Connecting the Dots: An Investigative Study on Linking Private User Data Across Messaging Apps

Junkyu Kang[1], **Soyoung Lee[1]**, Yonghwi Kwon[2], Sooel Son[1]

[1] KAIST [2] University of Maryland

NDSS 2026

Messenger usage

- Billions of users on mainstream messaging platforms

WhatsApp
(2 billion)

Telegram
(950 million)

Tinder
(50 million)

KakaoTalk
(48.7 million)

Messenger usage

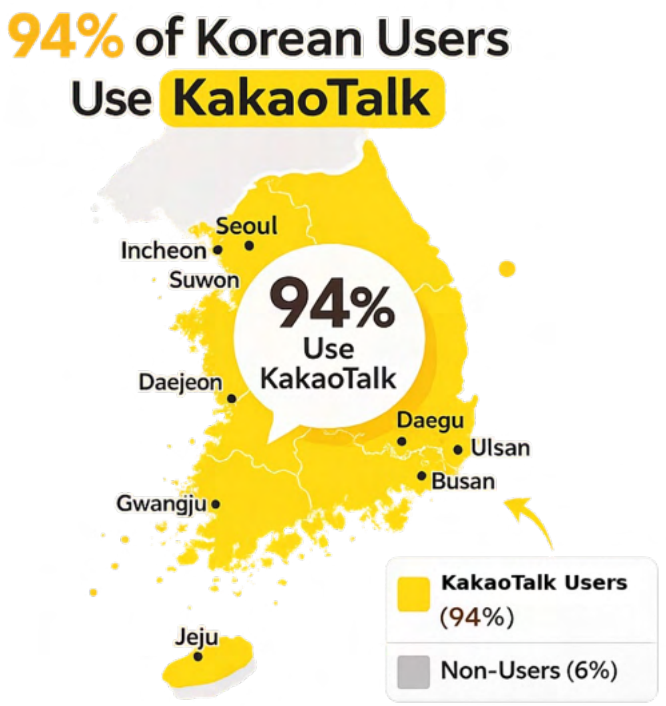
- Billions of users on mainstream messaging platforms

WhatsApp
(2 billion)

Telegram
(950 million)

Tinder
(50 million)

KakaoTalk
(48.7 million)

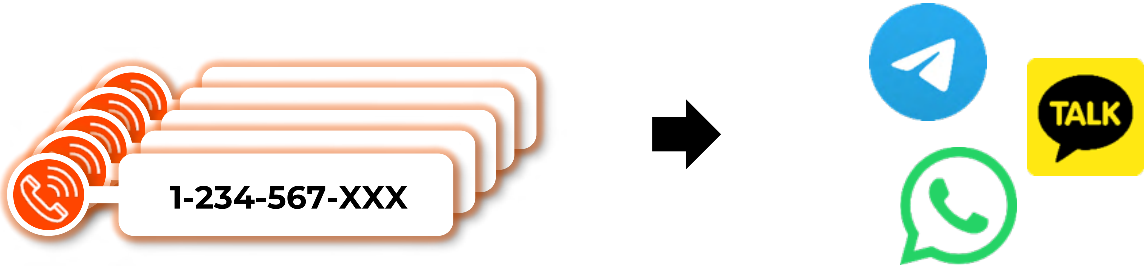


Privacy attacks targeting messengers

- Contact Discovery

Privacy attacks targeting messengers

- Contact Discovery



Phone numbers **Messenger apps**

Privacy attacks targeting messengers

- Contact Discovery

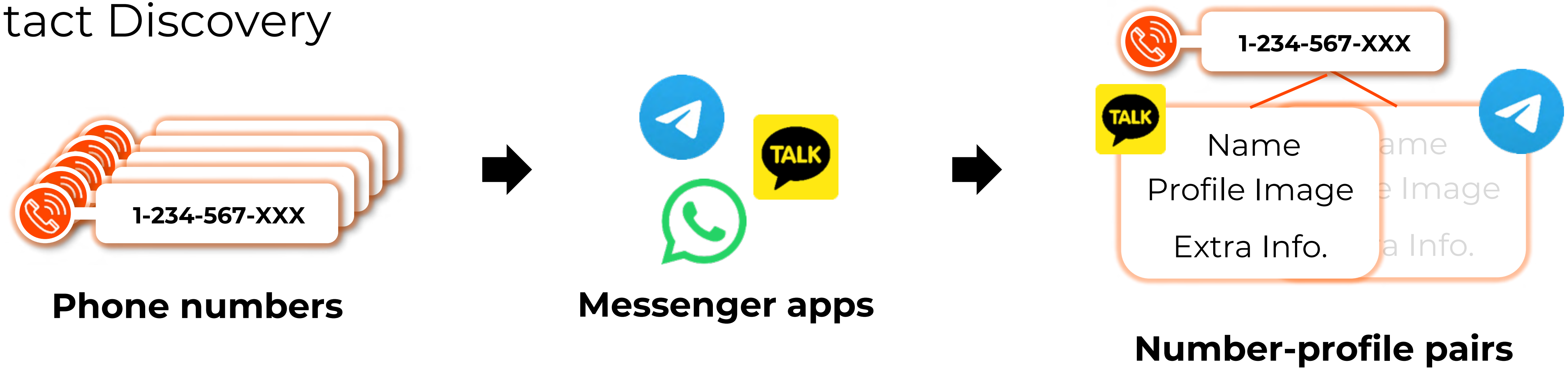
Phone numbers

Messenger apps

Number-profile pairs

Privacy attacks targeting messengers

- Contact Discovery



- Previous work

  - Targeting **WhatsApp**, **Signal**, and **Telegram**, enumerated **5M, 2.5M, and 908** profiles (Hagen *et al*., 2021)

  - Targeting **Facebook**, tested **200K** phone numbers (Kim *et al*., 2017)

  - Targeting **KakaoTalk**, enumerated over **50K** profiles (Kim *et al*., 2015)

Privacy attacks targeting messengers

- Location Inference

Privacy attacks targeting messengers

- Location Inference

Location-based Services (LBS)

Privacy attacks targeting messengers

- Location Inference

500m away

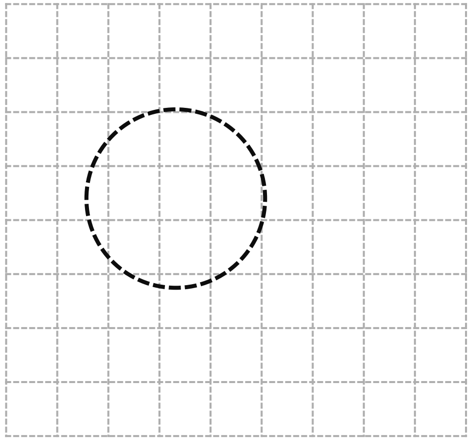
Location-based Services (LBS)

Nearby Signals

Privacy attacks targeting messengers

- Location Inference

500m away



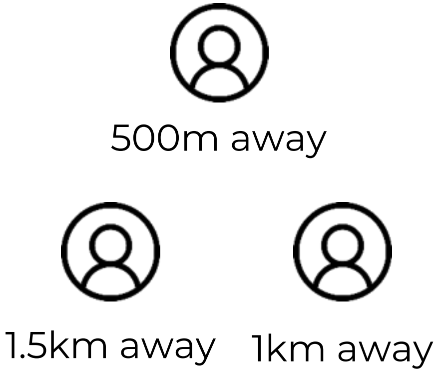
Location-based Services (LBS)

Nearby Signals

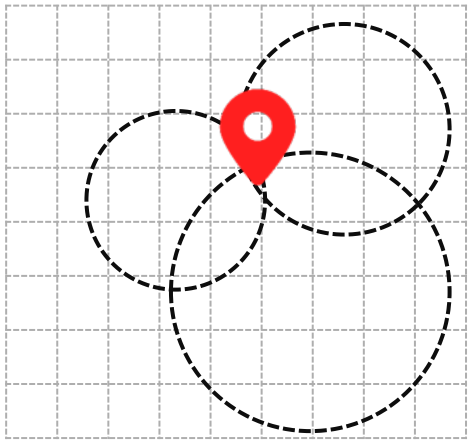
Privacy attacks targeting messengers

- Location Inference

Location-based Services (LBS)

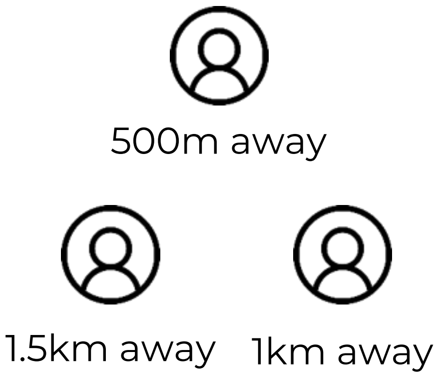


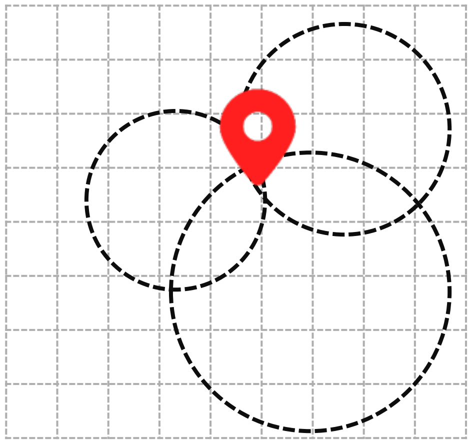
Nearby Signals



Privacy attacks targeting messengers

- Location Inference





Location-based Services (LBS)

Nearby Signals

- Previous work

  - API traffic leakage of LBS application (Dhondt *et al.*, 2024)

  - Precise localization attacks targeting Tinder (Carman *et al.*, 2017)

  - Automated user location tracking on location-based social networks (Li *et al.*, 2014)

Social & Messenger apps

- Widely used services

KakaoTalk Telegram Tinder

Social & Messenger apps

- Widely used services

KakaoTalk

Telegram

Tinder

Friend Registration

SSO Login

Social & Messenger apps

- Widely used services

KakaoTalk

Telegram

Tinder

Friend Registration

Friend Registration

SSO Login

Private Chatting

Social & Messenger apps

- Widely used services

KakaoTalk

Telegram

Tinder

Friend Registration

Friend Registration

Social Matching

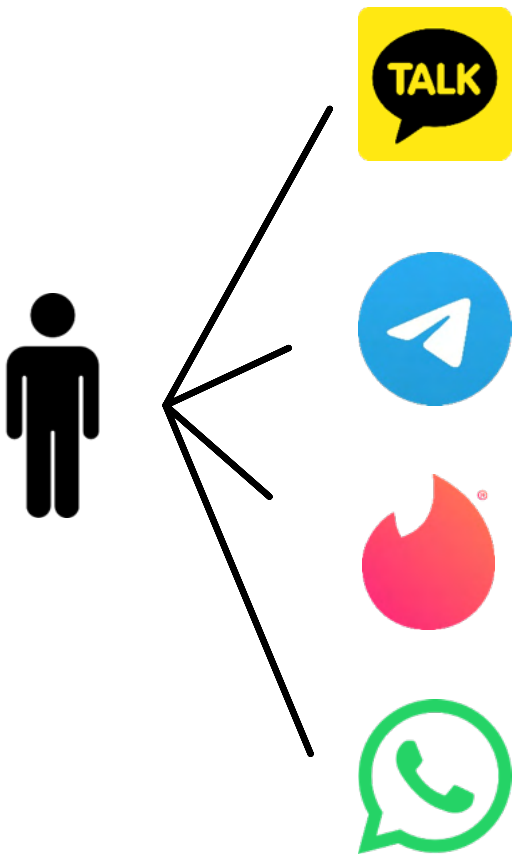
SSO Login

Private Chatting

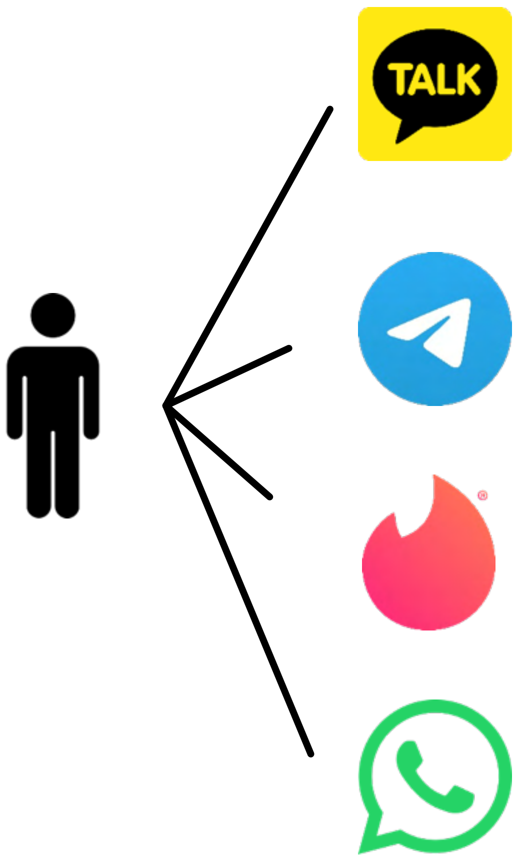
Location Service

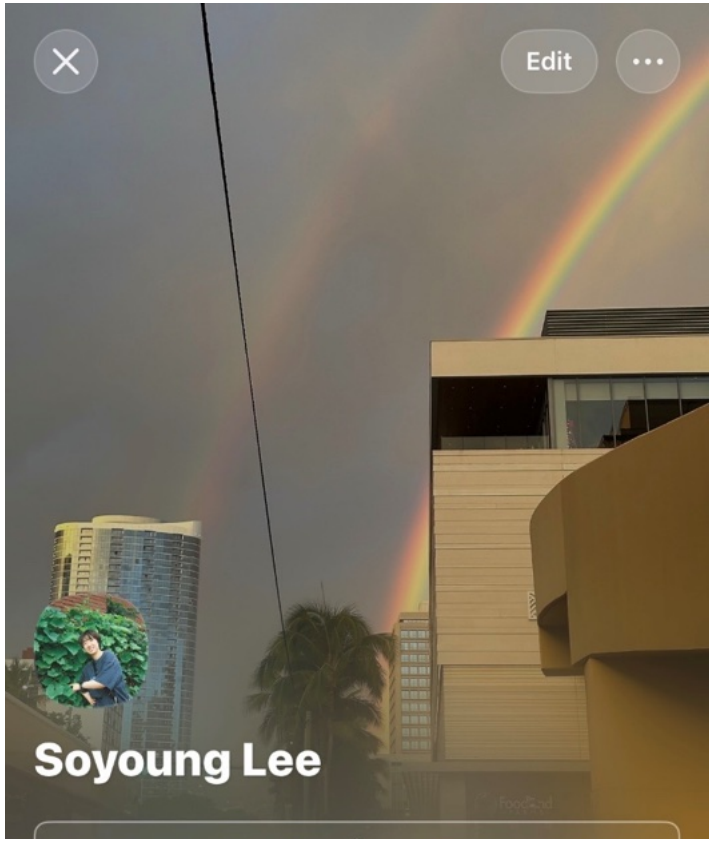
Multi-platform usage in real-world

Multi-platform usage in real-world



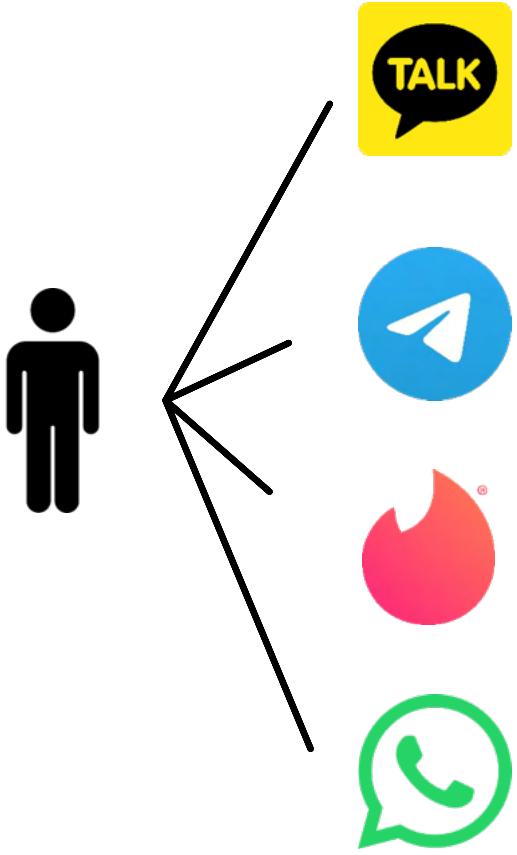
Multi-platform usage in real-world

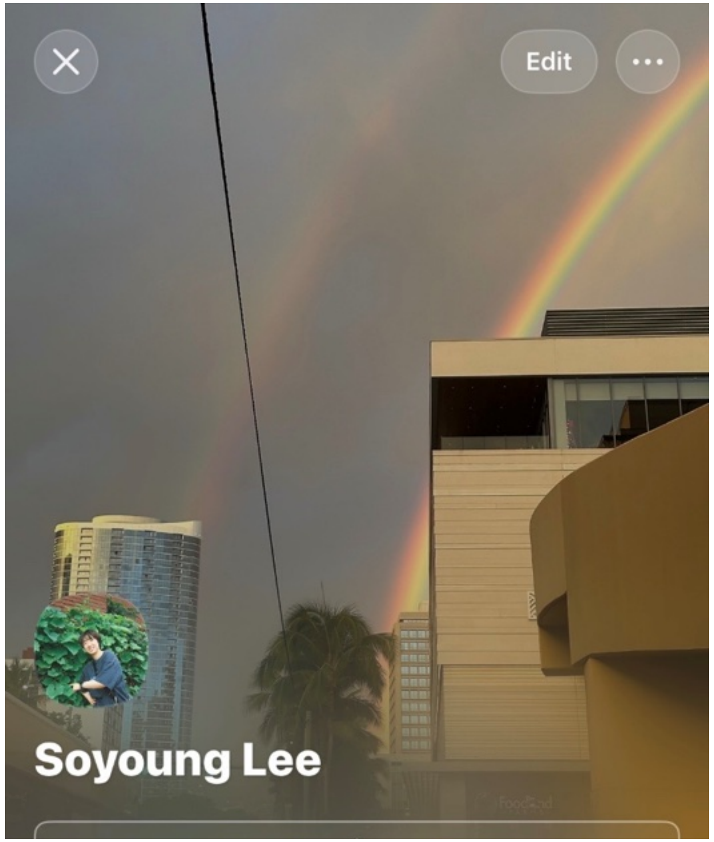


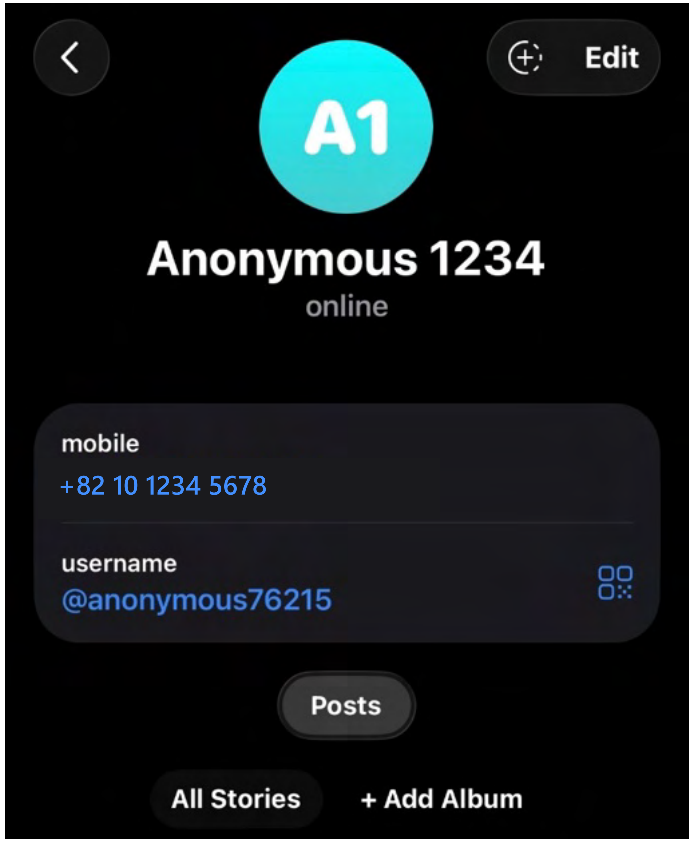


App A

Multi-platform usage in real-world



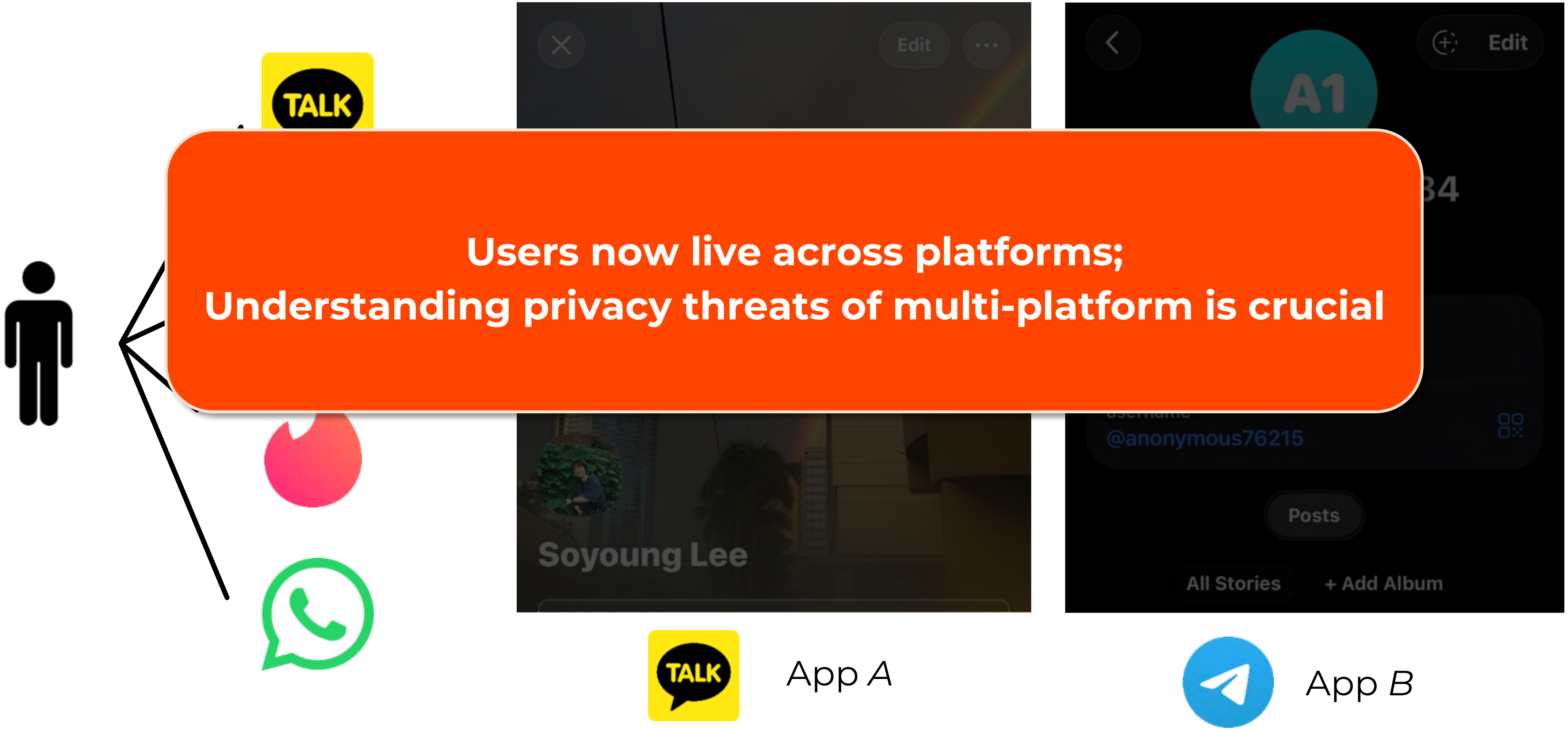




App *A*

App *B*

Multi-platform usage in real-world



Users now live across platforms;
Understanding privacy threats of multi-platform is crucial

Design: Apps, features, and pipeline

Design: Apps, features, and pipeline

Component-level Privacy Attacks

1. Contact Discovery

2. SSO Linking

3. Location Inference

Design: Apps, features, and pipeline

Component-level Privacy Attacks	1. Contact Discovery	2. SSO Linking	3. Location Inference
Threat Model			

Design: Apps, features, and pipeline

Component-level Privacy Attacks

1. Contact Discovery

2. SSO Linking

3. Location Inference

Threat Model

Design: Apps, features, and pipeline

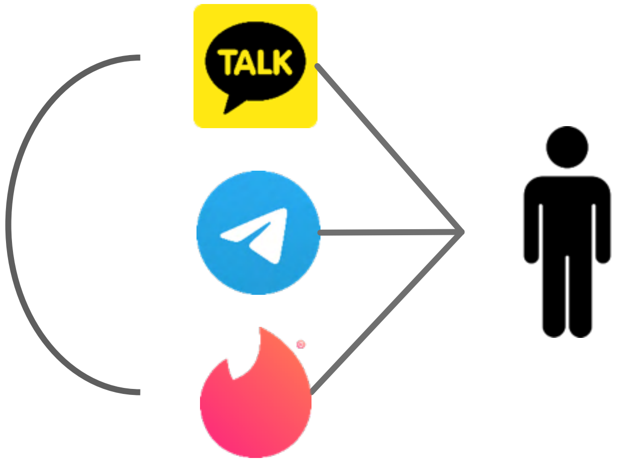
Component-level Privacy Attacks

1. Contact Discovery

2. SSO Linking

3. Location Inference

Threat Model



Design: Apps, features, and pipeline

Component-level Privacy Attacks

1. Contact Discovery

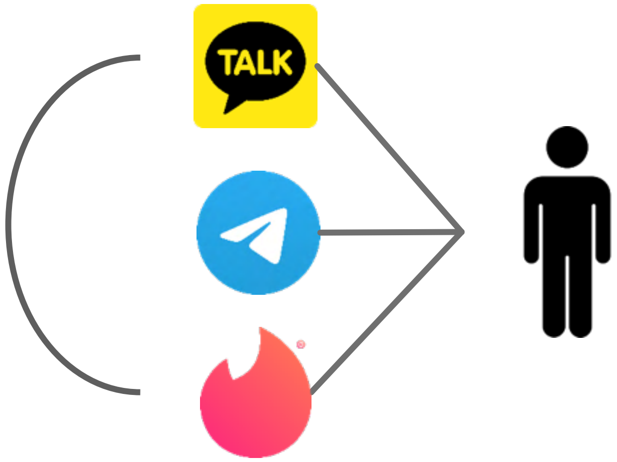
2. SSO Linking

3. Location Inference

Threat Model

Privacy-sensitive

Name, Profile image,
Location, ..



Design: Apps, features, and pipeline

Component-level Privacy Attacks

1. Contact Discovery

2. SSO Linking

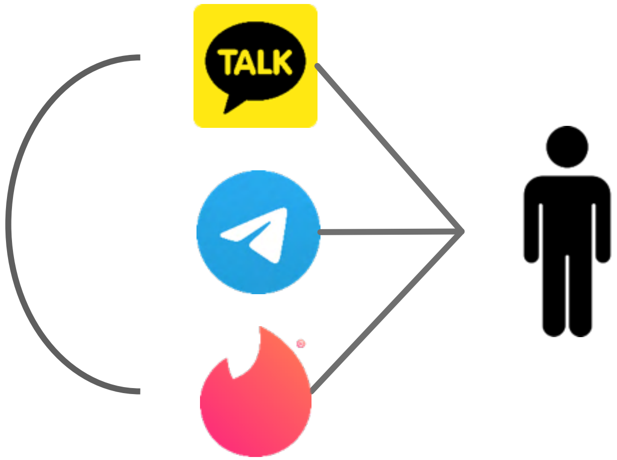
3. Location Inference

Threat Model

- ✓ Benign behaviors
- ✓ Same as regular users

Privacy-sensitive

Name, Profile image, Location, ..



Design: Apps, features, and pipeline

Component-level Privacy Attacks

1. Contact Discovery

2. SSO Linking

3. Location Inference

Design: Apps, features, and pipeline

Component-level Privacy Attacks

1. Contact Discovery
2. SSO Linking
3. Location Inference

Linking Keys

Phone numbers

Profile images

Design: Apps, features, and pipeline

Component-level Privacy Attacks

1. Contact Discovery
2. SSO Linking
3. Location Inference

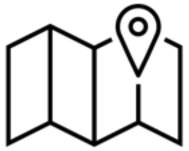
Linking Keys

Phone numbers

Profile images

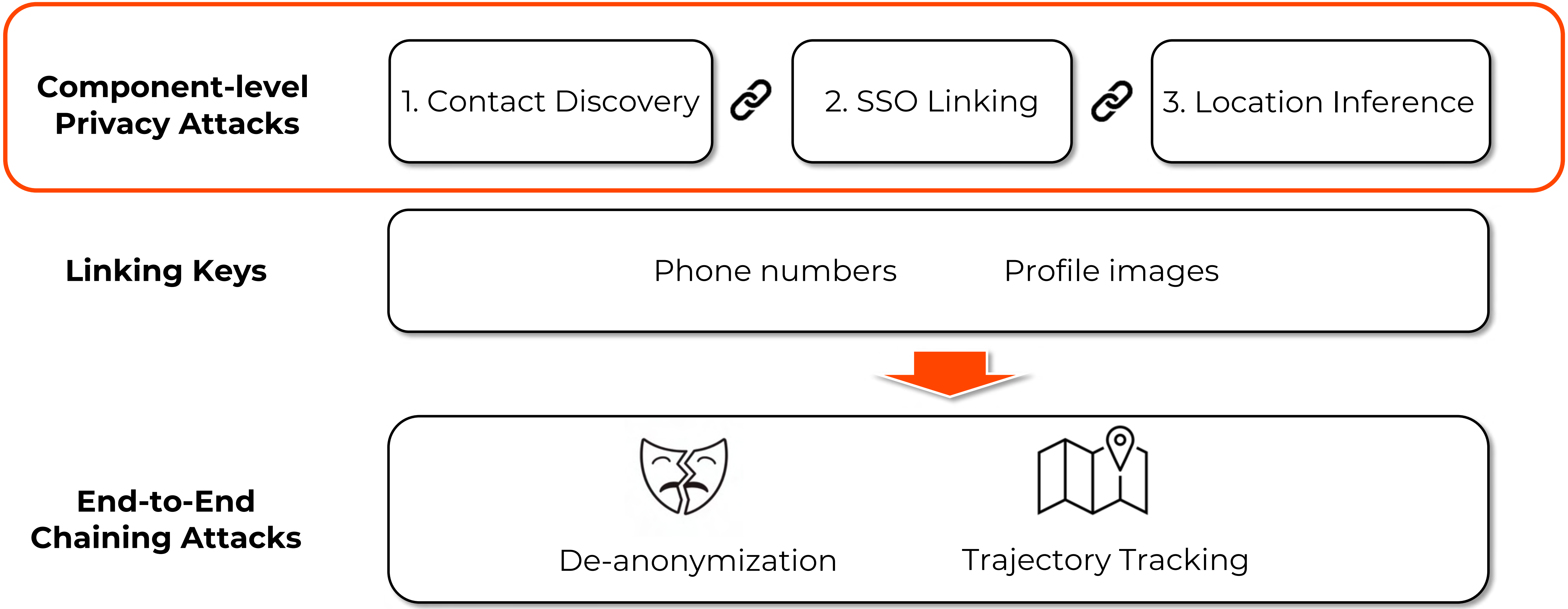
End-to-End Chaining Attacks

De-anonymization



Trajectory Tracking

Design: Apps, features, and pipeline



Attack 1: Contact discovery abuse

- Contact discovery attack

2015

Enumerated **50K** profiles
(Kim *et al.*, 2015)

2017

Tested **200K** numbers
(Kim *et al.*, 2017)

2021

5M, 2.5M, 908 profiles
(Hagen *et al.*, 2021)

Attack 1: Contact discovery abuse

- Contact discovery attack

2015

Enumerated **50K** profiles
(Kim *et al.*, 2015)

2017

Tested **200K** numbers
(Kim *et al.*, 2017)

2021

5M, 2.5M, 908 profiles
(Hagen *et al.*, 2021)

- Our Approach – Additional attack vectors
    - Address-book Syncing → **Still works!**

Attack 1: Contact discovery abuse

- Contact discovery attack

2015

15K limitation

Enumerated ~~**50K**~~ profiles
(Kim *et al.*, 2015)

2017

Tested **200K** numbers
(Kim *et al.*, 2017)

2021

5M, 2.5M, 908 profiles
(Hagen *et al.*, 2021)

- Our Approach – Additional attack vectors
  - Address-book Syncing → **Still works!**

Attack 1: Contact discovery abuse

- Contact discovery attack

2015

15K limitation

Enumerated ~~**50K**~~ profiles
(Kim *et al.*, 2015)

2017

Tested **200K** numbers
(Kim *et al.*, 2017)

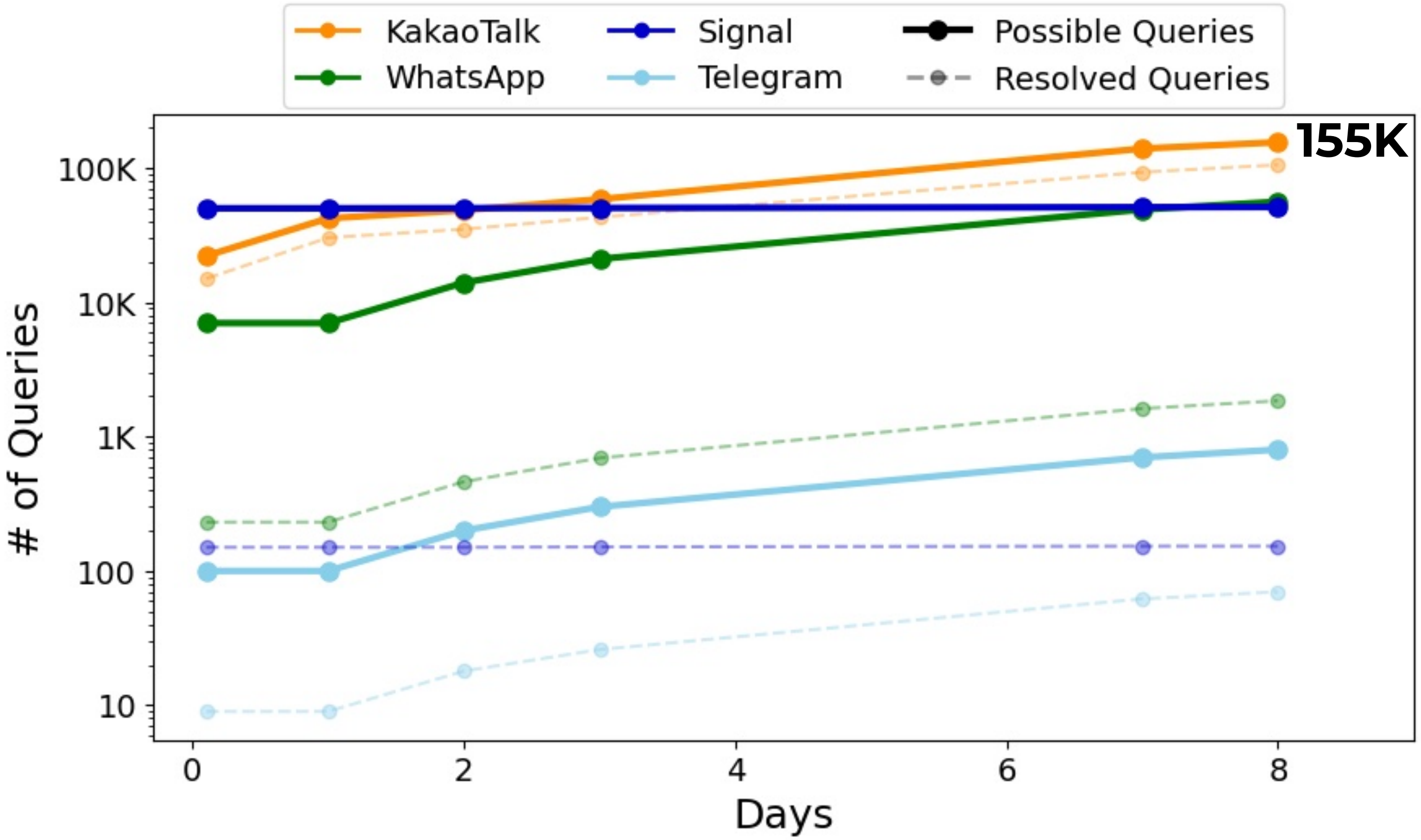
2021

5M, 2.5M, 908 profiles
(Hagen *et al.*, 2021)

- Our Approach – Additional attack vectors
  - Address-book Syncing → **Still works!**
  - **Deleting Friends**
  - **Block/Unblocking Friends**

Attack 1: Contact discovery abuse

- Using only one account



18K queries / day

7K queries / day

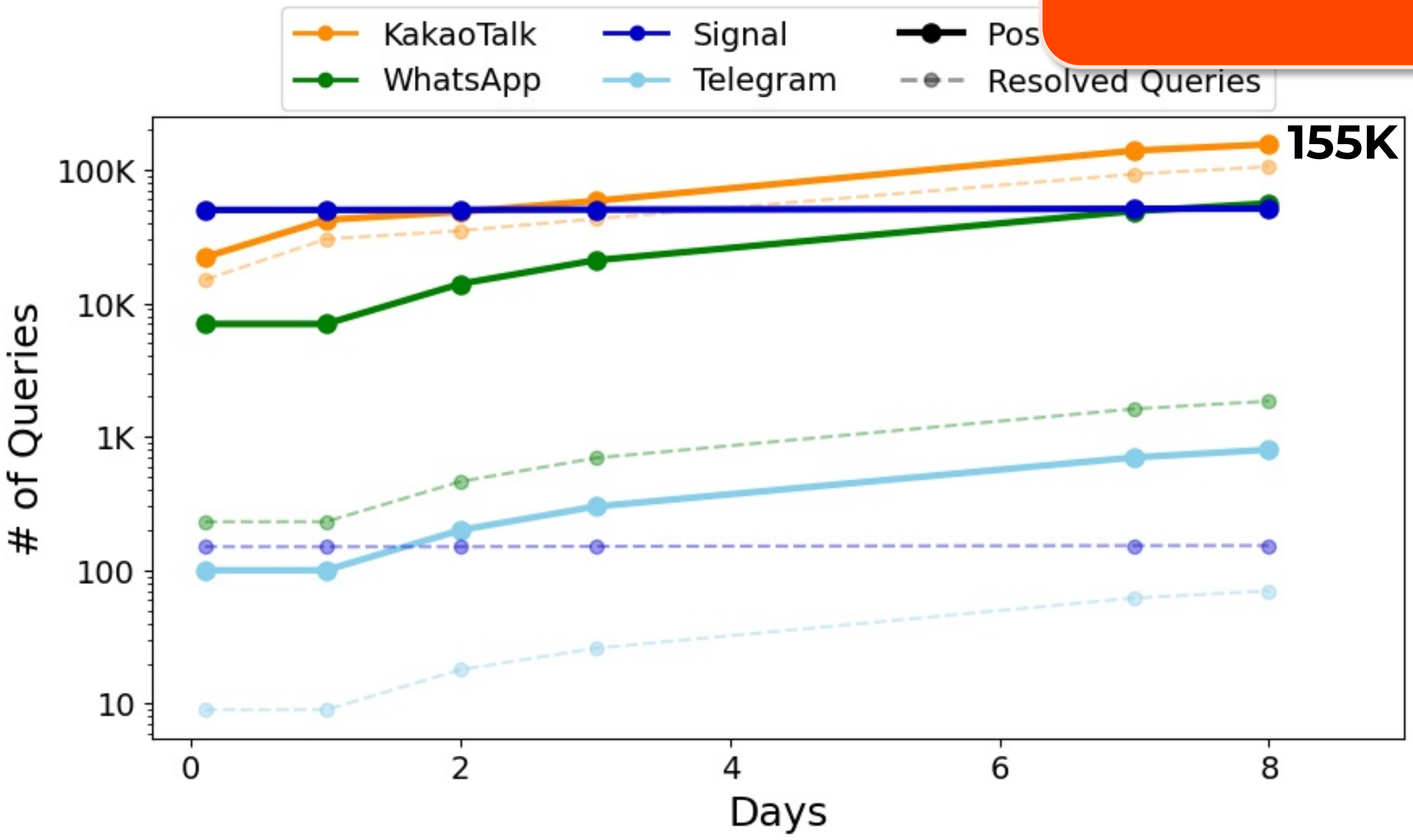
Initial 50K + 144 queries / day

100 queries / day

Attack 1: Contact discovery abuse

- Using only one account

Contact discovery still exists in modern messaging apps.



18K queries / day

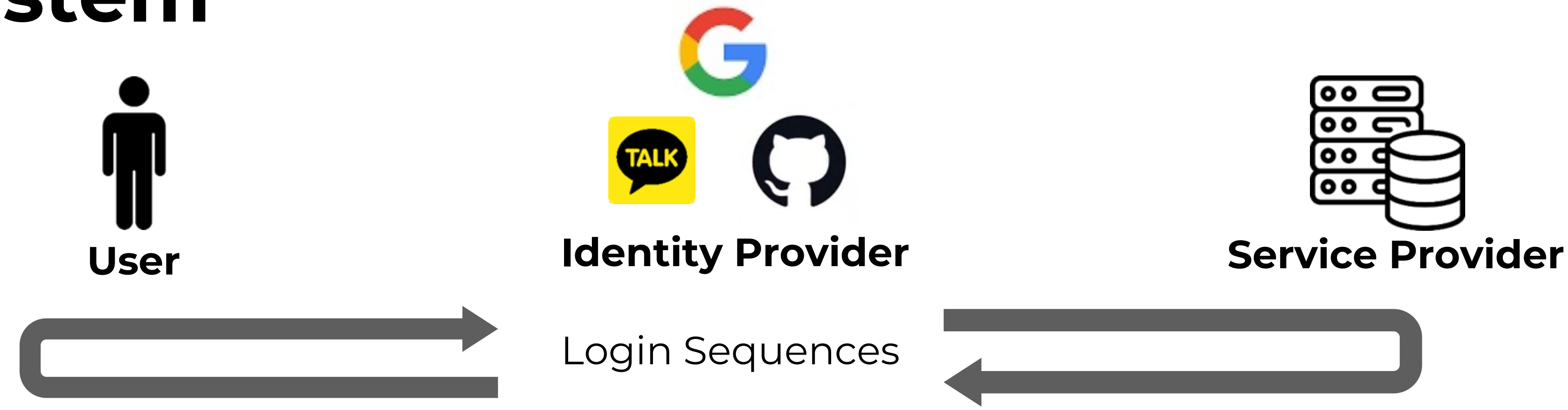
7K queries / day

Initial 50K + 144 queries / day

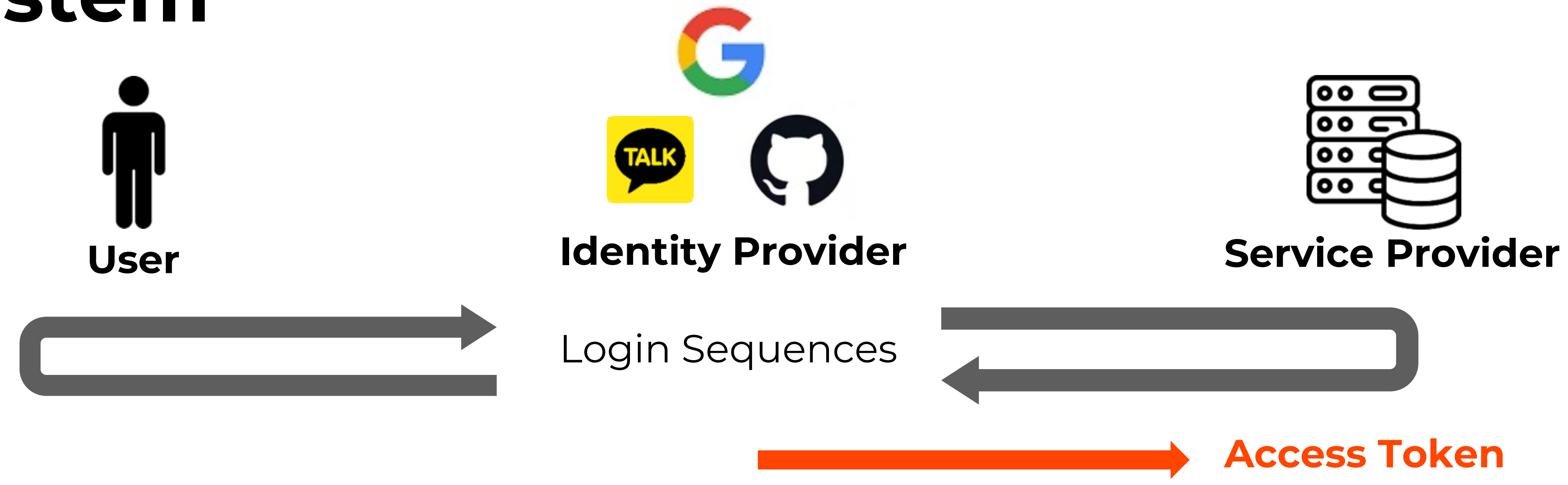
100 queries / day

Attack 2: OAuth token exposure in SSO ecosystem

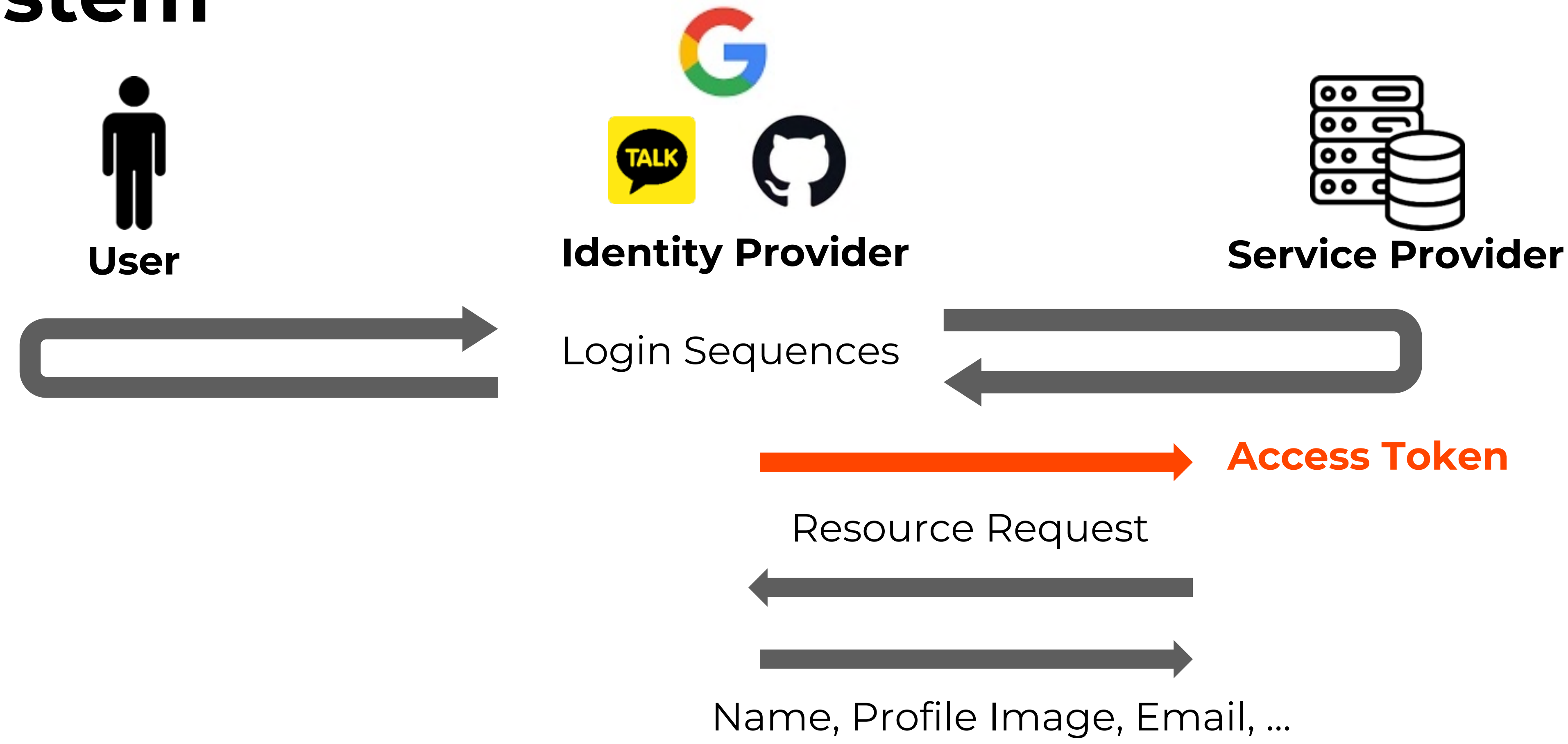
Attack 2: OAuth token exposure in SSO ecosystem



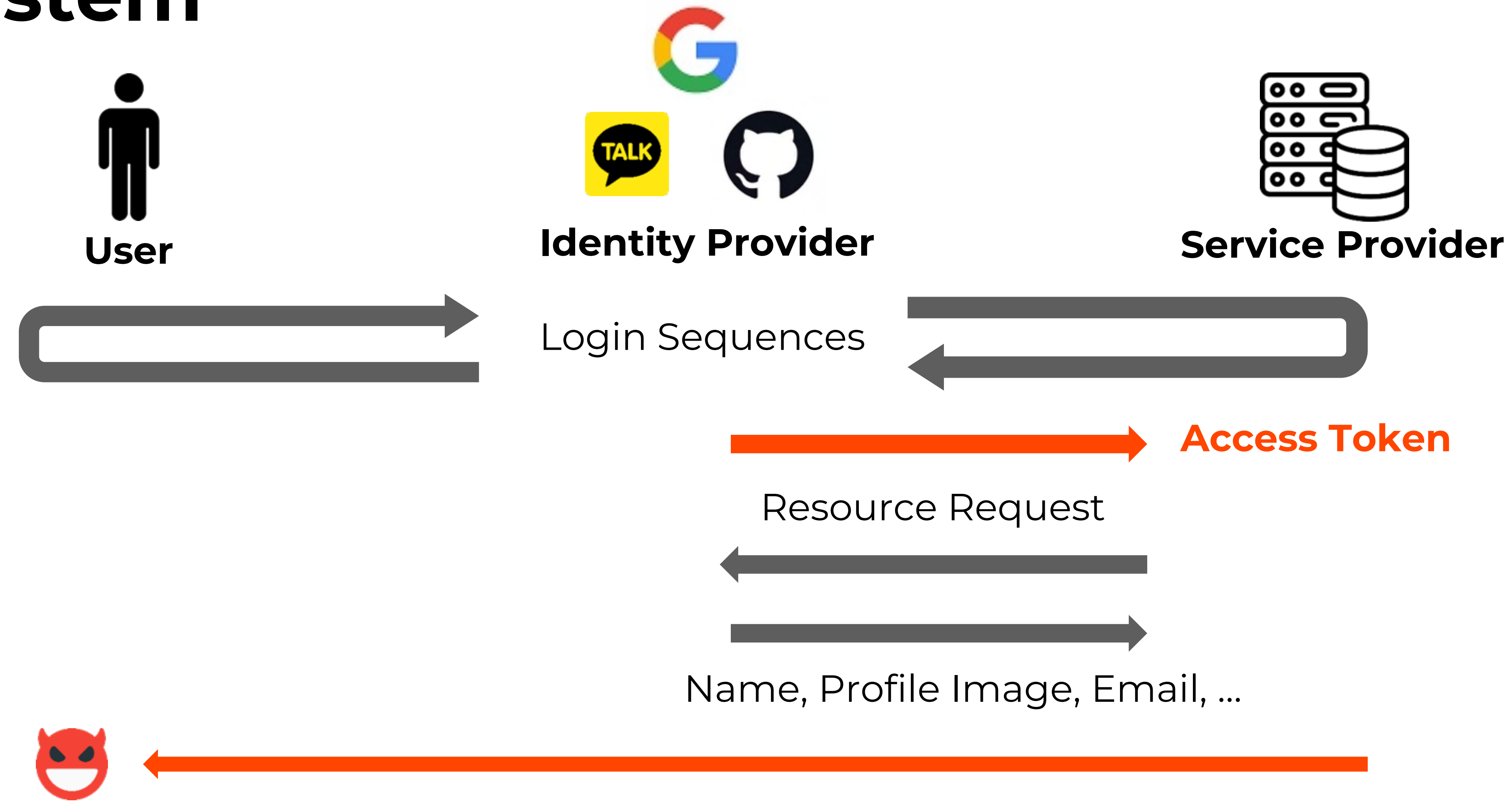
Attack 2: OAuth token exposure in SSO ecosystem



Attack 2: OAuth token exposure in SSO ecosystem



Attack 2: OAuth token exposure in SSO ecosystem



Attack 2: OAuth token exposure in KakaoTalk SSO ecosystem

- Targeted 14,102 websites using KakaoTalk SSO Login

Attack 2: OAuth token exposure in KakaoTalk SSO ecosystem

- Targeted 14,102 websites using KakaoTalk SSO Login

- Found **63 websites** with token exposure

Attack 2: OAuth token exposure in KakaoTalk SSO ecosystem

- Targeted 14,102 websites using KakaoTalk SSO Login

- Found **63 websites** with token exposure
- The attacker can obtain user’s name, profile image, email, etc.

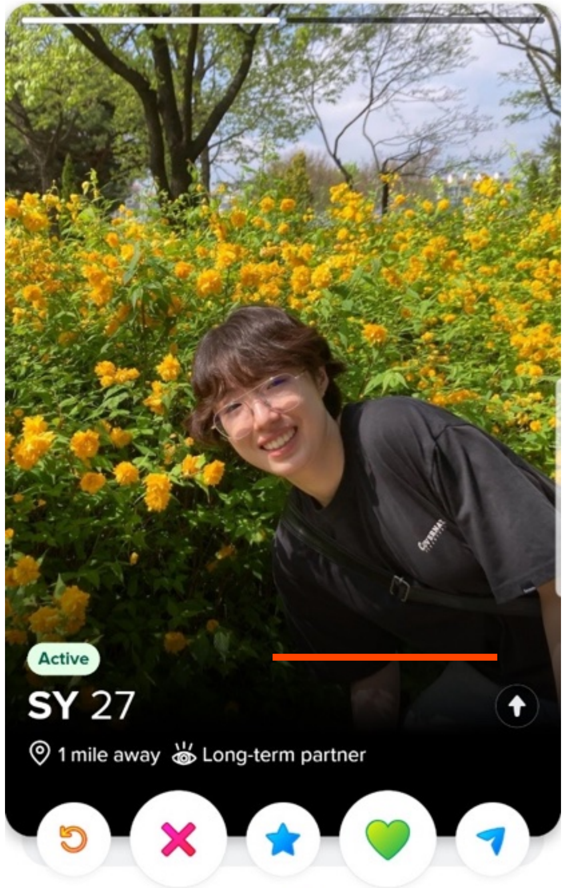
Attack 2: OAuth token exposure in KakaoTalk SSO ecosystem

- Targeted 14,102 websites using KakaoTalk SSO Login

- Found **63 websites** with token exposure
- The attacker can obtain user's name, profile image, email, etc.

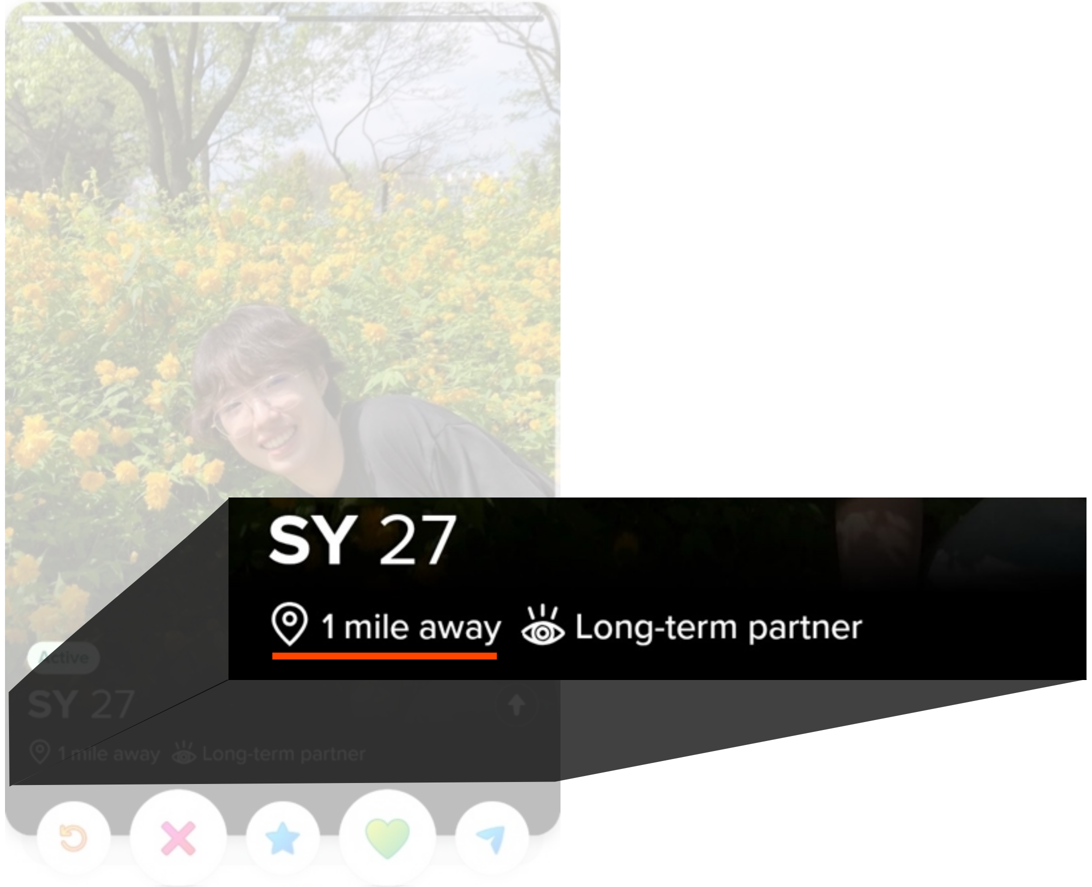
Token exposure is exploitable in the real world.

Attack 3: Efficient location inference from Tinder "nearby" signals



Tinder profile card

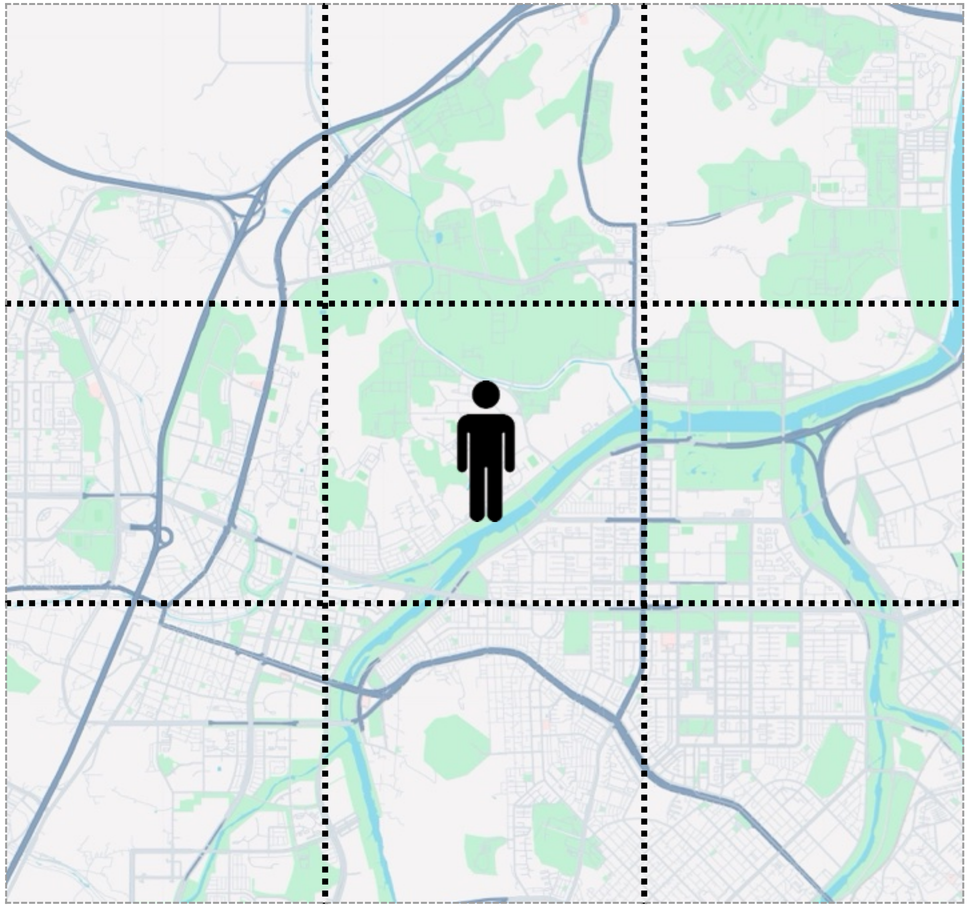
Attack 3: Efficient location inference from Tinder "nearby" signals



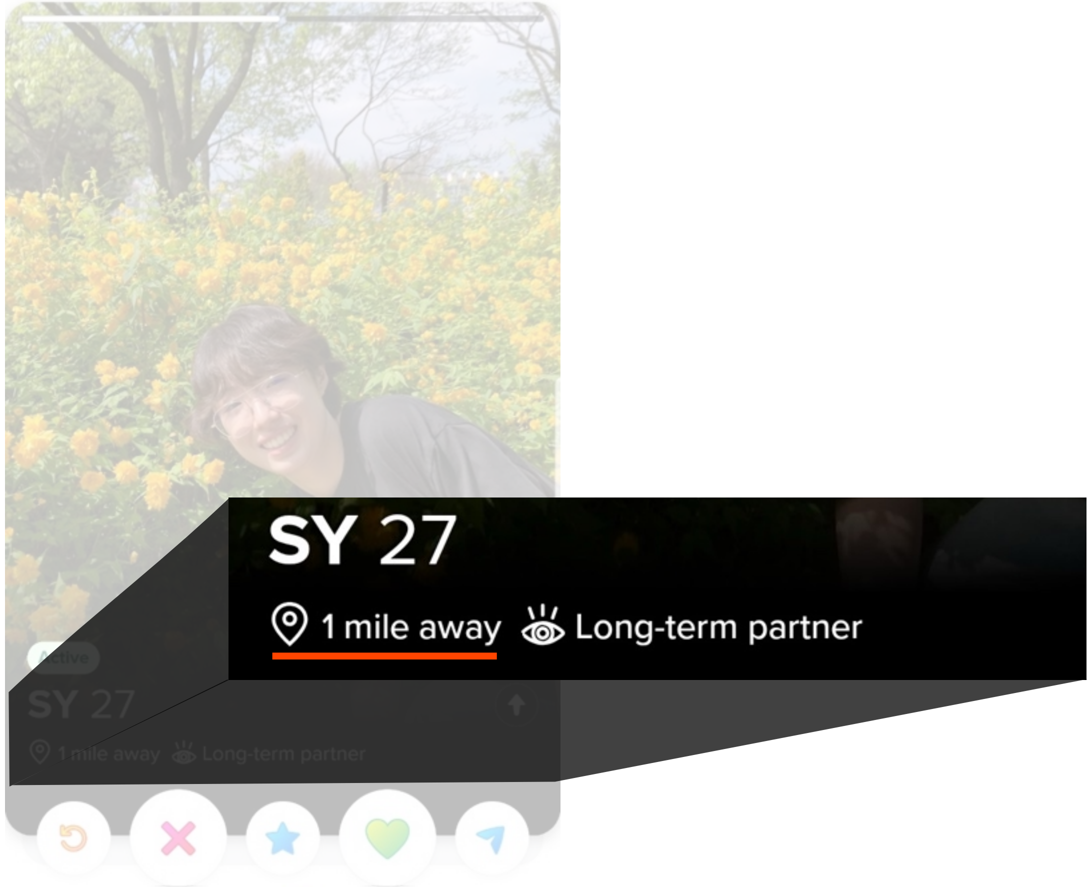
Tinder profile card

Attack 3: Efficient location inference from Tinder “nearby” signals

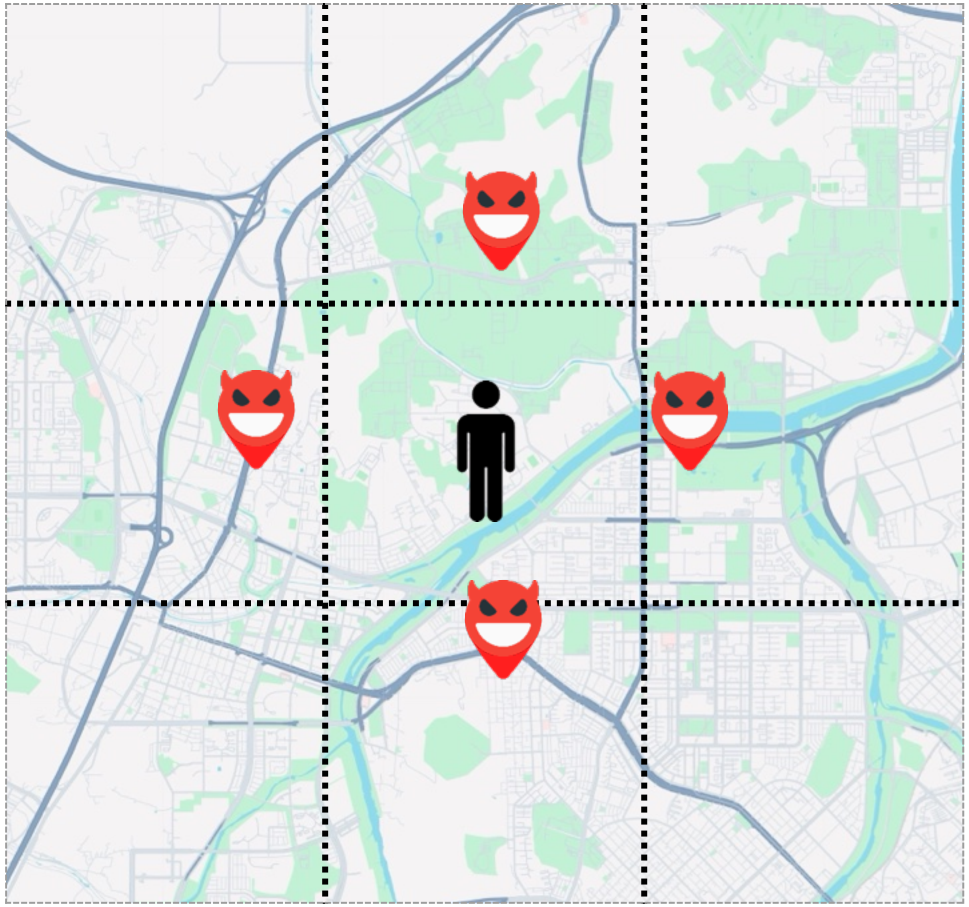
Tinder profile card



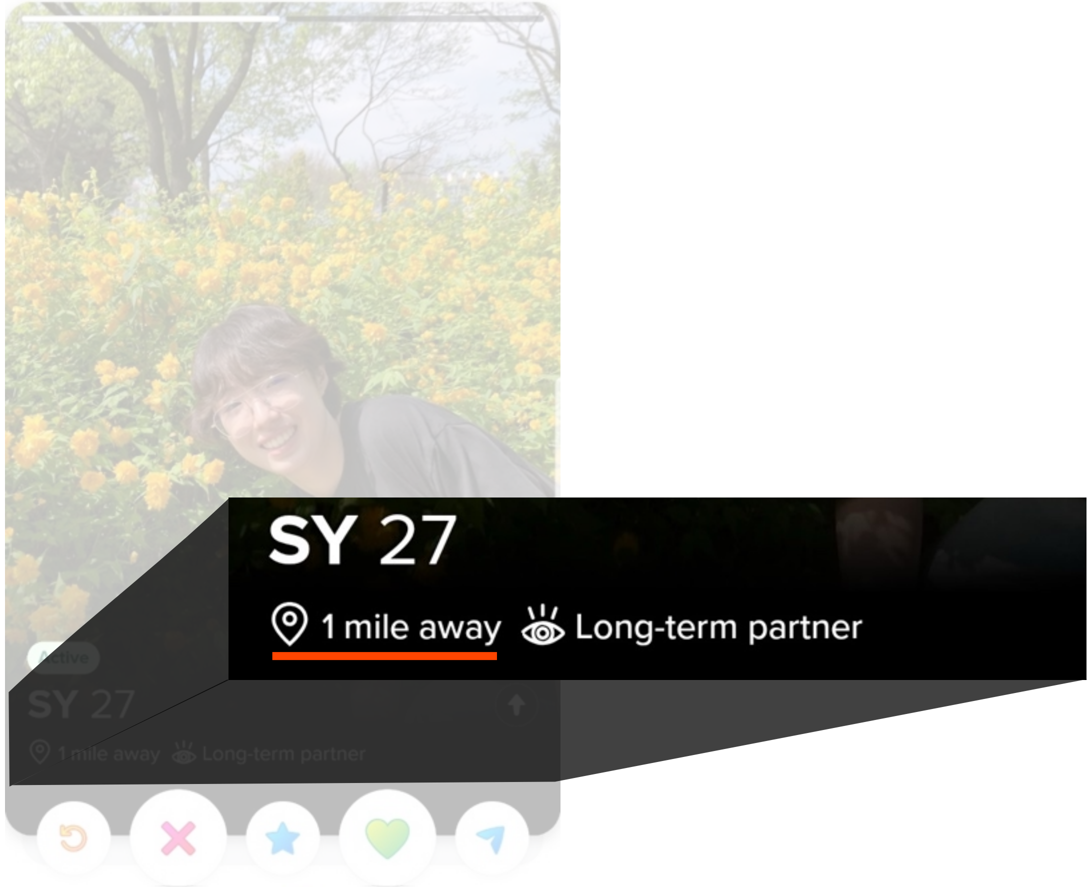
Attack 3: Efficient location inference from Tinder “nearby” signals



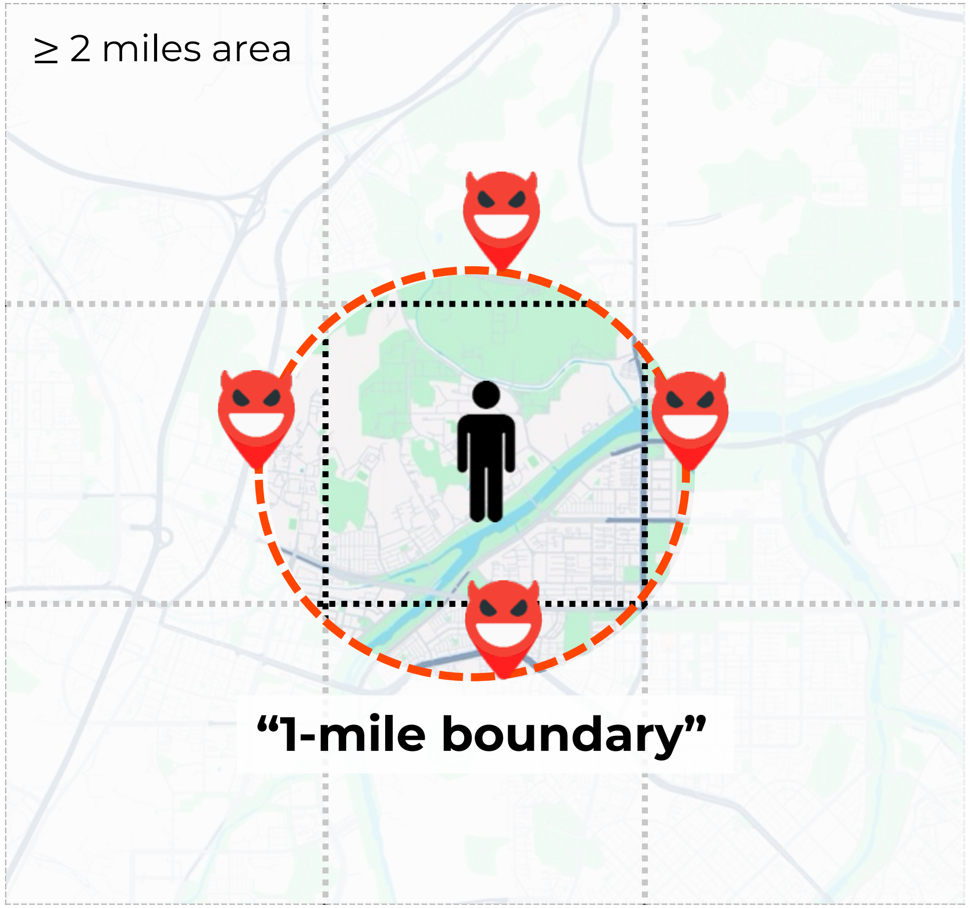
Tinder profile card



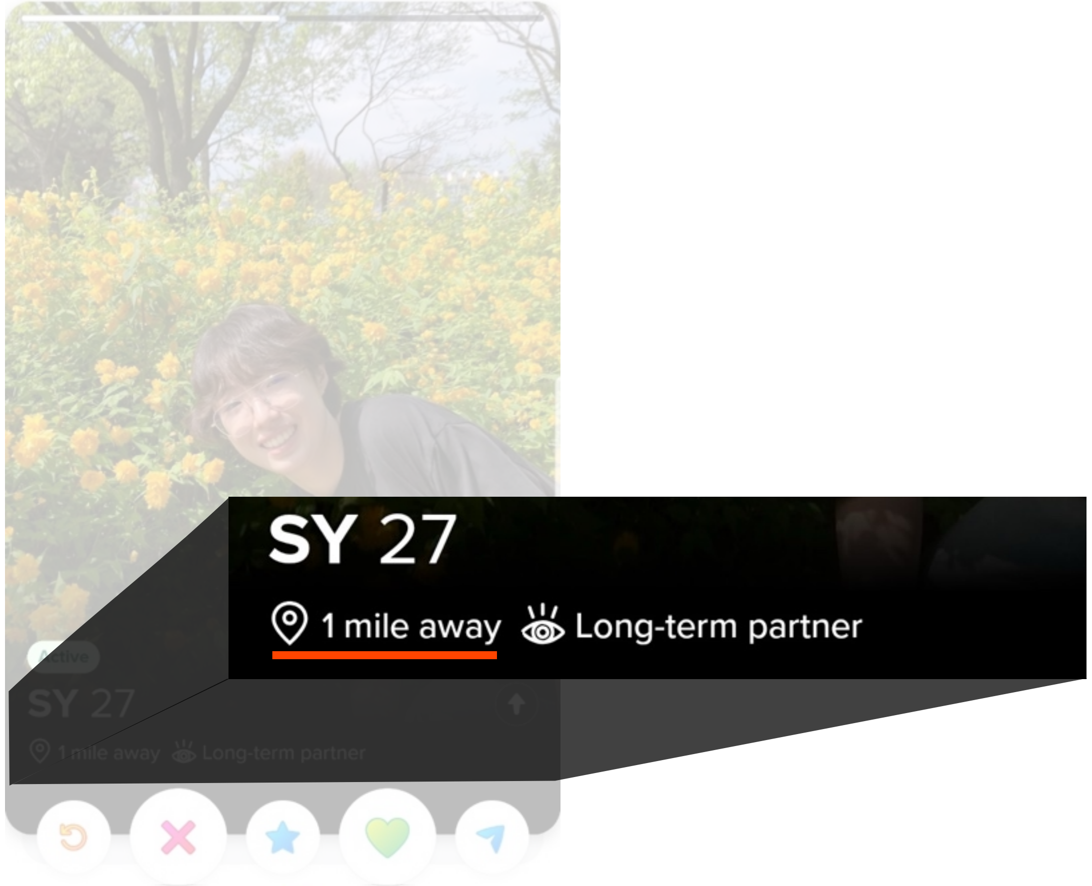
Attack 3: Efficient location inference from Tinder "nearby" signals



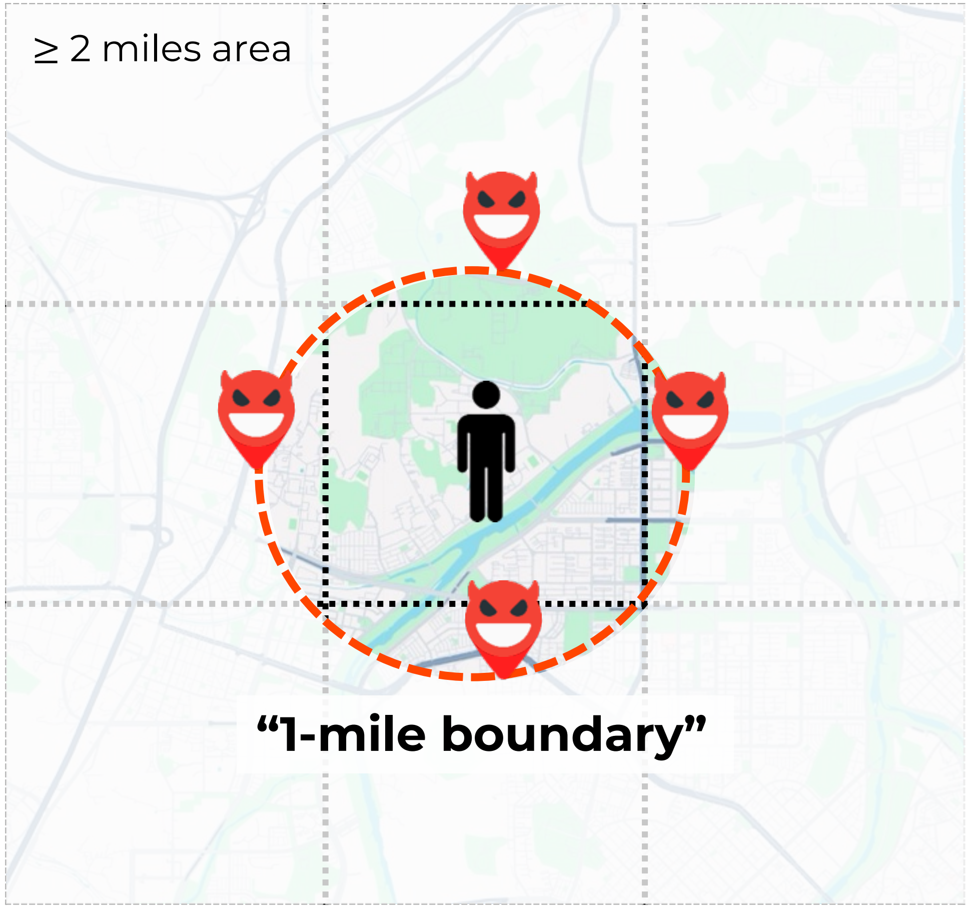
Tinder profile card



Attack 3: Efficient location inference from Tinder “nearby” signals



Tinder profile card



Attack 3: Efficient location inference from Tinder “nearby” signals

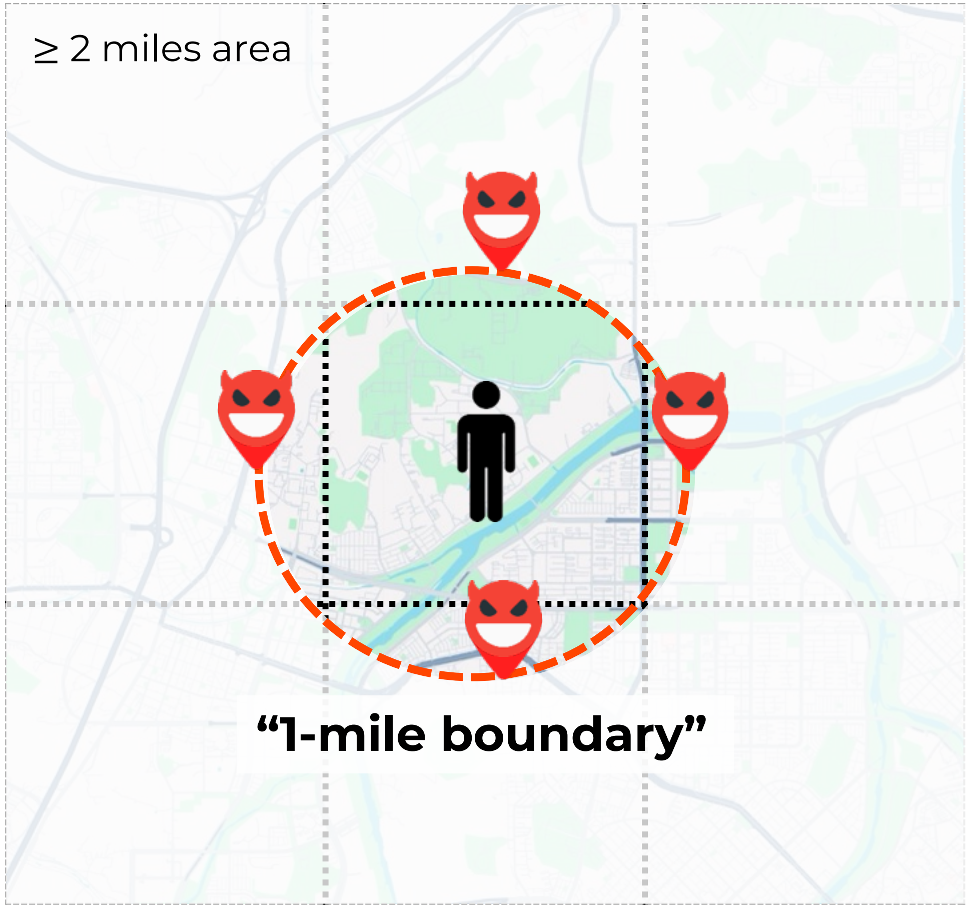
- Targeting 10 sampled locations
  - Previous work (Heaton, 2018)

    Average error: 371m with 676 queries

  - 1-mile boundary algorithm

    Average error: 385m with **12 queries (56x fewer)**

    Average error: 324m with **40 queries (17x fewer)**



Attack 3: Efficient location inference from Tinder “nearby” signals

- Targeting 10 sampled locations
  - Previous work (Heaton, 2018)

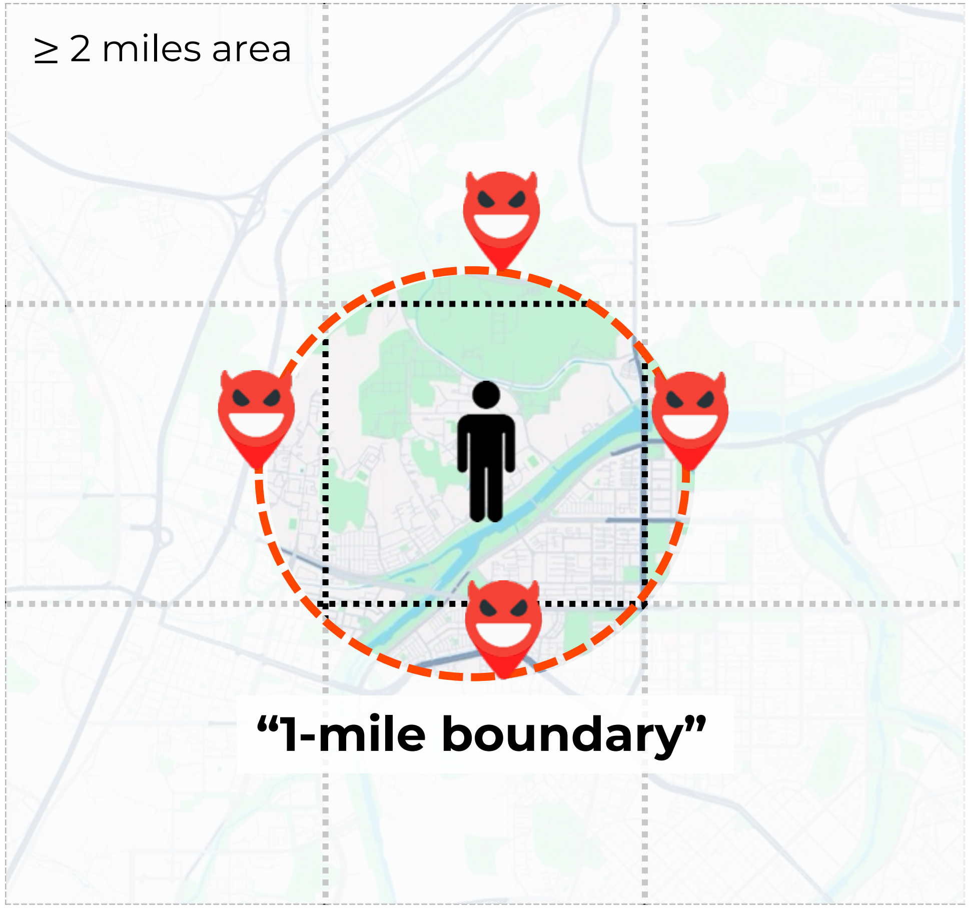
    Average error: 371m with 676 queries

  - 1-mile boundary algorithm

    Average error: 385m with **12 queries (56x fewer)**

    Average error: 324m with **40 queries (17x fewer)**

More precise location with fewer queries



“Linking keys” enable cross-platform chaining

1. Contact Discovery

2. SSO Linking

3. Location Inference

“Linking keys” enable cross-platform chaining

1. Contact Discovery

2. SSO Linking

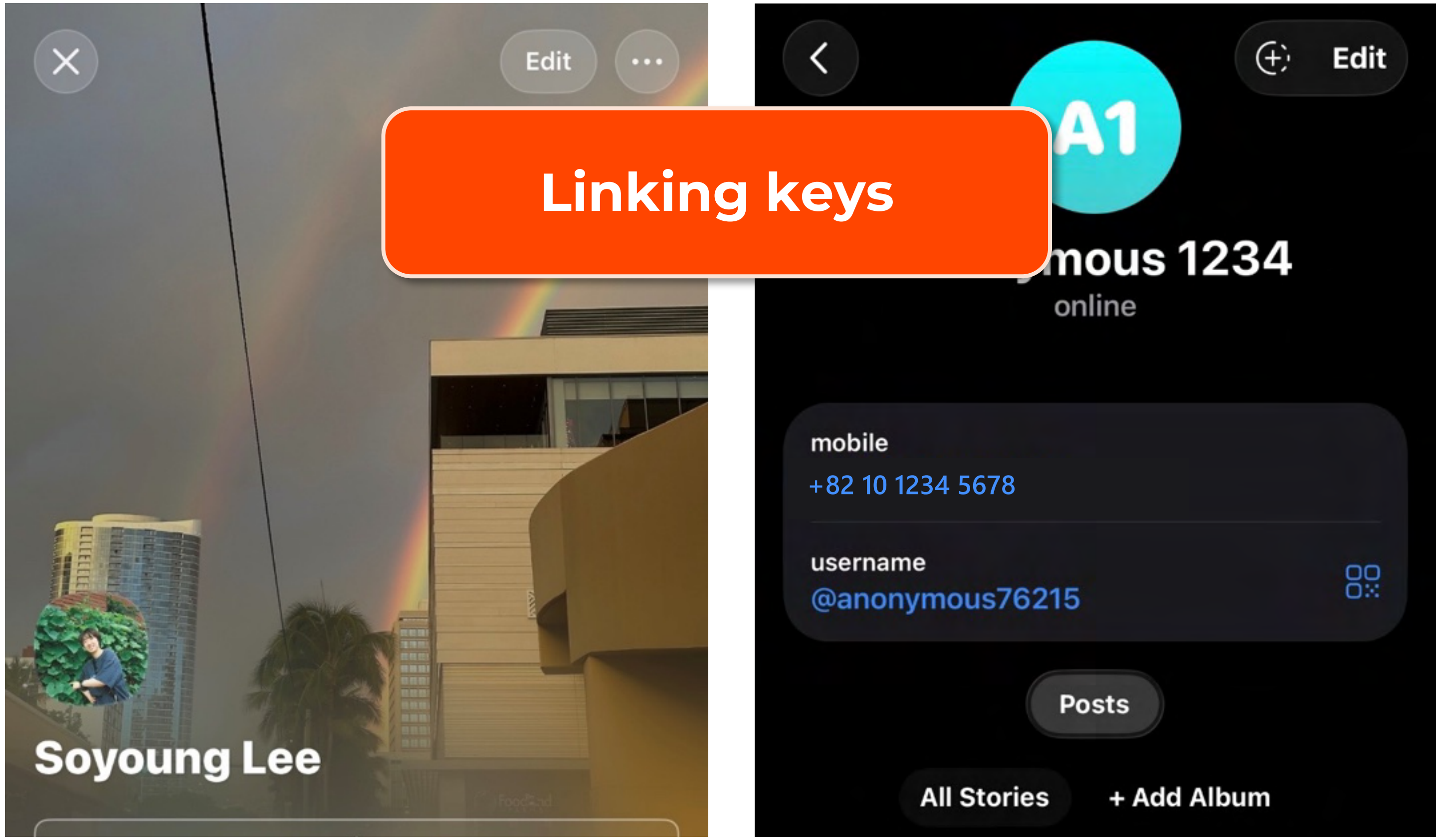
3. Location Inference

"Linking keys" enable cross-platform chaining

1. Contact Discovery

2. SSO Linking

3. Location Inference

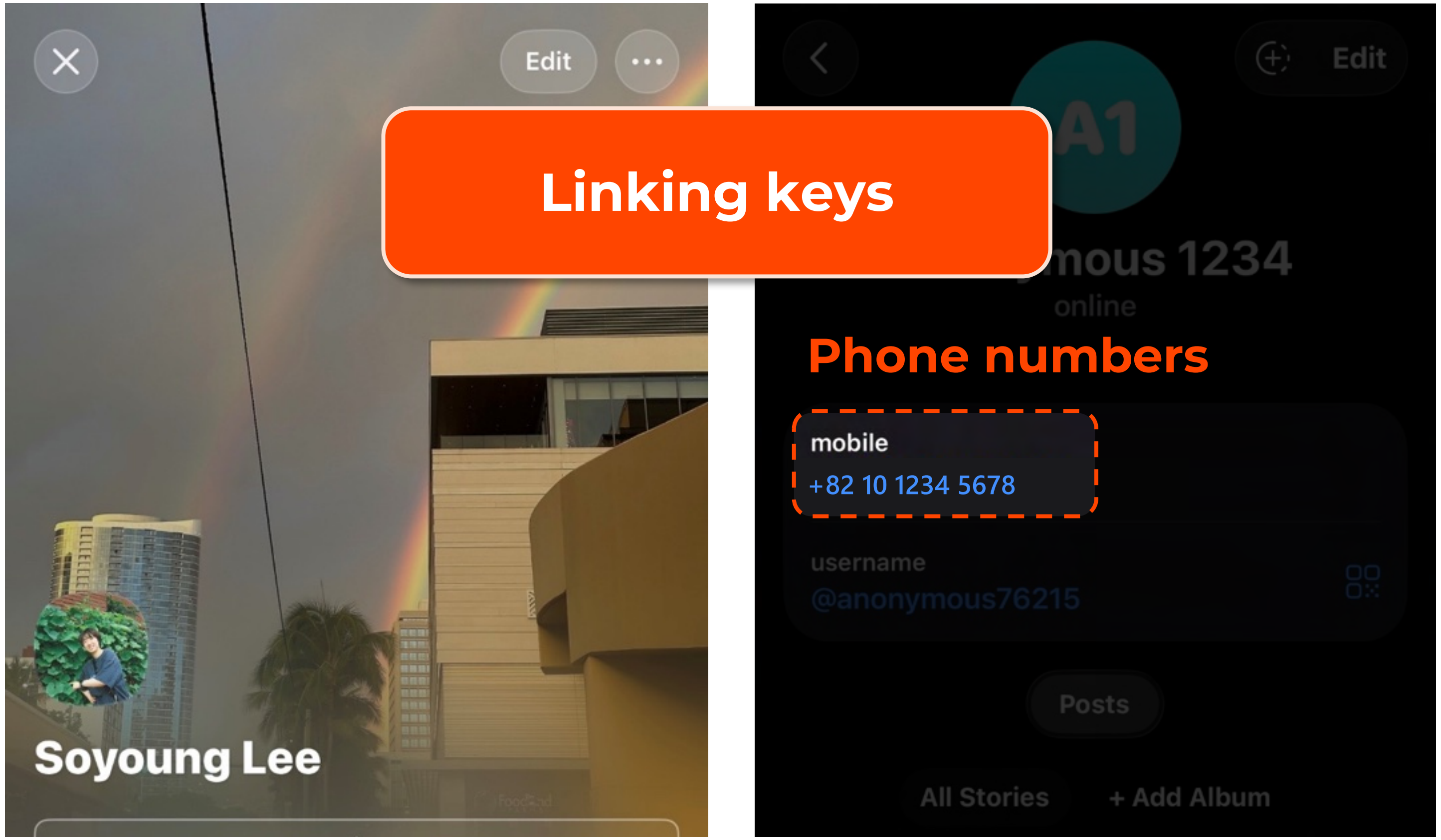


“Linking keys” enable cross-platform chaining

1. Contact Discovery

2. SSO Linking

3. Location Inference

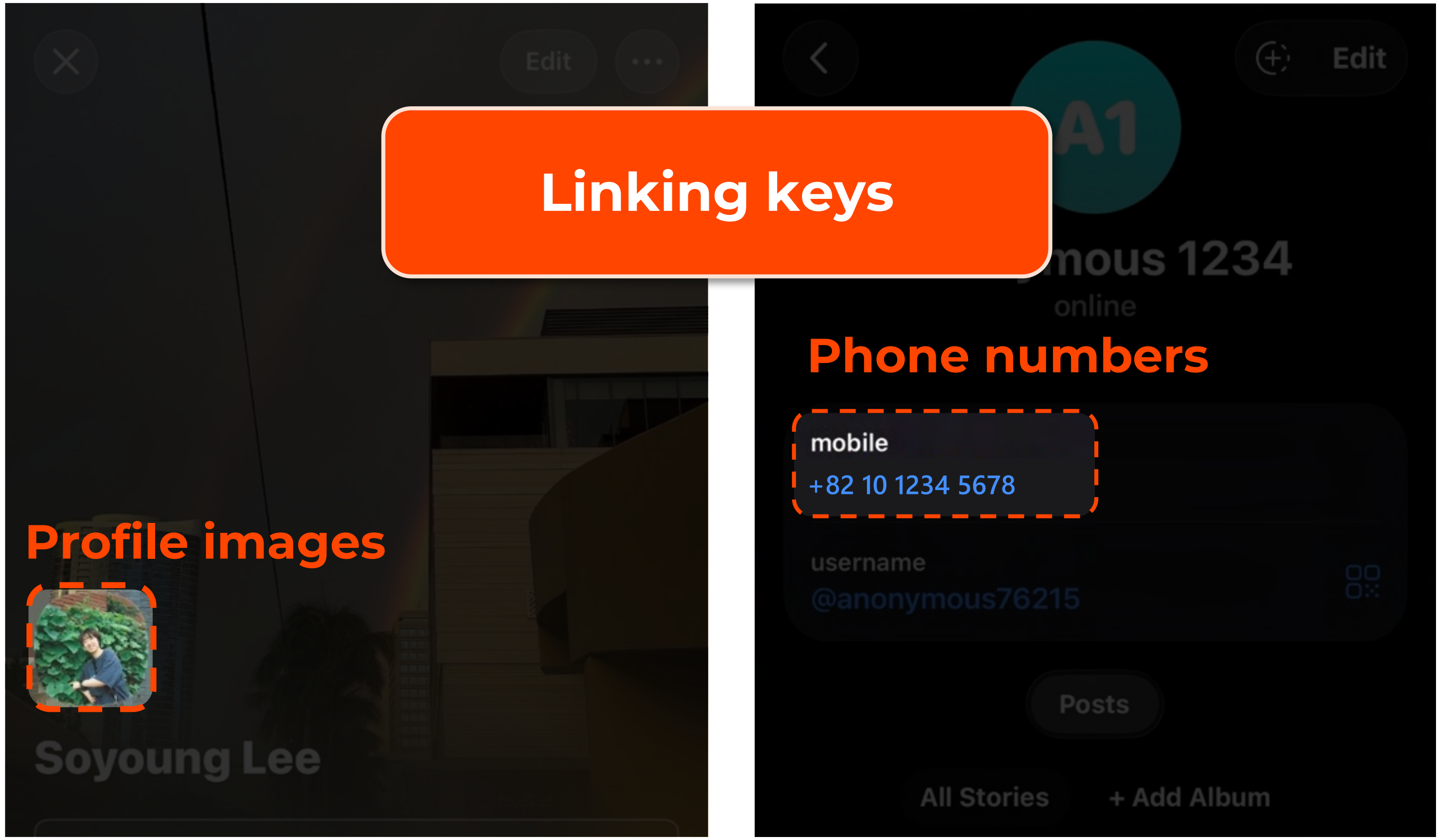


“Linking keys” enable cross-platform chaining

1. Contact Discovery

2. SSO Linking

3. Location Inference

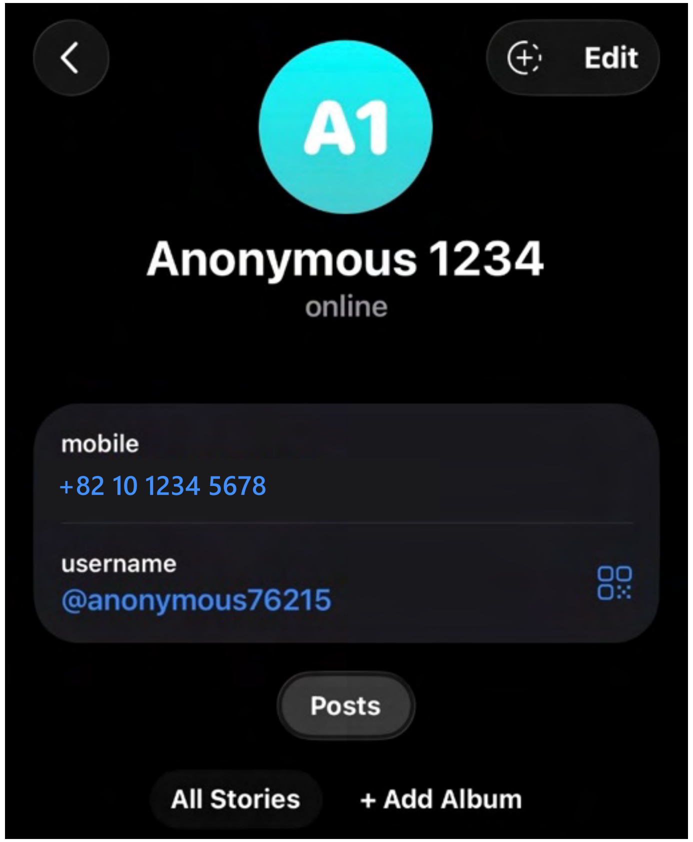


Chain 1: De-anonymization via cross-platform linking

Chain 1: De-anonymization via cross-platform linking

Who is this in App *B*?:
+82 10 1234 5678

App *B*



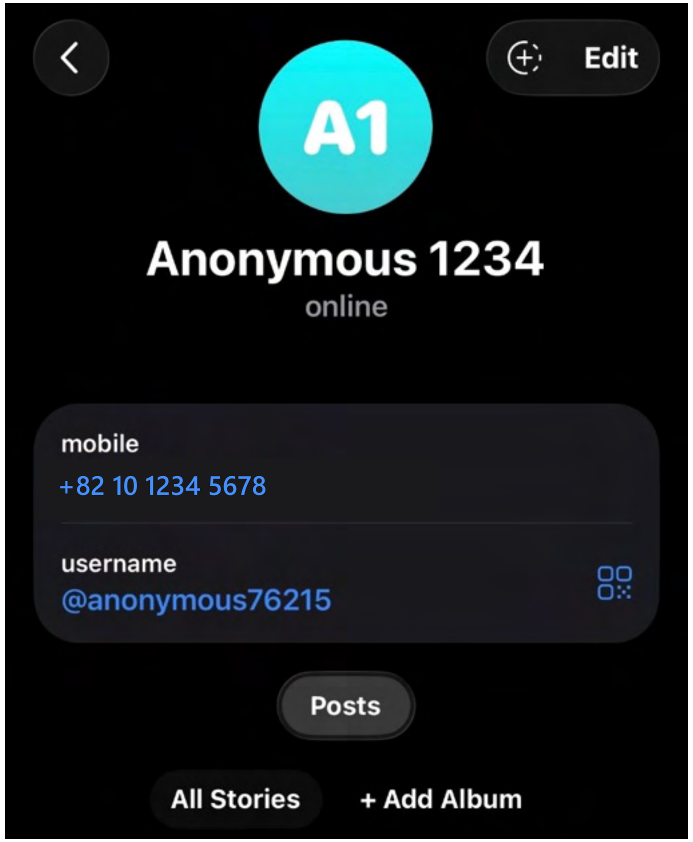
App *B*

Chain 1: De-anonymization via cross-platform linking

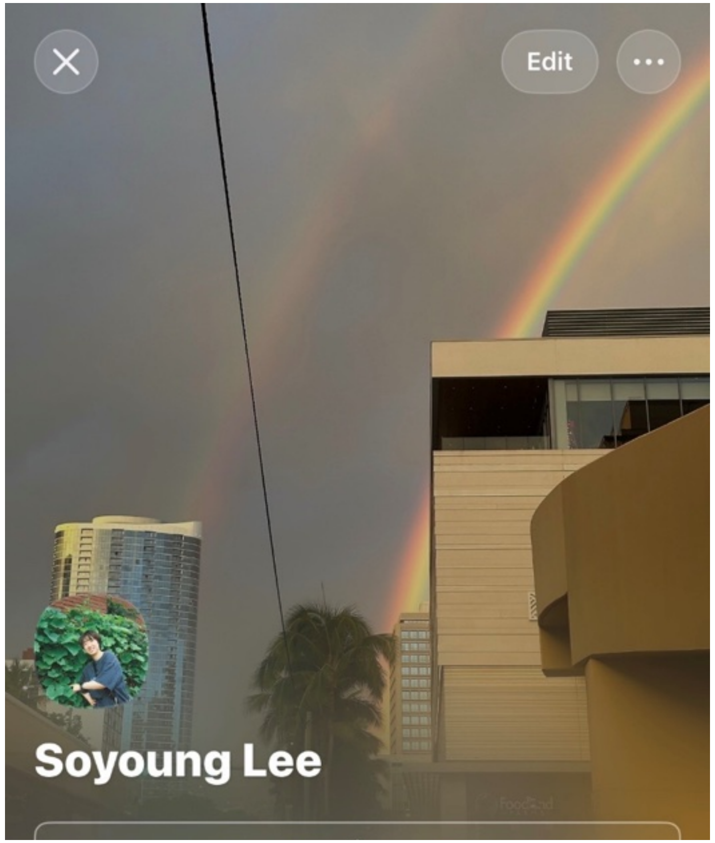
Who is this in App *B*?:
+82 10 1234 5678

App *B*

App *A*



App *B*



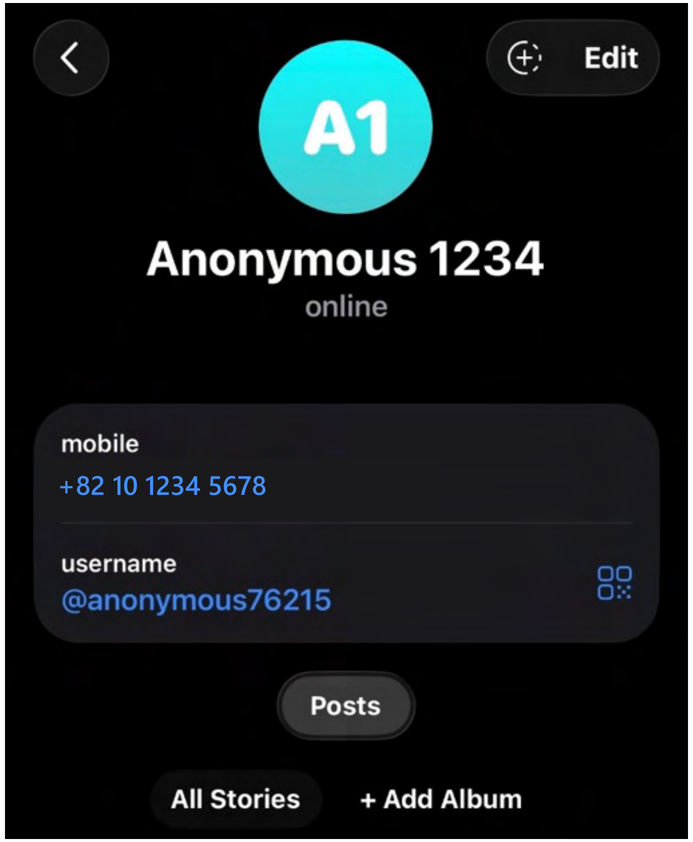
App *A*

Chain 1: De-anonymization via cross-platform linking

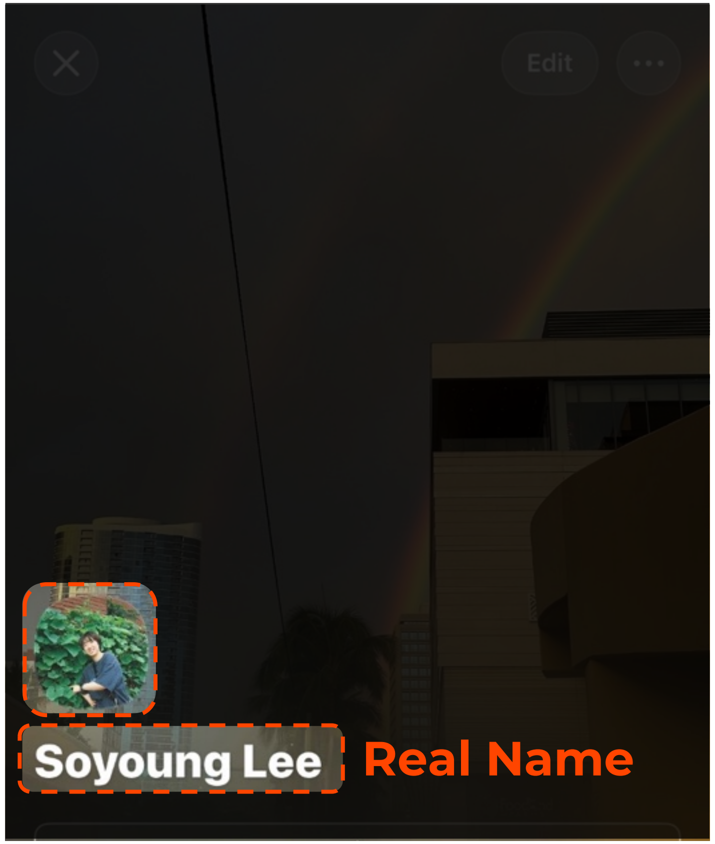
Who is this in App *B*?:
+82 10 1234 5678

App *B*

App *A*



App *B*

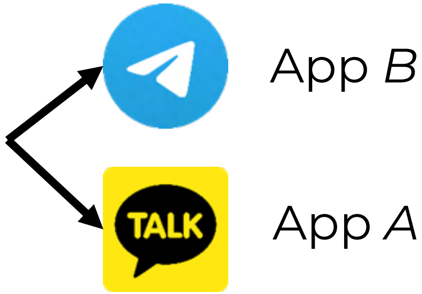


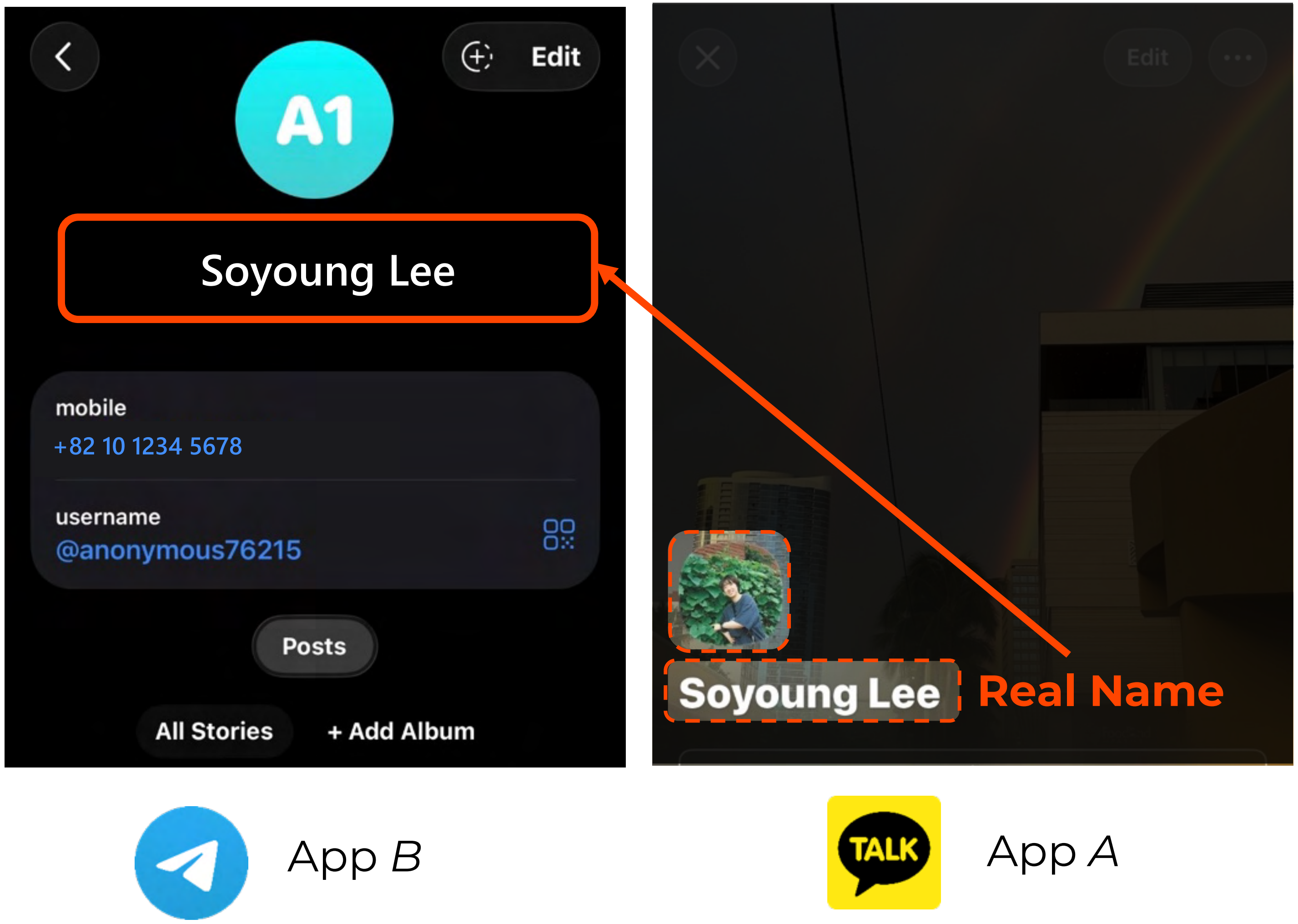
App *A*

Chain 1: De-anonymization via cross-platform linking

Who is this in App *B*?:
+82 10 1234 5678

Soyoung's Account!





Chain 1: De-anonymization via cross-platform linking

- Random samples 1,000 phone numbers

Chain 1: De-anonymization via cross-platform linking

- Random samples 1,000 phone numbers

88 profiles

40 anonymous (45%)

Chain 1: De-anonymization via cross-platform linking

- Random samples 1,000 phone numbers

88 profiles

40 anonymous (45%)

→

22 real-name profiles
(**55%** success rate)

Chain 1: De-anonymization via cross-platform linking

- Random samples 1,000 phone numbers

88 profiles
40 anonymous (45%)

22 real-name profiles
(**55%** success rate)

42 hidden profiles

5 hidden profiles

Chain 1: De-anonymization via cross-platform linking

- Random samples 1,000 phone numbers

88 profiles
40 anonymous (45%) → 22 real-name profiles (**55%** success rate)

42 hidden profiles → 30 real-name profiles (**71%** success rate)

5 hidden profiles → 4 real-name profiles (**80%** success rate)

Chain 1: De-anonymization via cross-platform linking

- Random samples 1,000 phone numbers

88 profiles
40 anonymous (45%) → 22 real-name profiles (**55%** success rate)

42 hidden profiles → 30 real-name profiles (**71%** success rate)

5 hidden profiles → 4 real-name profiles (**80%** success rate)

Why?

Chain 1: De-anonymization via cross-platform linking

- Random samples 1,000 phone numbers

88 profiles
40 anonymous (45%) → 22 real-name profiles (**55%** success rate)

42 hidden profiles → 30 real-name profiles (**71%** success rate)

5 hidden profiles → 4 real-name profiles (**80%** success rate)

Why?

Chain 1: De-anonymization via cross-platform linking

- Random samples 1,000 phone numbers

When a platform dominates a country, attacks become highly effective

Why?

88 pro

40 anony

42 hidden profiles

(**71%** success rate)

5 hidden profiles

4 real-name profiles
(**80%** success rate)

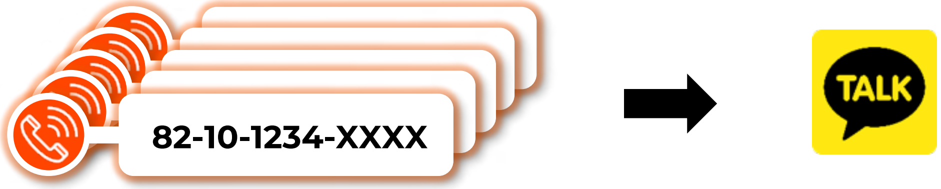
Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A (two authors)

Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A (two authors)

1. Contact discovery



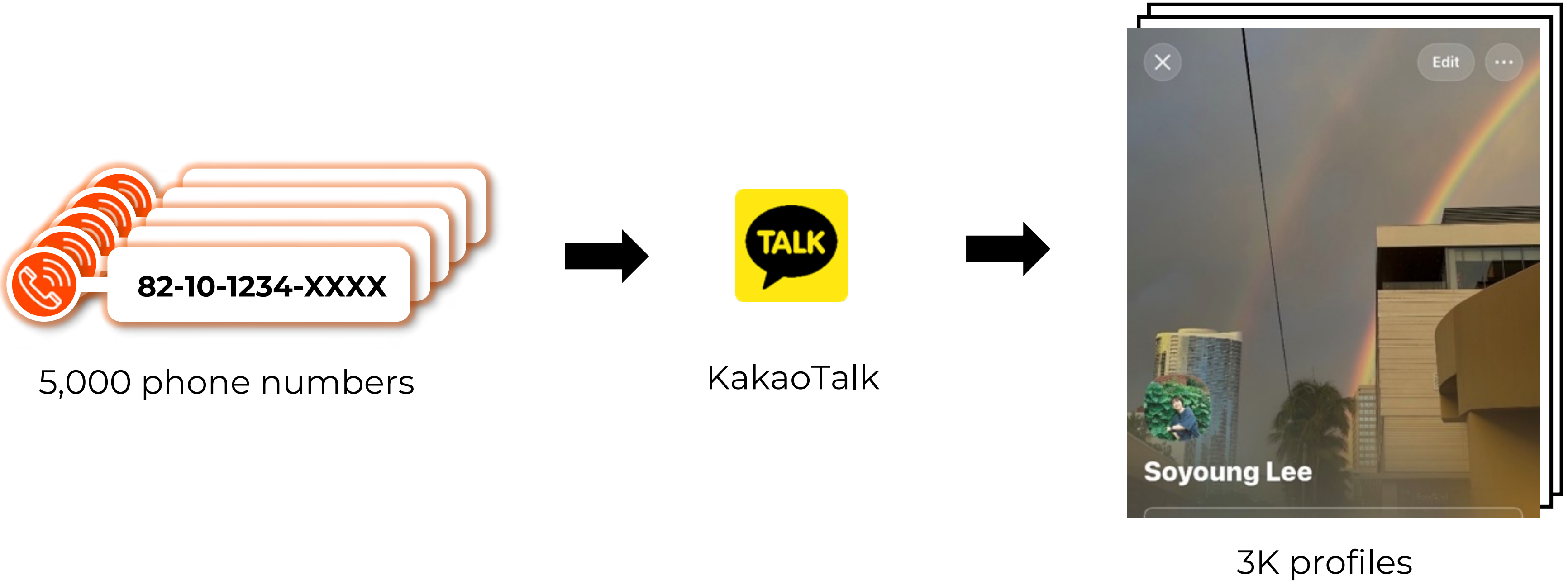
5,000 phone numbers

KakaoTalk

Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A
  (two authors)

1. Contact discovery

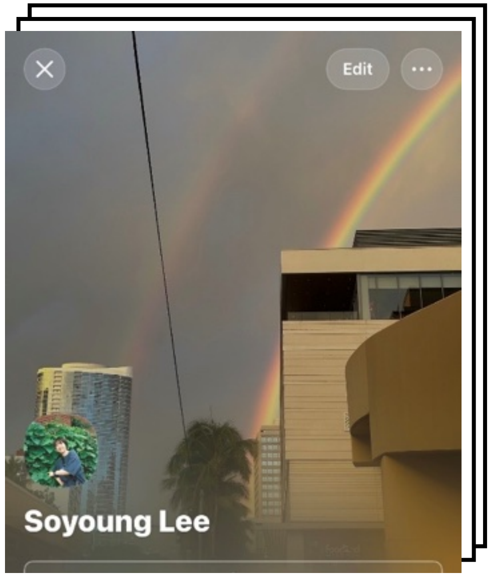


Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A

(two authors)

2. Matching profile images



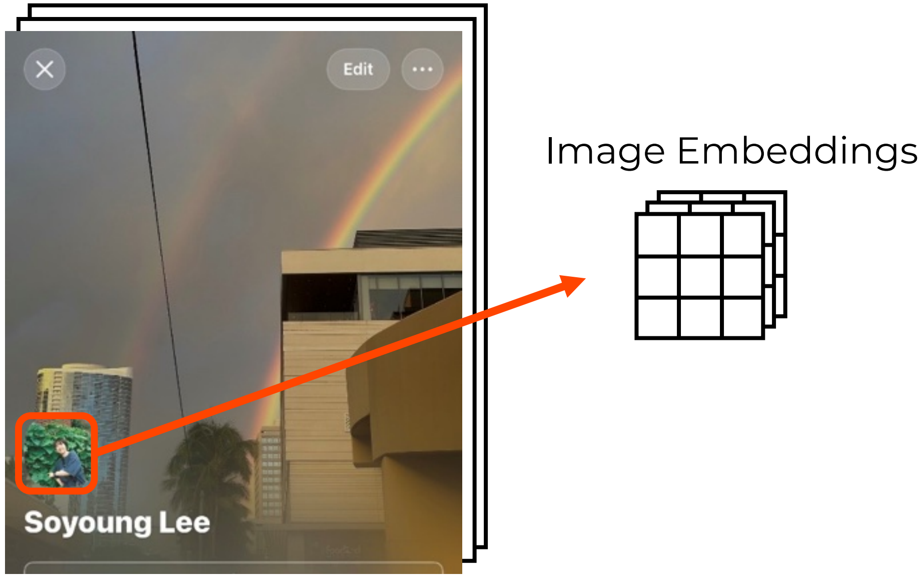
3K profiles

Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A

(two authors)

2. Matching profile images



3K profiles

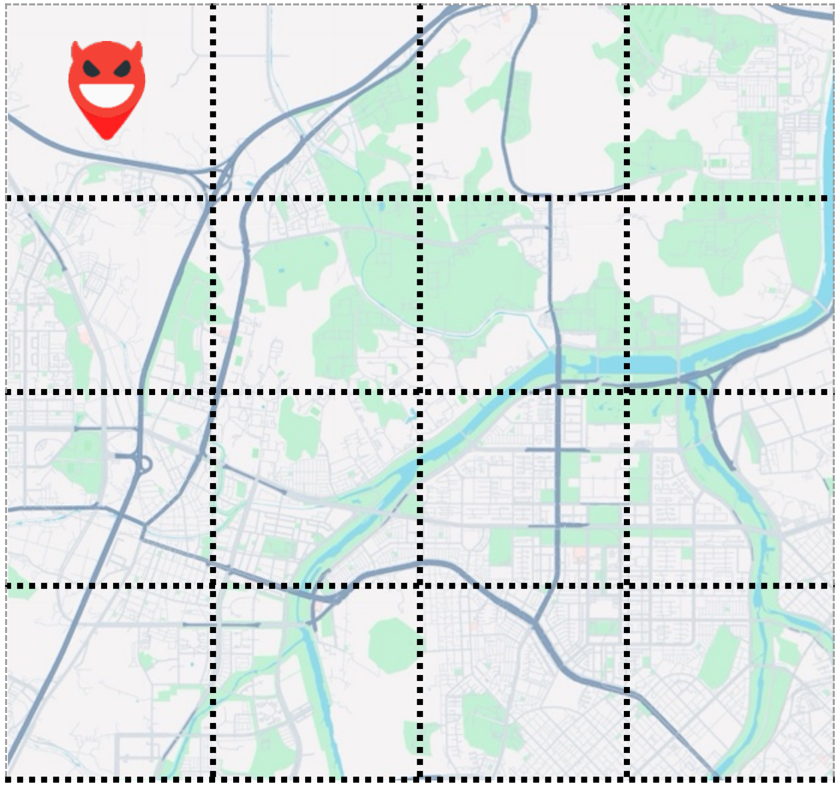
Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A (two authors)

2. Matching profile images

Find target

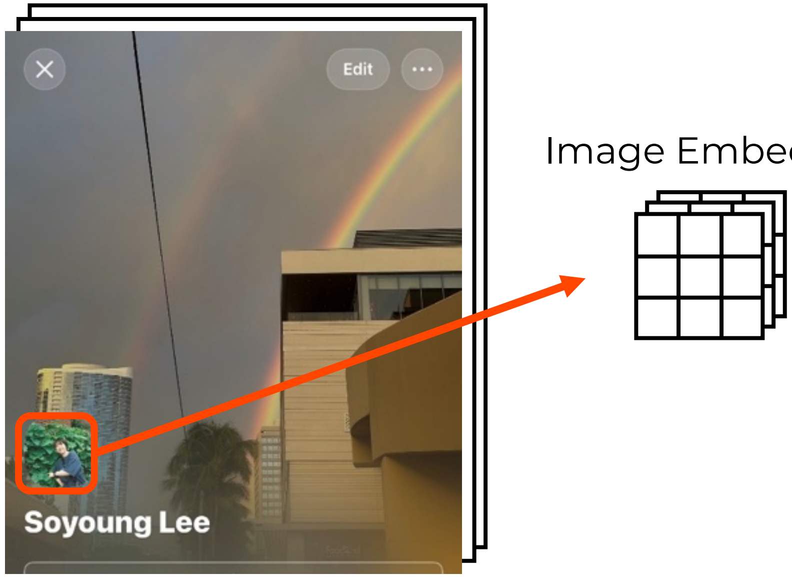
3K profiles



Chain 2: Untargeted tracking campaign

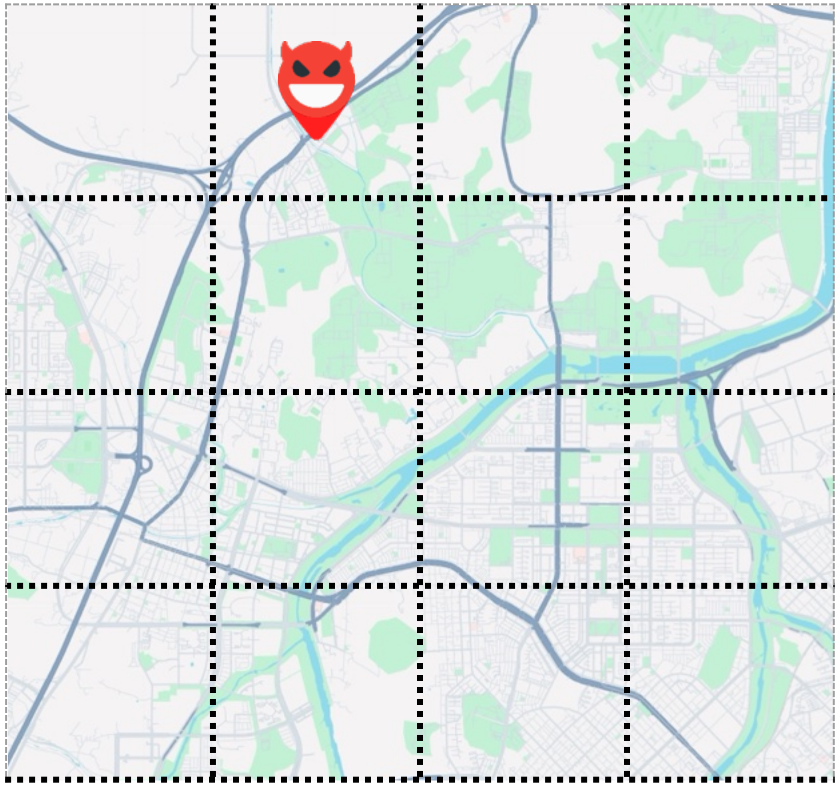
- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A (two authors)

2. Matching profile images



3K profiles

Find target

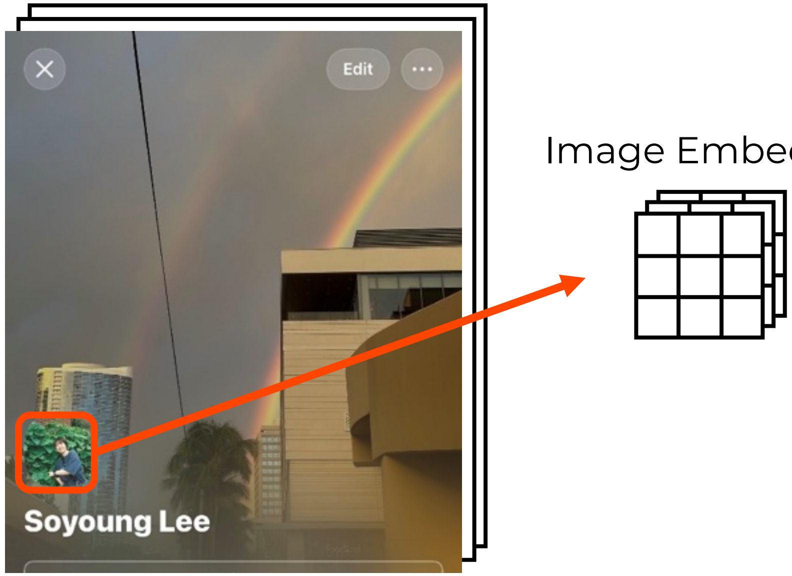


Chain 2: Untargeted tracking campaign

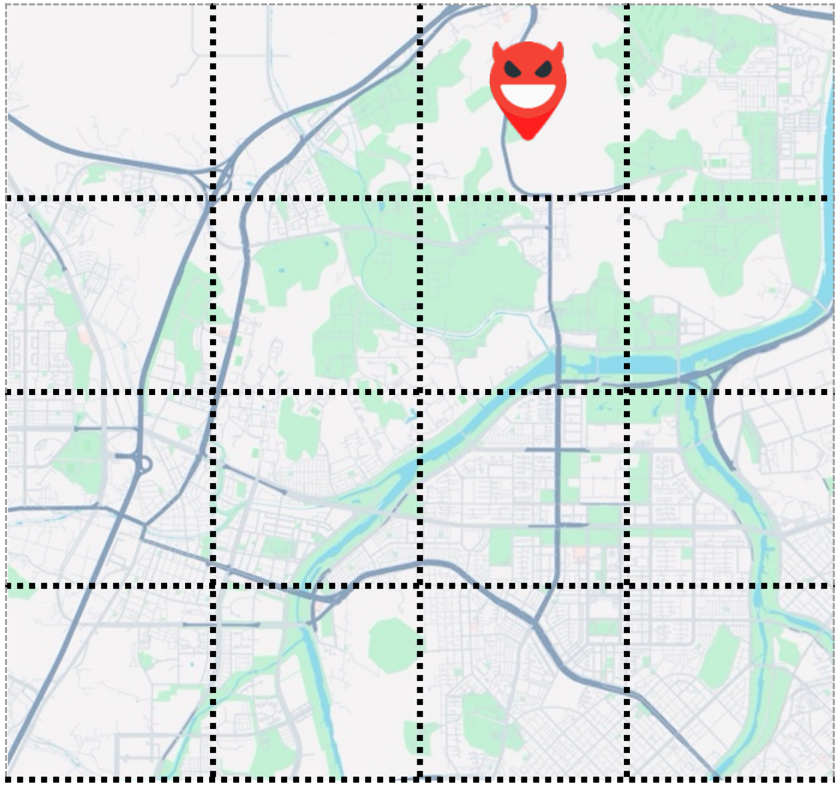
- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A (two authors)

2. Matching profile images

Find target



3K profiles

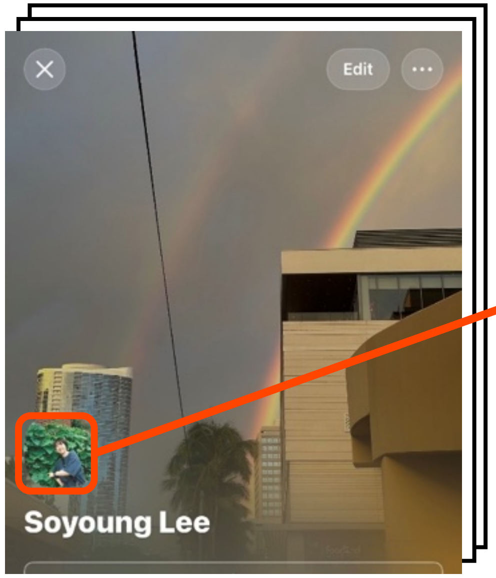


Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A (two authors)

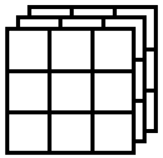
2. Matching profile images

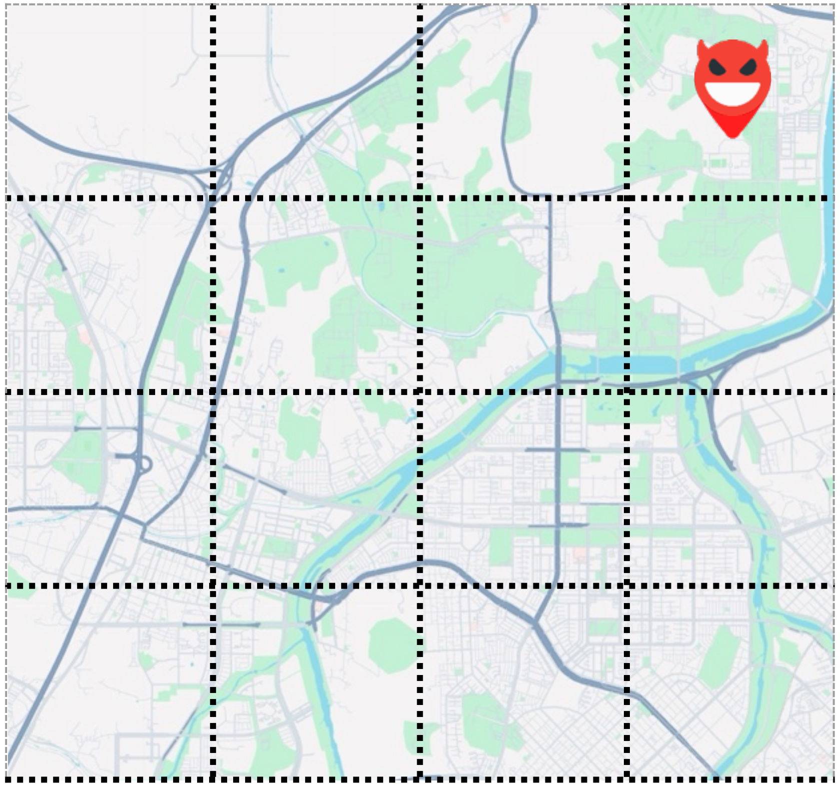
Find target



3K profiles

Image Embeddings





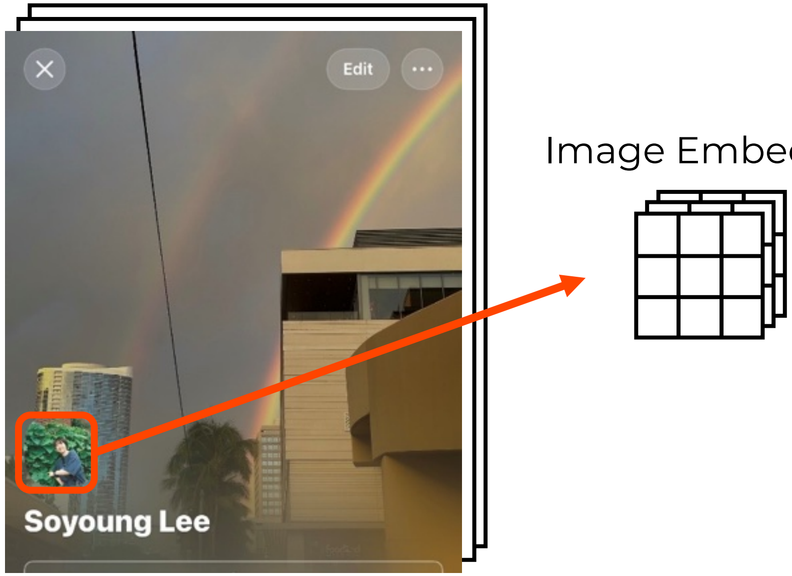
Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A
  (two authors)

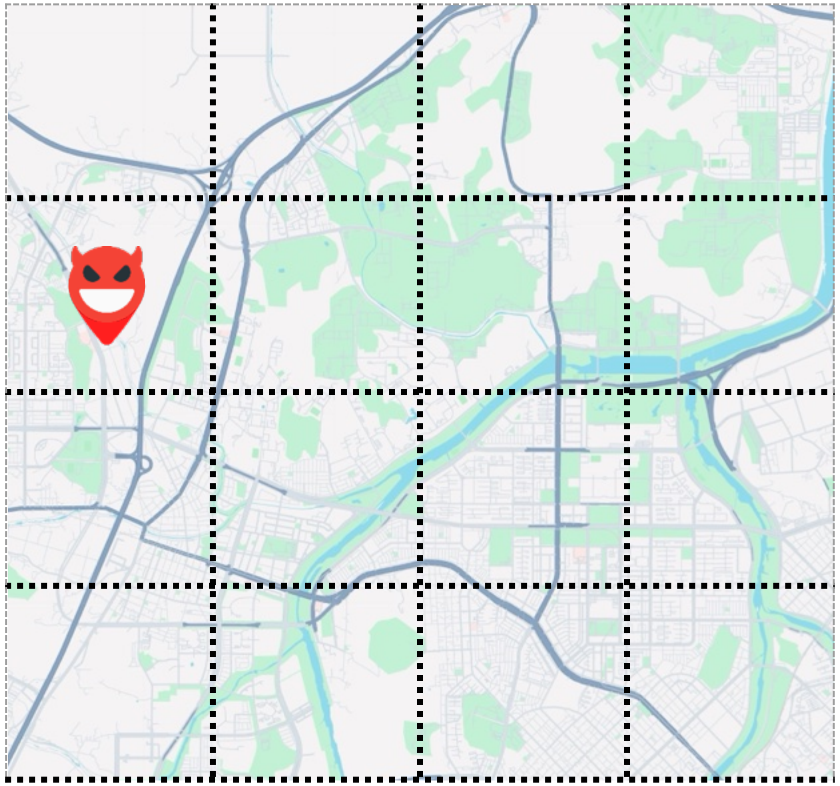
2. Matching profile images

Find target

Image Embeddings



3K profiles

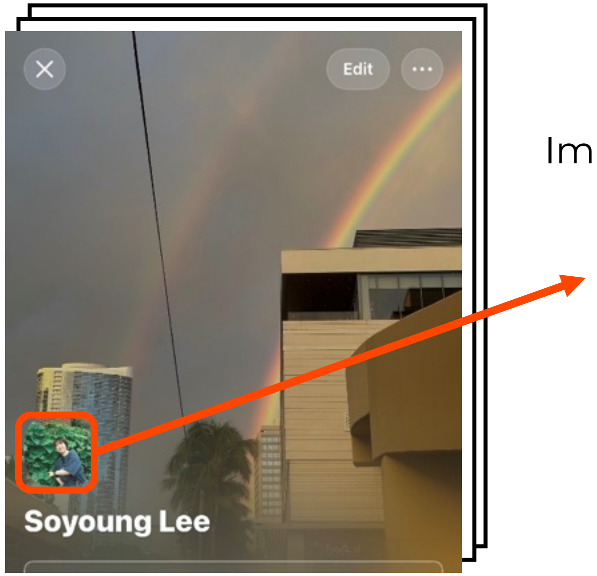


Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A

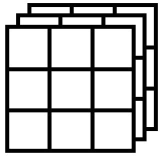
(two authors)

2. Matching profile images

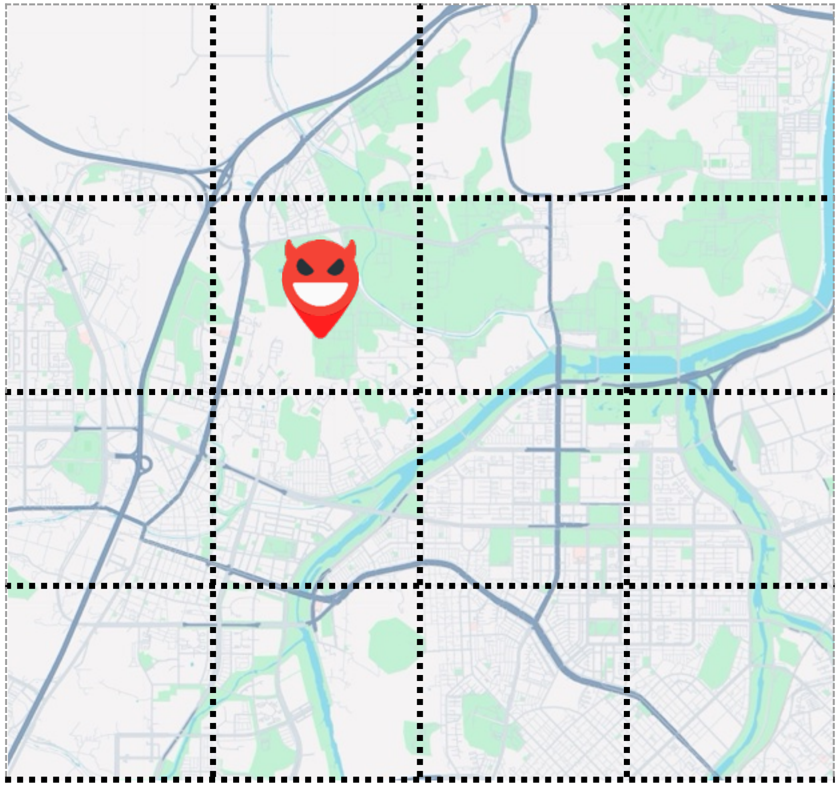


3K profiles

Image Embeddings



Find target



Chain 2: Untargeted tracking campaign

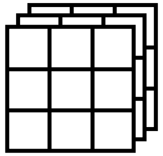
- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A

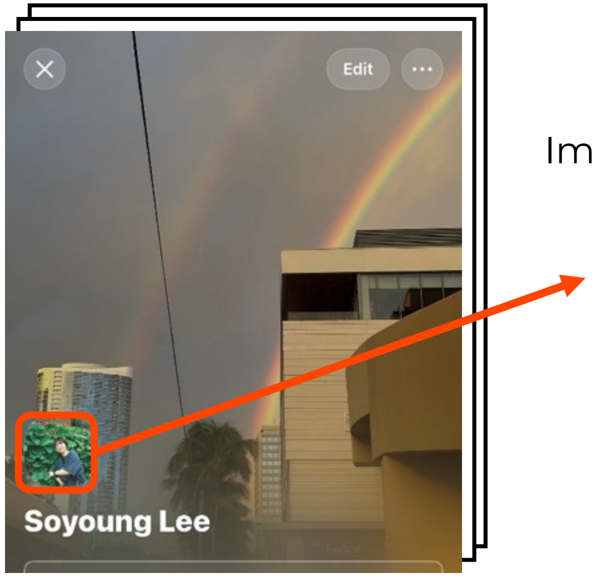
(two authors)

2. Matching profile images

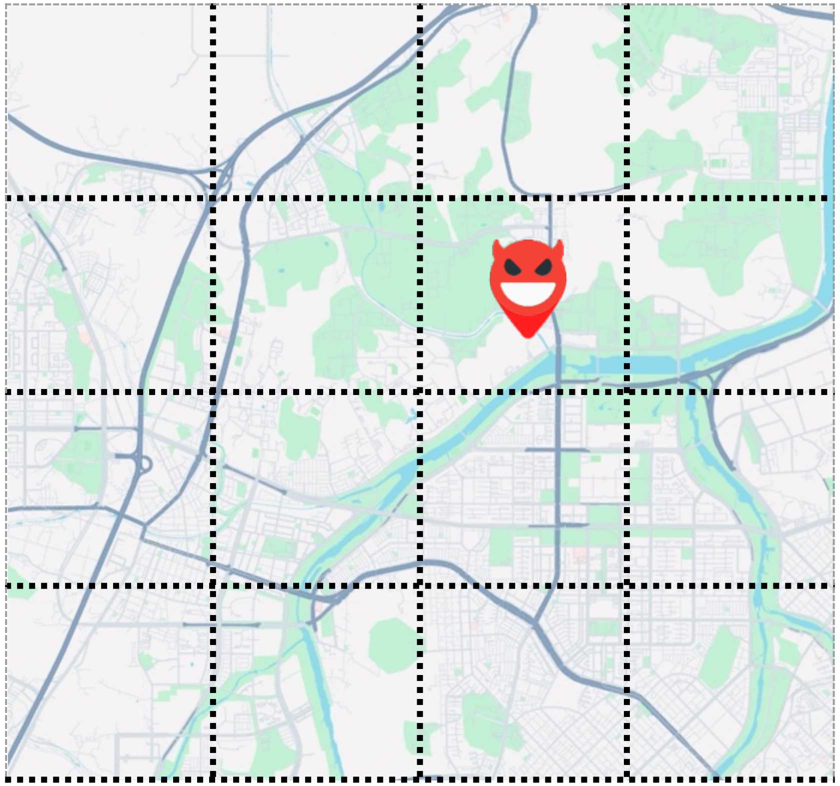
Find target

Image Embeddings





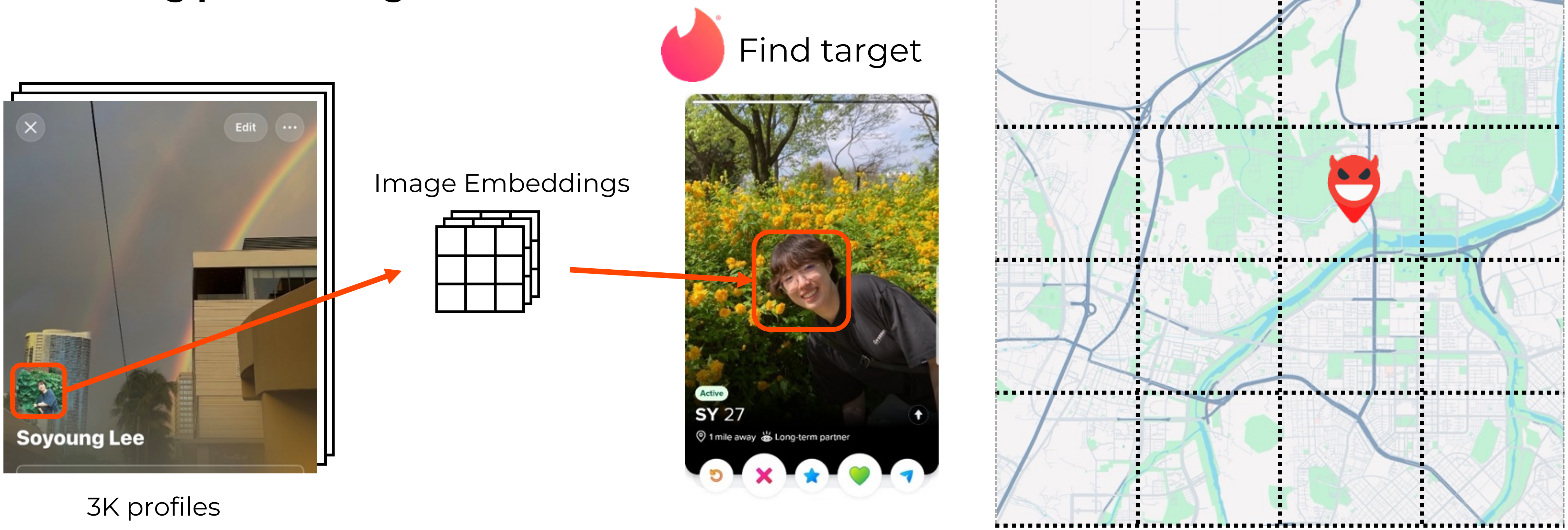
3K profiles



Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A (two authors)

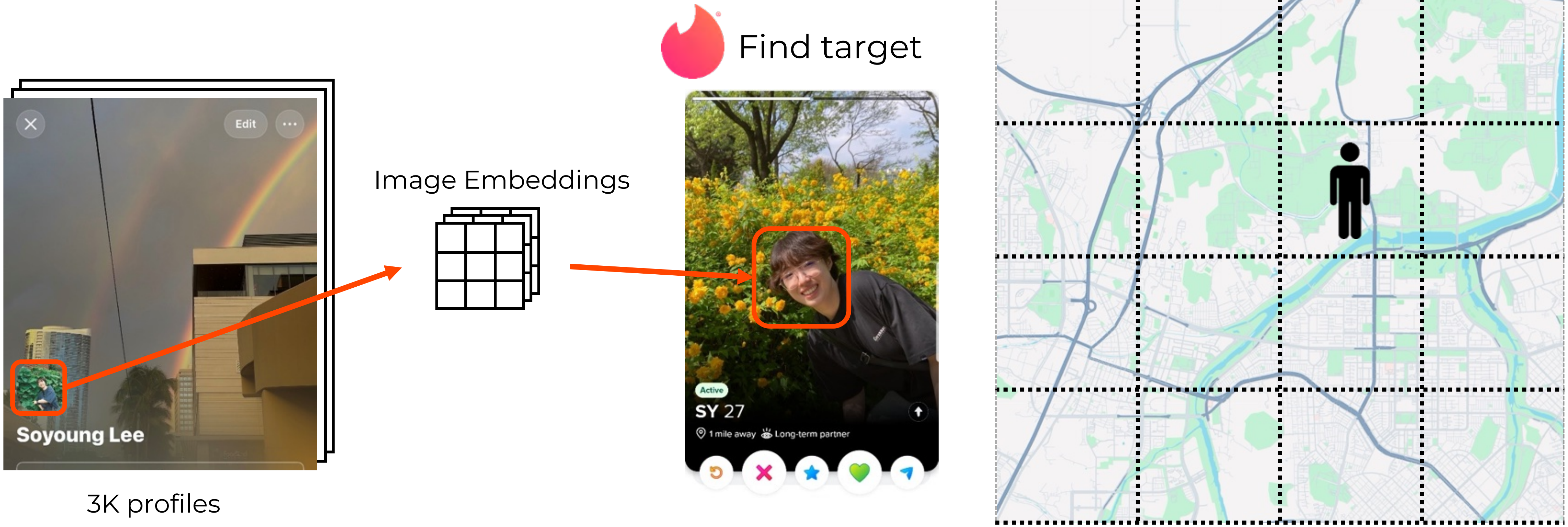
2. Matching profile images



Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A (two authors)

2. Matching profile images



Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A

(two authors)

3. Location Inference

Find target

Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area, two victims in City A (two authors)

3. Location Inference

- 1-mile boundary algorithm

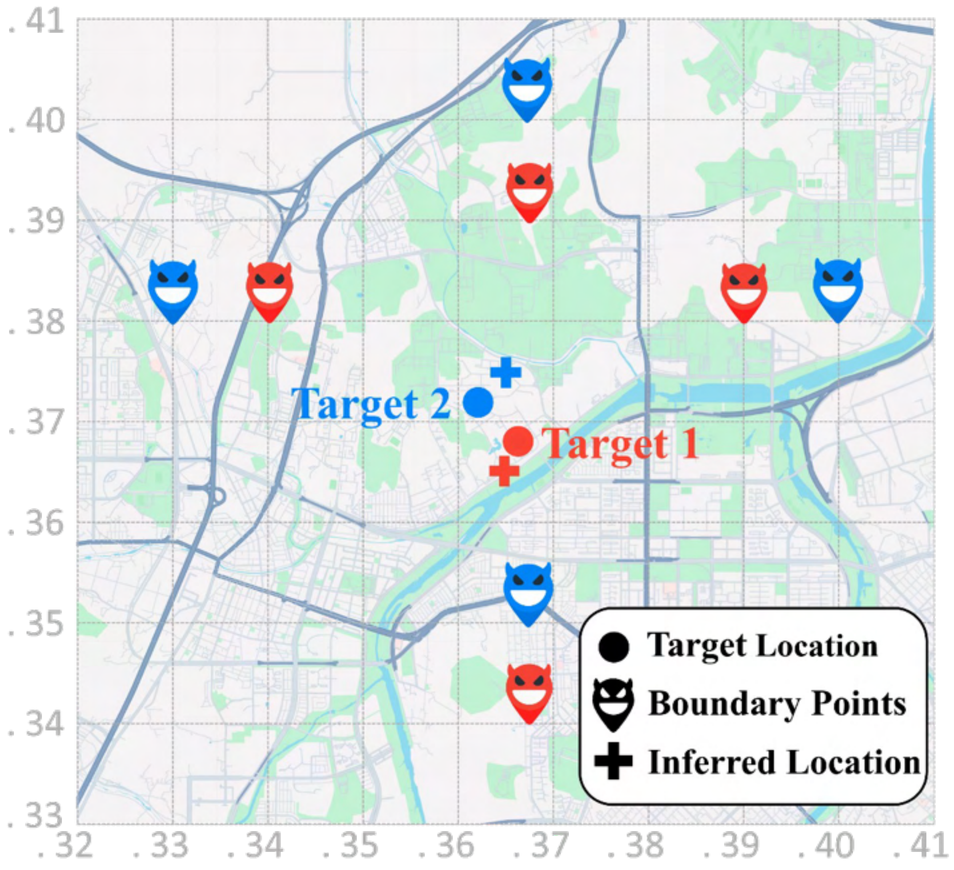
Find target

Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area (two authors in City A)

3. Location Inference

- 1-mile boundary algorithm

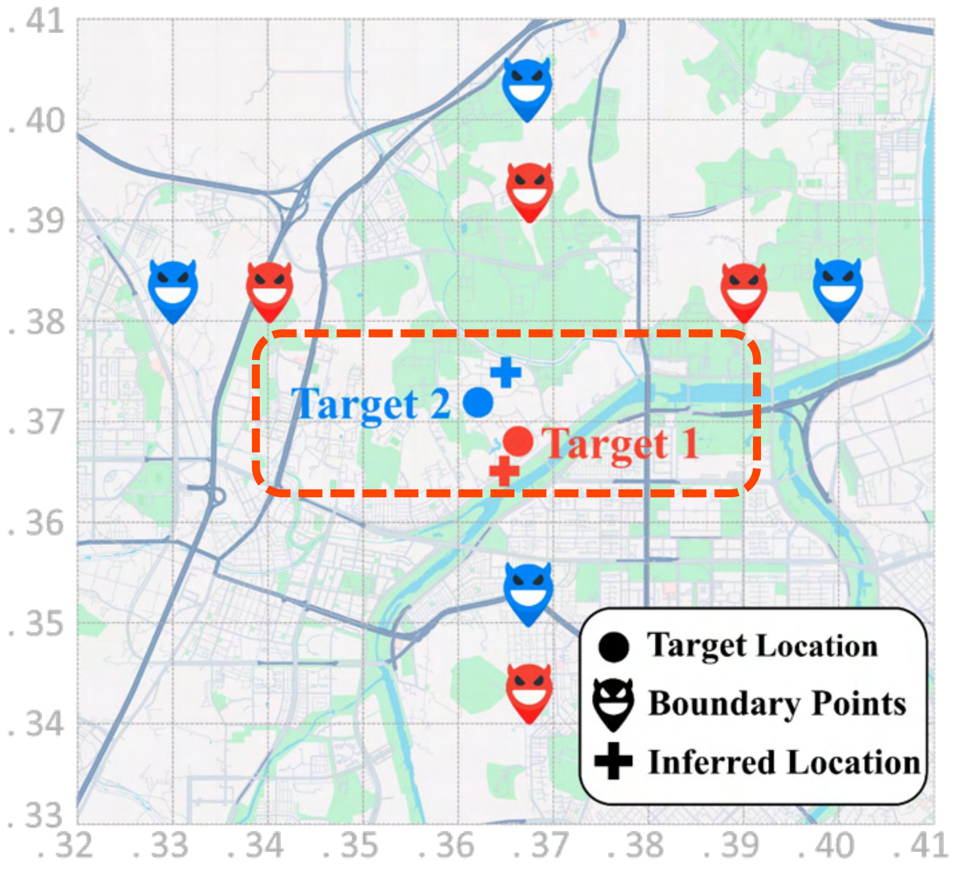


Chain 2: Untargeted tracking campaign

- Targeting 5,000 phone numbers
- Goal: Finding people in a certain area (two authors in City A)

3. Location Inference

- 1-mile boundary algorithm
- Successfully find two target users (336m and 418m errors)



Mitigation – Contact Discovery

1. Query throttling
    - Set a strict daily limit (e.g., under 100 registrations)
        → Disrupt the service (address-book sync)
        → Adversaries can easily bypass this using multiple accounts

Mitigation – Contact Discovery

1. Query throttling
   - Set a strict daily limit (e.g., under 100 registrations)
     - → Disrupt the service (address-book sync)
     - → Adversaries can easily bypass this using multiple accounts

Brute-force **contact discovery** attempts

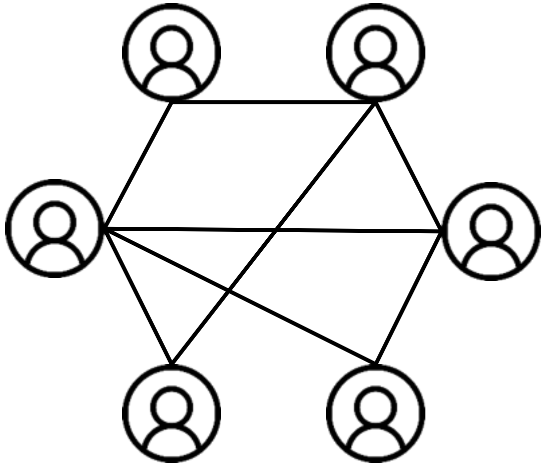
vs.

Registration attempts from **benign users**

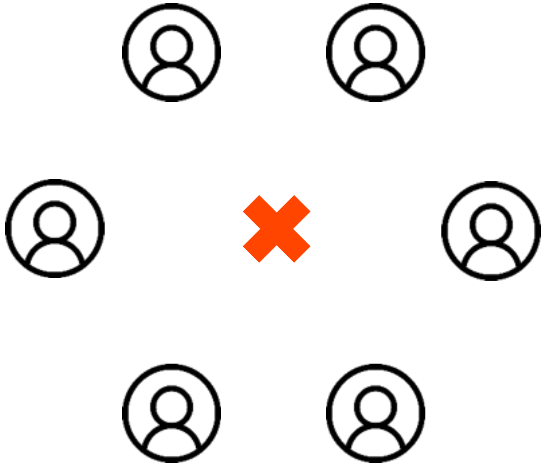
Mitigation – Contact Discovery

2. Social Circles

- Structural difference in social relationships



Benign user's address book

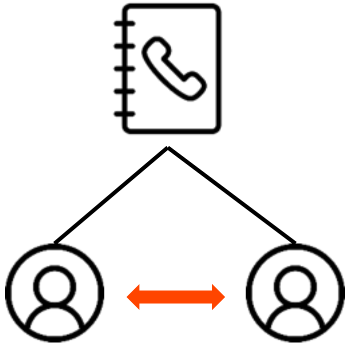


Adversary's address book
(Random generated)

Mitigation – Contact Discovery

2. Social Circles

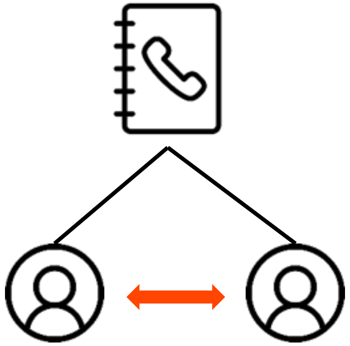
- Structural difference in social relationships



Mitigation – Contact Discovery

2. Social Circles

- Structural difference in social relationships

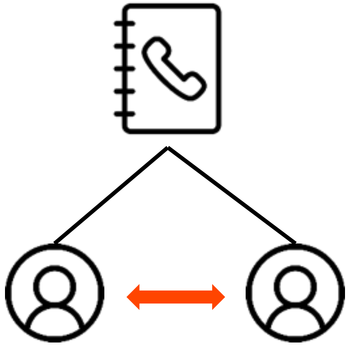


24%
(Mislove *et al.*, 2007)

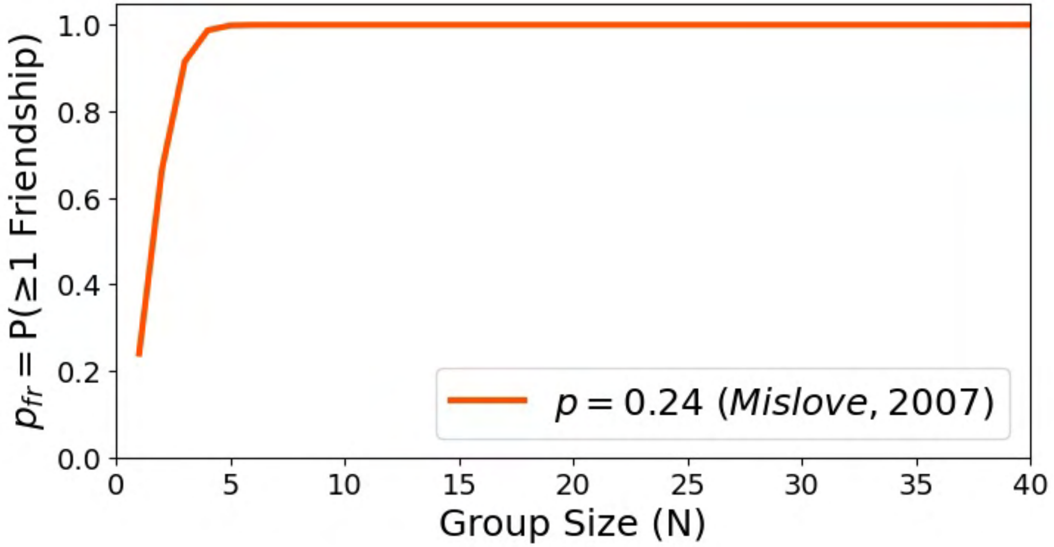
Mitigation – Contact Discovery

2. Social Circles

- Structural difference in social relationships



24%
(Mislove *et al.*, 2007)

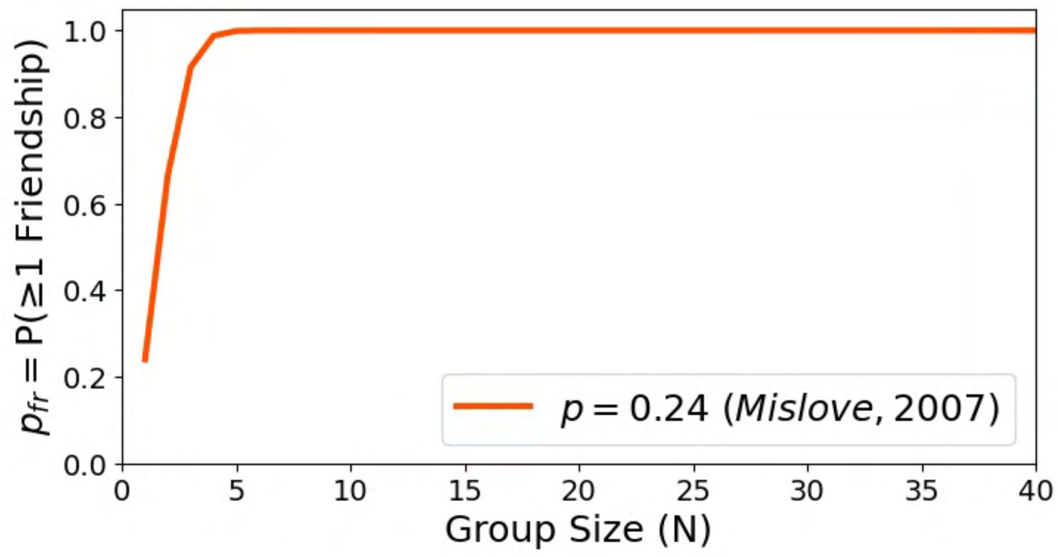


Simulation Results

Mitigation – Contact Discovery

2. Social Circles

- Structural difference in social relationships

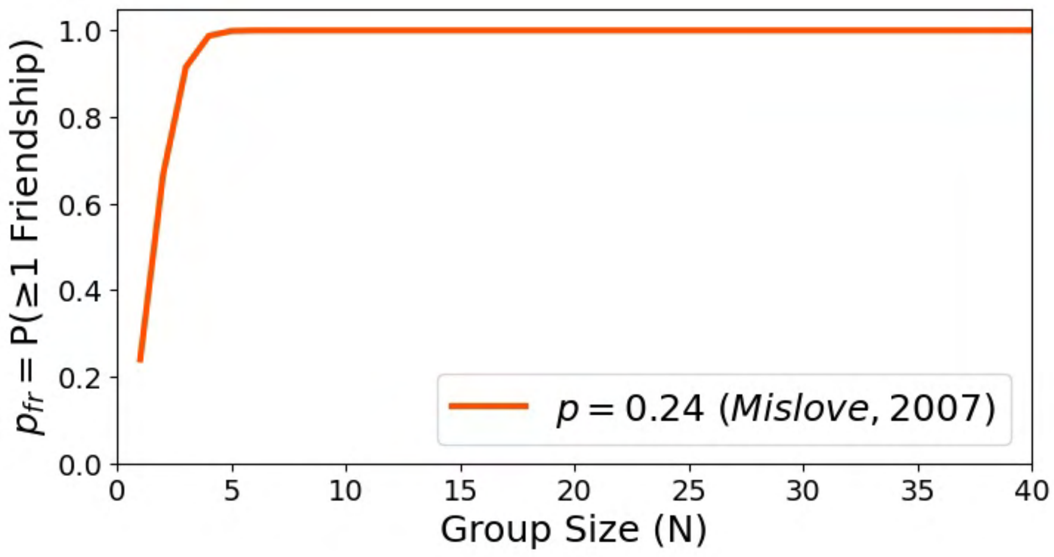


Simulation Results

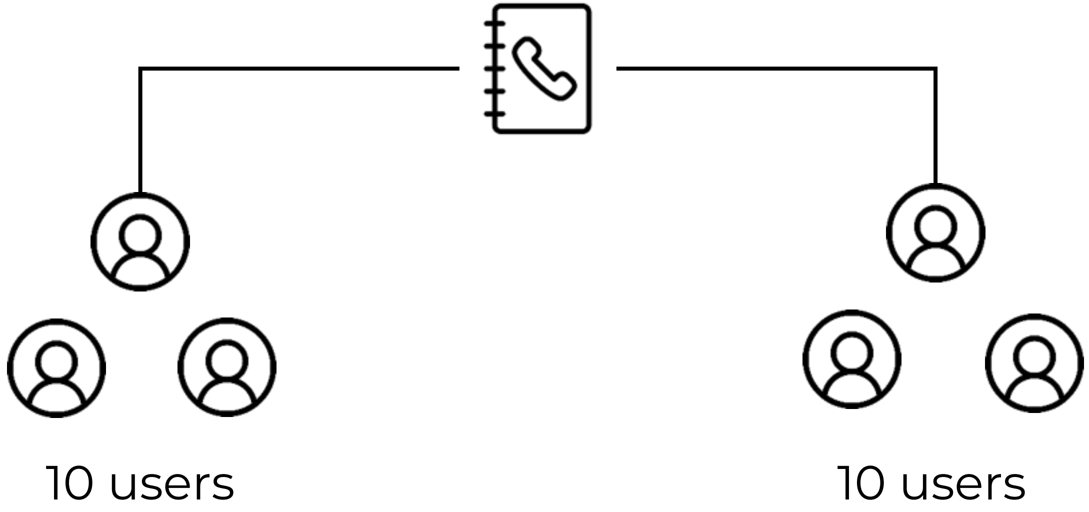
Mitigation – Contact Discovery

2. Social Circles

- Structural difference in social relationships



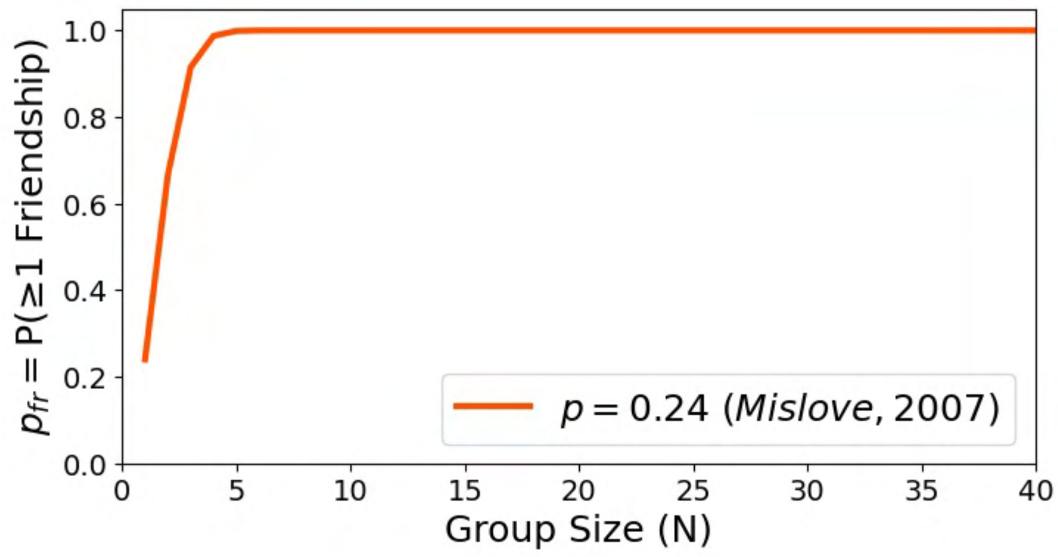
Simulation Results



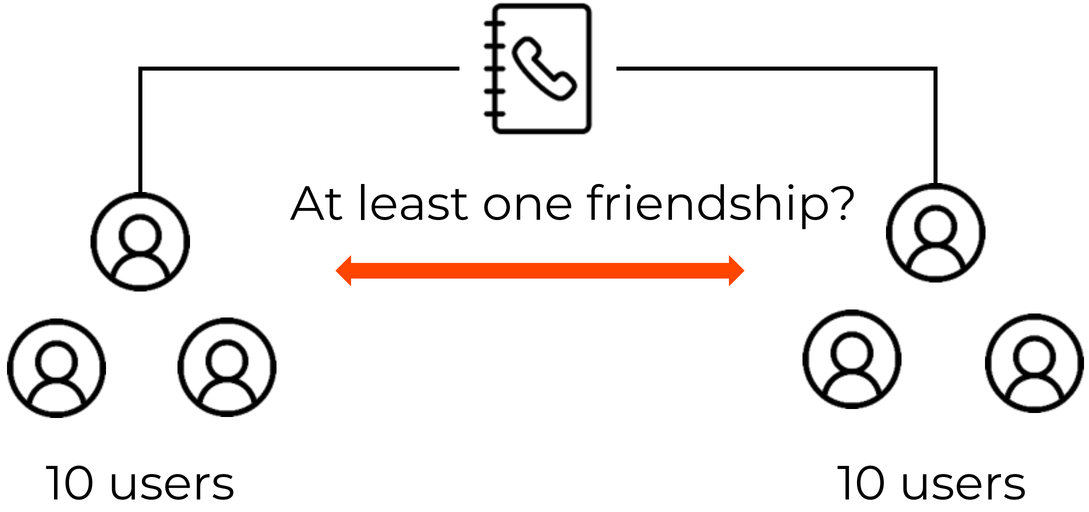
Mitigation – Contact Discovery

2. Social Circles

- Structural difference in social relationships



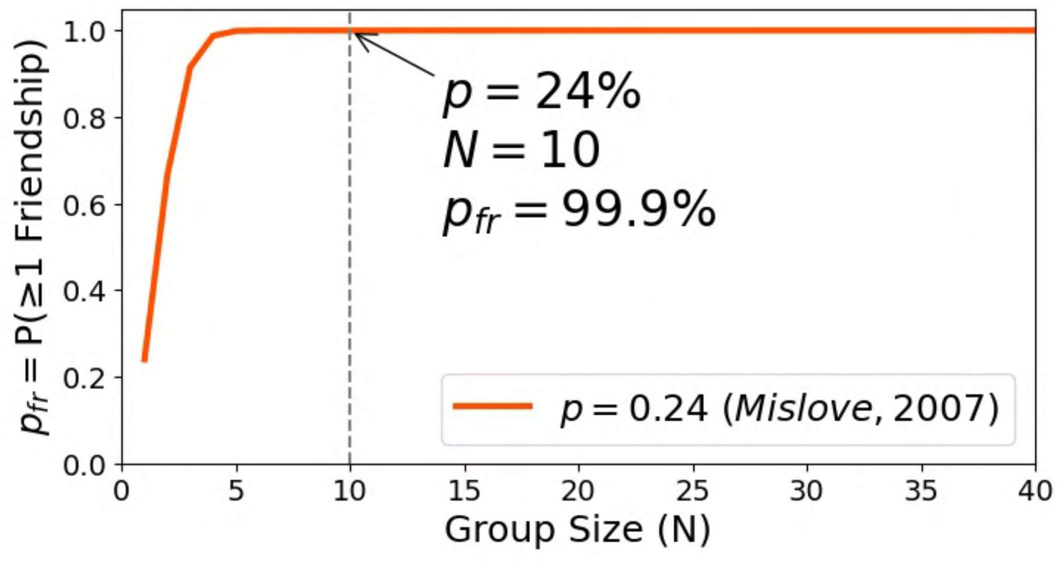
Simulation Results



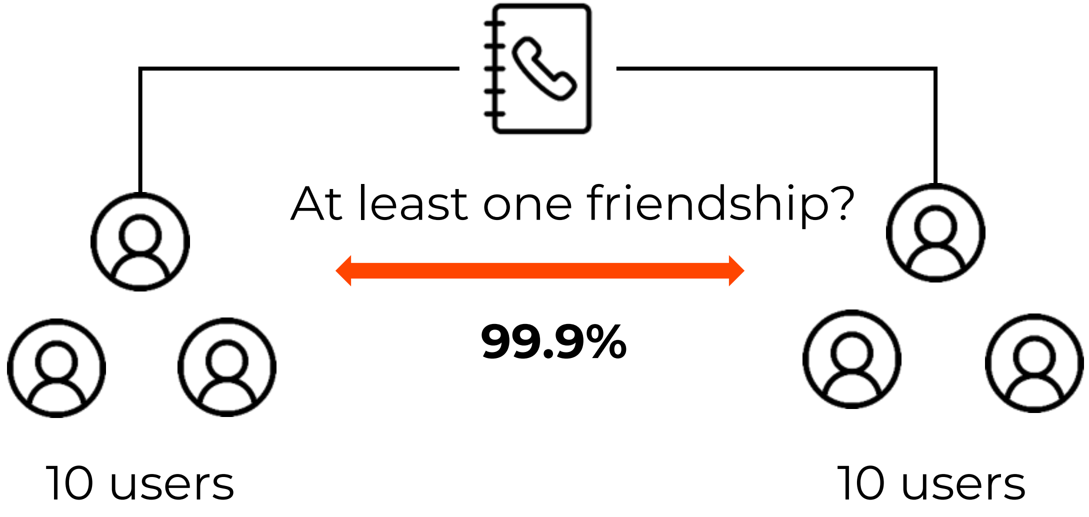
Mitigation – Contact Discovery

2. Social Circles

- Structural difference in social relationships



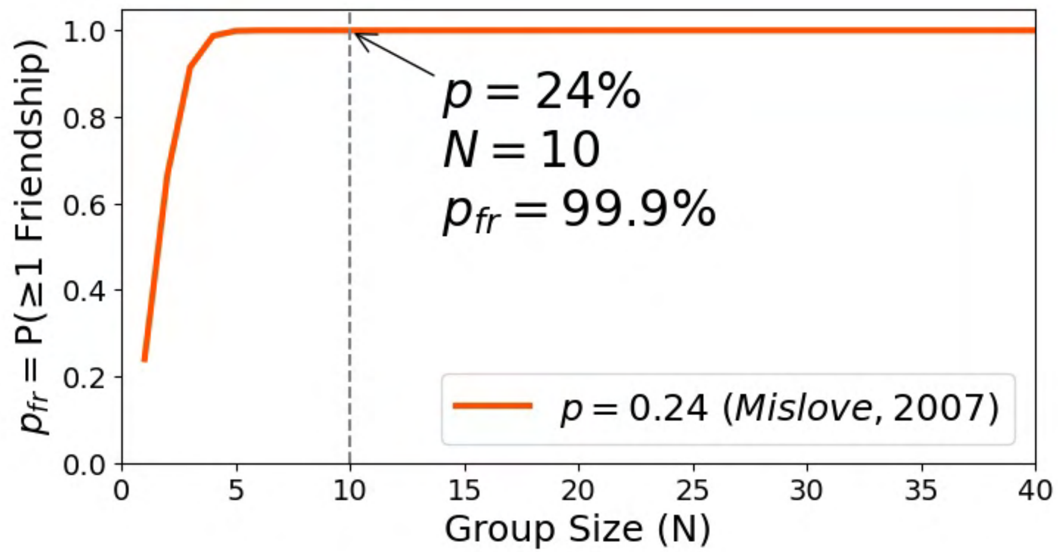
Simulation Results

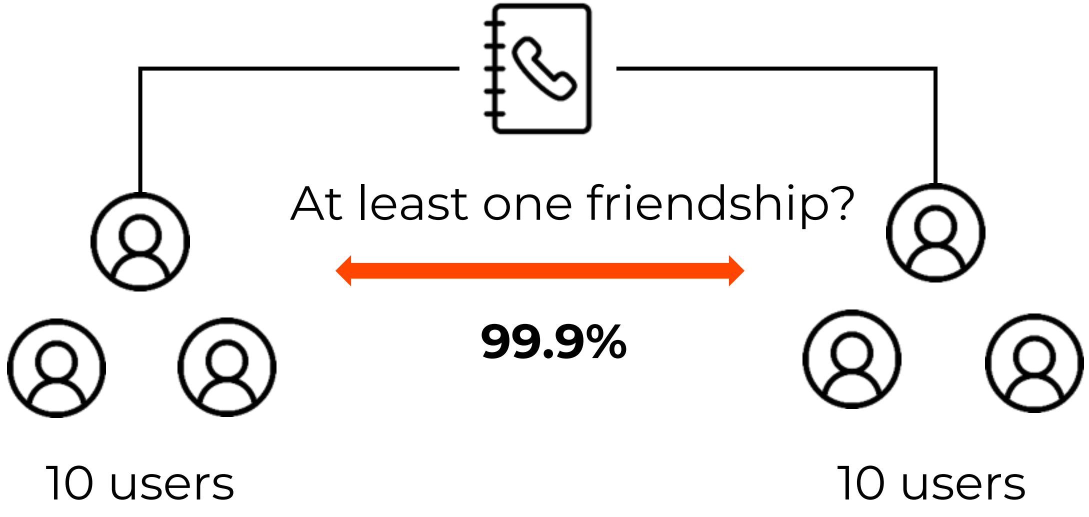


Mitigation – Contact Discovery

2. Social Circles

- Structural difference in social relationships

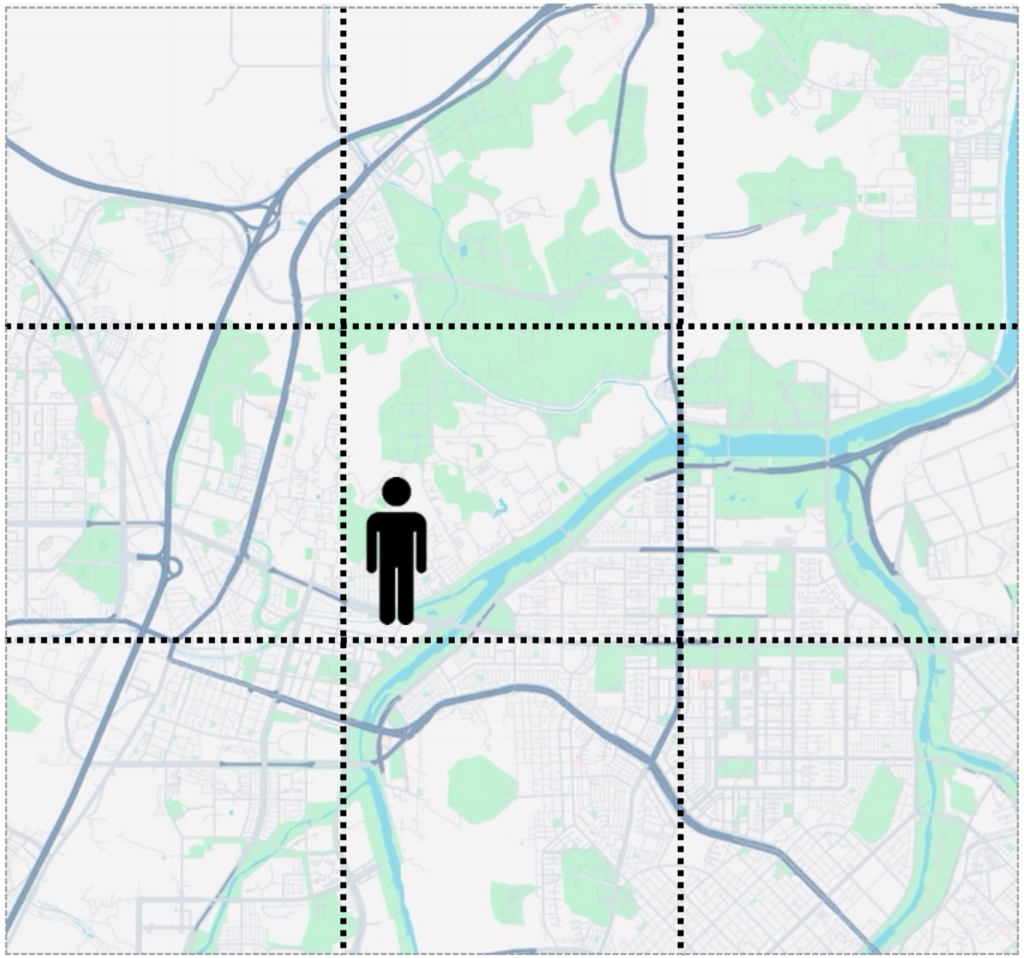




Identifying malicious attempts is possible by leveraging social circles

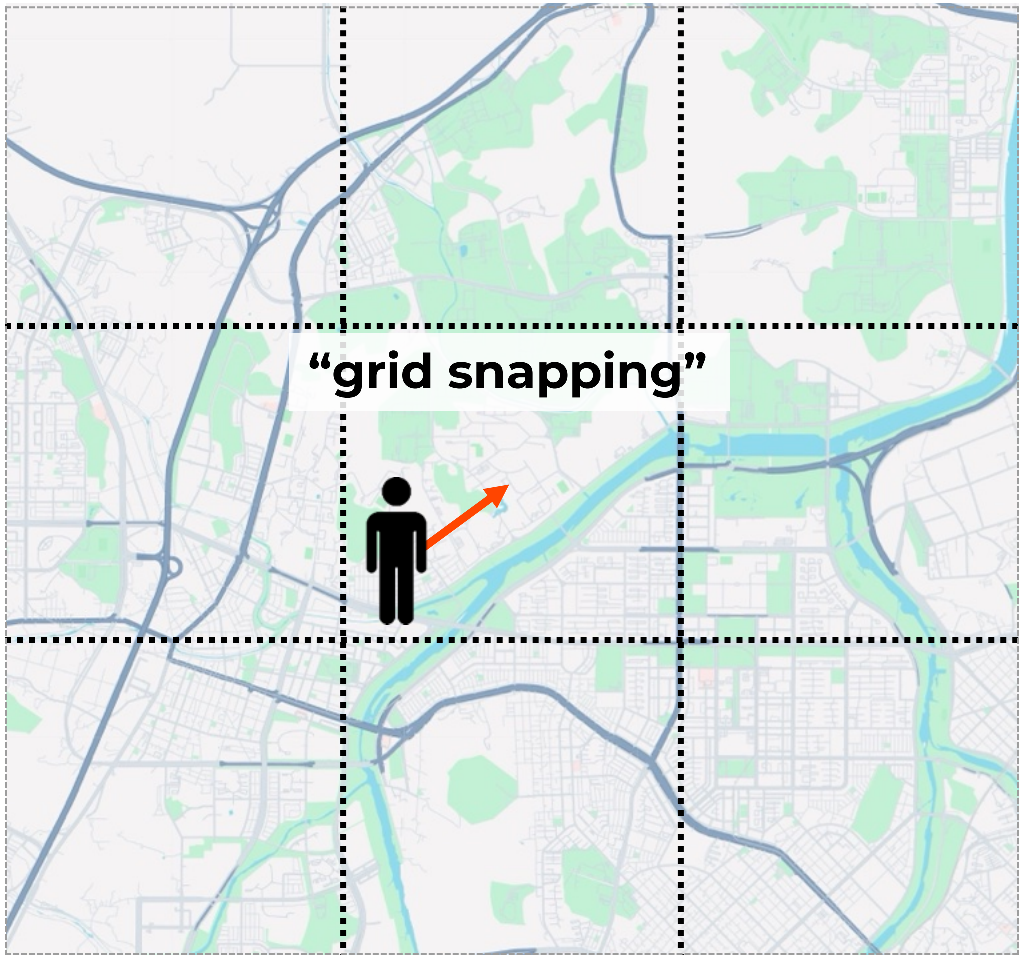
Mitigation – Location Inference

- Grid snapping



Mitigation – Location Inference

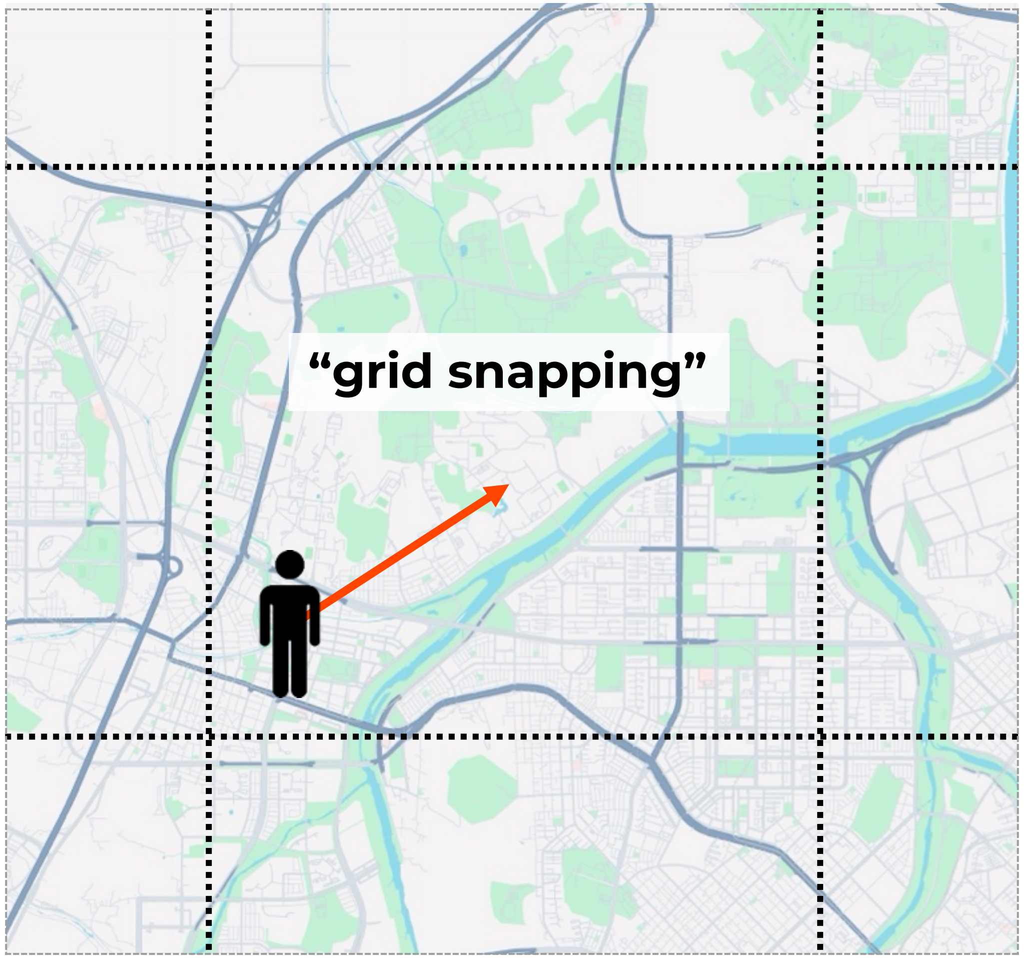
- Grid snapping
  - Mapping to the center point



Mitigation – Location Inference

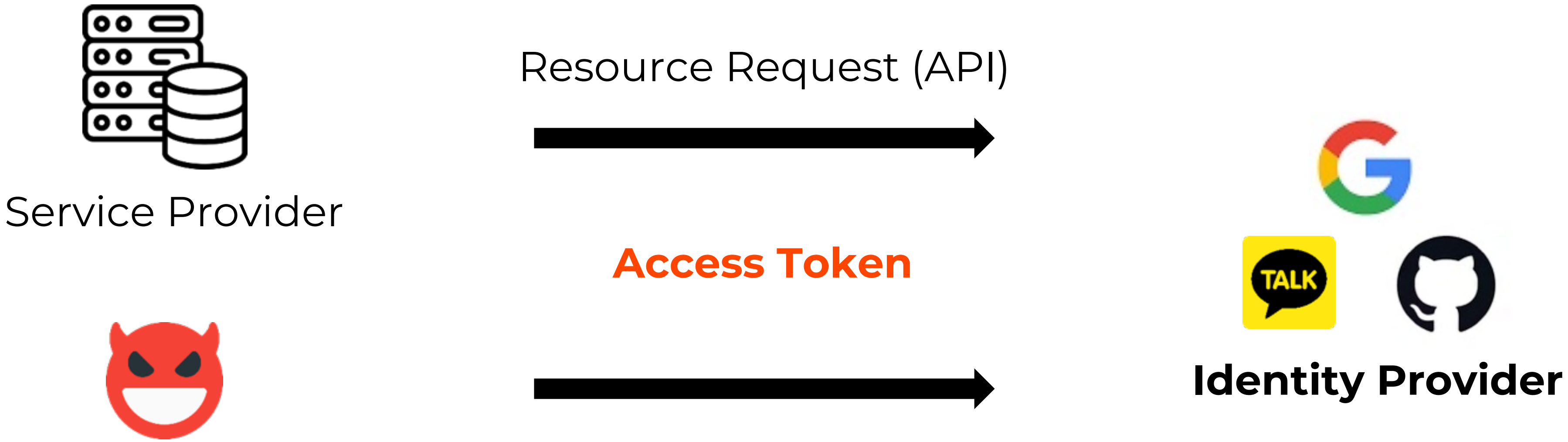
- Grid snapping
  - Mapping to the center point
  - Increasing the grid size increases the error margin

Tradeoff: Privacy vs. Usability



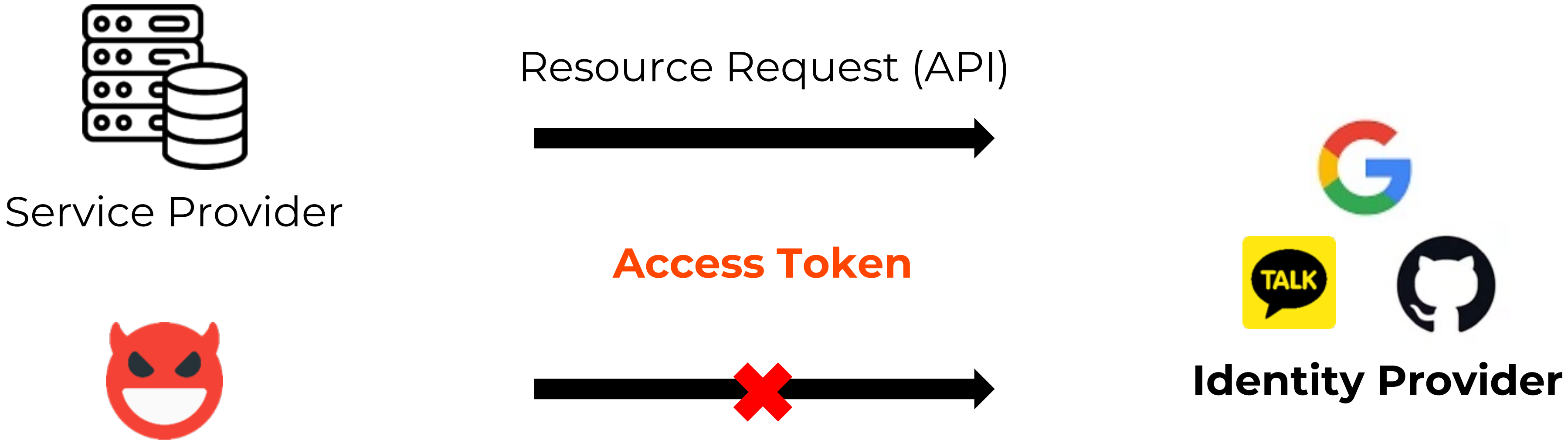
Mitigation – OAuth Token

- Using Mutual TLS protocol



Mitigation – OAuth Token

- Using Mutual TLS protocol



Conclusion

Conclusion

Evaluate **privacy attacks** and propose concrete **end-to-end attacks**

1. Contact Discovery
2. SSO Linking
3. Location Inference

De-anonymization

Trajectory Tracking

Conclusion

Evaluate **privacy attacks** and propose concrete **end-to-end attacks**

1. Contact Discovery
2. SSO Linking
3. Location Inference

De-anonymization

Trajectory Tracking

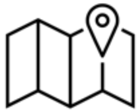
Privacy can fail by **composition** when attacks combine across apps

Conclusion

Evaluate **privacy attacks** and propose concrete **end-to-end attacks**

1. Contact Discovery
2. SSO Linking
3. Location Inference

De-anonymization



Trajectory Tracking

Privacy can fail by **composition** when attacks combine across apps

Geographically dominant messengers pose **privacy risks**.

Conclusion

Evaluate **privacy attacks** and propose concrete **end-to-end attacks**

1. Contact Discovery
2. SSO Linking
3. Location Inference

De-anonymization

Trajectory Tracking

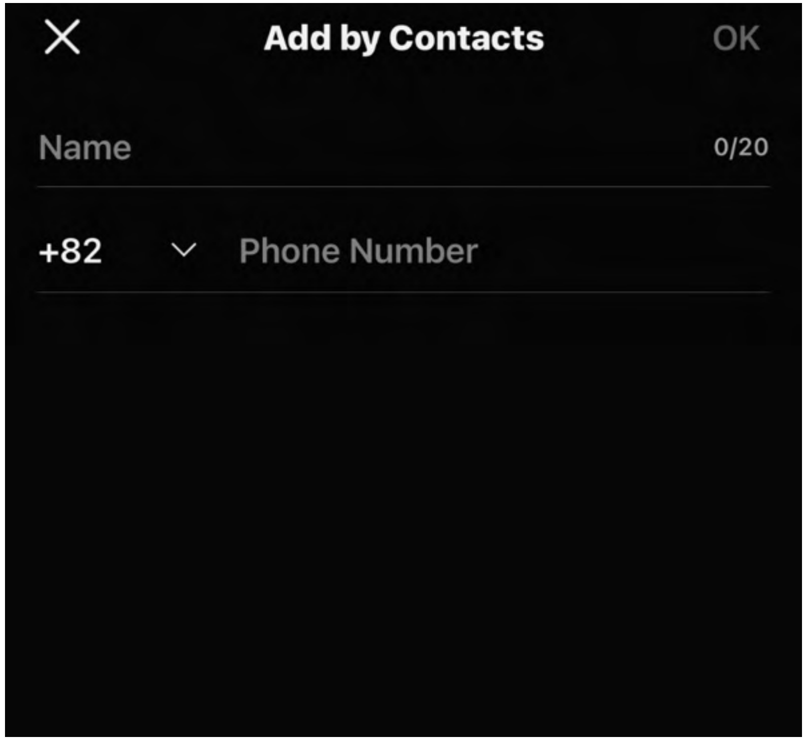
Privacy can fail by **composition** when attacks combine across apps

Geographically dominant messengers pose **privacy risks**.

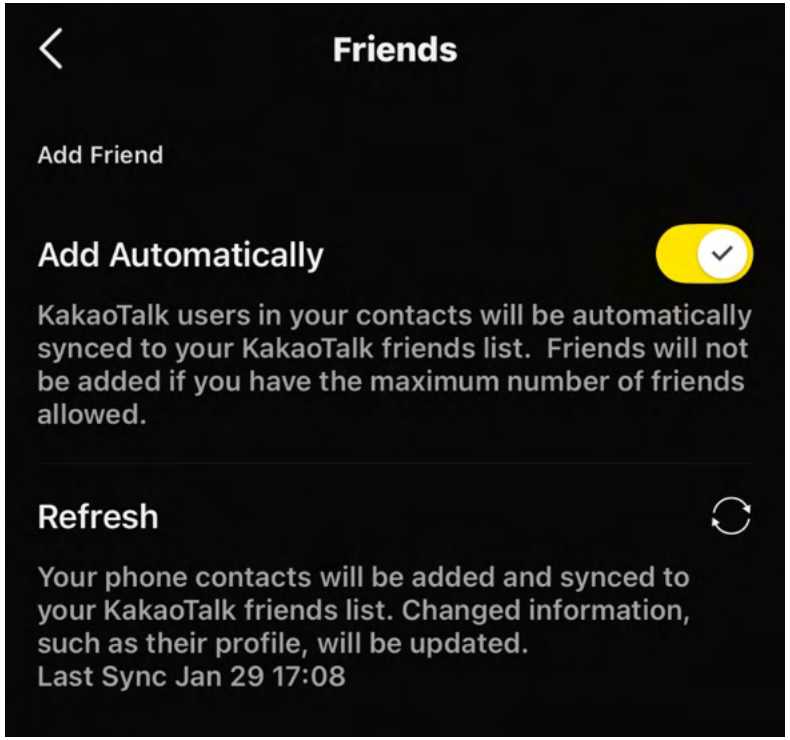
Extra slides

Attack 1: Contact discovery abuse (KakaoTalk)

- Friend Registration



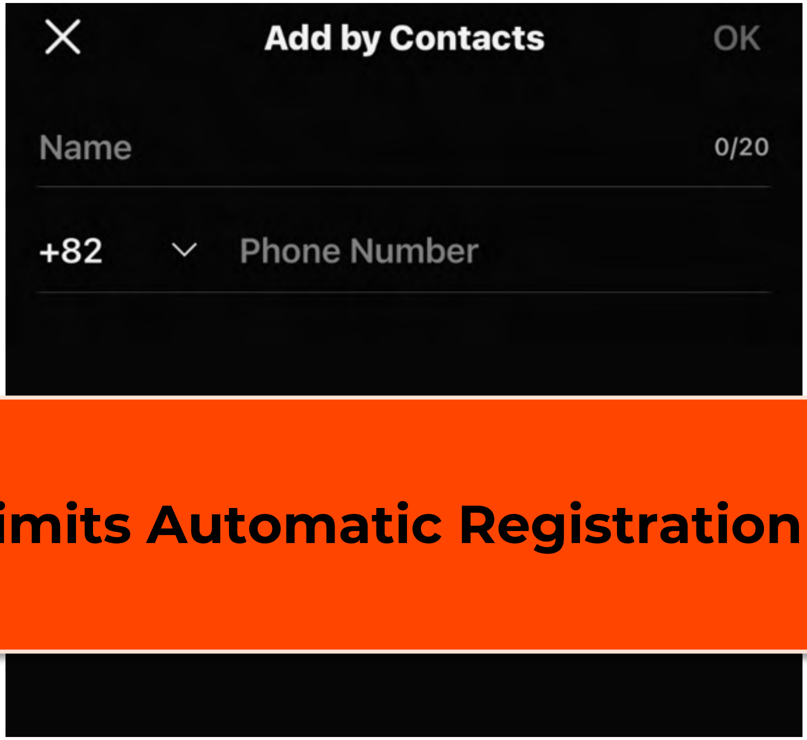
1. Manual Registration



2. Address-book syncing

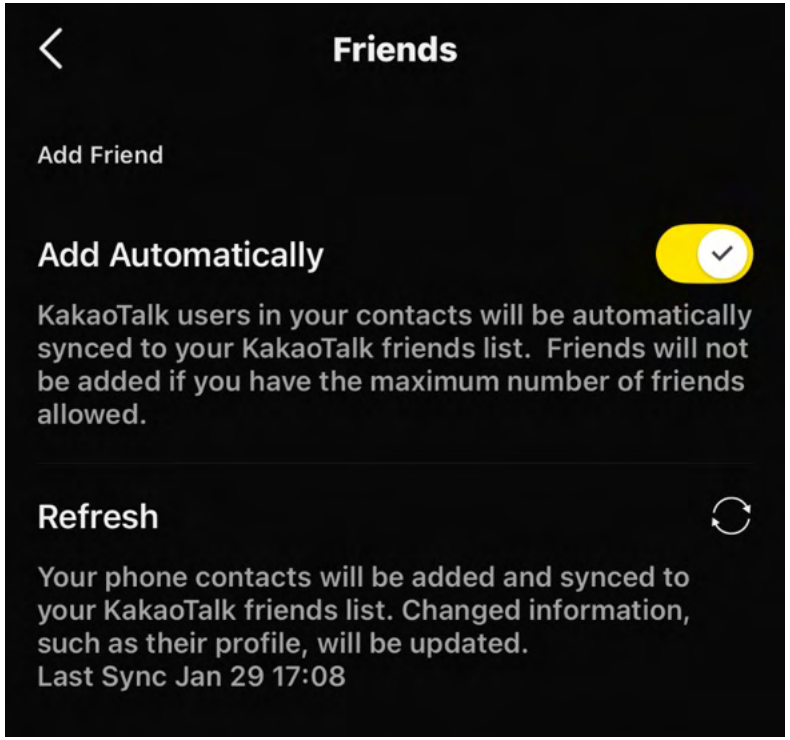
Attack 1: Contact discovery abuse (KakaoTalk)

- Friend Registration



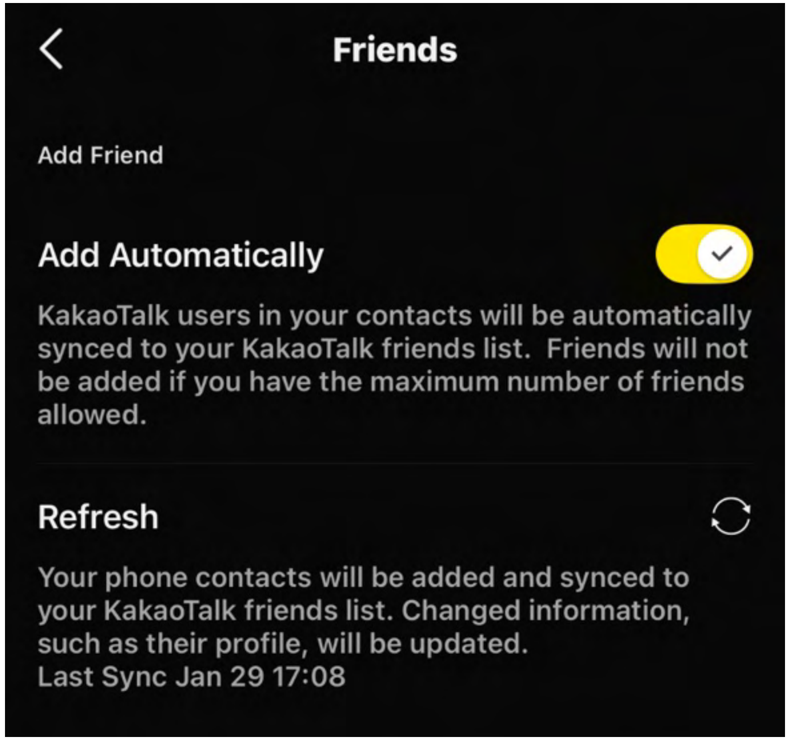
Limits Automatic Registration

1. Manual Registration



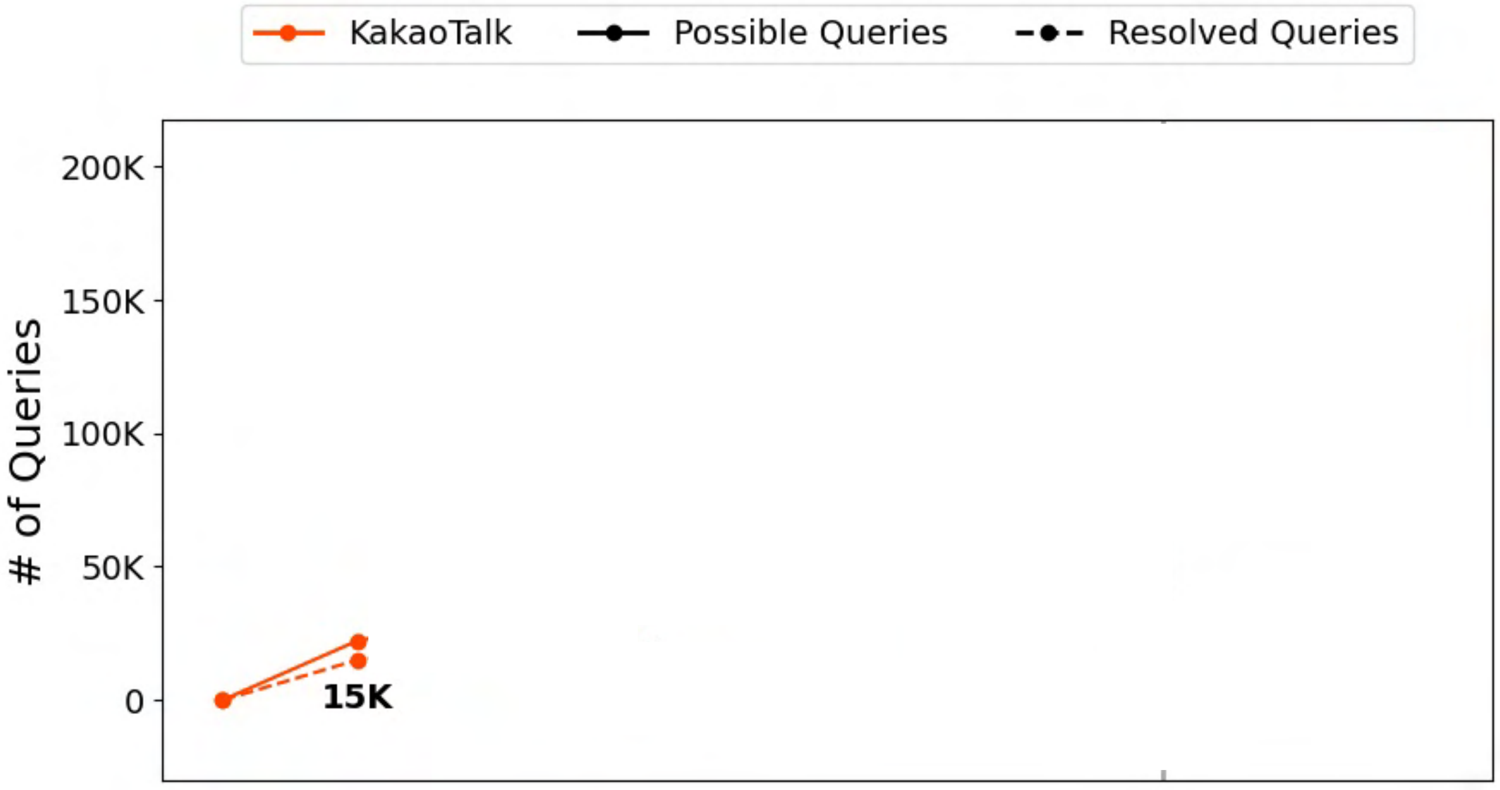
2. Address-book syncing

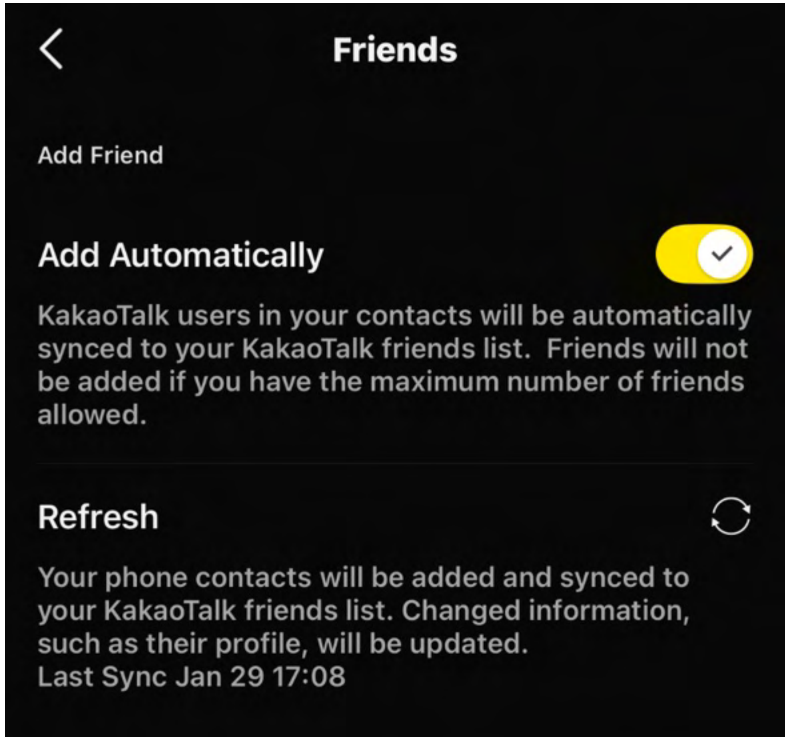
Attack 1: Contact discovery abuse (KakaoTalk)



1. Address-book syncing

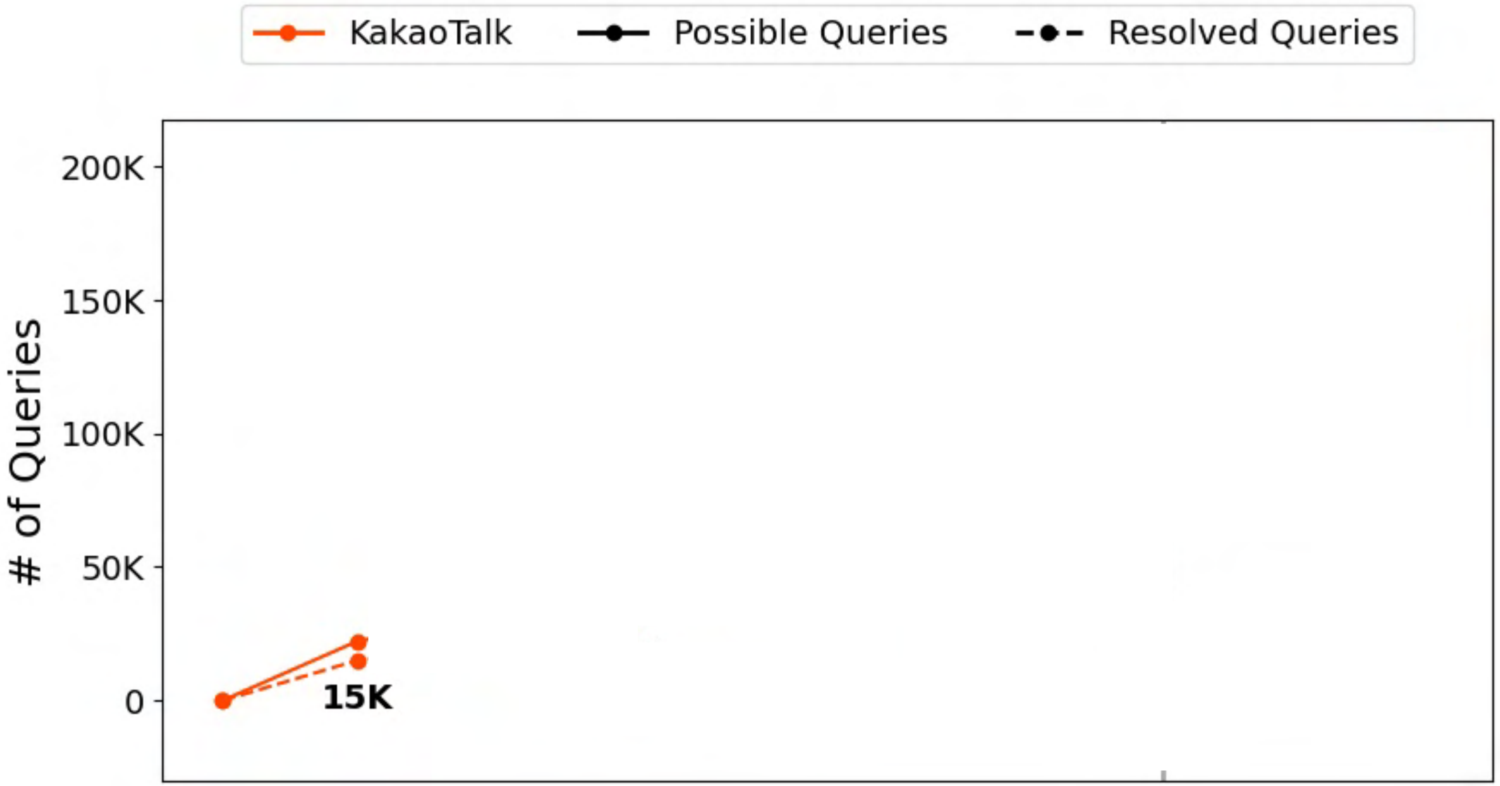
Attack 1: Contact discovery abuse (KakaoTalk)

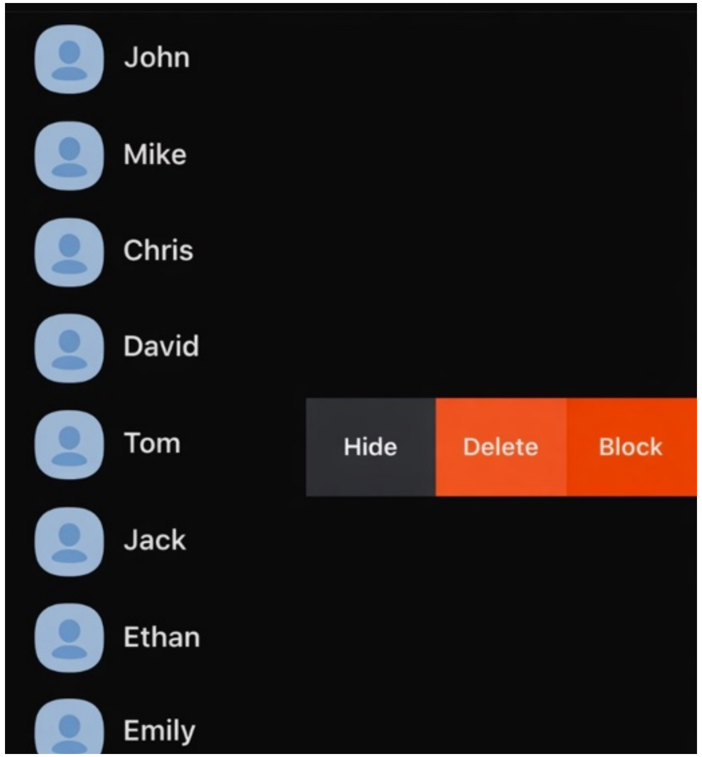




1. Address-book syncing

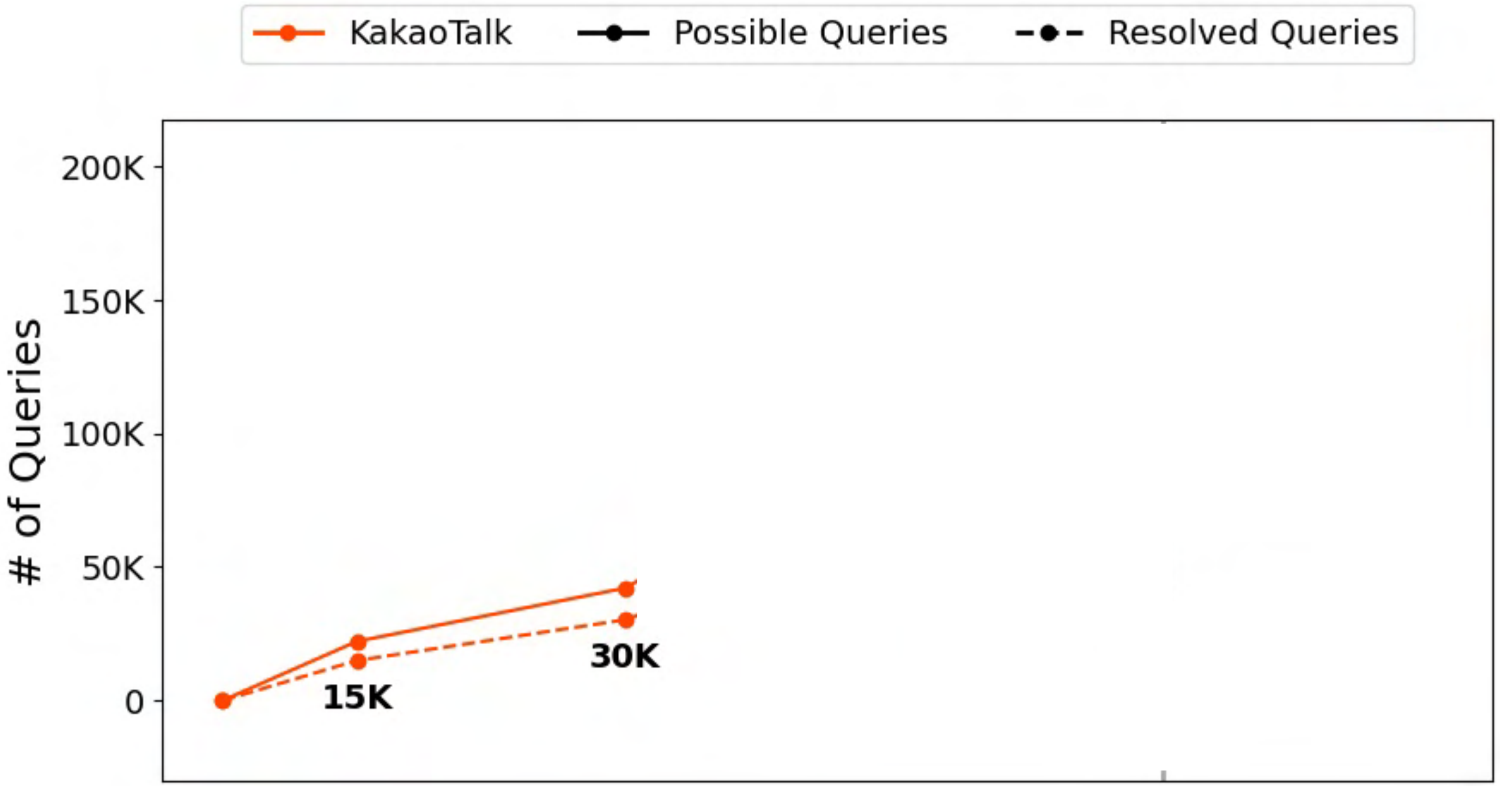
Attack 1: Contact discovery abuse (KakaoTalk)

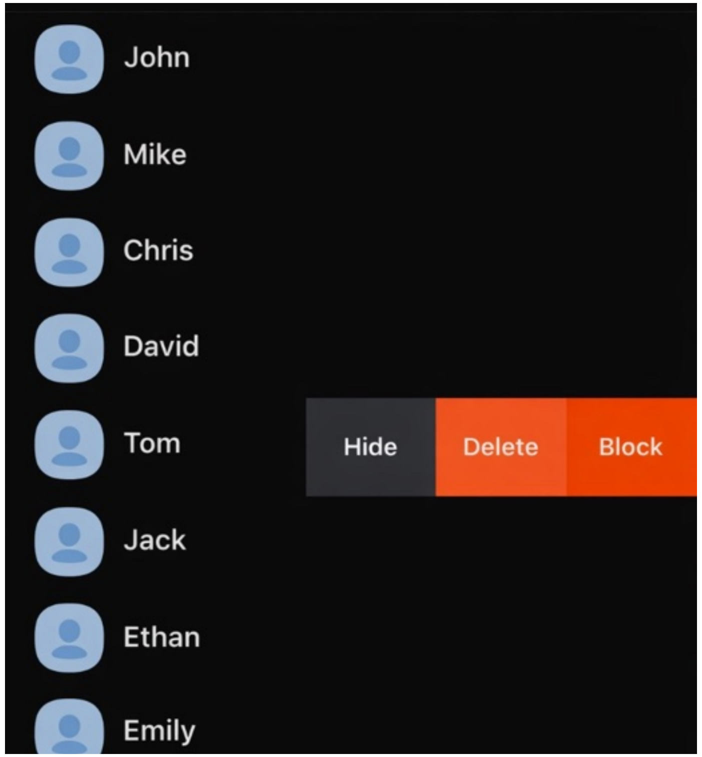




2. Deleting Friends

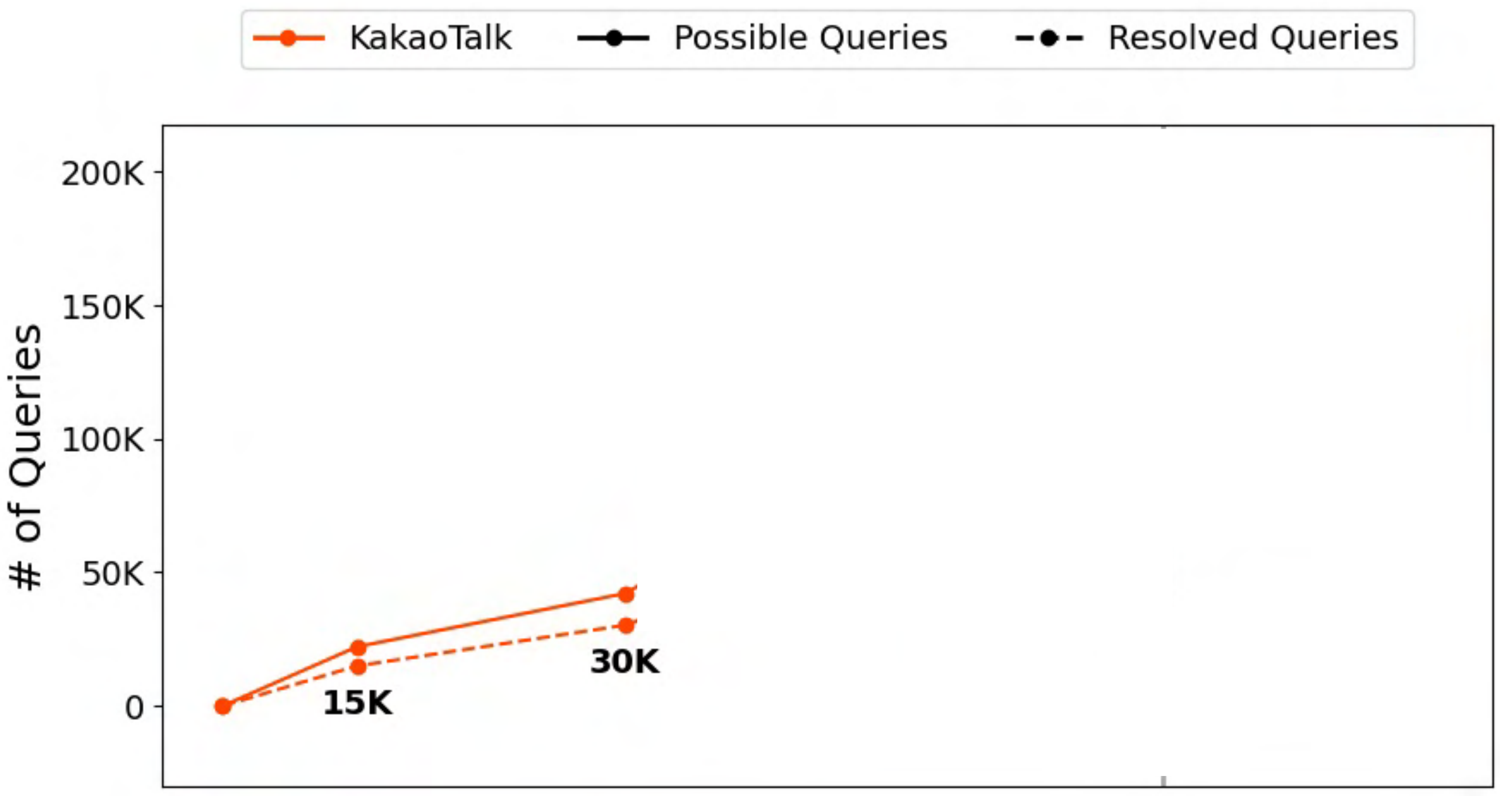
Attack 1: Contact discovery abuse (KakaoTalk)

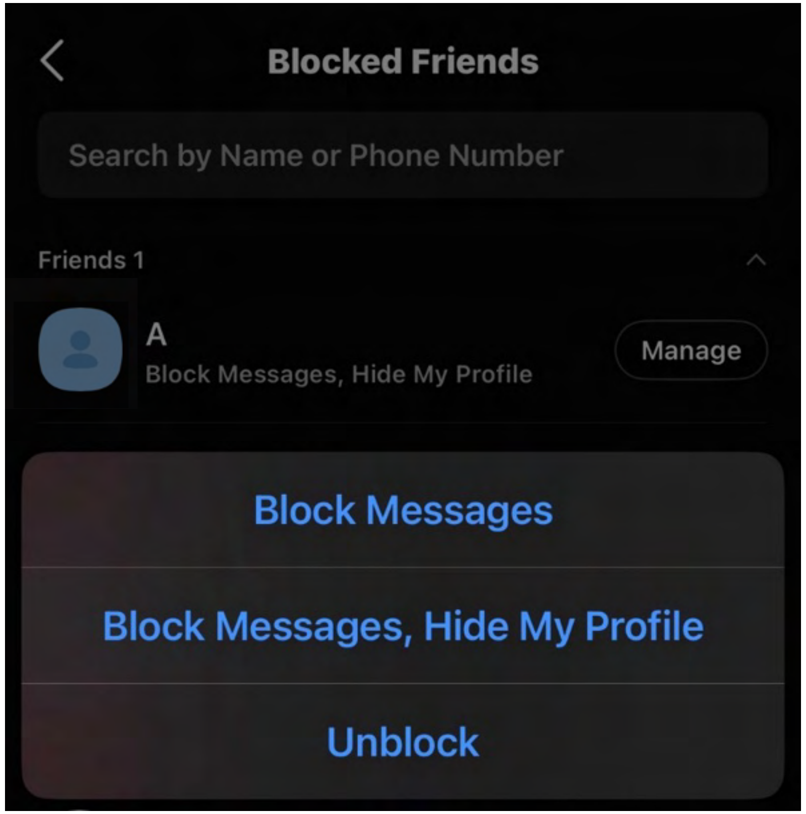




2. Deleting Friends

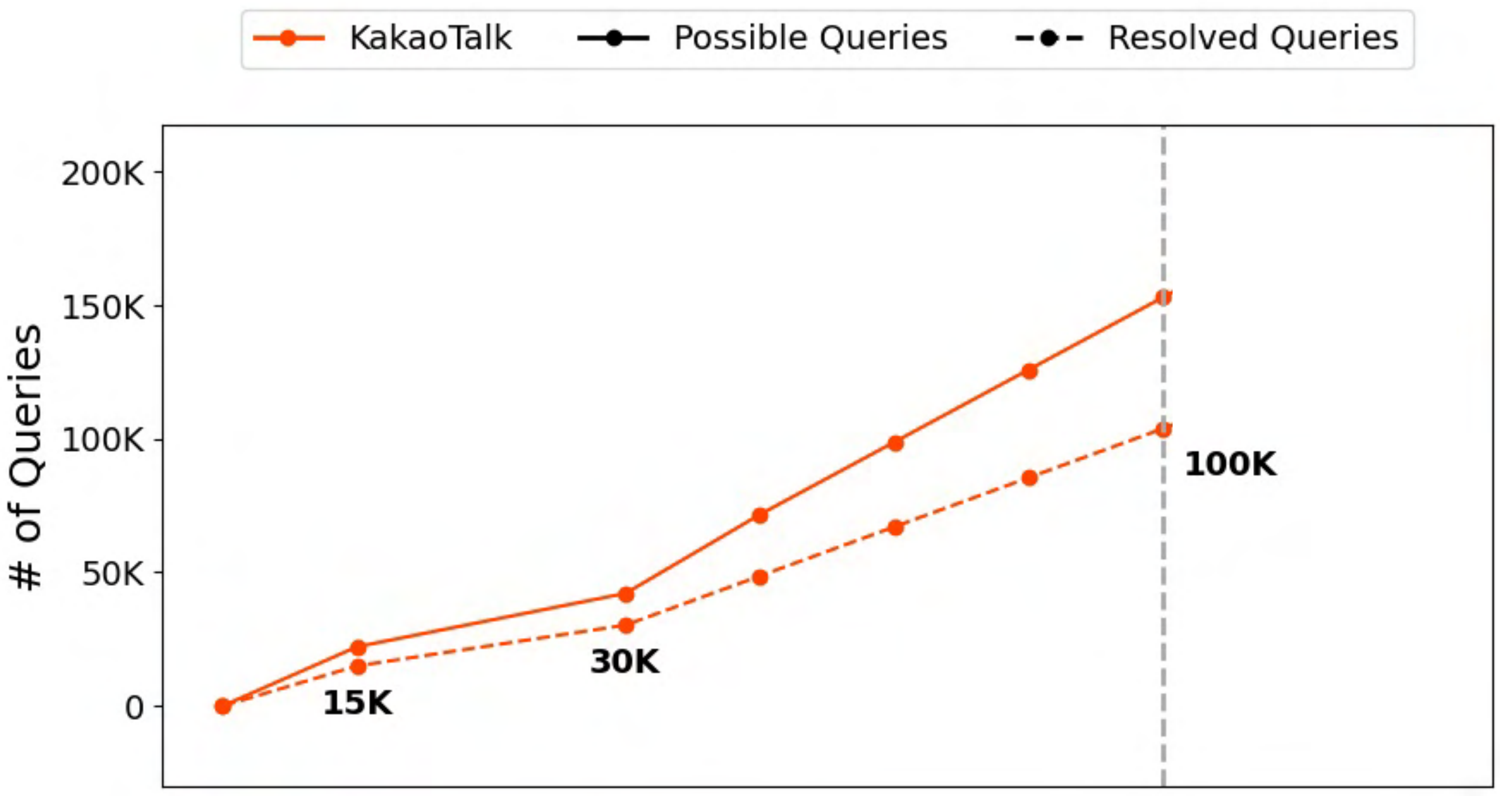
Attack 1: Contact discovery abuse (KakaoTalk)

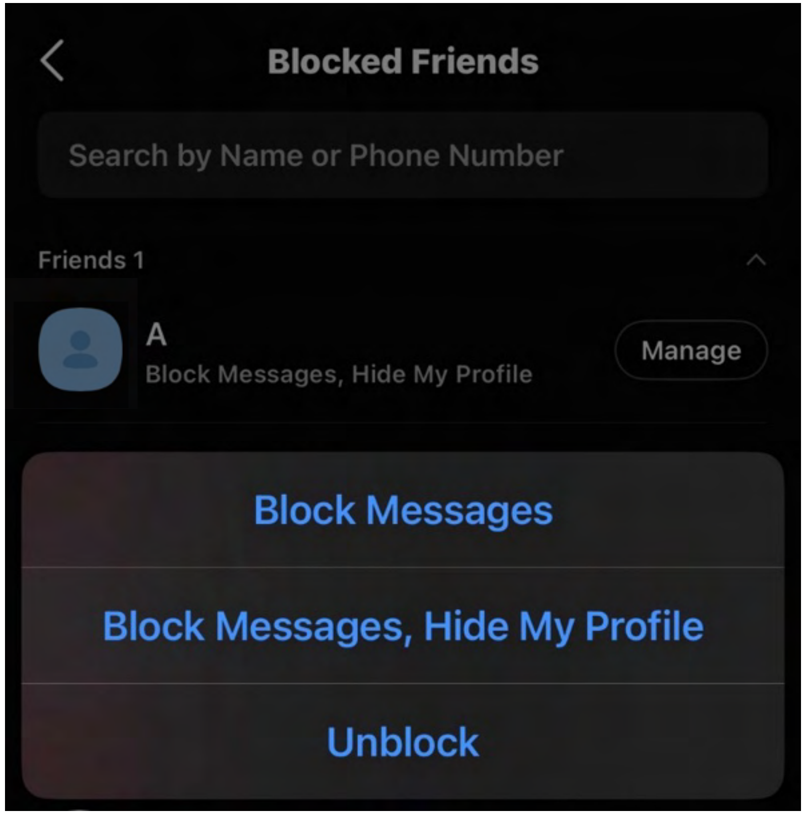




3. Block & Unblock Friends

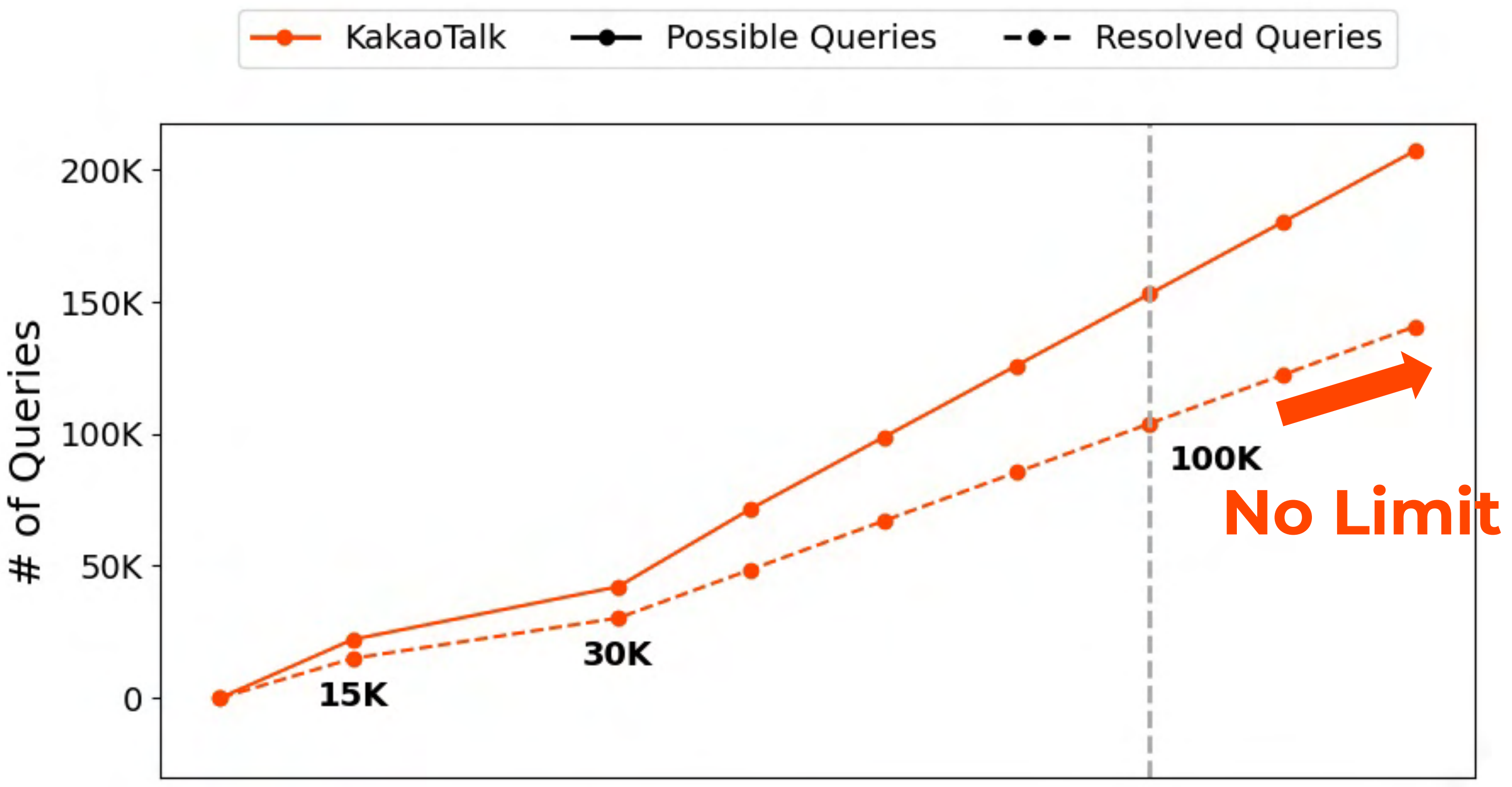
Attack 1: Contact discovery abuse (KakaoTalk)

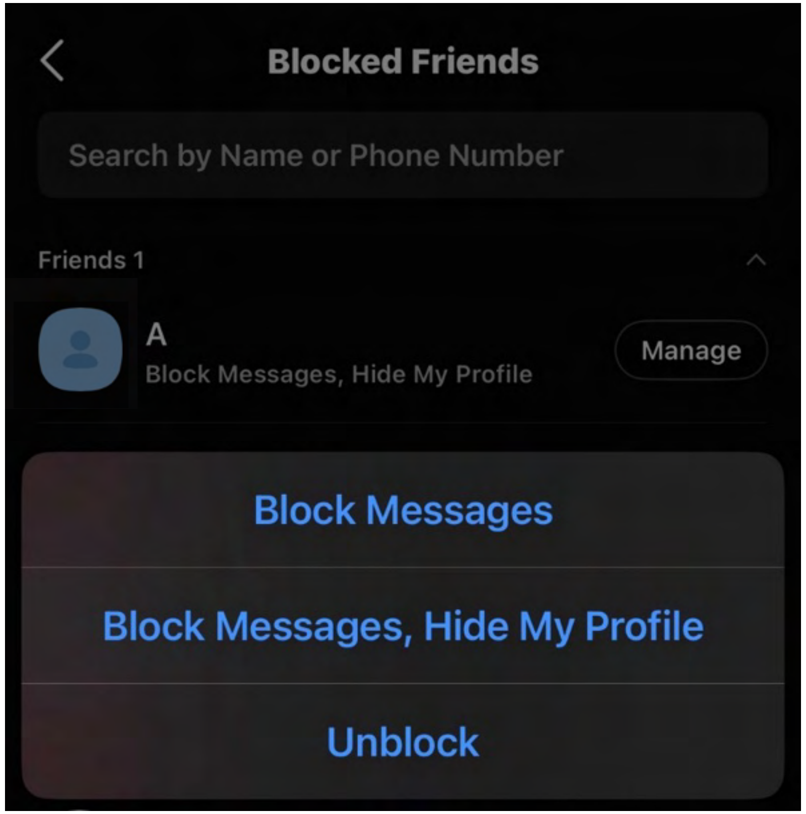




3. Block & Unblock Friends

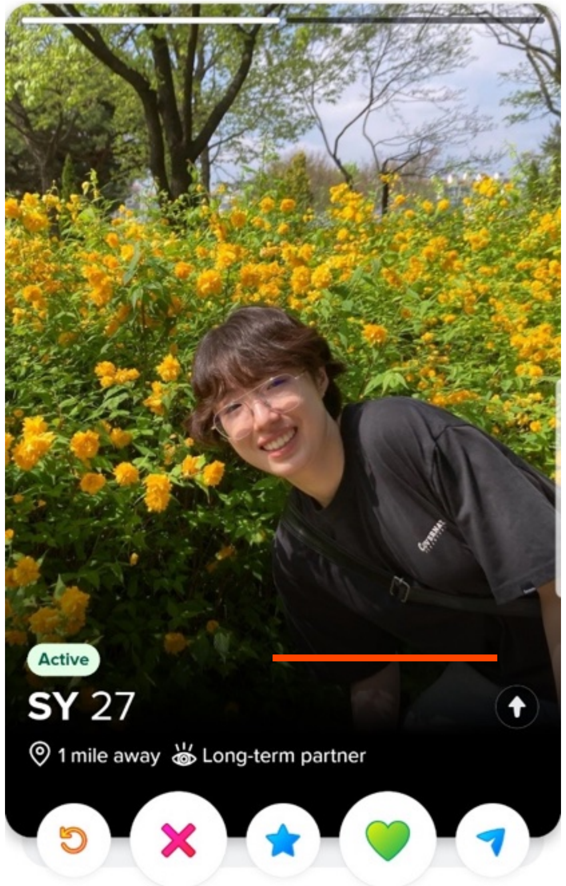
Attack 1: Contact discovery abuse (KakaoTalk)





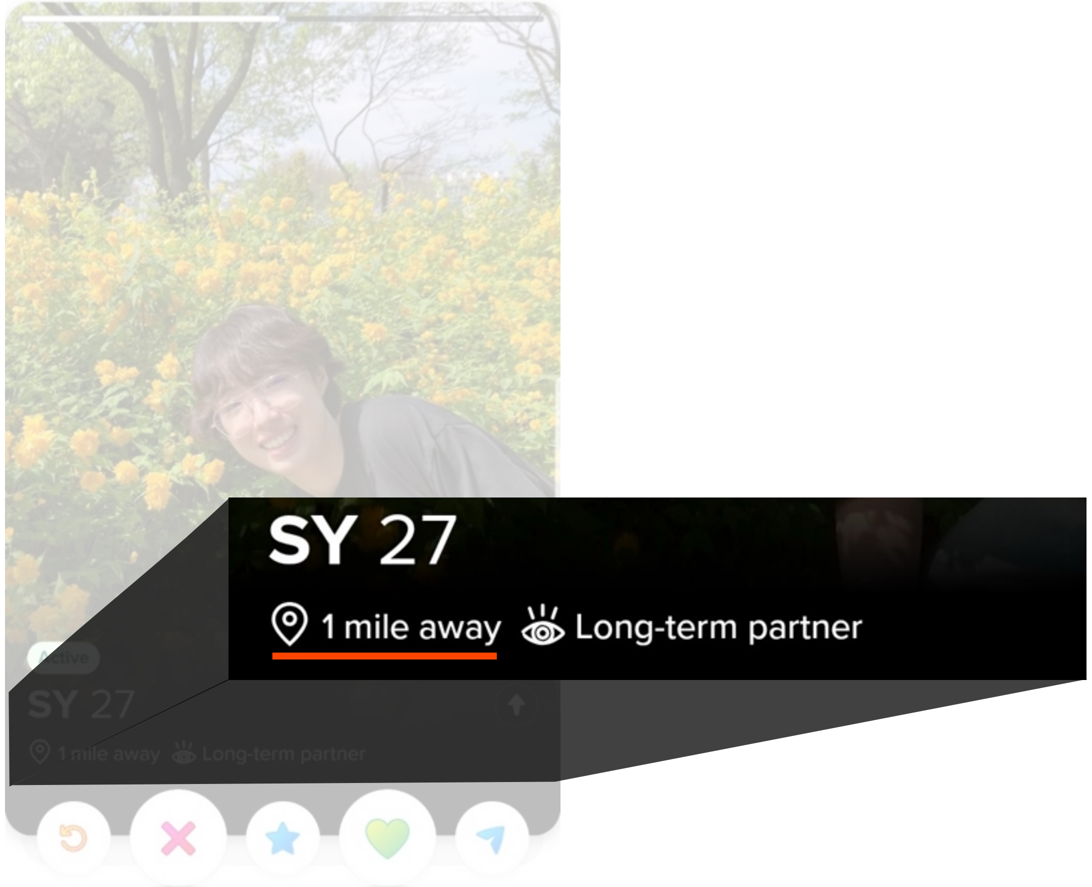
3. Block & Unblock Friends

Attack 3: Efficient location inference from Tinder “nearby” signals



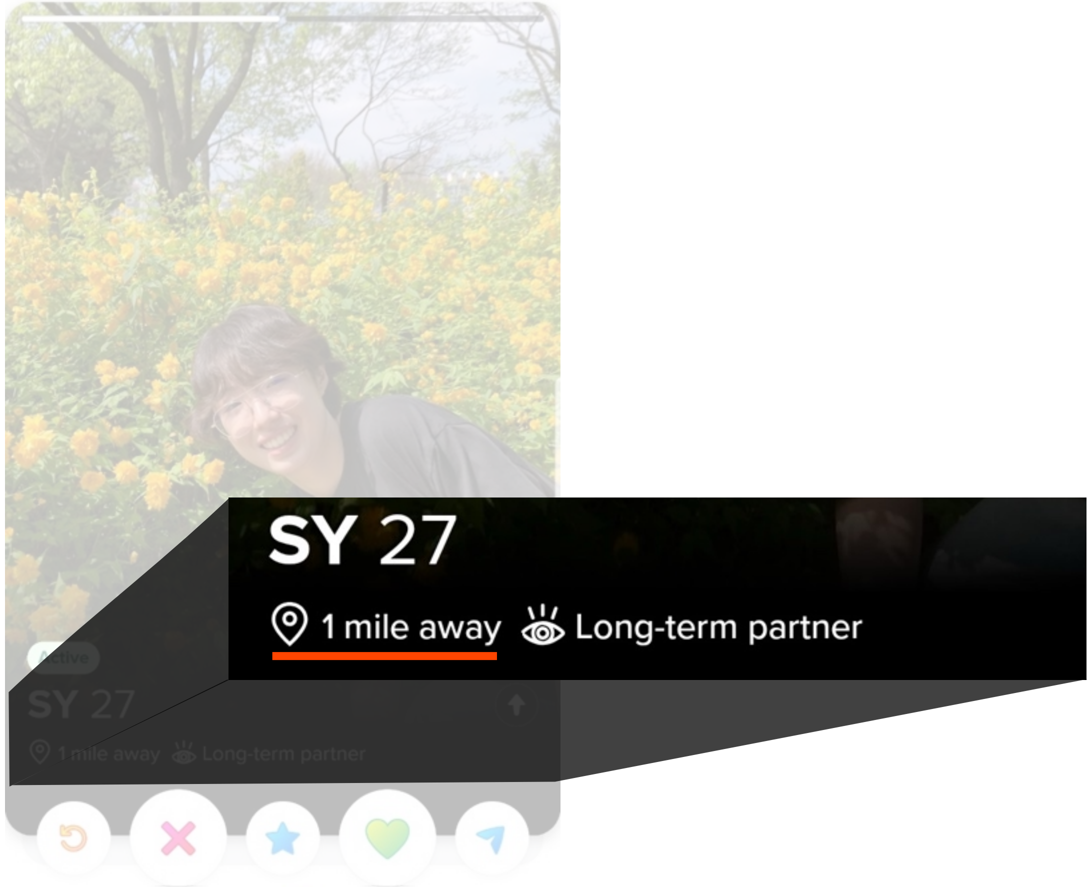
Tinder profile card

Attack 3: Efficient location inference from Tinder “nearby” signals

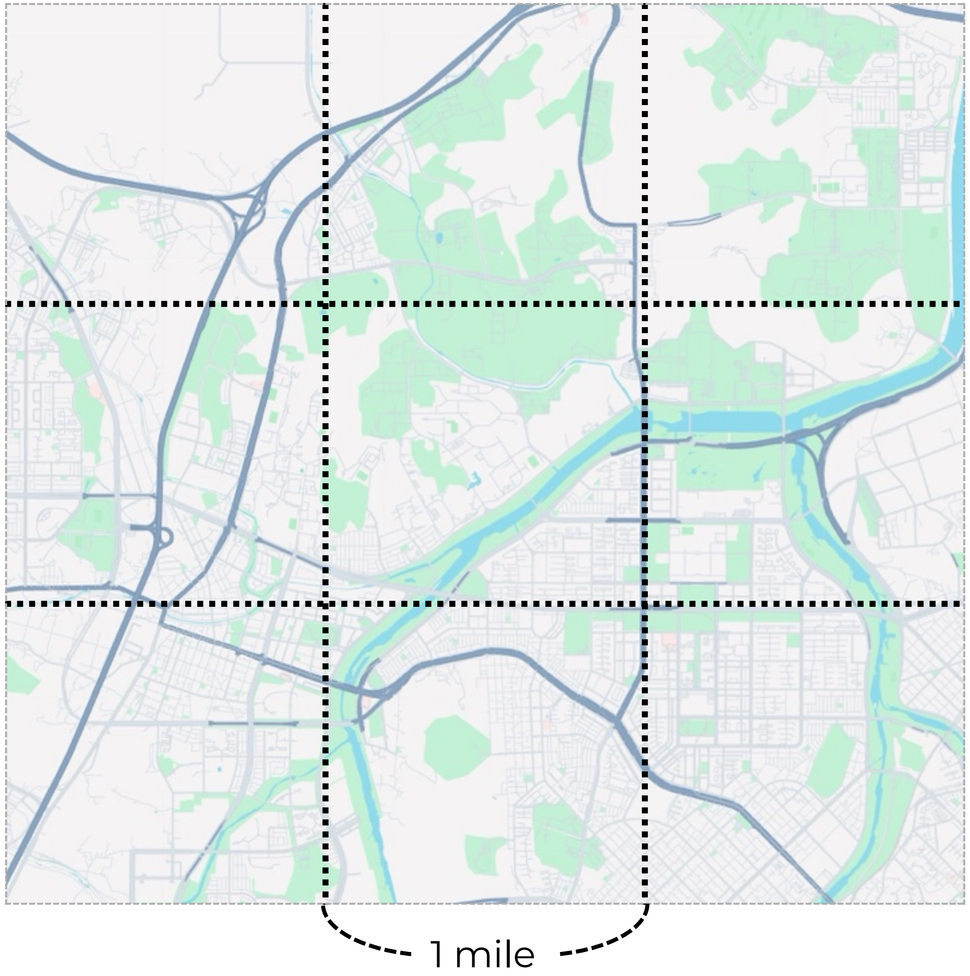


Tinder profile card

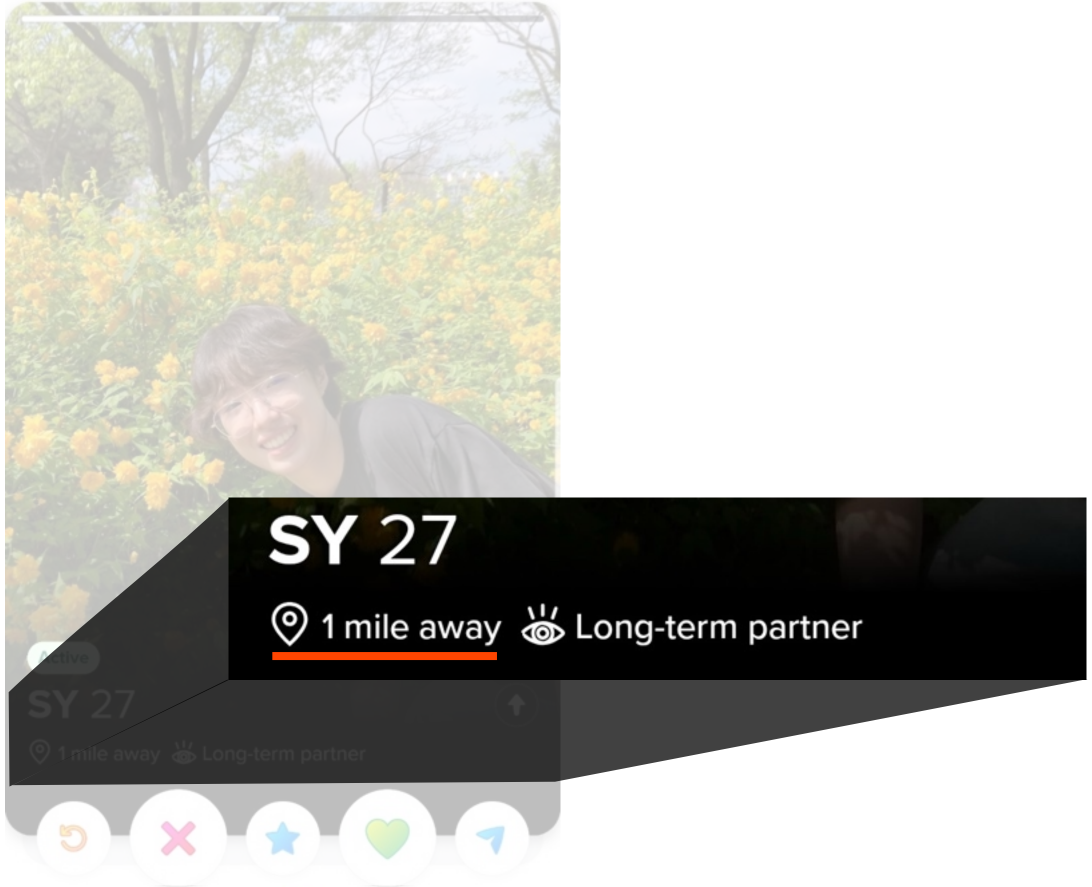
Attack 3: Efficient location inference from Tinder "nearby" signals



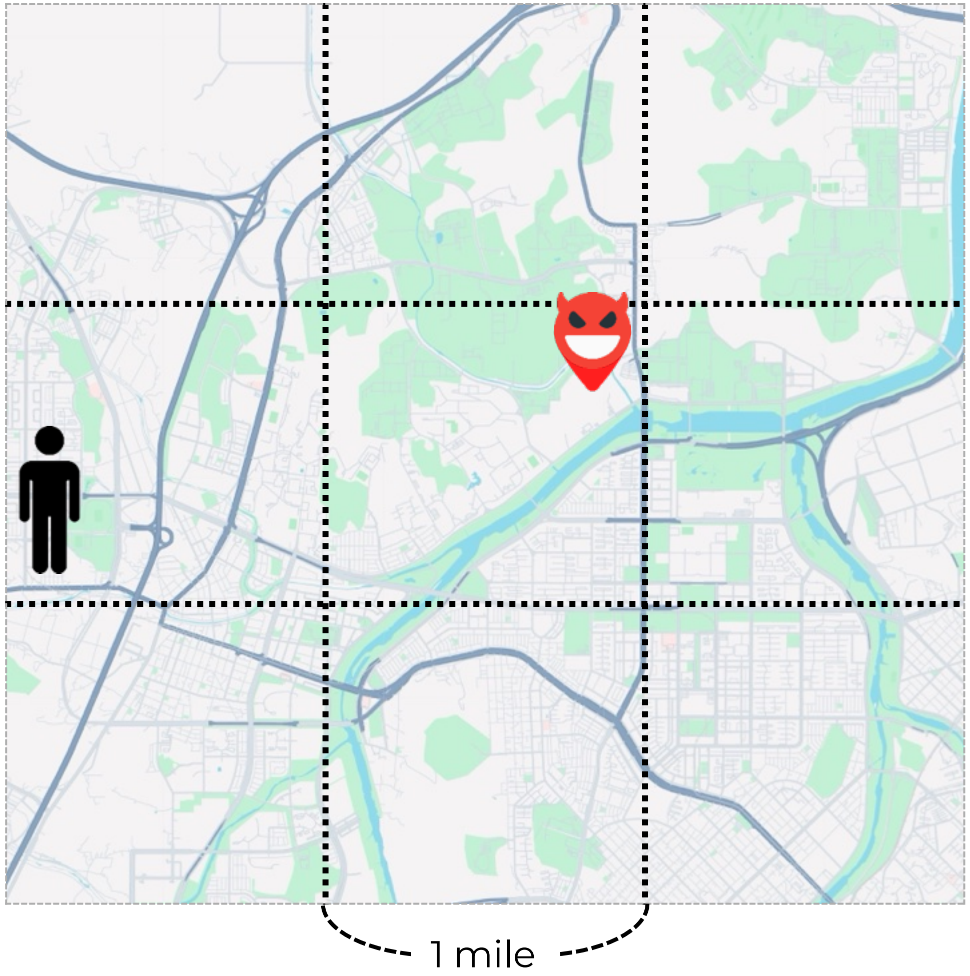
Tinder profile card



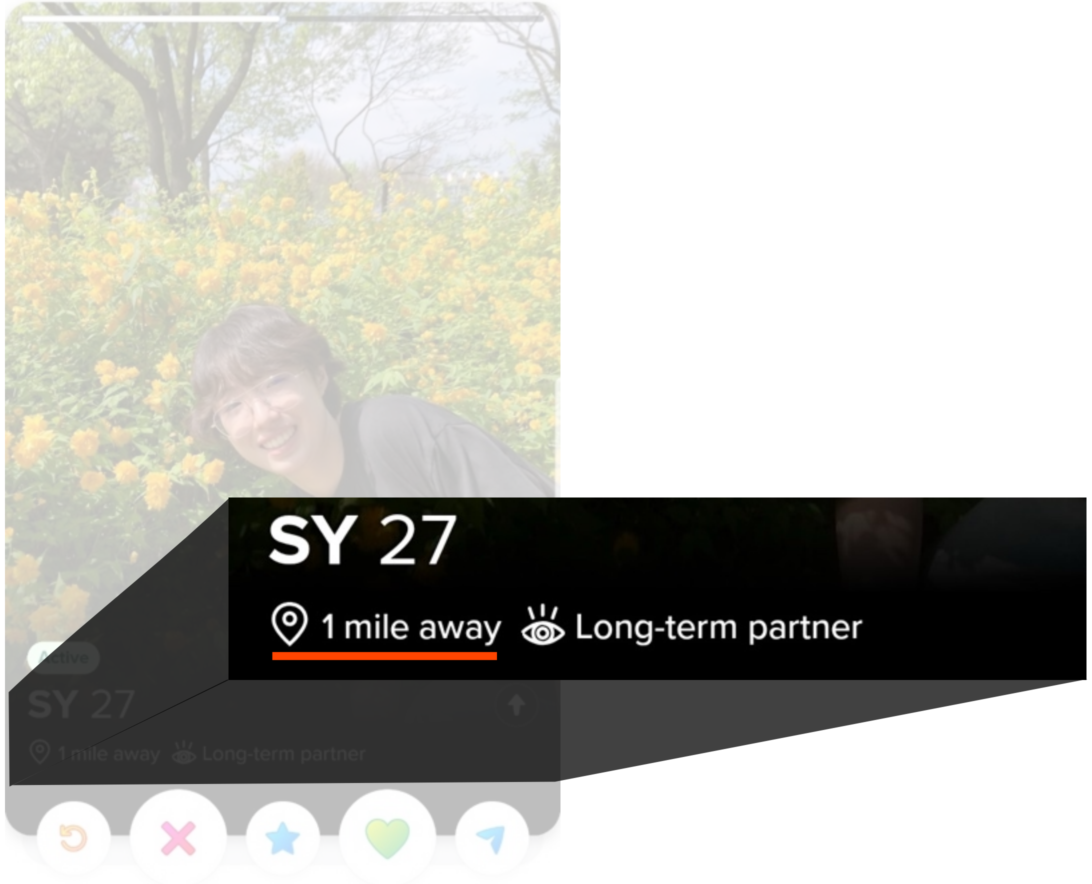
Attack 3: Efficient location inference from Tinder “nearby” signals



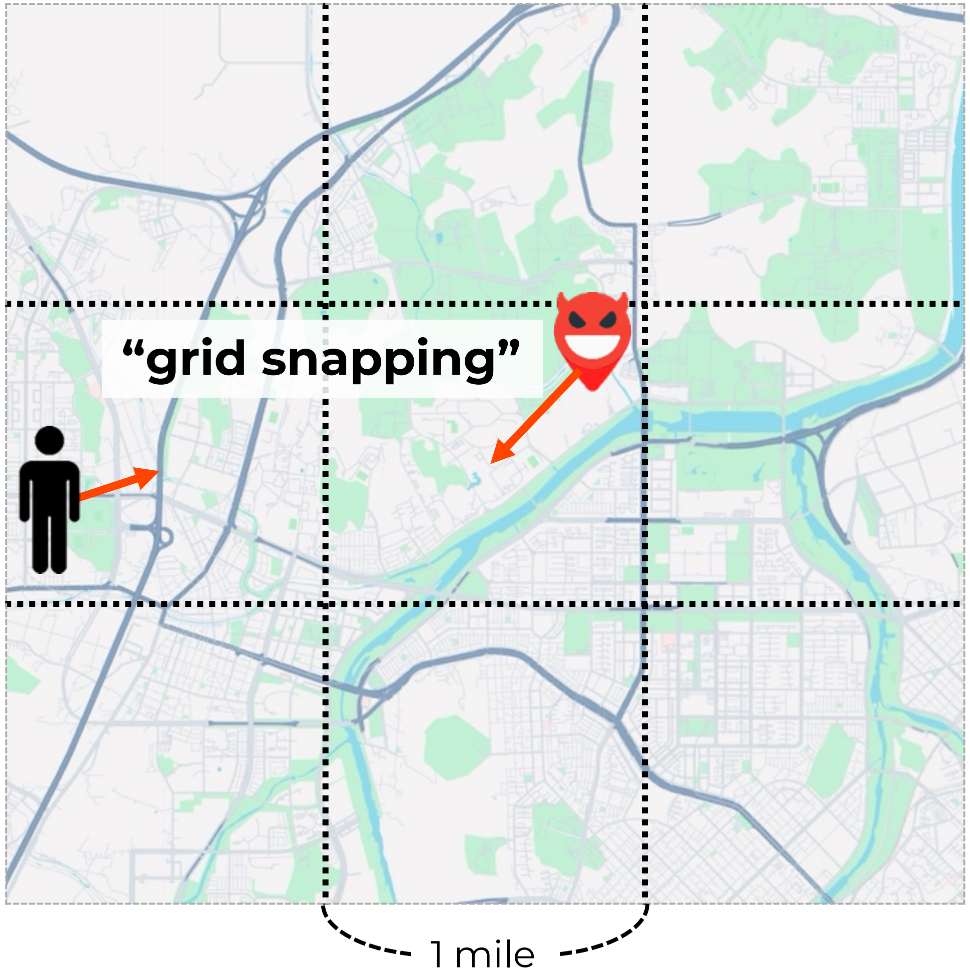
Tinder profile card



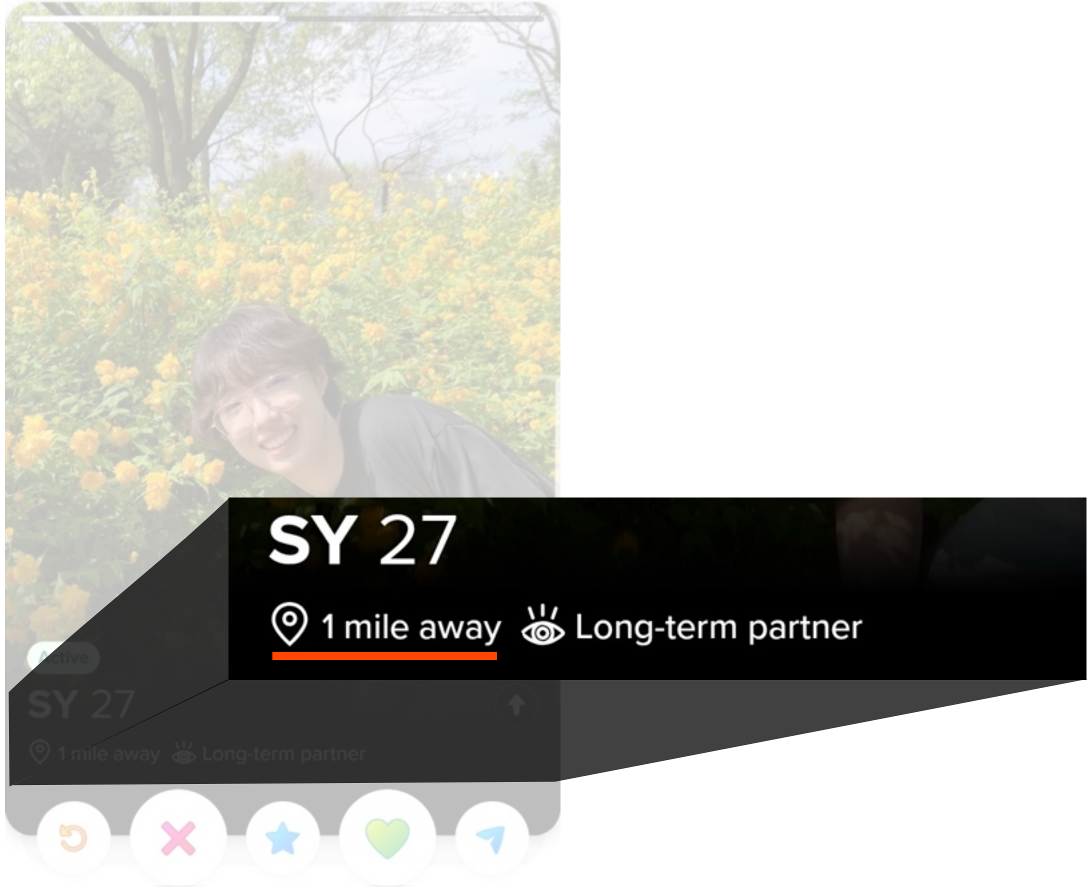
Attack 3: Efficient location inference from Tinder “nearby” signals



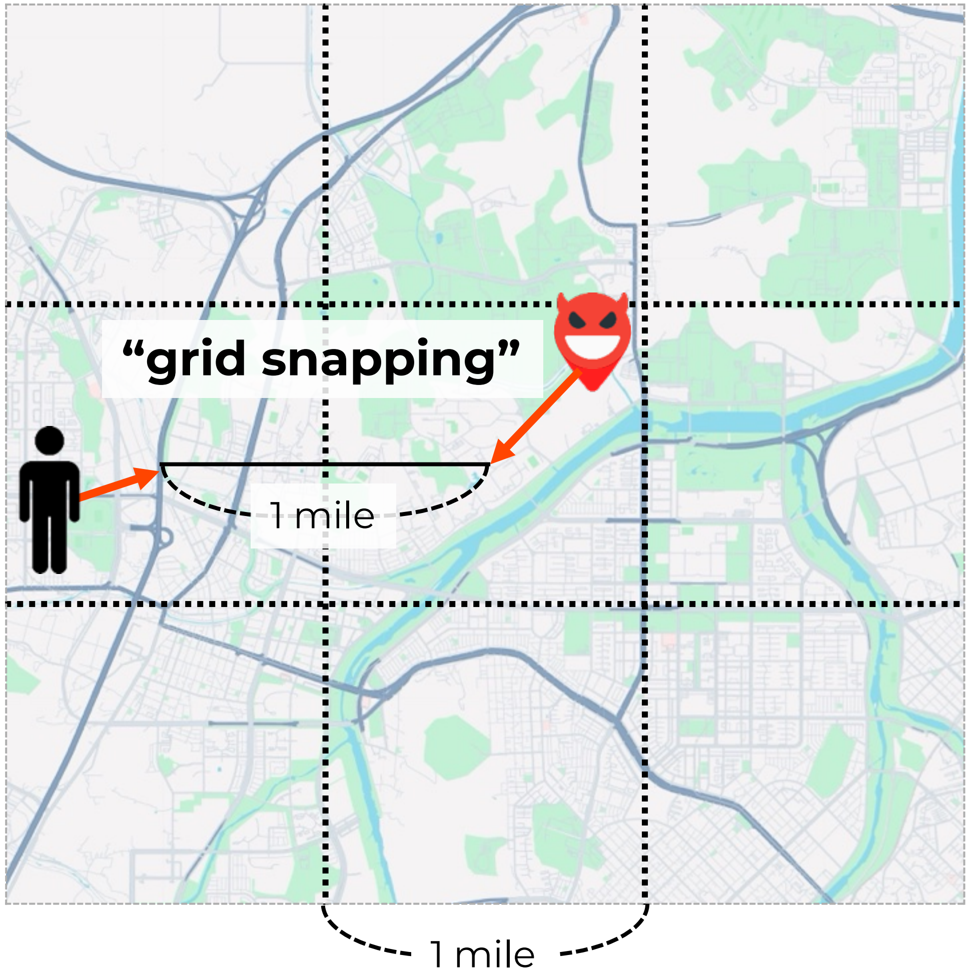
Tinder profile card



Attack 3: Efficient location inference from Tinder "nearby" signals

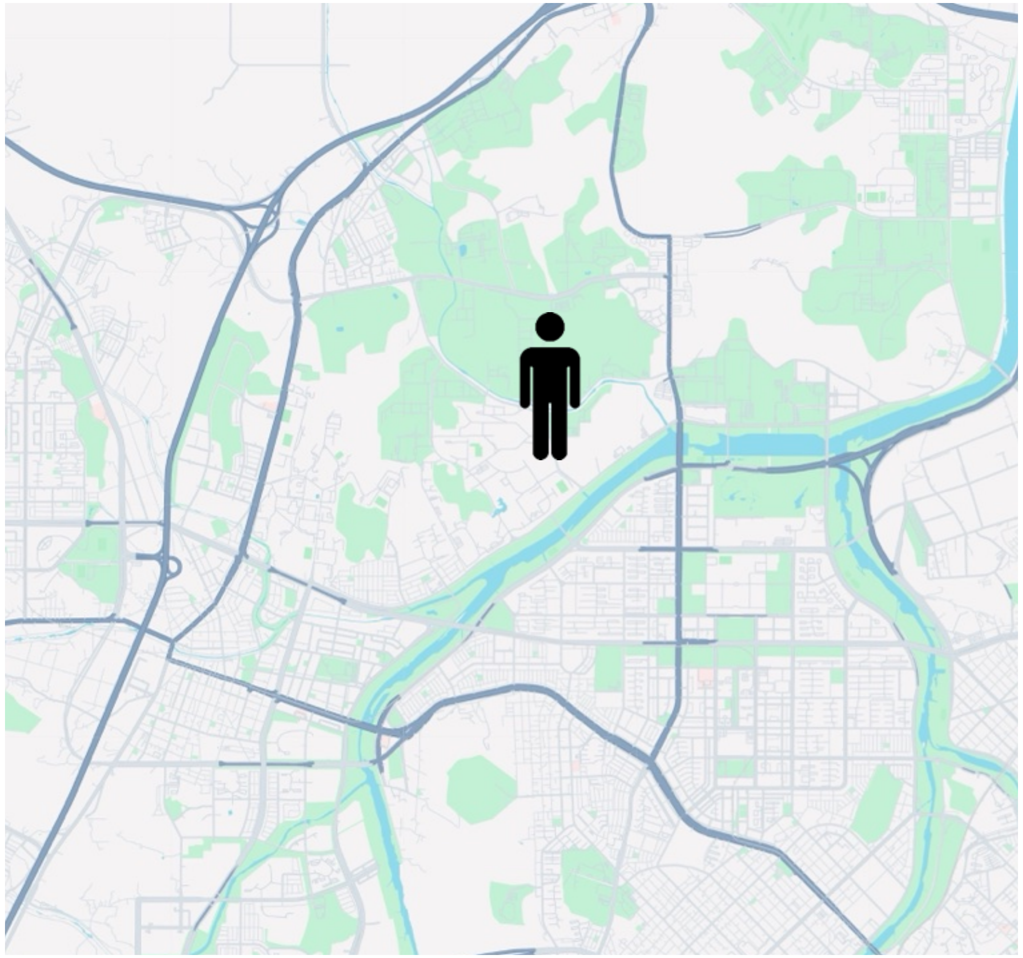


Tinder profile card



Attack 3: Efficient location inference from Tinder “nearby” signals

1. Create 2 miles x 2miles grid



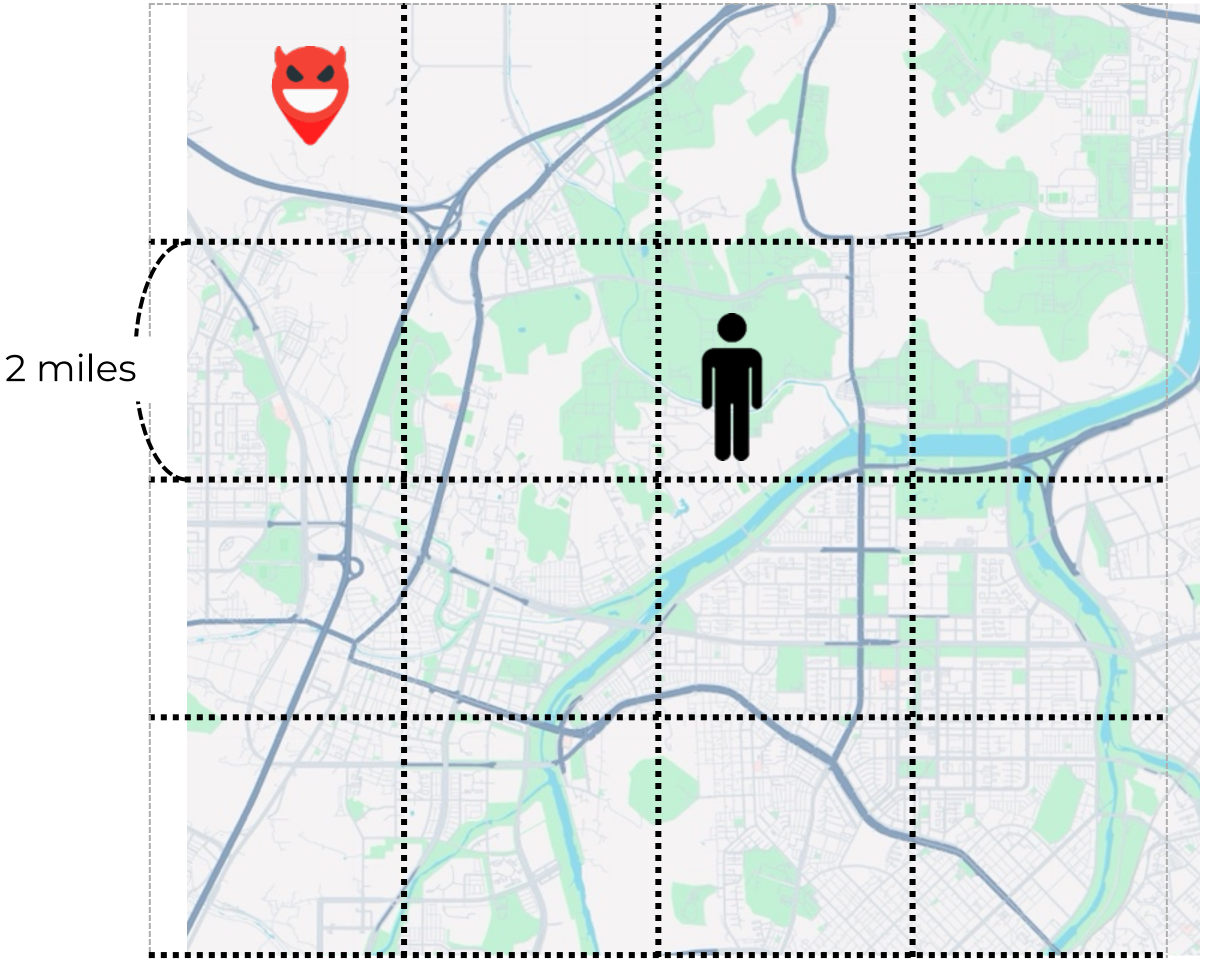
Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells



Attack 3: Efficient location inference from Tinder “nearby” signals

1. Create 2 miles x 2miles grid

2. Search grid cells

Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells

Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells

Attack 3: Efficient location inference from Tinder “nearby” signals

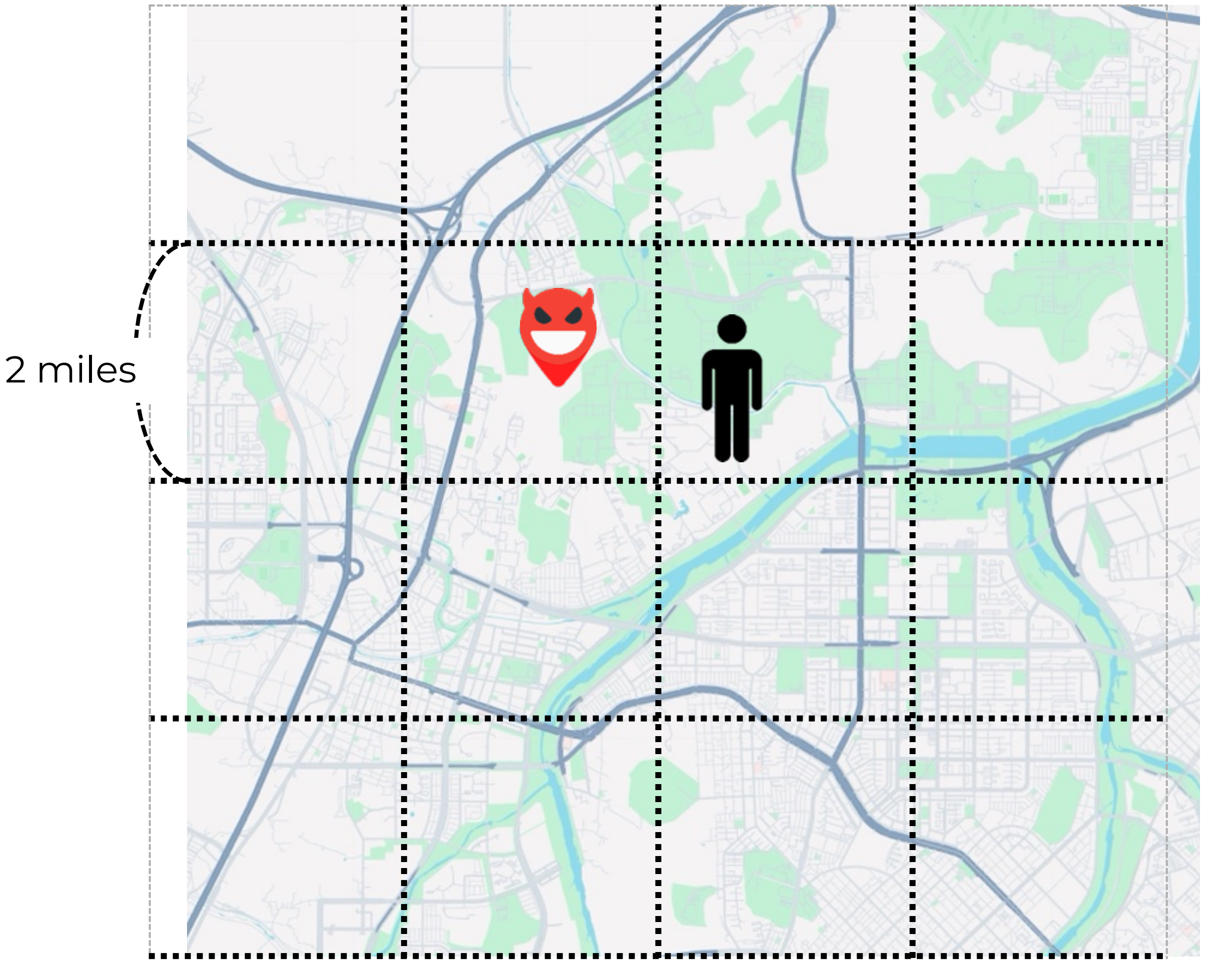
1. Create 2 miles x 2miles grid

2. Search grid cells

Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells



Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

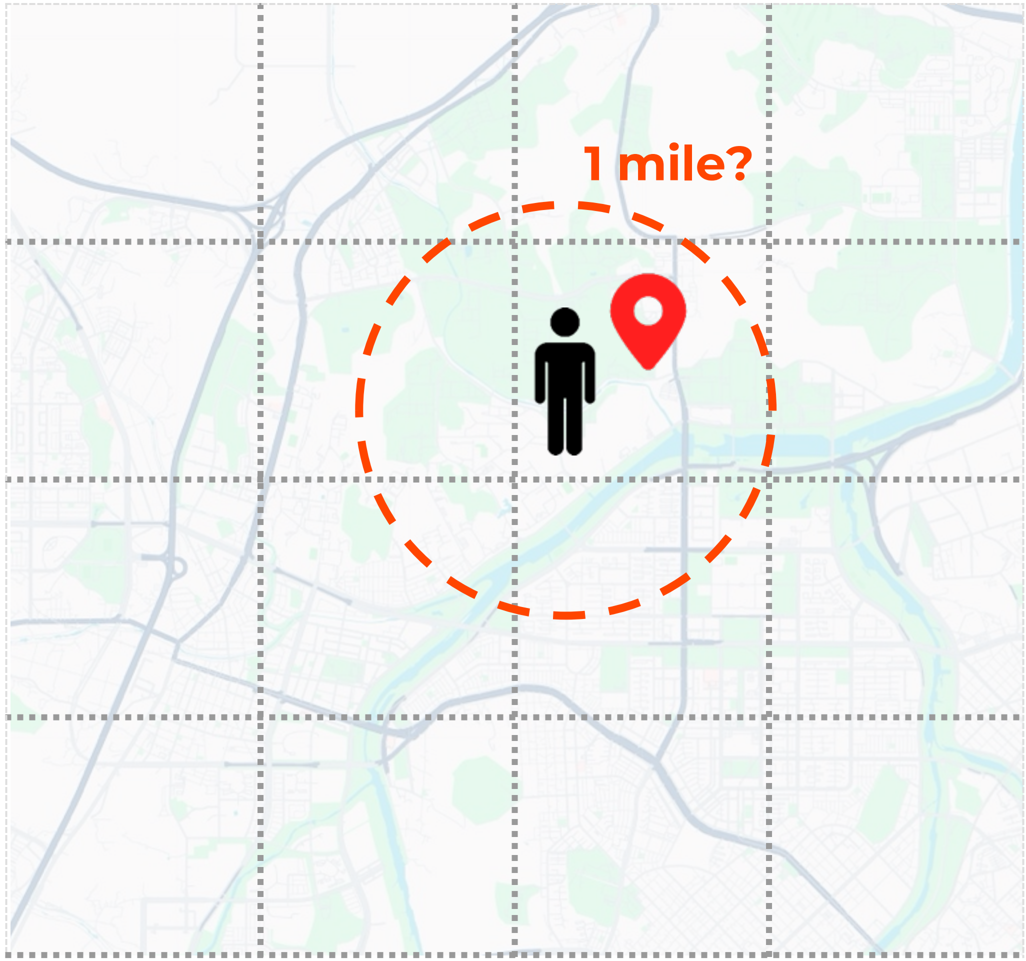
2. Search grid cells

Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points (East, West, North, South)
   - Binary Search
   - Stop when *high – low* < $epsilon$ *(9 meters)*



Attack 3: Efficient location inference from Tinder “nearby” signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points (East, West, North, South)
   - Binary Search
   - Stop when *high* – *low* < $epsilon$ *(9 meters)*

Attack 3: Efficient location inference from Tinder “nearby” signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points (East, West, North, South)
   - Binary Search
   - Stop when *high* – *low* < $epsilon$ *(9 meters)*

Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points (East, West, North, South)
    - Binary Search
    - Stop when *high – low* < $epsilon$ *(9 meters)*

Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points
(East, West, North, South)

- Binary Search
- Stop when *high* – *low* < $epsilon$ *(9 meters)*

Attack 3: Efficient location inference from Tinder “nearby” signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points (East, West, North, South)
   - Binary Search
   - Stop when *high* – *low* < $epsilon$ *(9 meters)*

Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points
(East, West, North, South)
   - Binary Search
   - Stop when *high* – *low* < $epsilon$ *(9 meters)*

Attack 3: Efficient location inference from Tinder “nearby” signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points (East, West, North, South)
   - Binary Search
   - Stop when *high* – *low* < $epsilon$ *(9 meters)*

Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells

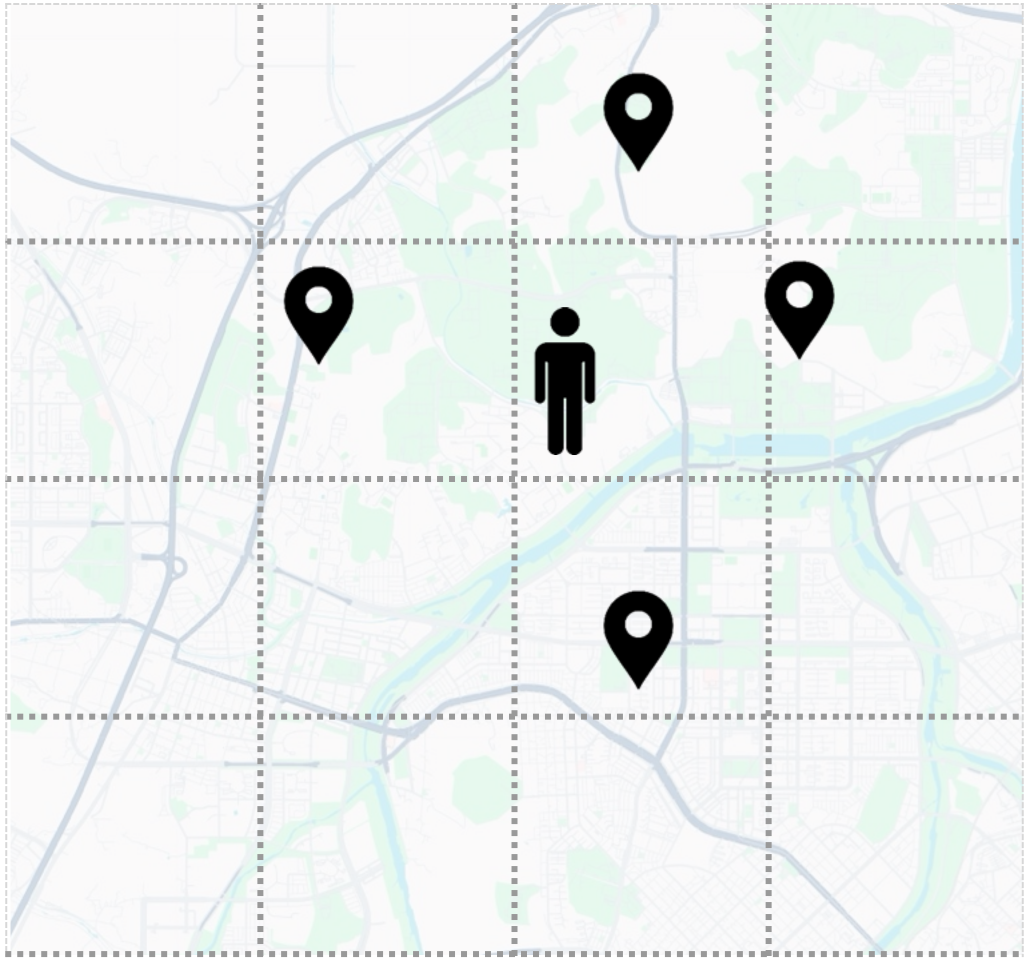
3. Find 1-mile boundary points (East, West, North, South)
    - Binary Search
    - Stop when *high* – *low* < $epsilon$ *(9 meters)*

Attack 3: Efficient location inference from Tinder “nearby” signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points
(East, West, North, South)
   - Binary Search
   - Stop when *high – low* < $epsilon$ *(9 meters)*



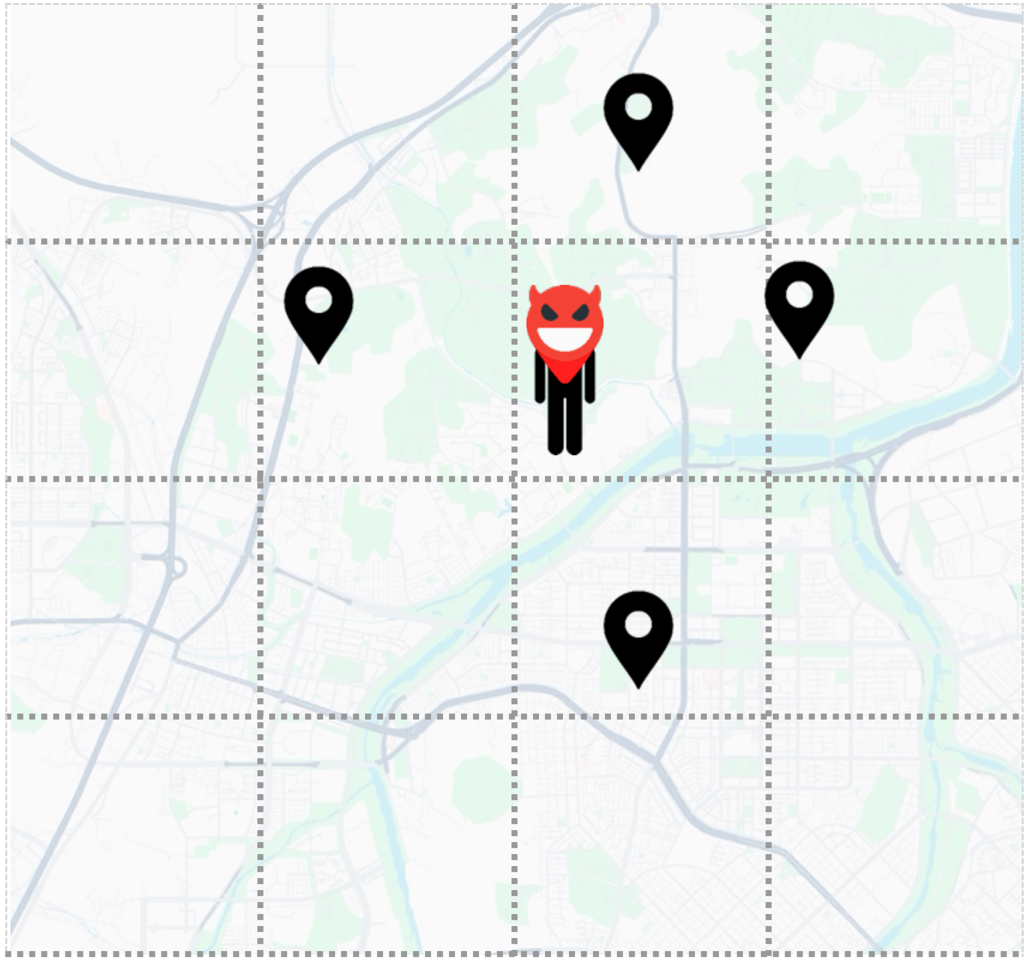
Attack 3: Efficient location inference from Tinder "nearby" signals

1. Create 2 miles x 2miles grid

2. Search grid cells

3. Find 1-mile boundary points
(East, West, North, South)

- Binary Search
- Stop when *high* – *low* < $epsilon$ *(9 meters)*



Chain 3: Targeted trajectory tracking

- Tracking specific user

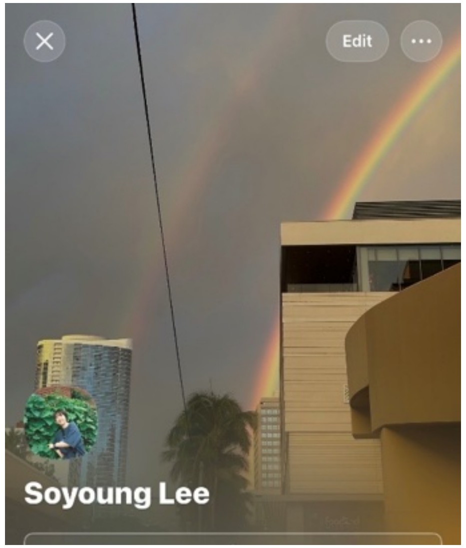
1. Access token exposure

Chain 3: Targeted trajectory tracking

- Tracking specific user

1. Access token exposure

TALK → Profile image

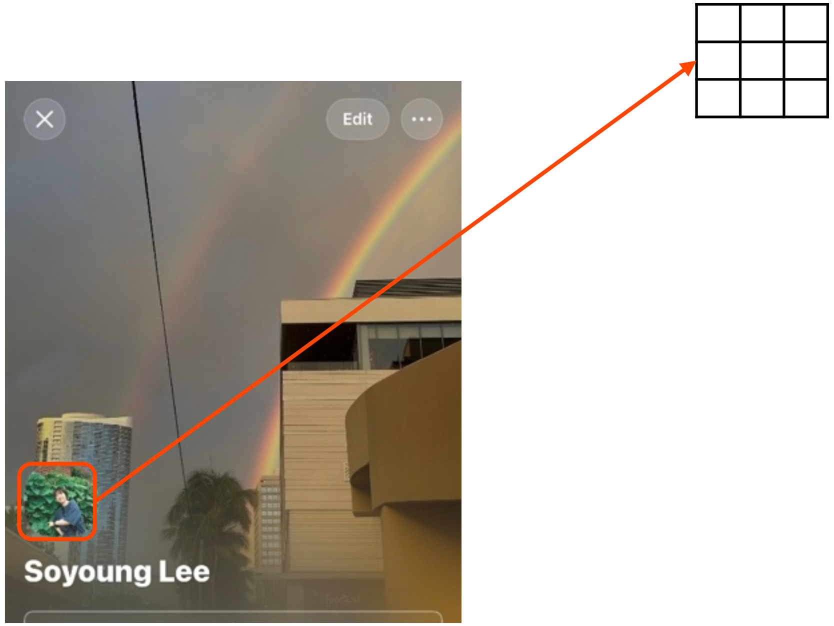


Chain 3: Targeted trajectory tracking

- Tracking specific user

1. Access token exposure

TALK → Profile image → Image Embeddings

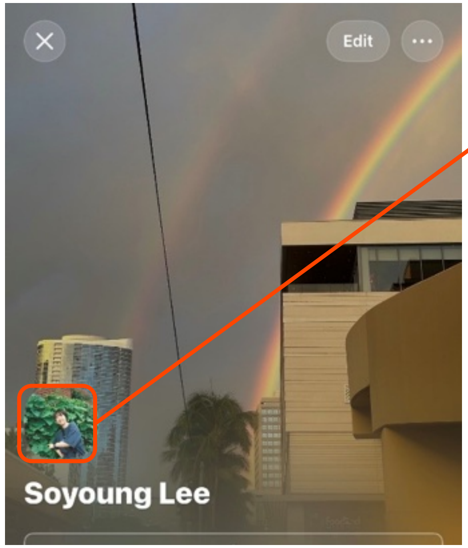


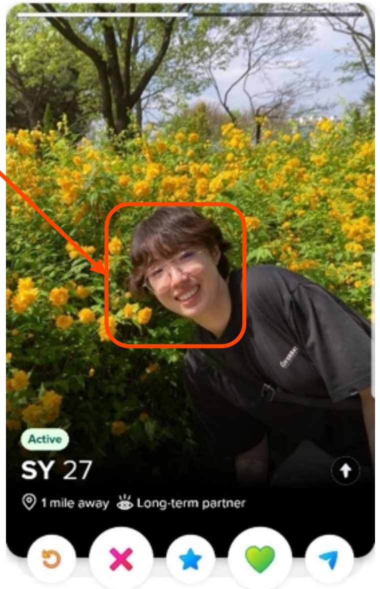
Chain 3: Targeted trajectory tracking

- Tracking specific user

1. Access token exposure

Profile image → Image Embeddings → Find target



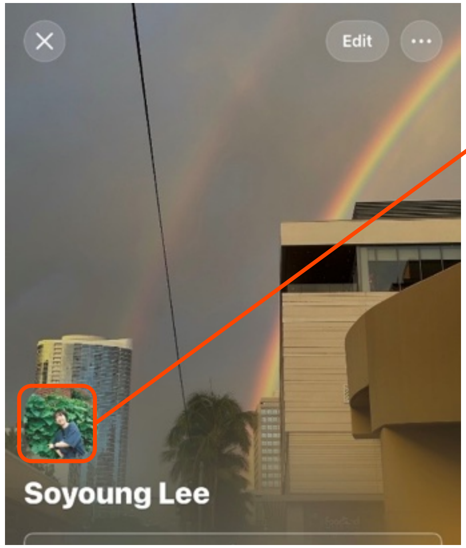


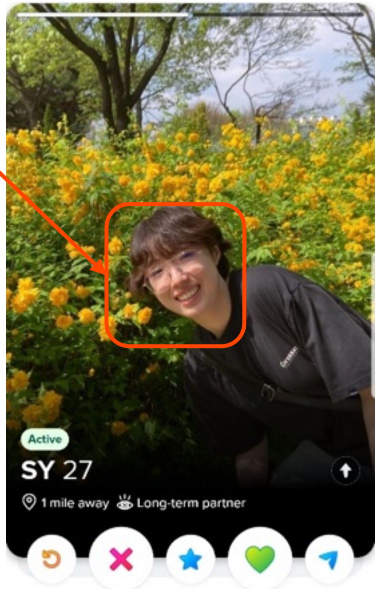
Chain 3: Targeted trajectory tracking

- Tracking specific user

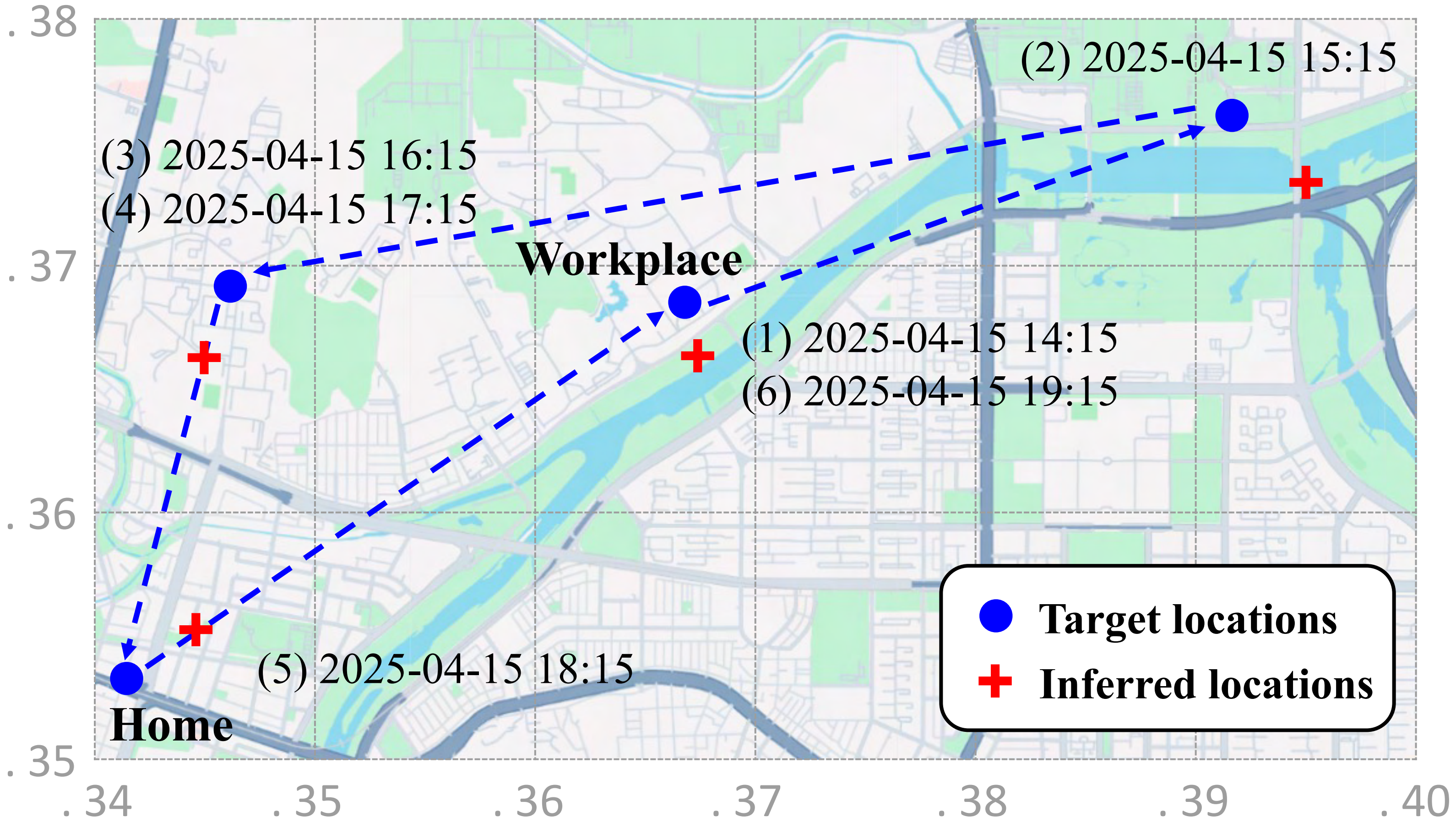
1. Access token exposure

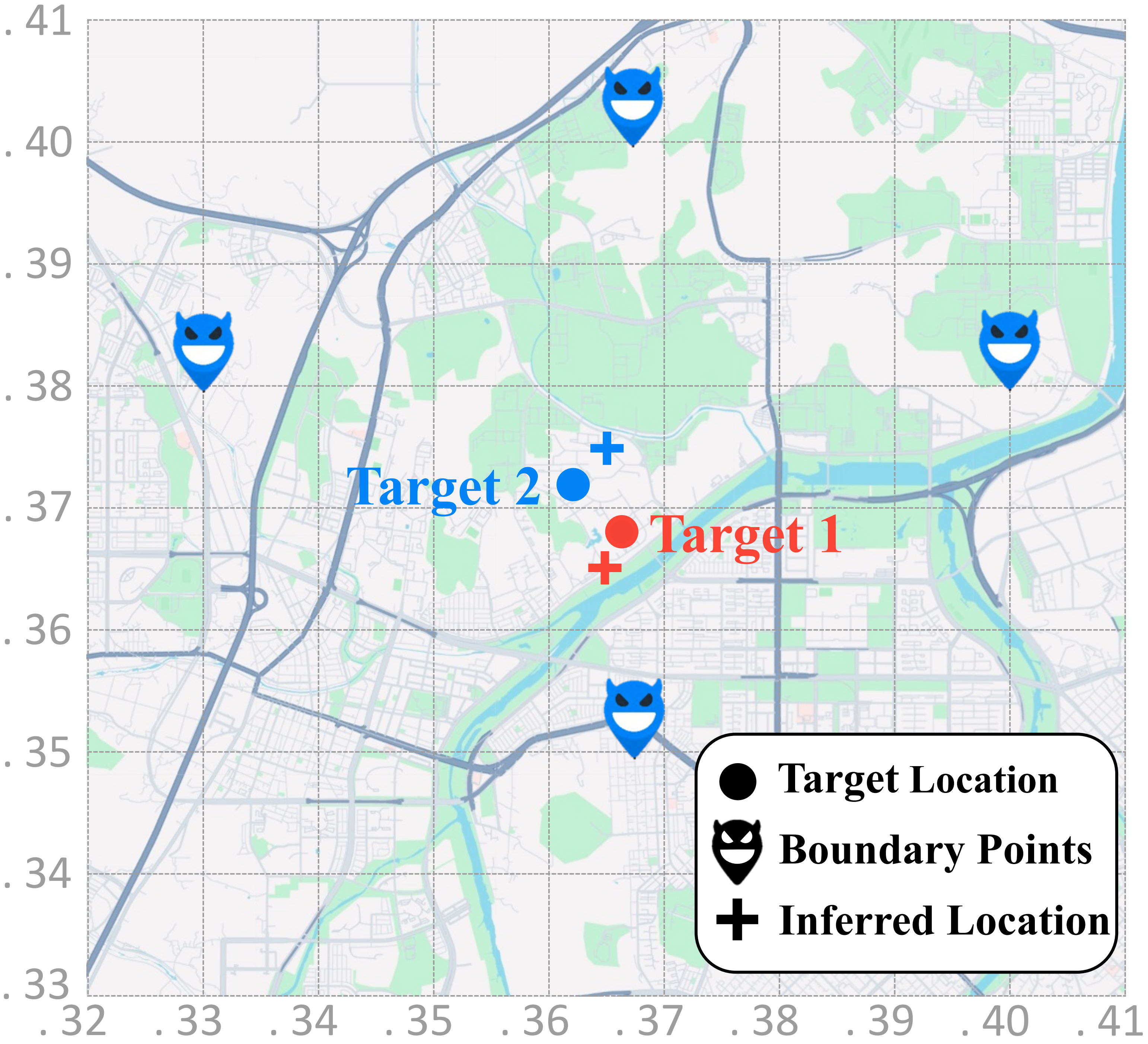
TALK → Profile image → Image Embeddings → Find target → **2. Trajectory Tracking (Location Inference)**



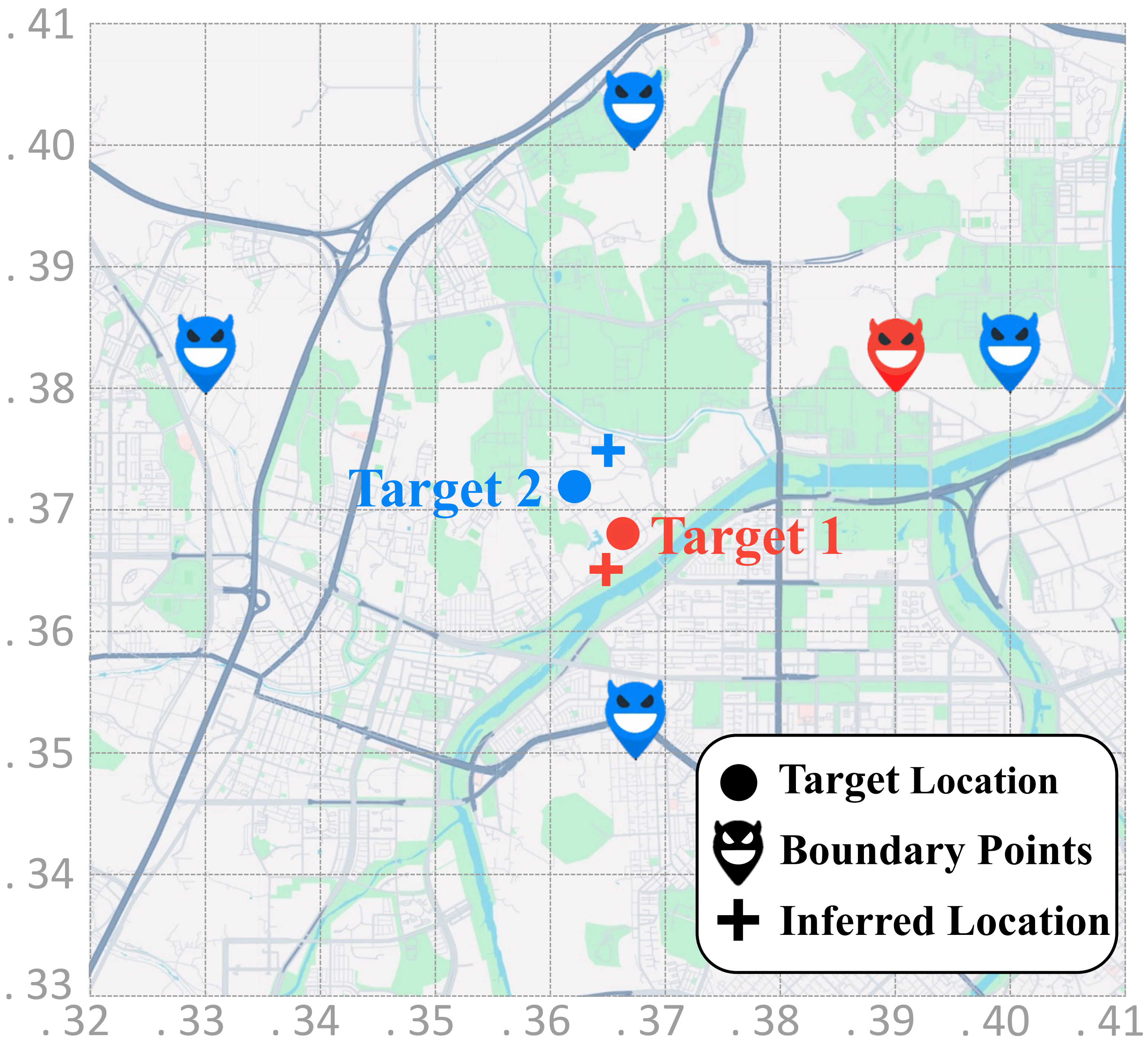


Chain 3: Targeted trajectory tracking (token → profile → Tinder)

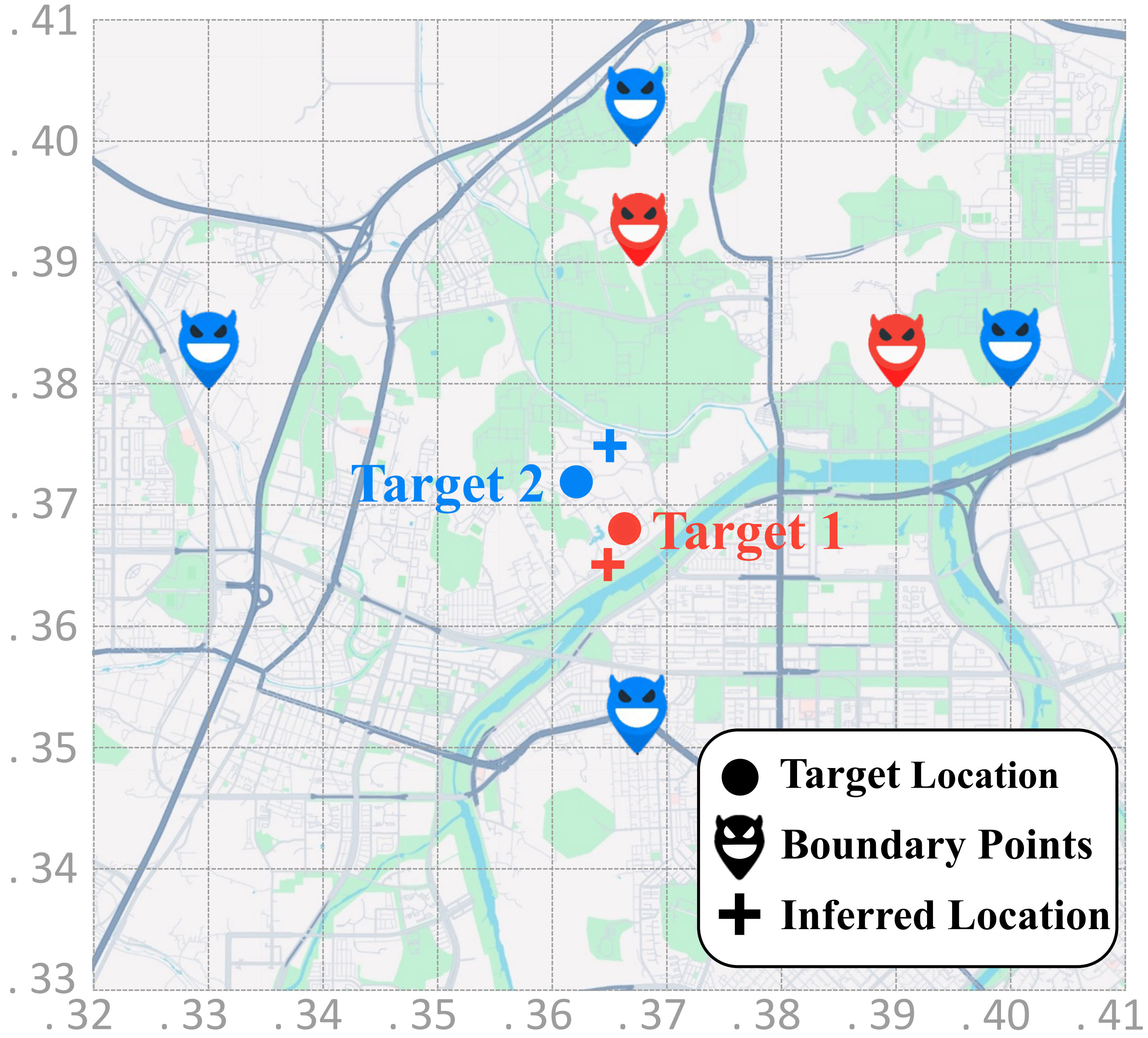




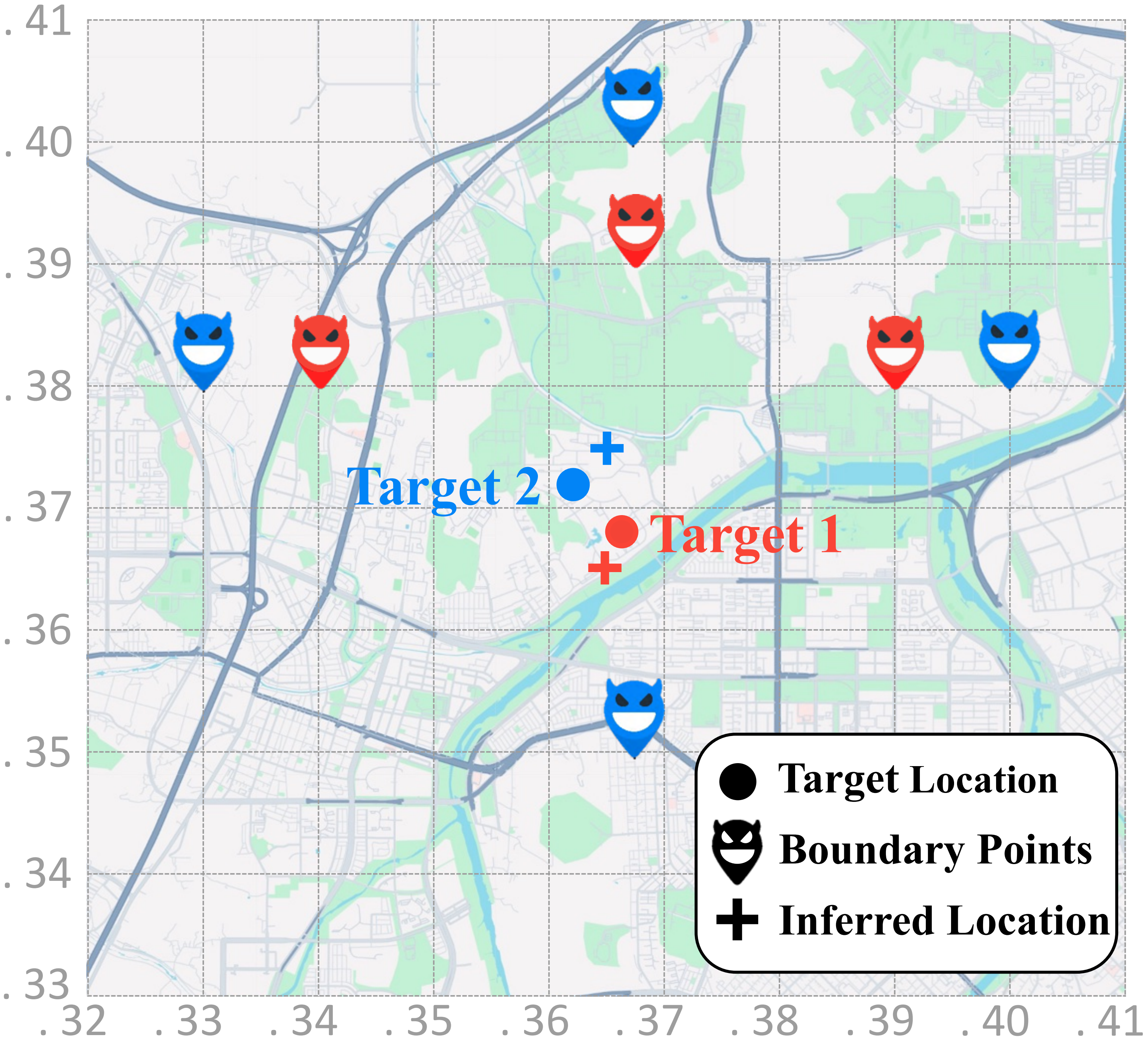
.41
.40
.39
.38
.37
.36
.35
.34
.33
.32
.33
.34
.35
.36
.37
.38
.39
.40
.41
Target 2
Target 1
Target Location
Boundary Points
Inferred Location



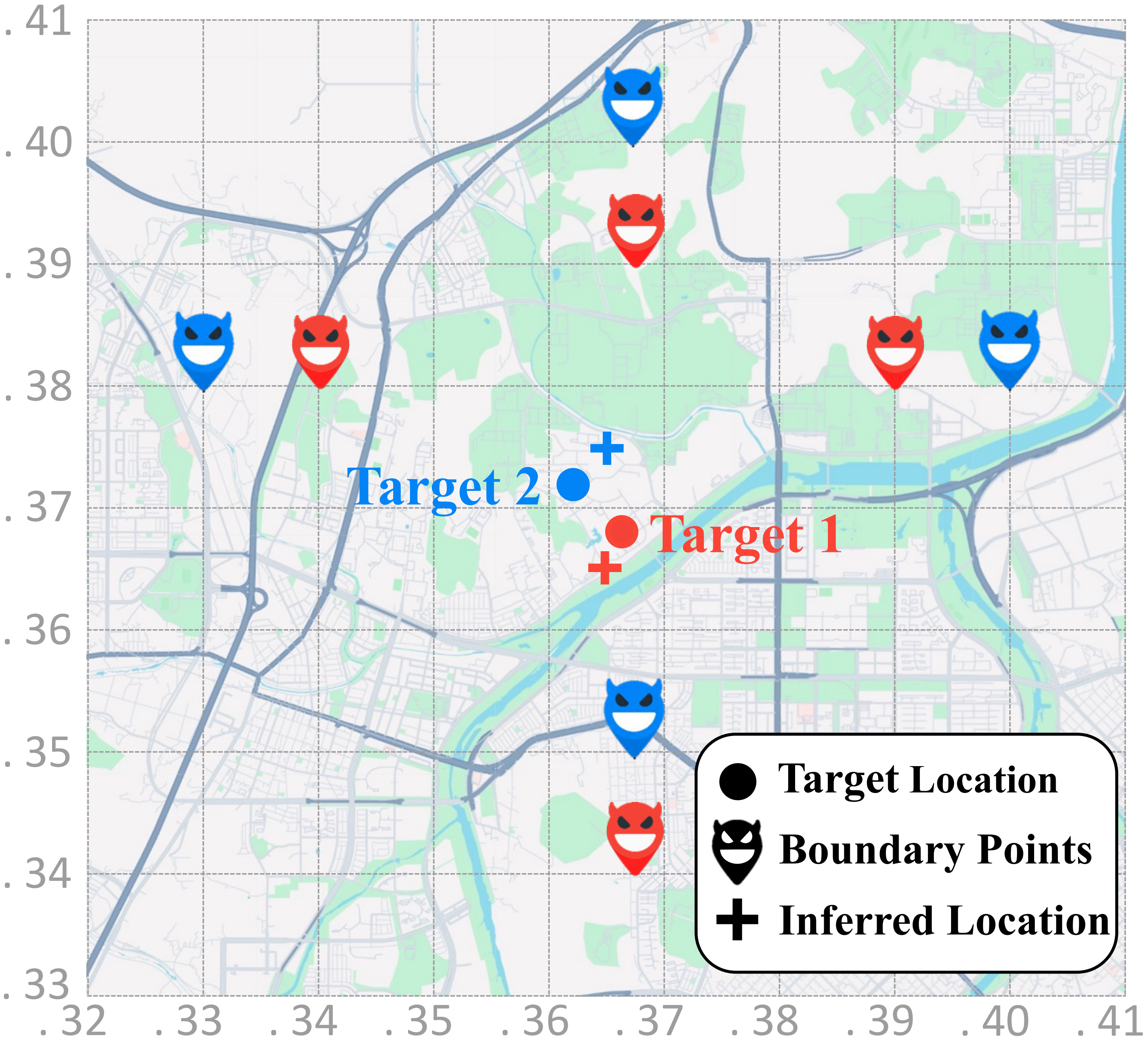
.41
.40
.39
.38
.37
.36
.35
.34
.33
.32
.33
.34
.35
.36
.37
.38
.39
.40
.41
Target 2
Target 1
Target Location
Boundary Points
Inferred Location

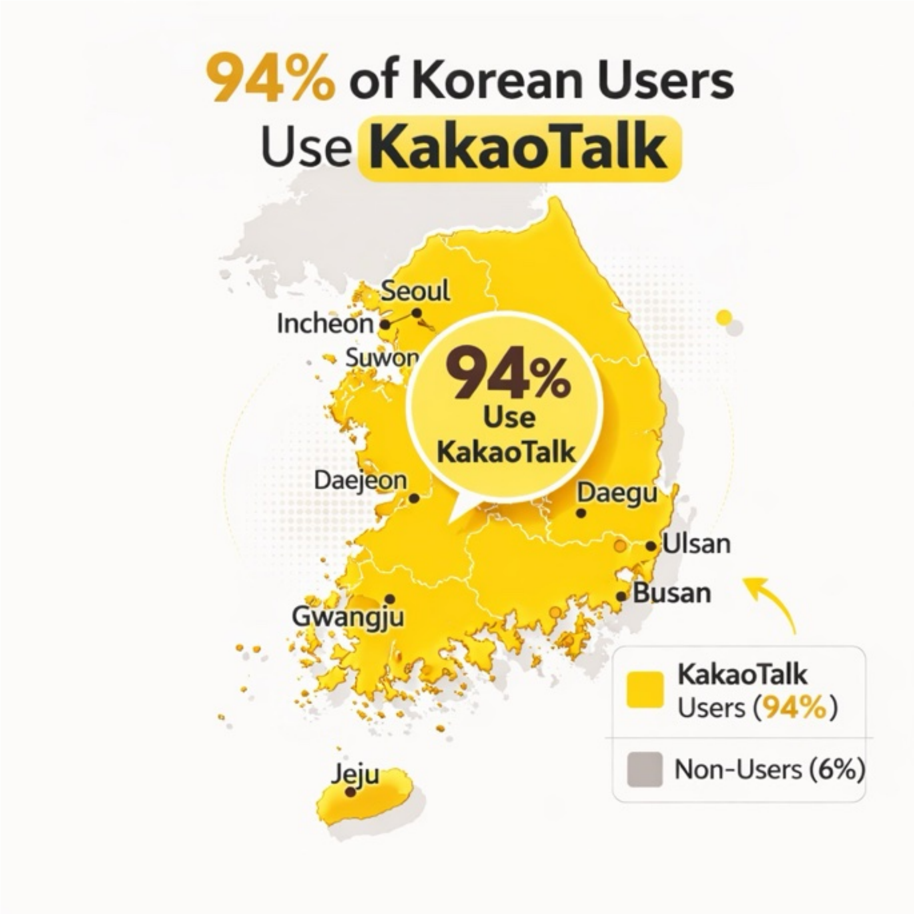


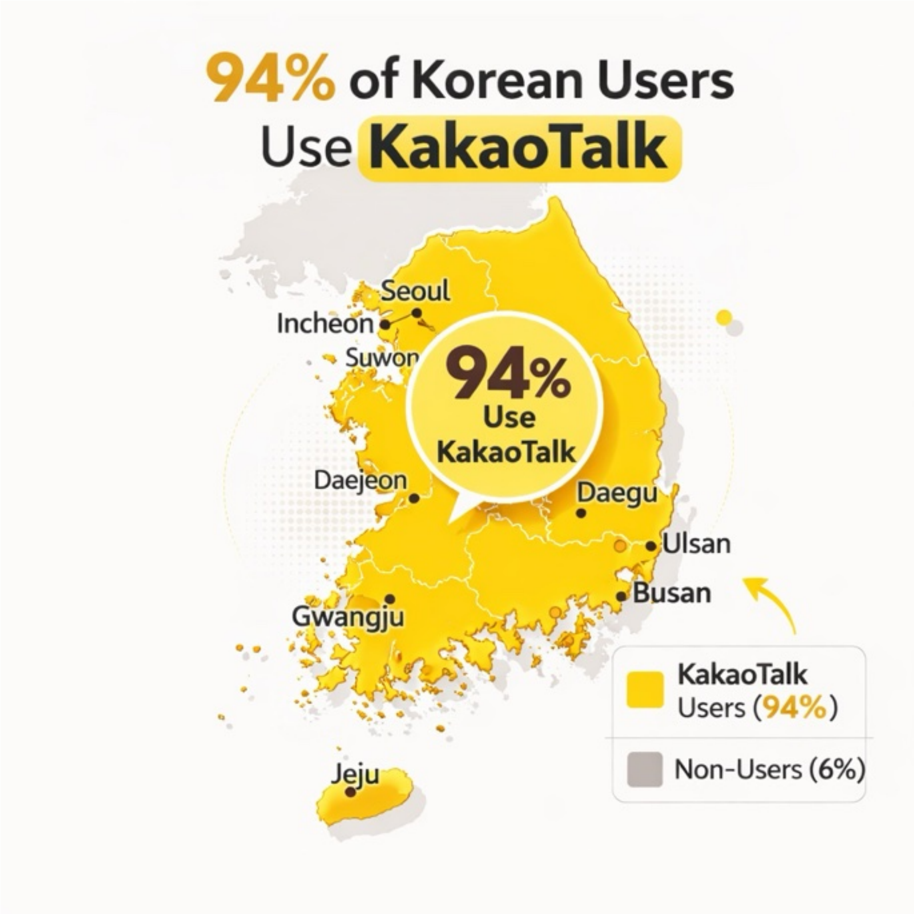
.41
.40
.39
.38
.37
.36
.35
.34
.33
.32
.33
.34
.35
.36
.37
.38
.39
.40
.41
Target 2
Target 1
Target Location
Boundary Points
Inferred Location



.41
.40
.39
.38
.37
.36
.35
.34
.33
.32
.33
.34
.35
.36
.37
.38
.39
.40
.41
Target 2
Target 1
Target Location
Boundary Points
Inferred Location


.41
.40
.39
.38
.37
.36
.35
.34
.33
.32
.33
.34
.35
.36
.37
.38
.39
.40
.41
Target 2
Target 1
Target Location
Boundary Points
Inferred Location


94% of Korean Users
Use KakaoTalk
Seoul
Incheon
Suwon
94%
Use
KakaoTalk
Daejeon
Daegu
Ulsan
Busan
Gwangju
Jeju
KakaoTalk
Users (94%)
Non-Users (6%)



KakaoTalk in S. Korea