Privacy-Preserving Twitter Browsing through Obfuscation

Panagiotis Papadopoulos
Master Thesis Presentation

University of Crete
School of Sciences and Engineering
Computer Science Department
Microblogging Services

A popular way for information sharing and communication. Users are able to have timely access to all information available from various providers.
Publish-Subscribe Model

• Information providers (*channels*)
  - politicians
  - news agencies or news reporters
  - hospitals or doctors
  - activists
  - artists
  - religious organizations
  - other communities

• Users **subscribe** to (or **follow**) channels
  ➢ In this way they receive **interesting** information in a timely manner
Publish-Subscribe Model (example)

channel subscription process

Channel 1

Barack Obama
@BarackObama
This account is run by Organizing for Action staff. Tweets from the President are signed -bo.
Washington, DC - barackobama.com

Channel 2

American Cancer Soc
@AmericanCancer
The official American Cancer Society Twitter stream. 100 years ago, we began the fight of a lifetime. Today, you can help us finish the fight.
1-800-227-2345
United States - cancer.org

The official American Cancer Society Twitter stream. 100 years ago, we began the fight of a lifetime. Today, you can help us finish the fight.
1-800-227-2345
United States - cancer.org
Publish-Subscribe Model (example)

User’s timeline

User’s following

Following

Barack Obama  @BarackObama
This account is run by Organizing for Action staff. Tweets from the President are signed -bo.

American Cancer Soc  @AmericanCancer
The official American Cancer Society Twitter stream. 100 years ago, we began the fight of a lifetime. Today, you can help us finish the fight. 1-800-227-2345
What about users’ privacy?

• The microblogging service knows a user’s interests based on the user’s channel subscriptions
  • Political preferences (e.g., Barack Obama)
  • Health issues (e.g., cancer)

• Detailed user profiling
  • Privacy-sensitive channels
  • Can be used for many purposes
  • Beyond the control of the users
Threat Model

• An “honest but curious” microblogging service
  ➢ capable of passively gain knowledge about users’ interests by monitoring the channels they follow.
  ➢ knowledge that can be given/sold to third parties e.g. advertisers

• Users that need access to timely information and they are able to follow individual channels.

• A channel can be the account of a physical person, a corporation, a politician’s office, and so on.
HOW CAN WE PROTECT USERS’ PRIVACY?
Existing approaches:

1. **No login**
   - Limited information available to non-logged in users.
   - Correlation of served content + IP address.

2. **Pseudonym or fake account**
   - IP address, third-party tracking cookies, browser fingerprints can reveal user’s identity

3. **Anonymization service (e.g., Tor)**
   - Logging into the service, possibility of Tor nodes blocking

4. **Tor + Fake account**
   - Cookies and fingerprints gathered through anonymous and eponymous browsing sessions

5. **Fake account + Tor + VM per browsing session**
   - Too complex for ordinary users and mobile devices
But…

How can we hide users’ interests in a world where it will be practically impossible to hide one’s real identity?

Our thesis is:

users’ interests can be protected using obfuscation
k-subscription

For each privacy-sensitive channel \( C_1 \) a user really wants to follow with k-subscription, the user will also randomly follow \( k - 1 \) additional sensitive channels acting as noise: \( C_1, C_2, C_3, \ldots, C_k \) (where \( C_2, C_3, \ldots, C_k \) are noise channels)

This way:

- The service cannot identify a user’s actual choices
- Hide the choices of other users as well
  - The service cannot identify the users that are actually interested in \( C_1 \)

**Note:** All channels \( C_1, C_2, C_3, \ldots, C_k \) belong to the same set \( S \) of privacy-sensitive channels
k-subscription in action

Following

- **Catholic Church**
  - @catholicEW
  - Serving the Catholic Bishops of England and Wales. Following/RTs ≠ endorsement.

- **American Cancer Soc**
  - @AmericanCancer
  - The official American Cancer Society Twitter stream. 100 years ago, we began the fight of a lifetime. Today, you can help us finish the fight. 1-800-227-2345

- **Everyday Health**
  - @HeartDiseases
  - Follow @HeartDiseases for the latest news and information on living a heart-healthy lifestyle, straight from the editors of @EverydayHealth.

- **HilaryClinton**
  - @HilaryClinton

- **Barack Obama**
  - @BarackObama
  - This account is run by Organizing for Action staff. Tweets from the President are signed -bo.

- **Alcohol Problems**
  - @AlcoholProbs
  - Alcohol, the cause and solution to all of life’s problems. For questions or promotion contact AlcoholProblems@gmail.com

Random noise

Real choices
Obfuscation algorithms

1. **Uniform sampling**
   - Randomly select every channel in $S$ as noise with *same* probability

2. **Proportional sampling**
   - Randomly select every channel in $S$ as noise with probability *proportional* to its *popularity*

**Multiple channels**

Following a set of *semantically-related* channels. *Can be easily identified by the service*

- Just choose proper $k$ so that there are *other* users that select the *same* set as noise
Implementation

- Browser extension for Google Chrome browser
- Using Twitter as case study

 Actual + Noise Channels

 twitter.com

 k-subscription

 Actual Channels

 Sensitive Channels Set $S$

Responsible for obfuscation, noise filtering, timeline construction
## Remove the effect of noise (1/2)

### What the microblogging service sees:

<table>
<thead>
<tr>
<th>Following</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catholic Church</td>
</tr>
<tr>
<td>American Cancer Soc</td>
</tr>
<tr>
<td>Everyday Health</td>
</tr>
<tr>
<td>Hilary Clinton</td>
</tr>
<tr>
<td>Barack Obama</td>
</tr>
<tr>
<td>Alcohol Problems</td>
</tr>
</tbody>
</table>

- **Real + noise channels**

### What the user sees:

<table>
<thead>
<tr>
<th>Following</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barack Obama</td>
</tr>
<tr>
<td>American Cancer Soc</td>
</tr>
</tbody>
</table>

- **Only real channels**
Remove the effect of noise (2/2)

What the microblogging service sees:

Barack Obama @BarackObama 7 Dec
"It shouldn't be a partisan issue."—President Obama on extending unemployment insurance. Watch the weekly address: OFA.BO/9hcWqE
View media
Reply Retweet Favorite More

Alcohol Problems @AlcoholProbs 7 Dec
All I want to do tonight is cuddle with YOU.

What the user sees:

American Cancer Soc @AmericanCancer 10h
Chemotherapy helps save lives. But how was it developed? Watch the video: bit.ly/1Swp6DC
View media
Reply Retweet Favorite More

Barack Obama @BarackObama 10h
Medical expansion and the new marketplace have helped these Americans get covered: OFA.BO/HRh1sD
Reply Retweet Favorite More

Tweets from real + noise channels

Tweets only from real channels
Disclosure Probability $P_C$

The probability that a user following channel $C$ is actually interested in $C$

Depends on

- channel’s popularity $p_C$
  - (e.g. number of followers)
- size of set $S$ ($|S|$)
  - Publicly released
- obfuscation level $k$
  - Can be inferred $\Rightarrow$ a user follows $k$ channels in short period
The $k$ parameter

Fine-tune the $k$ parameter to control the preferable privacy level and network overhead.
WHAT IS THE RIGHT K VALUE?
Choosing a value for $k$

- Analysis and simulation for disclosure probability as a function of $k$

- Experimental evaluation for network overhead as a function of $k$
ANALYTICAL EVALUATION
When channel popularity is high (10%), it is difficult to obfuscate with uniform sampling.

For channel popularity 1% and $k=100$, disclosure probability is 10%.

For channel popularity 0.1% and $k=100$, disclosure probability is 1%.
Proportional Sampling

|S| = 1000 channels

- For channel popularity 10% and $k=40$, disclosure probability is just 10%
- For channel popularity 1% and $k=50$, disclosure probability is 2%
Following Multiple Channels

For low values of $k$, disclosure probability increases with the number of sensitive channels because users interested in these $N$ channels are few.

For high values of $k$, disclosure probability decreases significantly with the number of channels because users interested in $N$ sensitive channels follow $k \times N$ channels in total, which leads to more noise.
SIMULATION-BASED EVALUATION
Use of set S with 7,000 sensitive channels using Twellow distribution can be approximated very well using a power law with exponential cutoff model.

\[ f(x) = x^a e^{bx} c \]

\((a=-0.073, b=-0.001, c=505747)\)
Number of sensitive channels users follow

0.85% follows 4 channels when 91.65% follows only 1 channel

can be approximated very well using a power law

$f(x) = ax^b$

$(a=55.6, b=-0.37)$
Simulation-Based Study

Disclosure probability decreases rapidly with $k$ for all channels and users.

- $|S| = 1,000$ channels
- $U = 1,000,000$ users
EXPERIMENTAL EVALUATION
Bandwidth Consumption Over Time

- Less than 1.5 Kbps to follow $k=100$ channels
- Large spike when loading Twitter’s page
- Between 0.5 – 1.0 Kbps
  Most of the bandwidth is used to download other information, which does not depend on $k$
Bandwidth Consumption: Initialization Stage

Twitter downloads content (like widgets, scripts, CSS profile images, trends)

k-subscription downloads tweets from all k channels in order to create a full timeline
Bandwidth consumption: Idle stage

![Graph showing bandwidth consumption vs. obfuscation level]
Browsing Latency

- 10 seconds to download and display Twitter page with Tor
- 7.7 seconds to download and display Twitter page when $k=100$
- 2.8 seconds to download and display Twitter page when $k=1$
- When the browser remains open (idle case) we do not observe any delay

$k$: Obfuscation level

Browsing Latency (seconds)
• Our tool is available as a Google Chrome browser extension

• This work has been published in the 29th Annual Computer Security Applications Conference (ACSAC ‘13) on Dec 2013 at New Orleans, USA
Conclusions

• *k-subscription*: an obfuscation-based approach for privacy-preserving Twitter browsing
  • Hide user’s subscriptions within selected noise
  • Hide user’s subscriptions within noise of *other* users
  • Add noise from a *common* set with sensitive channels

• Fine tuning the *k* parameter
  ➢ Disclosure probability
  ➢ Network overhead

• In a future where user’s identity cannot be hidden privacy could be achieved by:
  ➢ *obfuscation* and
  ➢ mutual collaboration between users.
BACKUP SLIDES
Posting Messages

k-subscription protects microblogging **browsing**:  
  - Does not aim to hide users’ interests when users want to post about a sensitive issue  
  - Does not aim to hide users’ interests when users want to retweet a post of a sensitive channel

For protecting user posts there are alternative solutions:  
  - Hummingbird, #h00t, etc. *(using post encryption)*
Short URLs

Short URL services usually cooperate with microblogging services. So these URL shortening services can be used to infer user’s interests based on user clicks on short URLs.

- k-subscription, when a user clicks on a short URL, resolves, transparently, on the background all short URLs in tweets from noise and real channels.
Formulas for Disclosure Probability $P_C$

**Uniform Sampling:**

$$P_C < \max\left(1/k, \frac{p_C}{p_C + (1 - p_C) \times (1 - (1 - 1/|S|)^{k-1})}\right)$$

**Proportional Sampling:**

$$P_C > \max\left(1/k, \frac{p_C}{p_C + (1 - p_C) \times (1 - (1 - p_C)^{k-1})}\right)$$

Following multiple channels $\mathbf{N}$:

$$P_{C_1,...,c_N} = \frac{p_{C_1,...,c_N}}{p_{C_1,...,c_N} + (1 - p_{C_1,...,c_N}) \times \left(\frac{|S|-N}{(k-1)N-N}\right) / \left(\frac{|S|}{(k-1)N}\right)}$$
Sensitive channels $S$

Maintained by a privacy-related organization

- Users may request, through k-subscription, new sensitive channels to be added in this set
- The set $S$ must be shielded against malicious users that tries to insert a large number of fake channels in order to increase disclosure probability
  - CAPTCHAs to avoid computer bots that inserts batches of fake channels
  - Use of Yahoo Term Extraction API in order to evaluate the channel’s sensitivity
  - Channel’s activity and channel’s audience validation
  - Channel’s audience evaluation: amount of followers to the amount of following ratio, number posts coming from API, duplicate or spam posts, posts with unrelated links.
Size of Sensitive Channels Set (1/2)

Linear relationship between obfuscation level and size of sensitive channels.

If we double the size of sensitive channels, we need twice as high $k$ for the same disclosure probability.

Channel popularity: 1%
Disclosure probabilities: 0.2, 0.1 and 0.05
Size of Sensitive Channels Set (2/2)

- in order to keep the disclosure probability constant:
  - if we double |S| -> we must double k value

- for a constant obfuscation level k:
  - larger |S| -> higher disclosure probability.

- very small |S| would easily give away a user’s true interests + limit the users’ choice for channels
  - if S contains n members, the microblogging service will be able to conclude with probability at least 1/n that the user is interested in the channel she follows.

- |S| must be enough so \(1/|S| < U_c/U\)
Why not N-tuples?

Whenever a user is interested in N related channels:

the \((k - 1) \times N\) noise channels that will follow will be selected in N-tuple groups, so that each N-tuple consists of N related noise channels.

However: the microblogging service may use different similarity metrics to identify related channels.
Time to follow a sensitive channel

It takes one minute to follow \( k = 100 \) channels.
CPU load ~ Initialization stage

less than 3% during the Idle stage
What about giving the wrong impressions?

User following illness-related channels or bankruptcy-related channels => worrying friends + family

• dummy account protects against worrying family
  - (but NOT against microblogging service, that can use IP tracking or cookies)

• followings can be organized in separate private lists (Twitter provides this option).
  - That option also, does not offer protection against Twitter
What about disappearing channels?

People close or delete their accounts:

• If users stop following channel D and it’s noise => correlate D’s disappearance with the users’ change of following patterns => users were interested in channel D.

• If users noise channels start disappearing => service will be in a better position to find the exact channel they are really interested in.
  ➢ add other noise? => NO, the service will figure out the noise channels

✓ users interested in D + users who not interested in D but have included D as noise => should do nothing! => the service will not be able to differentiate which users are interested in D and which are not.
k-subscription-UNIF

• when 10% of the users are interested in channel C:
  ➢ it would take a significant percentage of the rest 90% to include channel C among their noise channels,

• when popularity is around 1%:
  ➢ then it is much easier to obfuscate it.
  ➢ for $k = 100$ the disclosure probability is as low as 0.1