.com/home.php?ref=home
(2) http://www.facebook.com/ajax/profile/picture/upload.php?id=[userID]
(3) http://www.facebook.com/group.php?gid=[groupID]&v=info&ref=nf
(4) https://www.xing.com/net/[groupID]/forums
(5) http://www.amazon.com/tag/[groupID]/
(6) http://community.ebay.de/clubstart.htm?clubId=[groupID]

- **Query types:**
  1. **UID:** Ask if the unknown victim $u_J$ is user $u_i$ in the network.
Prior Work

(1) http://www.facebook.com/home.php?ref=home
(2) http://www.facebook.com/ajax/profile/picture/upload.php?id=[userID]
(3) http://www.facebook.com/group.php?gid=[groupID]&v=info&ref=nf
(4) https://www.xing.com/net/[groupID]/forums
(5) http://www.amazon.com/tag/[groupID]/
(6) http://community.ebay.de/clubstart.htm?clubId=[groupID]

- Query types:
  1. UID: Ask if the unknown victim $u_J$ is user $u_i$ in the network.
  2. CM: a CM asks if the victim $u_J$ is a member of the group $r_j$. 
Prior Work

(1) http://www.facebook.com/home.php?ref=home
(2) http://www.facebook.com/ajax/profile/picture/upload.php?id=[userID]
(3) http://www.facebook.com/group.php?gid=[groupID]&v=info&ref=nf
(4) https://www.xing.com/net/[groupID]/forums
(5) http://www.amazon.com/tag/[groupID]/
(6) http://community.ebay.de/clubstart.htm?clubId=[groupID]

- **Query types:**
  - 1. **UID**: Ask if the unknown victim $u_J$ is user $u_i$ in the network.
  - 2. **CM**: a CM asks if the victim $u_J$ is a member of the group $r_J$.

- **Naive strategy**: Query all users in the social network.
Prior Work

(1) http://www.facebook.com/home.php?ref=home
(2) http://www.facebook.com/ajax/profile/picture/upload.php?id=[userID]
(3) http://www.facebook.com/group.php?gid=[groupID]&v=info&ref=nf
(4) https://www.xing.com/net/[groupID]/forums
(5) http://www.amazon.com/tag/[groupID]/
(6) http://community.ebay.de/clubstart.htm?clubId=[groupID]

- Query types:
  1. **UID**: Ask if the unknown victim \( u_J \) is user \( u_i \) in the network.
  2. **CM**: a CM asks if the victim \( u_J \) is a member of the group \( r_j \).

- **Naive strategy**: Query all users in the social network.

- **Improved Strategy**: 

  1. Query all group memberships.
Query types:

1. UID: Ask if the unknown victim $u_J$ is user $u_i$ in the network.
2. CM: a CM asks if the victim $u_J$ is a member of the group $r_j$.

**Naive strategy:** Query all users in the social network.

**Improved Strategy:**


1. Query all group memberships.
2. Intersect groups and query users.
Contributions

- Previous efforts: ad-hoc strategies without performance analysis.
Contributions

• Previous efforts: ad-hoc strategies without performance analysis.
• Our work:
  1. Unifying statistical framework for devising and analyzing active attacks.
Contributions

- Previous efforts: ad-hoc strategies without performance analysis.
- Our work:
  1. Unifying statistical framework for devising and analyzing active attacks.
  2. Propose new attack strategies for active deanonymization.
Contributions

- Previous efforts: ad-hoc strategies without performance analysis.
- Our work:
  1. Unifying statistical framework for devising and analyzing active attacks.
  2. Propose new attack strategies for active deanonymization.
  3. Analyze the asymptotic performance of the new strategies.
Network Model

- The social network groups are modeled by a bipartite graph.
Network Model

- The social network groups are modeled by a bipartite graph.

- A bipartite graph $g^0$ consists of a triple of sets $(\mathcal{U}^0, \mathcal{R}^0, \mathcal{E}^0)$.
- The user set $\mathcal{U}^0 = \{u_1, u_2, \cdots, u_m\}$. 

\[
\begin{align*}
\text{Users} & \quad \text{Groups} \\
\bullet u_1 & \quad \bullet r_1 \\
\bullet u_2 & \quad \bullet r_2 \\
\bullet u_{m-1} & \quad \bullet r_{n-1} \\
\bullet u_m & \quad \bullet r_n
\end{align*}
\]
Network Model

- The social network groups are modeled by a bipartite graph.

A bipartite graph $g^0$ consists of a triple of sets $(\mathcal{U}^0, \mathcal{R}^0, \mathcal{E}^0)$.
- The user set $\mathcal{U}^0 = \{u_1, u_2, \cdots, u_m\}$.
- The class set $\mathcal{R}^0 = \{r_1, r_2, \cdots, r_n\}$. 
The social network groups are modeled by a bipartite graph.

A bipartite graph $g^0$ consists of a triple of sets $(\mathcal{U}^0, \mathcal{R}^0, \mathcal{E}^0)$.
- The user set $\mathcal{U}^0 = \{u_1, u_2, \cdots, u_m\}$.
- The class set $\mathcal{R}^0 = \{r_1, r_2, \cdots, r_n\}$.
- The set of edges is denoted by $\mathcal{E}^0 \subset \{ (i, j) | i \in [1, m], j \in [1, n] \}$. 
Network Model

- The social network groups are modeled by a bipartite graph.

A bipartite graph $g^0$ consists of a triple of sets $(U^0, R^0, E^0)$.

- The user set $U^0 = \{u_1, u_2, \cdots, u_m\}$.
- The class set $R^0 = \{r_1, r_2, \cdots, r_n\}$.
- The set of edges is denoted by $E^0 \subset \{(i, j)|i \in [1, m], j \in [1, n]\}$.

We assume a random graph edges have the same distribution.
Network Model

- The social network groups are modeled by a bipartite graph.

\[
\begin{align*}
\text{Users} & \quad \text{Groups} \\
\quad u_1 & \rightarrow \quad r_1 \\
\quad u_2 & \rightarrow \quad r_2 \\
\quad \vdots & \quad \quad \quad \vdots \\
\quad u_{m-1} & \rightarrow \quad r_{n-1} \\
\quad u_m & \rightarrow \quad r_n
\end{align*}
\]

- A bipartite graph \( g^0 \) consists of a triple of sets \((\mathcal{U}^0, \mathcal{R}^0, \mathcal{E}^0)\).
- The user set \( \mathcal{U}^0 = \{u_1, u_2, \cdots, u_m\} \).
- The class set \( \mathcal{R}^0 = \{r_1, r_2, \cdots, r_n\} \).
- The set of edges is denoted by
  \[ \mathcal{E}^0 \subset \{(i, j) | i \in [1, m], j \in [1, n]\} \].
- We assume a random graph edges have the same distribution.
Motivation

Problem Formulation

Attack Strategies

Models

Attacker’s Model

- **Offline phase**: attacker scans the network to get $g^1$.

  Each edge is sampled independently under identical noise.

Farhad Shirani, Siddharth Garg and Elza Erkip

New York University

An Information Theoretic Framework for Active De-anonymization in Social Networks Based on Group Memberships
Attacker’s Model

**Offline phase:** attacker scans the network to get $g^1$.

- Each edge is sampled independently under identical noise.
- **Online phase:** attacker queries the victim’s browser history.

![Diagram of users and groups with edges and labels](image-url)
Attacker’s Model

- **Offline phase**: attacker scans the network to get $g^1$.

- Each edge is sampled independently under identical noise.

- **Online phase**: attacker queries the victim’s browser history.

- We assume noisy responses to queries.
Objective

- We assume that the user index is chosen uniformly from the set $[1, m]$. 
Objective

- We assume that the user index is chosen uniformly from the set \([1, m]\).
- We consider zero-error deanonymization.
Objective

- We assume that the user index is chosen uniformly from the set $[1, m]$.
- We consider zero-error deanonymization.

**Definition**

The minimum expected queries is defined as

$$\bar{Q} \triangleq \min_{x_t, t \in \mathbb{N}} \mathbb{E}(Q),$$

where $x_t$ is the query function at time $t$. 
Objective

- We assume that the user index is chosen uniformly from the set $[1, m]$.
- We consider zero-error deanonymization.

Definition

The minimum expected queries is defined as

$$ \bar{Q} \triangleq \min_{x_t, t \in \mathbb{N}} \mathbb{E}(Q), $$

where $x_t$ is the query function at time $t$.

- Objective: characterize $\bar{Q}$ given $P_{Z|Y}$, $P_{E_0}$, $P_{E_1|E_0}$, $m$ and $n$. 
Objective

- We assume that the user index is chosen uniformly from the set $[1, m]$.
- We consider zero-error deanonymization.

Definition

The minimum expected queries is defined as

$$
\bar{Q} \triangleq \min_{x_t, t \in \mathbb{N}} \mathbb{E}(Q),
$$

where $x_t$ is the query function at time $t$.

- Objective: characterize $\bar{Q}$ given $P_{Z|Y}$, $P_{E_0}$, $P_{E_1|E_0}$, $m$ and $n$.
- Trivial bound: $\log m \leq \bar{Q} \leq \frac{m}{2}$.
Motivation

Problem Formulation

Attack Strategies

The CIS strategy

Class Intersection Strategy

- Assumptions:
  1. Attacker has noiseless access to network graph.
  2. Query responses are noiseless.
Class Intersection Strategy

- Assumptions:
  1. Attacker has noiseless access to network graph.
  2. Query responses are noiseless.

- We propose the class intersection strategy.

- Builds upon the algorithm in [Kruegel et. al. 2010]
The CIS strategy

Class Intersection Strategy

- **Assumptions:**
  1. Attacker has noiseless access to network graph.
  2. Query responses are noiseless.

- We propose the class intersection strategy.

- Builds upon the algorithm in [Kruegel et. al. 2010]

- **Fix** $n' < n$.

- **1st Step:** attacker sends CM queries for a subset of size $n'$ of the groups.
The CIS strategy

Class Intersection Strategy

Assumptions:
1. Attacker has noiseless access to network graph.
2. Query responses are noiseless.

We propose the class intersection strategy.

Builds upon the algorithm in [Kruegel et. al. 2010]

Fix $n' < n$.

1st Step: attacker sends CM queries for a subset of size $n'$ of the groups.

2nd Step: the attacker intersects the groups which the user is a member of.
The CIS strategy

Class Intersection Strategy

- Assumptions:
  1. Attacker has noiseless access to network graph.
  2. Query responses are noiseless.

- We propose the class intersection strategy.
- Builds upon the algorithm in [Kruegel et. al. 2010]
- Fix \( n' < n \).

- **1st Step:** attacker sends CM queries for a subset of size \( n' \) of the groups.
- **2nd Step:** the attacker intersects the groups which the user is a member of.
- **3rd Step:** the attacker sends additional UID queries until he finds the user index.
Class Intersection Strategy

- Assumptions:
  1. Attacker has noiseless access to network graph.
  2. Query responses are noiseless.
- We propose the class intersection strategy.
- Builds upon the algorithm in [Kruegel et. al. 2010]
- Fix $n' < n$.
- **1st Step:** attacker sends CM queries for a subset of size $n'$ of the groups.
- **2nd Step:** the attacker intersects the groups which the user is a member of.
- **3rd Step:** the attacker sends additional UID queries until he finds the user index.
If number of CM queries $n' = \frac{1}{\lambda} \log_2 m$, then
\[ E(Q_{CIS}) \leq \frac{1}{\lambda} \log_2 m + O(\sqrt{\log_2 m}), \text{ where } \lambda \text{ is constant in } m. \]
The CIS strategy

- If number of CM queries \( n' = \frac{1}{\lambda} \log_2 m \), then
  \[ \mathbb{E}(Q_{CIS}) \leq \frac{1}{\lambda} \log_2 m + O(\sqrt{\log_2 m}) \], where \( \lambda \) is constant in \( m \).
- Applicable when \( n \) grows at least logarithmically with \( m \).
If number of CM queries \( n' = \frac{1}{\lambda} \log_2 m \), then

\[
\mathbb{E}(Q_{CIS}) \leq \frac{1}{\lambda} \log_2 m + O(\sqrt{\log_2 m}),
\]

where \( \lambda \) is constant in \( m \).

- Applicable when \( n \) grows at least logarithmically with \( m \).
- Example: Facebook social network.
If number of CM queries $n' = \frac{1}{\lambda} \log_2 m$, then

$$E(Q_{CIS}) \leq \frac{1}{\lambda} \log_2 m + O(\sqrt{\log_2 m})$$

where $\lambda$ is constant in $m$.

Applicable when $n$ grows at least logarithmically with $m$.

Example: Facebook social network.

1. Number of Users $\sim$ 2 billion.
2. Number of groups $> 600$ million.
The CIS strategy

- If number of CM queries \( n' = \frac{1}{\lambda} \log_2 m \), then
  \[
  \mathbb{E}(Q_{CIS}) \leq \frac{1}{\lambda} \log_2 m + O(\sqrt{\log_2 m}), \text{ where } \lambda \text{ is constant in } m.
  \]
- Applicable when \( n \) grows at least logarithmically with \( m \).
- Example: Facebook social network.
  1. Number of Users \( \sim \) 2 billion.
  2. Number of groups > 600 million.
  3. Logarithm of number of users \( \sim \) 30.
The CIS strategy

- If number of CM queries $n' = \frac{1}{\lambda} \log_2 m$, then
  \[ \mathbb{E}(Q_{CIS}) \leq \frac{1}{\lambda} \log_2 m + O(\sqrt{\log_2 m}), \text{ where } \lambda \text{ is constant in } m. \]
- Applicable when $n$ grows at least logarithmically with $m$.
- Example: Facebook social network.
  1. Number of Users $\sim 2$ billion.
  2. Number of groups $> 600$ million.
  3. Logarithm of number of users $\sim 30$.
  4. Expected number of queries $\mathbb{E}(Q_{CIS}) \sim 300$. 
MAP Strategy

Assumptions:

1. Attacker has noiseless access to network graph.
2. Query responses are noiseless.
**MAP Strategy**

- **Assumptions:**
  1. Attacker has noiseless access to network graph.
  2. Query responses are noiseless.
- Use maximum a posteriori probabilities.
MAP Strategy

Assumptions:
1. Attacker has noiseless access to network graph.
2. Query responses are noiseless.

Use maximum a posteriori probabilities.

1st Step: attacker sends CM queries for a subset of size $n'$ of the groups.
MAP Strategy

Assumptions:

1. Attacker has noiseless access to network graph.
2. Query responses are noiseless.

Use maximum a posteriori probabilities.

1st Step: attacker sends CM queries for a subset of size $n'$ of the groups.

2nd Step: the attacker sorts users using the all of the signature vector.
MAP Strategy

- Assumptions:
  1. Attacker has noiseless access to network graph.
  2. Query responses are noiseless.

- Use maximum a posteriori probabilities.

1st Step: attacker sends CM queries for a subset of size $n'$ of the groups.

2nd Step: the attacker sorts users using the all of the signature vector.

3rd Step: the attacker sends additional UID queries and finds the user index.
MAP Strategy

- Assumptions:
  1. Attacker has noiseless access to network graph.
  2. Query responses are noiseless.

- Use maximum a posteriori probabilities.

- **1st Step:** attacker sends CM queries for a subset of size $n'$ of the groups.

- **2nd Step:** the attacker sorts users using the all of the signature vector.

- **3rd Step:** the attacker sends additional UID queries and finds the user index.

- Results in improved bounds on the logarithm’s coefficient.
Typical Set Strategy

- **1st Step**: attacker sends $n' < n$ CM queries for a subset of the groups.
Typical Set Strategy

- **1st Step**: attacker sends $n' < n$ CM queries for a subset of the groups.
- **2nd Step**: attacker finds the user indices which have signatures which are jointly typical with the received vector.
Typical Set Strategy

- **1st Step:** attacker sends $n' < n$ CM queries for a subset of the groups.
- **2nd Step:** attacker finds the user indices which have signatures which are jointly typical with the received vector.
- **3rd Step:** attacker sends additional UID queries for jointly typical users.
Typical Set Strategy

- **1st Step:** attacker sends $n' < n$ CM queries for a subset of the groups.
- **2nd Step:** attacker finds the user indices which have signatures which are jointly typical with the received vector.
- **3rd Step:** attacker sends additional UID queries for jointly typical users.
- **4th Step:** If user index not found, repeats the process for new CM queries.
Typical Set Strategy

1st Step: attacker sends $n' < n$ CM queries for a subset of the groups.

2nd Step: attacker finds the user indices which have signatures which are jointly typical with the received vector.

3rd Step: attacker sends additional UID queries for jointly typical users.

4th Step: If user index not found, repeats the process for new CM queries.

5th Step: If process fails in $l$ iterations do an exhaustive search.
The TSS strategy

Typical Set Strategy

- **1st Step:** attacker sends $n' < n$ CM queries for a subset of the groups.
- **2nd Step:** attacker finds the user indices which have signatures which are jointly typical with the received vector.
- **3rd Step:** attacker sends additional UID queries for jointly typical users.
- **4th Step:** If user index not found, repeats the process for new CM queries.
- **5th Step:** If process fails in $l$ iterations do an exhaustive search.
The TSS strategy

\[ E(Q_{TSS}) = \left(n' + m 2^{n'(I(U;Y) \pm \epsilon)}\right) \left(\frac{n' \epsilon^2}{n' \epsilon^2 - 1}\right) + \frac{m}{(n' \epsilon^2)^l}, \quad (1) \]

- For \( n' \triangleq \frac{1}{I(U;Y)+\epsilon} \log m, \quad \epsilon \triangleq n' - \frac{1}{3} \) and \( l \triangleq \frac{\log m}{\log n' \epsilon^2} \), the inequality

\[ E(Q_{TSS}) \leq \frac{1}{I(U;Y)} \log m + O(\log^{\frac{2}{3}} m) \]

holds.
Conclusion

- We constructed an information theoretic framework for the active deanonymization problem.
- Showed the number of queries grows logarithmically with $m$.
- For the noiseless scenario, the bound on the performance is tight.
We constructed an information theoretic framework for the active deanonymization problem.
Showed the number of queries grows logarithmically with $m$.
For the noiseless scenario, the bound on the performance is tight.

**Future Directions:**
1. Improvements using information thresholds.
2. Non-equiprobable user indices and edge probabilities.
3. Noise model which is correlated with the user-class pair.
4. Multiuser deanonymization involving several websites and users.
5. Test results on data sets from available social networks.
The TSS strategy

Seed-based de-anonymizability quantification of social networks.


A practical attack to de-anonymize social network users.


Exploiting innocuous activity for correlating users across sites.

The TSS strategy


Pedram Pedarsani and Matthias Grossglauser. On the privacy of anonymized networks.
The TSS strategy

In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’11, pages 1235–1243, New York, NY, USA, 2011. ACM.

E. Onaran, S. Garg, and E. Erkip.
Optimal de-anonymization in random graphs with community structure.

Daniel Cullina and Negar Kiyavash.
Improved achievability and converse bounds for erdos-renyi graph matching.

B. Zielosko M. Ju. Moshkov, M. Pilizczuk.
*Partial Covers, Reducts and Decision Rules in Rough Sets.*
The TSS strategy


I. Csizár and J. Korner.
Information Theory: Coding Theorems for Discrete Memoryless Systems.

V. Kostina, Y. Polyanskiy, and S. Verd.
Joint source-channel coding with feedback.

C. Vercellis.
A probabilistic analysis of the set covering problem.