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

Junkyu Kang[*]
KAIST
jkkang130@kaist.ac.kr

Soyoung Lee[*]
KAIST
soyoungleell@kaist.ac.kr

Yonghwi Kwon
University of Maryland
yongkwon@umd.edu

Sooel Son
KAIST
sl.son@kaist.ac.kr

***Abstract*—Mobile messaging apps have become an integral part of daily communication with massive user bases (e.g., over 950 million on Telegram and 48.7 million on KakaoTalk). To boost user engagement and user base, messaging apps offer diverse context-rich and platform-specific features, such as nearby user search, contact discovery, and single sign-on (SSO)-based account linking. While these features enable users to adopt multiple messaging apps on a single mobile device, they also introduce privacy risks of linking private user information across multiple message apps, which remains understudied.**

This paper presents an in-depth analysis of privacy threats in widely used messaging apps in South Korea, including KakaoTalk, Telegram, WhatsApp, Signal and Tinder, demonstrating concrete attacks exploiting their contact discovery, SSO-based account linking, and nearby user search features to compromise user privacy. More importantly, we chain the attacks to conduct the first cross-platform linking attack, which enables adversaries to deanonymize user names and infer users' physical locations with an average error margin of 324 meters for a large number of untargeted and targeted users. Our findings highlight that securing contact discovery is crucial as permissive contact discovery policies allow adversaries to exploit phone numbers and profile images as linking keys to connect private user information across multiple messaging apps. We discuss and propose mitigation strategies to alleviate the presented threats.

I. Introduction

Privacy attacks targeting messaging apps have become increasingly prevalent. To name a few, WhatsApp was exploited to install spyware via voice calls [1], and attackers in anonymous chat rooms on KakaoTalk linked anonymous identities to their phone numbers, which were then used for voice phishing and marketing calls [2], [3].

Given the inherent nature of messaging apps collecting privacy-sensitive user data, including names, phone numbers, profile images, and even physical locations, privacy attacks targeting these platforms pose significant privacy threats. Even worse, the vast number of messaging app users exacerbates these threats: Telegram, WhatsApp, and KakaoTalk have approximately 950 million, 2 billion and 48.7 million monthly active users, respectively [4]–[6]. Telegram and WhatsApp are widely used in India and U.S. [7]–[9], whereas KakaoTalk is predominantly used in S. Korea, covering about 94% of its population [10].

Recently, a growing trend has been observed in which users adopt multiple messaging platforms on a single mobile device to use each platform's unique features [11]. Users may choose Telegram for its support of anonymous communication and strict data non-disclosure policies, and Tinder to take advantage of its location-based matching features. KakaoTalk supports single sign-on (SSO) for third-party services, supporting users to avoid managing multiple credentials by leveraging their dominant 94% market share in S. Korea. Our measurement confirms this multi-platform usage trend: 84% of sampled phone number owners in Telegram, WhatsApp, and Signal are also found on KakaoTalk (§XI-B).

Despite the widespread multi-platform usage, the privacy risks of using multiple messaging apps together remain largely understudied. Prior research has focused on privacy threats within individual platforms [12], [13], examining large-scale identity revelation attacks on popular messaging apps.

In this paper, we revisit and extend privacy attacks on widely used messaging apps in S. Korea, with a particular focus on linking private information across the messaging apps. In particular, we present a case study demonstrating how a dominant messaging app vendor (i.e., one that occupies a significant portion of the phone number space) can serve as an effective linking key for combining private user information across different messaging platforms.

Our contributions. We present three concrete privacy attacks targeting five popular messaging apps in S. Korea: (1) identity disclosure via contact discovery abuse on Telegram, KakaoTalk, WhatsApp and Signal, (2) KakaoTalk identity harvesting by exploiting exposed SSO tokens, and (3) location trajectory inference of Tinder users.

(1) Identity disclosure: We evaluate the feasibility of retrieving user names and their profile images by querying brute-force-generated phone numbers through the contact discovery services of Telegram, KakaoTalk, WhatsApp and Signal. Notably, we reveal that KakaoTalk's permissive query budget and loophole allows over 18,376 contact registrations in a single day, enabling adversaries to identify the owners of arbitrary phone numbers. We demonstrated that attackers with only a single phone number are able to retrieve profile information for

[*] Both authors contributed equally to the paper.

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.230556
www.ndss-symposium.org

105,366 users, attaining a success rate of 67.9%. This threat is further exacerbated due to the dominant market share, 94% of the population registered on the platform in S. Korea, making large-scale profile database construction feasible.

(2) Identity harvesting: We discovered 63 popular websites that expose their SSO access tokens through URLs, cookies, local storage, and session storage to third parties and man-in-the-middle adversaries. These insecure patterns enable the adversary to exploit exposed tokens to authenticate as the token owner's KakaoTalk account, which in turn enables the exfiltration of various private information, including names, phone numbers, age, birth date, gender, and addresses. Alarmingly, in 16 out of the 63 vulnerable websites, we observe explicitly exposed access tokens in the URLs used by the third-party scripts.

(3) Location inference: We demonstrate that an adversary can infer the approximate location of a Tinder user based on a given profile image. This attack exploits Tinder's location-based service, which returns profile images of nearby users for a specified GPS location. By computing the embedding similarity between the given profile image and each nearby user's image, the adversary effectively confirms the presence of specific users at selected GPS locations. By repeating this process across multiple GPS locations systematically chosen to narrow the search space, adversaries can infer an accurate victim's location within an average error margin of 324 meters.

Beyond individual privacy attacks, we present the first cross-platform linking attacks that chain the privacy attacks aforementioned. Specifically, we introduce three novel end-to-end attacks: (1) deanonymizing Telegram, WhatsApp and Signal users via KakaoTalk, (2) inferring the locations of untargeted Tinder users using KakaoTalk, and (3) tracking targeted Tinder users whose KakaoTalk SSO tokens are leaked. Our deanonymization attack identifies the names and profile images of 22 out of 40 Telegram users who appeared under anonymous aliases. For the location inference attack, we accurately identified two authors from a pool of 5,000 random phone numbers, pinpointing their locations within 336 meters and 418 meters radius, and successfully reconstructing their commuting trajectories, including home and work locations.

Importantly, we believe that the presented chained privacy attacks have a large impact as they allow the construction of large-scale datasets containing sensitive attribute pairs, including names, phone numbers, profile images, and location trajectories. We attribute the success of our attacks to two environmental factors in S. Korea: (1) a small phone number space, which is largely saturated by mobile devices, and (2) the dominant market share of KakaoTalk. These conditions facilitate the presented linking attacks and leave important lessons for messaging platforms with dominant market presence.

We conclude with two mitigation strategies for preventing contact discovery abuse and one defense method for protecting location privacy. First, we recommend imposing strict query-rate limits on contact discovery services and disabling access to profile images by default. Second, we propose a novel detection mechanism that identifies adversarial queries involving randomly sampled phone numbers by leveraging the observation that legitimate contacts added by users highly likely form social circles with existing friend relationships. Lastly, to mitigate location inference attacks on Tinder, we apply existing differential privacy-based noise mechanisms to user-reported locations, effectively increasing the attacker's cost of location inference attacks.

II. Background

A. Contact Discovery of Messaging Apps

Contact discovery refers to an automated procedure that mobile messaging apps perform to *retrieve contacts from a user's address book* on their mobile device [13]. For each phone number in the address book, a contact discovery service finds a matching user and registers them as a *friend*. This process facilitates the seamless registration of users without manually adding individual phone numbers.

Telegram. Telegram implements a contact discovery service. For a given phone number, it returns the registered account holder name, nickname, profile image, the number of Telegram users who have attempted to register this user, and other miscellaneous private information [13].

KakaoTalk. KakaoTalk is a mobile messaging app that dominates the messaging app market in S. Korea, with 48.7 million users—approximately 94% of the South Korean population—using this mobile app for daily communication [6]. KakaoTalk offers a contact discovery service that provides a profile image and name for each phone number in a user's address book.

WhatsApp. WhatsApp is a popular messaging app with approximately 2 billion users across over 180 countries [14]. It provides a contact discovery service. When a queried phone number belongs to a registered user, their profile appears in the contact list of the messaging app. Unlike other messaging platforms, WhatsApp limits the visibility of profile information; by default, only the profile image is shown, while the account names set by other users remain inaccessible unless explicitly shared through privacy settings.

Signal. Signal is an open-source, privacy-focused messaging app that offers end-to-end encryption. It also supports friend addition through contact discovery. The profile image and name of each account holder remain hidden by default and are only disclosed to those who have added them as a contact, if the user explicitly consents to share.

Tinder. Tinder is an online dating app that facilitates communications between registered users. Tinder does not provide any contact discovery services. However, it offers a nearby user search functionality, which displays profile images of other users located near the app user. Tinder has been downloaded over 630 million times, serving approximately 50 million monthly active users [15]. In S. Korea, Tinder's monthly active users are approximately 170,000 [16].

Phone numbers. South Korean cellular phone numbers typically begin with the prefix '010' and are followed by an 8-digit unique subscriber number (e.g., 010-XXXX-XXXX). Out of 100 million possible numbers in this system, 80 million are available for allocation under government policies [17]–[19].

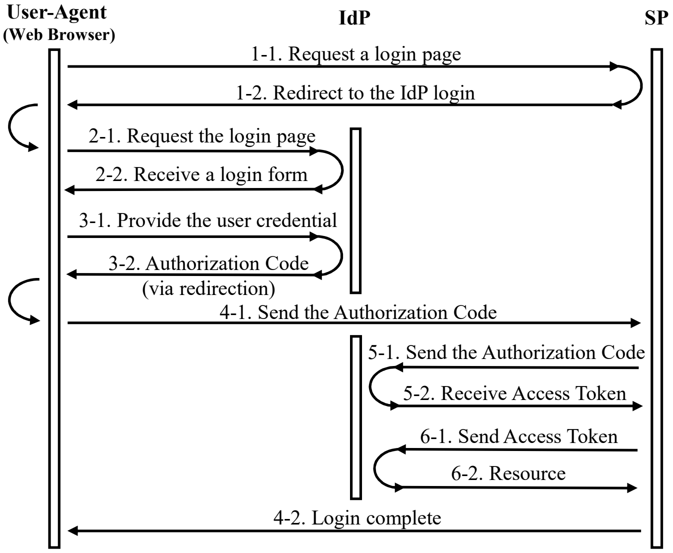

User-Agent
(Web Browser)
IdP
SP
1-1. Request a login page
1-2. Redirect to the IdP login
2-1. Request the login page
2-2. Receive a login form
3-1. Provide the user credential
3-2. Authorization Code
(via redirection)
4-1. Send the Authorization Code
5-1. Send the Authorization Code
5-2. Receive Access Token
6-1. Send Access Token
6-2. Resource
4-2. Login complete

Fig. 1: A simplified account linking process using the OAuth 2.0 authorization code grant method [21].

Among them, 73.9 million are assigned to cellular carriers with approximately 63.7 million in use, indicating that 79.6% of the available numbers are already assigned [20].

B. SSO-based Account Linking Service

Account linking refers to a process that enables a user to connect their account on one identity provider (IdP) website to an account on a different service provider (SP) website. This process uses an SSO mechanism, enabling authentication across multiple websites with user credentials from a single IdP. Similar to Google and Facebook, KakaoTalk also provides an SSO-based account linking service, enabling registered users to use their credentials across multiple SP websites.

Figure 1 illustrates an authentication process using KakaoTalk's account linking mechanism, which is based on the OAuth 2.0 authorization code grant method [21]. This process involves three parties: user, SP, and KakaoTalk IdP. It starts by initiating the authentication of the user to the SP website (Step 1). Then, the SP redirects the user to the IdP's login page, along with their SP identifier and redirection URI (Step 2). After the user successfully authenticates on the login page of the IdP, the IdP redirects the user back to the SP using the redirection URI provided earlier to deliver the authorization code to the SP (Step 3,4). The SP uses this one-time authorization code to obtain an *access token* from the IdP and then uses this token to access the user's private information stored by the IdP (Step 5,6). The SP then authenticates the user using this information. We emphasize that this access token should not be exposed to the user and should be used for accessing IdP-provided information through an out-of-band channel rather than being exposed to user-accessible web requests or responses [22].

Each access token has a scope that defines the level of access to the private information held by the IdP. Table II lists 16 categories of private information and privileges accessible to a token holder. Given that these tokens grant access to privacy-sensitive user information, SPs should protect OAuth access tokens from adversaries seeking to exfiltrate and misuse them.

Many OAuth 2.0-based services, including KakaoTalk, have adopted the bearer token authentication mechanism [23], [24]. A bearer token is a security token that allows any holder to use it without requiring proof of token ownership. This property makes bearer tokens particularly vulnerable to their misuse if they are exposed [25]. To mitigate these risks, SPs should implement strict token management practices by avoiding to pass such access tokens to the user side, thereby minimizing the likelihood of token exposure.

III. Motivation

A. Threat Model

We assume an adversary of which goal is to extract privacy-sensitive information from popular messaging apps, specifically Telegram, KakaoTalk, WhatsApp, Signal and Tinder. The adversary's objective is to collect users' names, phone numbers, profile images, and location trajectories.

Our adversary is motivated to conduct large-scale surveillance by linking identities to phone numbers and locations at minimal cost. We demonstrate that this adversary is capable of identifying 105,366 KakaoTalk users and their phone numbers, and infer the locations of those users if they are also active on Tinder, enabling the construction of large user databases containing names, phone numbers, profile images, and mobility patterns. Similar datasets have been sold for amounts ranging up to $3.5 million [26], depending on their size.

Weak adversary capability. The adversary is able to register a limited number of accounts on Telegram, KakaoTalk, WhatsApp, Signal, and Tinder using their own phone numbers. With these registered accounts associated with a phone number, they can access the legitimate app functionalities, including contact discovery, SSO-based account linking, and nearby user services. This weak adversary has no additional capabilities beyond those of typical messaging app users. We consider the number of phone numbers leveraged by the adversary as the cost of an attack campaign. Accordingly, the adversary seeks to maximize the collection of privacy-sensitive information from messaging app users while minimizing their attack cost.

Strong adversary capability. In addition to the capabilities of the weak adversary, we assume a strong adversary is able to access exposed SSO access tokens (§IV-B), such as KakaoTalk access tokens, to conduct privacy attacks. The adversary is either (1) the owner of a third-party script that is embedded in SP websites using SSO-based account linking or (2) a classic web attacker [27], [28], capable of exploiting cross-site scripting vulnerabilities to exfiltrate exposed access tokens.

We note that any third-party script owners embedded in SP websites using SSO-based account linking can become this strong adversary. In our measurement (§IV-B), we identified that 135 third-party script providers embedded in vulnerable SP websites along with Google, Facebook, and Naver are capable of acting as this adversary.

TABLE I: Policies of messaging apps regarding query limits in contact discovery and default profile visibility.

Platforms	Query limit criteria	Conditions	Observed enforcement	Profile Img	Username
Telegram	Query rate	100 / day	Allowing queries for up to 100 phone numbers per day.	✓	✓
KakaoTalk	Account age & Activity	90 days & high activity	Friends registration: Prohibiting the registration after reaching 15,000 friends. Friends deletion: Allowing additional 20,000 registrations after friends deletion. Friends block & unblock: Allowing additional 18,376 registrations per day.	✓	✓
		90 days & no activity	Prohibiting the registration of new contacts.		
		7 days & high activity	Prohibiting the registration of new contacts after reaching 1,000 phone numbers.		
		7 days & no activity	Prohibiting the registration of new contacts after reaching 1,000 phone numbers.		
WhatsApp	Query rate	20,000 / day	Allowing queries for up to 20,000 phone numbers per day. However, a total query cap (i.e., approximately 40,000) applies over a fixed period. Once this limit is reached, additional queries are only permitted after certain time has elapsed. We were able to add 15,000 contacts over two days after this cap is reached.	✓	✗
Signal	Query rate	Init 50,000 & 144 / day	Allowing queries for up to 50,000 phone numbers initially. After this, queries for an additional 144 phone numbers are permitted per day.	✗	✗

IV. Privacy Attacks Targeting Messaging Apps

We present three privacy attacks that (1) retrieve the profile information of Telegram, KakaoTalk, WhatsApp and Signal users (§IV-A), (2) extract the private information of KakaoTalk users using exposed OAuth access tokens (§IV-B), and (3) infer the location trajectories of Tinder users (§IV-C).

A. Obtaining Identities via Contact Discovery

Telegram, KakaoTalk, WhatsApp and Signal provide a contact discovery service that allows users to retrieve registered profiles that match provided phone numbers and then add these matching profiles as contacts (i.e., friends). By abusing this feature, the adversary is able to obtain a list of phone numbers and profile pairs $(phone, profile)$, referred to as phone-profile pairs. This $profile$ contains a user name registered by the account holder and their profile image.

Attack method. An adversary starts by enumerating phone numbers starting with '010.' Recall that there are only 73.9 million available mobile phone numbers in S. Korea, covering 63.7 million 4G/5G cellular phones [17]–[19] (§II-A). The adversary randomly selects a generated number, queries it on a target messaging app by adding it as a contact, and then retrieves the corresponding profile. This enables the adversary to build a large number of phone-profile pairs, as long as the discovery services permit the queries. Note that Telegram, KakaoTalk, WhatsApp and Signal impose query limits on their contact discovery services for each device with a phone number, thus limiting the adversary's capability.

Telegram. As previous research [13] has investigated, Telegram imposes a strict limit of allowing only 100 contact discovery queries per day. We confirmed that this condition remains valid through an experiment in which we attempted to add 1,000 randomly generated phone numbers to Telegram using a single phone number. This process took 10 days, during which 88 (8.8%) of the sampled phone numbers were successfully added as contacts to the account associated with the phone number used. This result implies that the adversary is able to expect to successfully retrieve profiles for 8.8% of queried phone numbers. Once a user exceeds the query limit of 100 per day twice, their account becomes shadowbanned, preventing further attempts to add new contacts.

KakaoTalk. KakaoTalk provides two ways of using its contact discovery feature. The first method asks users to manually add friends by providing explicit contact information, including names and phone numbers, through the app interface. The second one enables users to automatically synchronize all contacts from their address book to KakaoTalk.

Given the lack of recent studies on checking the query limit on KakaoTalk's contact discovery service, we conducted experiments to measure this query limit. For the first contact discovery channel (i.e., manual friend addition), we observed that KakaoTalk imposes a strict limit of 50 users per day, imposing the most restrictive rate limit.

In contrast, for the second channel (i.e., syncing KakaoTalk with a user's address book), we identified specific conditions under which users are capable of registering a large number of phone numbers without restriction. These conditions are influenced by two factors: account age and account activity. For account age, we compared accounts older than 90 days with those only 7 days old. For account activity, we compared active accounts (with more than 999 chat rooms) to inactive accounts (with no chat rooms). We attempted to register 1,000 phone numbers at once, and the results are as follows.

A 90+ day account with high activity was able to register an unlimited number of phone numbers. Using our method, we observed no strict limits on registering a large number of phone numbers. While KakaoTalk imposes a contact limit of 15,000 per account, this restriction can be bypassed by blocking existing contacts to free up space for new additions. KakaoTalk also enforces a cap of 15,000 on the number of blocked contacts. However, the adversary is able to unblock previously blocked contacts, which both preserves available contact slots and resets the blocked contact count. This loophole enables continuous registration of new phone numbers without violating either limit. Accordingly, this method effectively enables the continuous registration of new phone numbers; during our testing, we observed no upper limit on unblocking requests within 64,000 unblocking attempts.

Since blocking contacts involves individual UI interactions with the KakaoTalk app, we developed a Python script to automate this process. The script reduced the blocking time to two seconds per contact. Also, we used Android Auto Clicker [29] for unblocking with two seconds. These time constraints of requiring four seconds per number render a query limit of approximately a maximum of 21,600 queries per day through this contact discovery channel. In our experiment, we attempted to register contacts with 155,147 phone numbers; among these, 105,366 users were successfully added as contacts over 10 days (registration rate: 67.9%). On the other hand, the 90+ day account with no activity was immediately banned when we tried to register phone numbers. Similarly, the 7 day account successfully registered 1,000 phone numbers, but was banned when attempting to register additional phone numbers. These results show that the adversary with an active chat history over 90 days is capable of generating approximately 18,376 phone-profile pairs per day.

WhatsApp. WhatsApp are known to employ a leaky bucket rate-limiting system with an estimated bucket size of approximately 120,000, allowing up to 60,000 contact queries per day as of 2019 [13]. However, our experiments revealed a more restrictive policy; they now permit adding only 20,000 contacts per day. Once the cumulative total reaches around 60,000, their rate limiting activates, enforcing a cooldown period before registering additional contacts. In our study, we successfully synced 98,000 phone numbers over a 14-day period, averaging 7,000 contacts per day. Among these, 3,194 (3.3%) were registered as WhatsApp users, and their profile images were accessible through the contact discovery feature.

Signal. Since Signal's server-side code is open source, we analyzed its contact discovery implementation and found that it employs a leaky bucket rate-limiting mechanism with an estimated bucket size of 50,000. To validate this, we conducted measurements using randomly sampled phone numbers. We were initially able to send up to 50,000 contact discovery queries. After that, the server permitted approximately 1,000 additional queries per week. Based on our measurements and analysis of Signal server responses with log messages, we concluded that Signal allows 144 additional queries per day once the initial bucket limit is reached.

Out of 51,000 phone numbers queried, 153 (0.3%) were successfully added as contacts within a single day. Compared to other platforms, Signal has relatively low adoption in S. Korea. Additionally, we note that retrieved profile images and names were inaccessible due to the Signal's contact discovery policy (§II-A). Therefore, this privacy attack on Signal only confirms the existence of account holders and does not reveal their profile images and names. However, we show that such inaccessible information can still be retrieved by cross-referencing phone numbers with a dominant platform when those users are on two messaging platforms (§V-A).

Costs. Figure 2 shows the number of contact discovery queries allowed and the estimated number of successfully registered users for Telegram, KakaoTalk, WhatsApp and Signal over time. When assuming our observed registration rates of 8.8%, 67.9%, 3.3%, and 0.3% for Telegram, KakaoTalk, WhatsApp, and Signal respectively, the adversary is able to retrieve the phone-profile pairs for 88 users in Telegram, 199,136 users in KakaoTalk, 2,310 users in WhatsApp, and 154 users in Signal over 10 days by using only one phone number. Even worse, we note that the adversary is able to coordinate an attack campaign using multiple phone numbers in S. Korea.

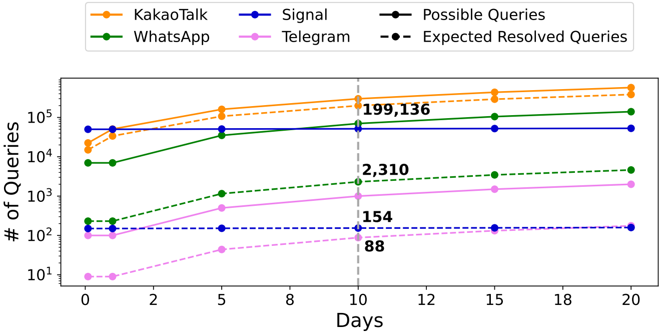


Fig. 2: Numbers of allowed contact discovery queries and estimated successfully registered users in Telegram, KakaoTalk, WhatsApp and Signal over time.

The adversary has various ways of obtaining new phone numbers. They can exploit phone number rental services [30], [31] and SMS verification bypass services [32], [33] to acquire new phone numbers quickly and on a large scale. Furthermore, in S. Korea, low-cost mobile carriers provide new numbers for just $0.10 [34]. Therefore, these services facilitate attacks by allowing the adversary to create many accounts at a low price; the attack is expected to obtain profile images and names for approximately 1.8 million phone numbers using 10 valid phone numbers for the attack campaign over 10 days only with $1 (18,376 * 10 numbers * 10 days).

Kim *et al.* [35] tested the efficacy of their privacy attacks that harvest Facebook profiles associated with phone numbers, demonstrating the feasibility of testing 200,000 Californian and Korean phone numbers within 15 days, reporting success rates of 12% and 25%, respectively. Hagen *et al.* [13] also reported success rates of 9.8%, 0.5%, and 0.9% for their contact discovery attacks against WhatsApp, Signal, and Telegram, respectively. Comparing these results, our attack success rates of 8.8% for Telegram and 67.9% for KakaoTalk significantly outperform those reported in prior studies. We attribute these success rates to two environmental factors in S. Korea: (1) the relatively small phone number space (i.e., 73.9 million), which is largely occupied by the South Korean population and their mobile devices (i.e., 63.7 million), and (2) the dominant market share of KakaoTalk.

We note that both Telegram and WhatsApp impose strict limits on contact discovery queries; Telegram permits only 100 queries per day, and WhatsApp allows adding average 7,000 contacts per day. However, these restrictions do not eliminate successful contact discovery attacks. Due to the low cost of acquiring mobile phone numbers, the adversary can scale the attack operation by distributing their queries across multiple devices. For instance, with 30 phone numbers, the adversary can still retrieve profile information for up to 264 Telegram users and 6,930 WhatsApp users. These numbers increase proportionally with the user base of each platform. Assuming a contact discovery success rate of 67.9% as observed in KakaoTalk and accounting for market share differences, the estimated numbers increase to 2,037 for Telegram and 142,590 for WhatsApp.

Our findings highlight the need for stricter query rate limiting in contact discovery services, particularly in countries where a single messaging platform dominates the market. We believe that similar risks are evident in regions such as India and Vietnam, where WhatsApp and Zalo hold market shares of 79.59% and 60.24%, respectively.

Attack implementation. For Telegram, we leveraged TDLib, a cross-platform library for building Telegram clients [36], [37]. In TDLib, the **getUser** API function provides access to the profiles of users either in the client's Telegram contact list or those who have previously interacted with the client. However, the original name set by the target user is not accessible in this profile information. However, when we removed this user from the client's contacts list using the **removeContacts** function, we were able to access the original name set by this target user. We thus implemented a script to automate this process, thus generating a list of phone-profile pairs for given phone numbers. For WhatsApp, we used a web crawler to retrieve profile information from its web service.

KakaoTalk does not provide open APIs or web version service for accessing the profile information of registered contacts. Furthermore, neither the mobile nor PC versions of KakaoTalk offer direct access to usernames within the retrieved contact profile data. Therefore, we leveraged an open-source version of KakaoTalk client [38]. Specifically, we parsed responses to HTTP POST requests that retrieve the profile information of registered contacts. This approach enables the extraction of the registered names and profile images of the registered KakaoTalk users.

B. Obtaining Identities via Exposed Tokens

We investigate the privacy risks posed by the adversary attempting to exfiltrate OAuth access tokens due to insecure practices of SPs in deploying the SSO-based account linking service that KakaoTalk offers. Considering that OAuth access tokens grant access to victims' profile images and phone numbers based on their scopes, the adversary is able to misuse these tokens to obtain profiles and corresponding private information. Table II shows the possible scopes of KakaoTalk OAuth access tokens.

TABLE II: Information categories on KakaoTalk accessible via their OAuth access tokens based on their scopes.

Category	Information
Profile Info	Nickname, Profile image
Private Info	Name, Gender, Age ranges, Birthday, Phone number Email, Birth year, Connecting Info, Shipping Info
KakaoTalk Related	Friends list, Service channel addition status Send messages in KakaoTalk Manage Talk calendar and events Manage tasks in Talk Calendar

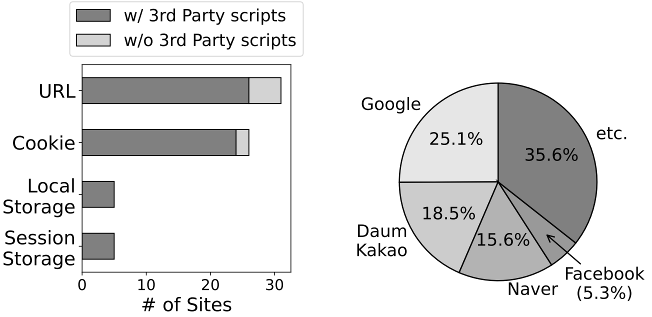
Analysis method. To assess the current status of token leakage, we first crawled login-related pages from popular real-world websites. To identify webpages that support KakaoTalk account linking, we implemented and applied a KakaoTalk logo classifier, thus producing a list of 14,102 webpages likely to deploy KakaoTalk account linking. We then examined these webpages for potential access token exposures.

Crawling. We investigated popular websites integrated with KakaoTalk's account linking service. To compile a comprehensive list of high-traffic websites offering the KakaoTalk account linking service, we crawled the top 300 websites in S. Korea by traffic, based on Similarweb statistics (from May 2024 to July 2024). We performed Google keyword searches using 9,920 common keywords, which include approximately 6,000 basic Korean words commonly used in education, curated by the National Institute of Korean Language [39], and 3,000 English words from the Oxford Basic Word List [40].

For this keyword search process, we used Google Search APIs with our own site-filtering rules to collect URLs containing any of keywords related to authentication (e.g., login, signin, members). To compile a list of authentication-related keywords, we analyzed the top 300 high-traffic Korean websites, examined the path keywords used in login pages, and selected the top 20 most frequent keywords. We identified these login-related keywords by analyzing the authentication pages of the Similarweb top 300 websites. After this crawling process, we compiled a set of 85,053 authentication webpages for further investigation.

Accordingly, from these 85,053 webpages, we identified 14,102 webpages that integrate the account linking service of KakaoTalk. For this, we configured our crawler to click icons or buttons displaying KakaoTalk logos and to confirm whether these clicks were redirected to the authentication page of KakaoTalk. Specifically, we used Selenium with Chrome WebDriver to identify link or button elements and to extract images corresponding with these HTML elements. We then fed each extracted image into our image classifier that outputs a confidence vector, indicating the likelihood that the image is a KakaoTalk logo. Using this classifier, we sorted the images in descending order by likelihood. Our crawler then proceeded to click each HTML element on this sorted list. If the crawler detects a redirection to KakaoTalk's login page, it flags the website for further investigation and stops clicking the remaining HTML elements. In Appendix XI-A, we provide details of our logo classifier. We note that even if this classifier misclassifies logo images, it only affects the order in which the HTML elements are clicked; this does not lead to missing clickable KakaoTalk authentication icons.

Results. We further analysed these 14,102 webpages and observed two insecure practices that lead to expose OAuth access tokens: (1) exposing tokens in URLs and (2) placing tokens where they are easily accessible to third-party scripts. During the crawling process, we collected all URLs, cookies, and storage items for detailed analysis. We observed that KakaoTalk OAuth tokens follow a consistent pattern, where the access token is concatenated with a specific string marker. This pattern enabled us to automatically locate access tokens within URLs, cookies, and local/session storage using regular expressions. After identifying all websites exposing OAuth tokens on the client side, we manually examined the third-party scripts embedded in the authentication pages using KakaoTalk's account linking service.



(a) Locations where access tokens are exposed.

(b) Third-party domains that can access exposed tokens.

Fig. 3: Numbers of vulnerable websites.

Figure 3a shows the identified websites that expose their access tokens to various sources accessible to third-party scripts. We found that 31 websites expose OAuth access tokens in their URLs when loading post-authentication content. This insecure practice exposes tokens in the client browser's browsing history. This browsing history effectively enables the adversary to access these tokens when users log in from shared computers. Furthermore, this practice allows third-party scripts to access these access tokens. We emphasize that this practice is highly discouraged according to the OAuth 2.1 standard and OAuth 2.0 Security Best Current Practice [41], [42].

Furthermore, 26, 5, and 5 websites store their tokens in cookies, local storage, and session storage, respectively. This implementation choice allows third-party scripts to access these tokens, leaving users who authenticate via KakaoTalk's identity service vulnerable to unauthorized access by embedded third parties. Among the 63 websites that expose their tokens, 56 websites deploy third-party scripts. Alarmingly, among them, the third-party scripts in 16 websites actively collect these access tokens and transmit them to their corresponding servers in the process of harvesting the current URLs.

Figure 4 depicts websites implementing each insecure practice for each important scope of KakaoTalk access tokens. Respectively, 16 and 39 websites expose OAuth tokens that grant access to users' phone numbers and profile images. Additionally, 13 websites expose OAuth tokens that enable access to users' real names. Furthermore, among the 26 websites that store access tokens in cookies, 24 did not set the HttpOnly property, and 22 did not set the Secure property, leaving the tokens vulnerable to man-in-the-middle attacks [28].

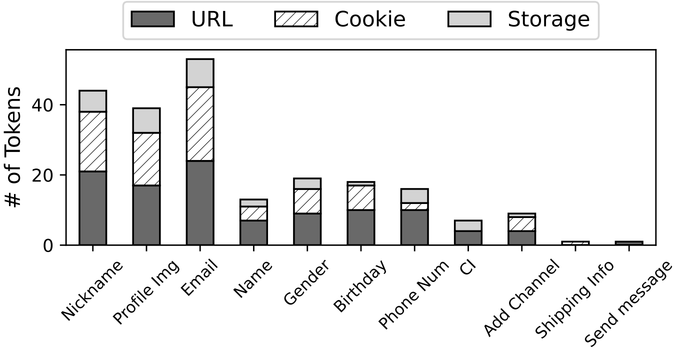


Fig. 4: Number of websites exposing OAuth access tokens categorized by their scopes.

Table III shows the accumulated visitors across vulnerable websites for each location of the exposed access tokens. 31 websites exposing their tokens via URLs have 75.31 million visitors per month, demonstrating the impact of the proposed threats.

Figure 3b shows the domain distribution of third-party script sources. A total of 379 third-party scripts were found on websites where access tokens were exposed. Among these, Google-related scripts were the most common, with 95 scripts. This was followed by Daum with 70 scripts, Naver with 59 scripts, and Facebook with 20 scripts. The remaining 135 scripts were categorized as Other. Of the remaining 135 third-party scripts, 65 are related to data analytic and marketing solutions, 58 are related to script and image CDNs, and the rest are related to various other solutions (including GitHub open-source projects).

TABLE III: Total numbers of visitors to the vulnerable websites in November 2024.

Exposed location	# of total visitors
URL	75.31M
Cookie	777.71K
Storage	16.68M

Our investigation reveals the privacy risks posed by the adversary attempting to exfiltrate KakaoTalk OAuth tokens due to insecure practices in deploying the KakaoTalk identity-providing service.

C. Inferring Location

Tinder offers a geosocial feature that allows users to view the profile images of other Tinder users located within a user-specified distance. The minimum user-specified distance is 1 mile, offering a coarse proximity estimate within a 1-mile radius, protecting the location privacy of their users. To enable this feature, Tinder collects the latitude and longitude of each user's mobile device. Upon each user request of searching for dating partners, Tinder retrieves the profiles of nearby users. Each profile includes the user ID, name, birth date, profile image URL, gender, occupation, education, height, drinking/smoking status, hobbies, vaccination status, and the *approximate distance* to this profile owner's mobile device.

Attack method. We present a location inference attack that exploits the functionalities above to approximate the physical location of a target Tinder user. Specifically, given the profile image of a Tinder user, the adversary queries nearby dating partners on Tinder and retrieves their profile images while varying the GPS location of the adversary's mobile device. By strategically changing the GPS locations in multiple queries until a retrieved profile image matches the target profile image, the adversary infers the physical location of the target user.

We note that this GPS location is provided by a client-side mobile device. The adversary is thus able to forge this location by using fake GPS apps [43] or modifying client requests for reporting latitude and longitude data to the Tinder servers. Moreover, Tinder supports the browser-supported services, allowing Tinder users to update their locations via desktop browsers [44]. Accordingly, this enables the adversary to forge their location using browser developer tools, such as the sensor overriding feature in Chrome [45]. We leveraged this particular web channel to vary the adversary's location.

Location inferring. We present a novel location inference algorithm that reduces the target area from an initial 1,600-meter by 1,600-meter region [46] to a circular area with an average radius of 324 meters. Our algorithm begins with four rectangular points defining an initial area to search for target users. We used this initial area large enough to encompass an entire city. The algorithm then divides this rectangular area into a grid of 2 miles × 2 miles cells and then queries each grid cell using its center as the initial search location. When there exist grid cells that contain Tinder users located within 1 mile with facial images similar to the target profile image, the adversary performs more precise location narrowing by conducting location inferring in Algorithm 1.

Algorithm 1: Inferring the location of a target user.

```
1  $P_{init} \leftarrow$ Given grid point
2  $\epsilon \leftarrow$ Boundary threshold
3  function ComputeInferredPosition $(P_{init}, \epsilon)$:
4      $P_i \leftarrow P_{init}$
5      $R_i$.low $\leftarrow 0$, $R_i$.high $\leftarrow$ 4 miles over
         $i \in \{East, West, South, North\}$
6      while $\exists_i$, $R_i$.high $-$ $R_i$.low $> \epsilon$ do
7          for $i \in \{East, West, South, North\}$ do
8              mid $\leftarrow (R_i$.low$+R_i$.high$)/2$
9              if $\| (P_{init}$ + mid in $i) - P_i \| < 1$
                 mile then
10                 $P_i \leftarrow$ MovePosition $(P_{init}, -$mid$, i)$
11             $P_i \leftarrow$ MovePosition $(P_{init}$, mid$, i)$
12             if Distance $(P_{\text{new}}$, target$) > 1$
                 mile then
13                 $R_i$.high $\leftarrow$ mid
14             else
15                 $R_i$.low $\leftarrow$ mid
16     return $\{(R_{East}+R_{West})/2,\ (R_{North}+R_{South})/2\}$
```

Algorithm 1 is designed to compute four cardinal points and report their center as an inferred target location. For each cardinal point, the algorithm computes a range in which this cardinal point exists and decreases this range by conducting binary search based on the query response from Tinder.

Note that the adversary is able to check whether a target individual in the retrieved profiles is located within 1 mile or 2 miles from a GPS location that the adversary chooses, which serves as a distance oracle. The algorithm exploits this oracle feedback to narrow the boundary for each cardinal point until it reaches $epsilon$ (i.e., 9 meters).

Specifically, given an initial GPS location P_{init} (i.e., the grid center), the algorithm begins by setting an initial bounding range of 0 to 6,440 meters (i.e., 4 miles) for each cardinal direction (Ln 5). The algorithm iterates the following logic for each cardinal direction (i.e., east, west, south, and north) until this boundary range becomes lower than $epsilon$ (Lns 6–7). It calculates the median distance within this range (Ln 8) and derives a new location by adding this median distance to the initial point in the current cardinal direction (Ln 11). The algorithm then queries the distance oracle to determine whether the target user is within a 1-mile or 2-mile radius (Ln 12). If the target user is outside a 1-mile radius, the upper boundary of the range is updated to the median distance. Otherwise, if the target user is within a 1-mile radius, the lower boundary is updated to the median distance (Lns 13–15). The algorithm repeats this process until this boundary range becomes lower than $epsilon$.

One notable tweak in the algorithm involves the adversary moving their vetting point more than 1 mile from the previous location. This adjustment is necessary due to Tinder's internal policy of only updating nearby users when the user's location changes by over 1 mile. Consequently, Lns 9–10 update the adversary's position by 1 mile in the opposite direction from the previous location, making that the distance oracle returns up-to-date results.

Finally, for each cardinal direction, the algorithm calculates a boundary point by adding the median distance of the corresponding boundary range to the initial point. It then compiles these four boundary points and reports their center as the inferred location (Ln 16).

In previous studies, Carman *et al.* [47] proposed a location attack that calculates boundary points by moving GPS points within a precomputed Tinder grid. However, they reported significant difficulties due to the significantly slow process of precomputing grids from Tinder server responses. Heaton [46] introduced a boundary inference method that determines grid snapping boundaries by collecting location values at intervals of 0.01°, approximating a grid size of 1 mile × 1 mile. To evaluate the efficacy of our approach, we compared it to the attack method proposed by Heaton [46], measuring distance error margins over varying queries. For this attack, we adjusted the adversary's moving increments to 0.01°, 0.02°, 0.03°, and 0.06°, as the method necessitates identifying the entire boundary to accurately determine the center location.

Figure 5 depicts the performance of our proposed location inference attack as the number of queries to Tinder varies. For this evaluation, we sampled 10 random locations and reported the average distances between the inferred locations and their corresponding true locations. For each query budget, the queries were evenly distributed across the computations of the four boundary ranges. As illustrated in the figure, our approach requires only 12 queries to accurately infer the target's location, with an average error of 385 meters. Additionally, with 40 queries, the average error is 324 meters.

In contrast, the prior method [46] necessitates at least 676 queries to achieve a comparable average error of 371 meters. Consecutive queries require a time interval due to Tinder's query rate limit since queries made without this interval are rejected. We observed that this interval is approximately 60 seconds, meaning each query should be fired 60 seconds apart to perform location inference. Therefore, increasing the number of queries introduces significant time overheads. On average, our method completes the attack in 18 minutes (12 queries), outperforming the prior approach, which requires 900 minutes (676 queries) per target user. These experimental results emphasize the importance of minimizing the number of queries to conduct practical location inferring attacks.

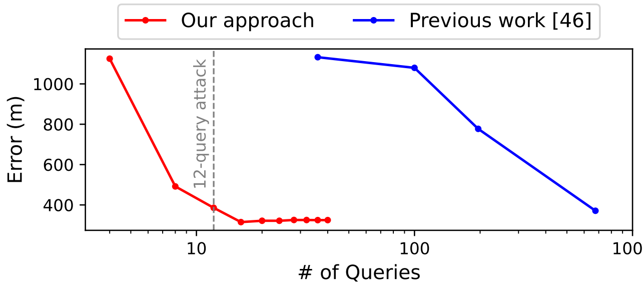


Fig. 5: Distance errors between the inferred and ground-truth locations across 10 sampled points.

V. Chaining Privacy Attacks

We test the feasibility of chaining the presented privacy attacks (§IV), demonstrating the privacy risks of using multiple messaging apps. We present three end-to-end attacks: (1) deanonymizing messaging app users (§V-A), (2) tracking the locations of untargeted individuals (§V-B), and (3) tracking the locations of targeted individuals (§V-C).

A. Revealing Anonymity

The adversary carries out the privacy attack (§IV-A) against Telegram, KakaoTalk, WhatsApp, and Signal by querying randomly sampled phone numbers and linking the retrieved profile information associated with the same phone number across two contact discovery services. By leveraging a dominant messaging app, the attacker is able to uncover the profiles of anonymous users registered in other messaging platforms.

Previous studies have reported that 56% of Telegram users have used pseudonyms and default profile images rather than personal photographs [13]. In contrast, 73% users in S. Korea rely on KakaoTalk for daily communication, using their real names for their profile [48], [49]. This behavioral difference enables the adversary to query anonymous names on one platform through the contact discovery service of another, thereby retrieving their real-name profiles.

Attack method. We evaluated the feasibility of this profile information linking attack. Specifically, we paired each of Telegram, WhatsApp, and Signal with KakaoTalk and identified phone numbers registered on both platforms using the same phone number. For each pair (i.e., Telegram–KakaoTalk, WhatsApp–KakaoTalk, and Signal–KakaoTalk), we queried 1,000 randomly sampled phone numbers through the corresponding contact discovery services to collect profiles. We then identified accounts using anonymous names, defined as names composed only of special characters or alphabetic initials. Finally, we cross-referenced these anonymous profiles with the other platform in each pair to determine whether the corresponding account has a different real-name-like name.

Telegram. We randomly sampled 1,000 phone numbers and added them to our Telegram contact list. This resulted in 88 users being successfully added, of which 40 (45.45%) used anonymous names. Among these 40 anonymous users on Telegram, we were able to retrieve the profile information, including their names in Korean, of 22 users on KakaoTalk. Among the 88 successfully added phone numbers, 73 (82.95%) were also registered on KakaoTalk. Using these 73 users, we additionally retrieved the names of 11 Telegram users whose names were anonymous on KakaoTalk. Overall, 70 out of 88 users were identified via cross-referencing, achieving a success rate of 79.55%.

WhatsApp. Among the 1,000 sampled phone numbers, we retrieved 42 user profiles (4.2%) from WhatsApp. Notably, WhatsApp does not reveal user-set names unless the account holders explicitly grant access. However, 37 of the 42 identified users (88.09%) were also registered on KakaoTalk, and 30 of their KakaoTalk profiles displayed registered names.

Signal. Out of the 1,000 sampled phone numbers, five users were registered on Signal. Although Signal does not disclose profile images or names without the account holder's consent, four of these users (80%) were also registered on KakaoTalk, where their names are visible.

Table IV summarizes our experimental results, and Appendix XI-B provides further details on the user intersections across platforms. These results demonstrate notable differences in user behavior between KakaoTalk and Telegram. Telegram exhibited a lower prevalence of real names (54.5%) compared to KakaoTalk (80.8%), indicating a stronger tendency for anonymity on Telegram and a preference for real names on KakaoTalk. We also note that although WhatsApp and Signal restrict the disclosure of user names, this defense becomes ineffective due to our chaining attack with KakaoTalk.

These experimental results also suggest that messaging app users who seek anonymity on only one platform are at risk of having their identities revealed. This risk is further exacerbated by the usage pattern of these messaging apps in S. Korea; a large proportion of users adopt anonymous names on Telegram while 94% of the entire South Korean population are registered on KakaoTalk and tend to use their legal names [48], [49].

TABLE IV: Experimental results on the numbers of accounts with real names identified through cross-referencing Telegram, WhatsApp and Signal with KakaoTalk (KT).

Platform (A)	A Reg. (KT Reg.)	Identified A→KT	Identified KT→A	Total Identified
Telegram	88 (73)	11/14	22/40	70 (79.55%)
WhatsApp	42 (37)	N/A	30/42	30 (71.42%)
Signal	5 (4)	N/A	4/5	4 (80%)

B. *Untargeted Individual Tracking*

For this end-to-end attack, the adversary chains two privacy attacks: one that extracts untargeted phone-profile pairs (§IV-A) and another that infers the location trajectories of the individuals identified (§IV-C). In this attack scenario, the adversary aims to track untargeted individuals residing in a specific city. Without prior information about these users, the attacker begins by retrieving profile images associated with randomly sampled phone numbers through the contact discovery service of KakaoTalk. Using these retrieved profile images, the attacker then tracks the users of these profile images on Tinder by exploiting its geosocial feature.

For this chaining attack to succeed, the victims should be registered on both Tinder and KakaoTalk. Given KakaoTalk's dominant market share of 94% in S. Korea, it is highly likely that most Tinder users in the region also use KakaoTalk.

To demonstrate the feasibility of chaining these attacks, we conducted end-to-end attack evaluations by tracking the locations of two of the authors.

Attack method. We selected City A as the target search area, which is set to a crowded college town in the target users' locations. City A is used here to comply with the anonymous submission requirement. To simulate the attack scenario, we registered two phone numbers for the authors as Tinder users and moved the two mobile devices with these numbers. For these registered accounts, we used distinct facial profile images on KakaoTalk and Tinder to evaluate the effectiveness of the attack across platforms. We limited the search area to a 8 miles × 8 miles square (2 miles × 2 miles grids) to avoid excessive queries to Tinder.

Extracting phone-profile pairs. We randomly sampled 5,000 phone numbers from a pool of properly formatted phone numbers. Using the contact discovery of KakaoTalk using automated contact synchronization, we retrieved profile images and names associated with these phone numbers. From this process, we identified 3,738 valid numbers with corresponding profiles, and were able to obtain 956 face image embeddings, including the images of the two authors. We then tracked the locations of these two authors using Tinder. We processed only the embedding vectors of the retrieved profile images and stored no profile images.

Image embedding. To compute embedding vectors for the given profile images, we used the RetinaFace and FaceNet512 models. The RetinaFace model [50] was used to detect the facial region within a given profile image, which was then resized to a resolution of 160 × 160. This resized image was subsequently fed into the FaceNet512 model [51], [52], generating a 512-dimensional embedding vector. The FaceNet512 model uses the Inception-ResNet-v1 [53] architecture. It has 23 million parameters and achieved 98.4% accuracy on the LFW dataset [52], [54].

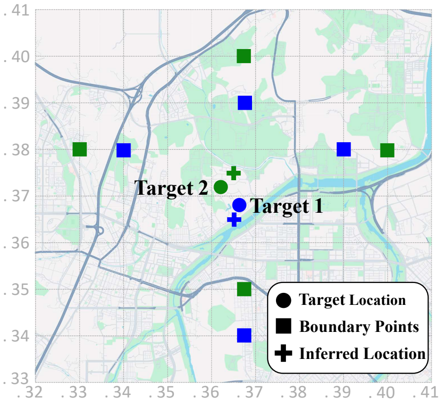


Fig. 6: Inferred location of two targets

We did not retrain or fine-tune these pretrained models, as using those was sufficient to distinguish our targets from other profile images, demonstrating the practical risk posed by the adversary leveraging off-the-shelf models.

Inferring locations. We inferred the locations of the two authors residing in City A by examining Tinder profiles that matched the embeddings of profile images obtained from KakaoTalk. To search for Tinder users in the target area, we divided the search area into 2 miles × 2 miles grids. At each grid point, we sent a query to retrieve Tinder users located within 1-mile. When a user profile was retrieved, we extracted the image embedding from the corresponding profile image. We then fed the profile image into the Facenet512 model to compute its embedding. Finally, we compared all the image embeddings from KakaoTalk and identified users whose cosine similarity exceeded 0.75, indicating a match.

Figure 6 illustrates the inferred locations of the two authors in City A. The square markers represent the boundaries where the distance to the target changes from 1-mile to 2-mile, the circle markers indicate the actual locations of the targets, and the cross markers represent the inferred locations. The localization errors between the actual and inferred positions were 336 meters and 418 meters for the two targets, respectively. As the figure shows, our simulated adversary was able to accurately infer the targets' locations.

Attack cost. For the phone-profile pair extraction, we used 5,000 numbers to register the target numbers to KakaoTalk. For the image recognition and target profile search, we used two Tinder accounts, it took about 22 minutes and 15 location updating queries to Tinder to identify the target profile. Finally, for further location inference via Algorithm 1, it took 40 location-updating queries over 16 minutes. In total, we spent about 52 minutes and $0.8 to launch the entire attack campaign. We argue that this is a significantly low-cost attack and feasible for real-world scenarios. When compared to previous studies [46], [47], our approach demonstrates reasonable cost-efficiency, highlighting the practicality and potential impact of this attack in real-world scenarios.

C. Targeted Individual Tracking

We present the third chaining attack that links two privacy attacks: one that extracts user information from a KakaoTalk OAuth token (§IV-B), and another that infers the target's location with Tinder (§IV-C). In this attack scenario, we assume that the victim is already signed in to a vulnerable website that inadvertently exposes their OAuth access token.

Attack method. We created an author profile on the Interpark website [55], which supports authentication via KakaoTalk identities. This website requires user consent to access profile information, including a nickname and a profile image. The website has gained large traction, with 20M registered users. However, we observed that the website exposes its OAuth access tokens in URLs. We also signed up for Tinder using one author's profile but with a different image from the one used for the KakaoTalk profile. Additionally, we assume the attacker already knows the target resides in City A. However, the adversary is able to initiate this attack across multiple cities to expand surveillance coverage.

Extracting profiles using OAuth tokens. Using the leaked OAuth tokens for one author, we accessed the corresponding KakaoTalk profile. By sending an API request with the leaked token, we extracted the target author's user information, including the phone number, profile image URL, and name. We converted the profile image into an embedding vector to facilitate further image matching processes.

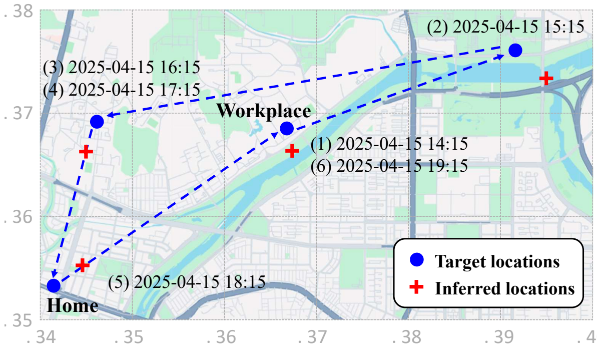


Fig. 7: Trajectory tracking of a single target

Trajectory tracking. Figure 7 depicts the inferred trajectory of a moving target. The blue circle markers represent the actual locations, while the red cross markers denote the inferred locations. The average error between the actual and inferred locations was approximately 335 meters. This result demonstrates that the trajectory tracking mechanism using Tinder is effective not only in accurately pinpointing a target's location but also in reliably tracking their trajectories. This capability poses a significant privacy threat, potentially exposing the home and workplace locations of target individuals [56]. We further conducted an experiment in City B to demonstrate that the same approach can be applied across different regions. The target's location was successfully tracked with an average error of 407 m (see Appendix XI-C for details).

Attack cost. We tracked the target every 90 minutes and kept an 80~100 seconds interval between each movement of the adversary. In total, we sent 84 location-updating queries over 9 hours to complete the entire attack process.

VI. Mitigation

Query throttling. A well-known approach to mitigating the misuse of contact discovery services is to enforce a query limit for each user. Previous studies [12], [13], [57] have proposed various query-throttling strategies, which impose restrictions on the number of requests. Similar to Telegram, we recommend that KakaoTalk enforce a daily limit of 100 phone number registrations. This restriction leads to requiring 180 days in attempting to register 18,000 phone numbers.

However, enforcing a query budget may disrupt the service when users attempt to sync address books with a large number of registered contacts. For instance, enforcing this limit would require 150 days for a user with 15,000 phone numbers to sync their address book with KakaoTalk, potentially undermining user experiences. In contrast, Telegram imposes a query limit even during the address book syncing process. User accounts exceeding their query budget are subjected to additional verification steps, which require the target users to add the querying account to their contact lists.

We contend that enforcing a strict query limit does not eliminate the threat. An attacker is still able to afford $1 to retrieve profile information by leveraging 10 accounts with different phone numbers. The existence of adversaries using 20 to 100 accounts in previous studies [13] demonstrates that this remains a practical threat. Another effective approach is to implement mutual contacts by default [12], [13], [57], which requires bidirectional agreement before a phone number is registered as a contact. However, both KakaoTalk and Telegram only provide an opt-out option for this feature in their settings rather than enabling it by default.

Leveraging social circles. We propose a lightweight detection method for identifying brute-force contact discovery attempts targeting random phone numbers. Our key idea is to exploit the structural difference in social relationships between benign and malicious address book syncing behaviors. Legitimate users typically query phone numbers belonging to individuals within their social circles, where the queried numbers are likely connected by friendship edges. In contrast, adversarial queries that exploit random phone numbers, making intra-query friendships rare. Our method leverages this distinction.

Given a set of queried phone numbers from a user via address book syncing, the defender samples two non-overlapping groups of size N and checks whether at least one friendship exists between them. If no such relationship is found, the contact discovery queries with the queried numbers are all rejected; otherwise, contact information is returned.

Assume a user selects a pair of individuals from their address book, and let p denote the probability that this pair has a friend relationship. Based on this, we compute p_{fr}, the probability that two randomly selected groups of N people from the user's address book share at least one friendship link.

Given that there are $N \times N$ possible inter-group pairs between two groups of size N, the probability that none of these pairs are connected as friends is $(1-p)^{N \times N}$. Therefore, the probability that at least one friendship exists between the two groups is the negation of this probability, as Equation 1 shows.

$$p_{fr} = 1 - (1-p)^{N \times N}. \tag{1}$$

We analyze the efficacy of this test across varying values of p and N (Figure 8). When $N = 25$ and $p > 1\%$, the test succeeds with near certainty (p_{fr} about 99.8%), accurately accepting benign queries. Prior work on social graphs estimates p to be approximately 24%, averaged over four popular social networks, each involving an average of 2.84 million users [58].

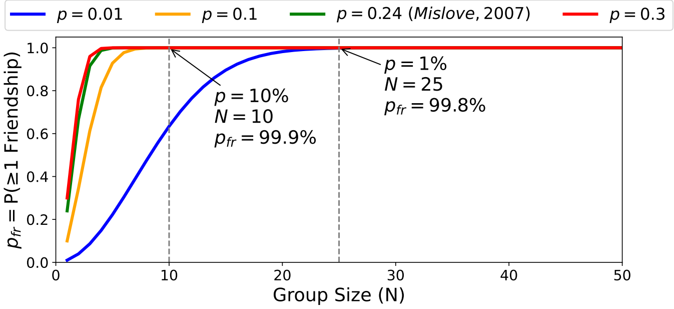


Fig. 8: Probability that at least one friendship exists between randomly sampled groups, as a function of group size (N) and friendship probability (p).

For adversarial queries composed of random phone numbers, p drops significantly. Assuming a KakaoTalk user base of 45 million and an average of 200 friends per user—consistent with prior studies on social network connectivity [59]–[61], the probability of observing any intra-group friendship among two random sets of 25 phone numbers falls below 0.278%.

While our method requires $O(N^2)$ friendship lookups, we argue this is justified to effectively mitigate large-scale contact discovery abuse. A prior study demonstrated the feasibility of subgraph computations, such as triangle counting, on large-scale social graphs [62], achieving an average of 0.33 seconds per million edges. Based on these results, our defense mechanism—which analyzes only incoming queries instead of the entire graph—is expected to be practical for real-world deployment. Moreover, the defender can tune N based on the underlying friendship probability p, as shown in Figure 8. For instance, when $p = 10\%$, setting $N = 10$ yields $p_{fr} = 0.999$, ensuring near-certain detection of legitimate address book synchronization.

Reporting non-deterministic distances. A widely known technique for mitigating trajectory tracking threats is grid snapping [56], [63], [64]. This method obscures a user's location by mapping it to the center point of a predefined grid. Tinder has deployed grid snapping, which maps users' locations to the center of server-defined grids [63]. However, if the grid size is sufficiently small and the grid boundaries are known, an attacker can deduce the mapping points of each user and infer their location with an error margin of less than half the grid size.

Our location inference algorithm identifies the target grid and its center where target users reside, uncovering Tinder's internal method for grid snapping; we estimate that Tinder uses a grid size of approximately 0.01° (about 900m × 1100m in S. Korea). Our location inferring attack is able to accurately approximate this grid center.

One approach to mitigating the proposed attack is to increase the grid size. Increasing the grid size introduces a higher margin of error for attackers, thereby complicating location-based tracking. While a larger grid size reduces the precision of inferred locations, it undermines the functionality of finding nearby users. We contend that this trade-off justifies increasing the grid size to enhance user privacy.

Tinder provides user location information measured in n miles, enabling grid size adjustments just before the reported value changes. We propose increasing the distance between grid center points to just under 2 miles, given that the minimum reported user distance is 1 mile. In S.Korea, the current grid size is approximately 900m × 1100m, with a maximum distance of 1,421 meters between neighboring center points and a maximum error margin of 711 meters. Doubling this distance expands the grid size to 0.02°. This adjustment introduces a displacement of 1,421 meters, which is sufficient to impede precise location tracking while maintaining the functionality of location-based services.

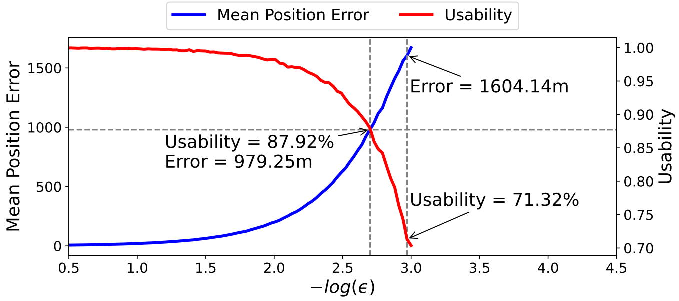


Fig. 9: Tradeoff between location privacy and usability.

We also explored an alternative defense mechanism based on differential privacy. Andres *et al.* [65] proposed a method for injecting differentially private noise into user coordinates to protect location privacy. We adapt this approach to the Tinder setting by adding noise to the distances to other Tinder users.

To evaluate its effectiveness, we simulated a scenario with 10 Tinder users located within a 2-mile radius, including a designated target. We applied noise to the distance values using the differential privacy mechanism, varying the privacy budget ϵ, and measured how many users would remain observable within a 1-mile query radius, representing utility. Specifically, we applied noise to all user coordinates and measured utility by calculating the difference in the proportion of observable users, with and without differential privacy, from the perspective of the target user. In addition, we measured the distance estimation error of our attack on the target, thereby capturing the trade-off between privacy and utility.

Figure 9 presents the results. We varied the privacy budget (ϵ) from 0.001 to 0.3; a lower ϵ provides stronger privacy guarantees. From our results, we observe that the number of observable users can be effectively constrained when the privacy parameter ϵ is set to 0.002 or lower.

OAuth access tokens. To mitigate the security risks posed by adversaries abusing leaked access tokens, IdPs like KakaoTalk may adopt the mutual TLS protocol [66] to validate whether API requests using OAuth access tokens originate from legitimate service providers, rather than attackers with leaked tokens. Another effective application-layer defense is implementing the Demonstrating Proof-of-Possession (DPoP) protocol [67]. DPoP is an authentication protocol that binds access tokens to the client. When sending a request to the server, the client includes a JSON Web Token containing its ID, public key, and other metadata, signed with the client's private key. Upon receiving the request, the server verifies the client's possession of the private key, ensuring token integrity and preventing replay attacks.

VII. Discussion and Lessons

Cross-platform privacy risks. Mobile messengers face persistent privacy and security challenges, including contact discovery attacks [12], [13], [48], [49], token exposure [68], [69], and location-based privacy violations [47], [56], [63], [70]. Our concerns lie in the use of phone numbers and facial images as linking identifiers across multiple platforms. Gupta *et al.* [57] demonstrated how phone numbers serve as linking keys between mobile apps. Our findings extend this observation, revealing that profile images, often perceived as non-sensitive, can also act as linking keys.

Bearer tokens. Due to their simplicity and ease of integration, many organizations, including X, GitHub, and KakaoTalk, have adopted bearer tokens as an authentication mechanism. However, by design, bearer tokens grant access to any party that possesses them; thus, if a token is leaked, its security is immediately compromised. Thus, the IETF has recommended implementing additional measures, such as DPoP [67].

Adversary scalability. Phone numbers serve as a crucial resource for the proposed attacks. The cost of executing these attacks is directly linked to the expense of acquiring phone numbers. If applications impose strict limitations on the number of user requests per phone number, attackers may attempt to bypass these restrictions using a large pool of phone numbers. For instance, KakaoTalk requires a valid phone number for user authorization verification. As a result, temporary or internet-based phone numbers (IP numbers) are generally ineffective for KakaoTalk registrations. Nevertheless, the attacks may leverage low-cost mobile carriers, which provide phone numbers for as little as $0.10 per month. These numbers are tied to real user identities and satisfy all authorization requirements.

VIII. Related Work

Contact discovery attacks. Previous research has investigated privacy risks associated with contact discovery services in messaging apps. Cheng *et al.* [12] demonstrated how address book matching features in messengers like WeChat can be exploited for user profiling and large-scale identity revelation. Similarly, Kim *et al.* [48], [49] conducted an in-depth analysis of KakaoTalk, uncovering attack methods for extracting profile information through automated contact synchronization. Hagen *et al.* [13] analyzed three popular messengers and revealed that large-scale crawling attacks targeting the contact discovery services of these platforms were feasible. A unique aspect of our investigative study is its focus on testing the adversary's capability to chain privacy attacks using profile images and phone numbers, further demonstrating the feasibility of revealing user identities and inferring location trajectories.

Location-based services. Caprolu *et al.* [70] analyzed Telegram's "People Nearby" feature, reverse-engineered its distance calculation algorithm, and demonstrated that the actual location privacy offered is consistently lower than the levels claimed by Telegram. Dhondt *et al.* [63] systematically examined 15 LBS applications, identifying API traffic leaks that enable adversaries to deduce precise user locations through trilateration. Li *et al.* [56] exposed vulnerabilities in platforms such as WeChat, Momo, and Skout, where adversaries could achieve high-accuracy tracking and reconstruct a target's top five frequently visited locations. Carman *et al.* [47] revisited prior attacks on Tinder demonstrating that advanced defenses like grid snapping remain vulnerable to precise localization attacks using more computational resources.

Token security. OAuth tokens are a key of modern SSO authentication, and their management has been extensively studied. Sun *et al.* [68] analyzed vulnerabilities in OAuth-based SSO implementations, highlighting how design flaws and poor practices by OAuth IdPs lead to unauthorized access to victims' private information. Similarly, Farooqi *et al.* [69] uncovered large-scale exploitation of Facebook OAuth tokens in collusion networks, where attackers exploit weak application security settings to retrieve and abuse access tokens for reputation manipulation. Zhang *et al.* [22] examined the exposure of sensitive keys and tokens on client-side platforms in mobile and web ecosystems, such as the leakage of "AppSecret" keys in WeChat mini-programs.

IX. Conclusion

In this paper, we present the first investigative study on connecting partially available private information across different messaging apps. In particular, we highlight practical privacy threats posed by permissive contact discovery services, which allow adversaries to extract profile images and names using forged phone numbers. These pieces of identity information are then used as keys to link private information across different messaging apps.

To demonstrate the threats, we evaluate three privacy attacks on the popular messaging apps in S. Korea and propose three concrete end-to-end attacks that chain these privacy attacks to either track the locations of messaging app users or reveal the users' identities. We also suggest novel mitigation strategies against the presented attacks. Lastly, our findings in S. Korea highlight the dire need to secure contact discovery and location-based functionalities, which also leave lessons for other countries dominated by single messaging apps.

X. Ethics Considerations

We obtained IRB approval for our investigative studies. To avoid disruption of normal service operations, we strictly limited our query rates. Specifically, location update queries to Tinder were limited to one per minute. For address book syncing, we adhered to the permitted usage policies of Telegram, KakaoTalk, WhatsApp, and Signal, and followed previous studies [48], [49], avoiding to add an excessive number of phone numbers. In total, we added no more than 120,000 phone numbers. For experimental purposes, we did not store any profile images. Instead, we temporarily stored the embeddings of profile images along with corresponding phone numbers on encrypted hard drives on two isolated and dedicated local machines. All data were permanently deleted following the completion of the experiments and analysis.

Additionally, we have responsibly disclosed our findings to both KakaoTalk and Tinder. KakaoTalk acknowledged our concerns, expressed appreciation for the report, and committed to deploying a fix. KakaoTalk has strengthened the restriction on the number of friend registrations and enhanced its policy for detecting abnormal patterns in friend additions and deletions. KakaoTalk also noted that they will implement additional safeguards to reduce the risk of mass friend registration within a short time frame. Furthermore, they plan to monitor adversarial behaviors continuously and introduce supplementary policies and rules as necessary. Tinder noted that users voluntarily share approximate location information through their app. We did not notify Telegram, WhatsApp, and Signal, as our findings confirm similar restrictions that Hagen *et al.* [13] have observed. Also, we notified 16 websites that expose their access tokens to third parties; two of them responded, and we confirmed that six have patched (including the two that responded) the reported vulnerable usage of access tokens.

Acknowledgment

We thank anonymous reviewers for their concrete feedback. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-NR076965).

References

[1] D. Lee, "WhatsApp discovers 'targeted' surveillance attack," https://www.bbc.com/news/technology-48262681, 2019, [Online].

[2] U. Minjae, "Open chat room personal information leaked "65,000 cases"," https://news.sbs.co.kr/news/endPage.do?news_id=N1007638572, SBS News, 2024, [Online].

[3] ——, "Open KakaoTalk room personal information 'more than 65,000 people' period... How did you get leaked?" https://news.sbs.co.kr/news/endPage.do?news_id=N1007648469, SBS News, 2024, [Online].

[4] L. Ceci, "Most popular global mobile messenger apps as of February 2025, based on number of monthly active users," https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/, Statista, 2025, [Online].

[5] Meta, "Two Billion Users — Connecting the World Privately," https://about.fb.com/news/2020/02/two-billion-users/, Meta, 2025, [Online].

[6] D. Kim, "The Crisis of the National App? The Number of People Not Using KakaoTalk Has Increased..." https://www.hankyung.com/article/202405134509g, 2024, [Online].

[7] A. Fleck, "Who Uses Telegram?" https://www.statista.com/chart/32923/share-of-respondents-who-regularly-use-telegram/, 2024, [Online].

[8] G. Sevilla, "WhatsApp outage results in a scramble for messaging alternatives," https://www.emarketer.com/content/whatsapp-outage-results-scramble-messaging-alternatives, 2022, [Online].

[9] WhatsApp, "100 million using WhatsApp across the United States," https://blog.whatsapp.com/100-million-using-whatsapp-across-the-united-states, 2024, [Online].

[10] J. H. Choi, "National Messenger 'KakaoTalk' is shaking... Generation 1020 "To Instagram and Telegram"," https://www.edaily.co.kr/News/Read?newsId=03145526638983056, edaily, 2024, [Online].

[11] Y. Xu, "A dynamic network perspective on the evolution of the use of multiple mobile instant messaging apps," *Communication Monographs*, vol. 90, no. 1, pp. 25–45, 2023.

[12] Y. Cheng, L. Ying, S. Jiao, P. Su, and D. Feng, "Bind your phone number with caution: Automated user profiling through address book matching on smartphone," in *Proceedings of the ACM SIGSAC Symposium on Information, Computer and Communications Security*, 2013.

[13] C. Hagen, C. Weinert, C. Sendne, A. Dmitrienko, and T. Schneider, "All the Numbers are US: Large-scale Abuse of Contact Discovery in Mobile Messengers," in *Proceedings of the Network and Distributed System Security Symposium*, 2021.

[14] WhatsApp, "About WhatsApp," https://www.whatsapp.com/about, 2025, [Online].

[15] Tinder, "About Tinder," https://www.tinderpressroom.com/about, 2024, [Online].

[16] D. Jeong, "No to self-satisfaction! The era of app-satisfaction! Dating app, everything," https://www.mk.co.kr/economy/view.php?sc=50000001&year=2024&no=63682, 2024, [Online].

[17] SK Telecom, "SK Telecom Phone Number Management," https://www.tworld.co.kr/web/support/service/fee-guide/27, 2024, [Online].

[18] KT, "KT Phone Number Management," https://help.kt.com/serviceinfo/ServiceJoinGuideL5.do, 2024, [Online].

[19] LG Uplus, "LG Uplus Phone Number Management," https://www.lguplus.com/support/service/use-guide/areacode, 2024, [Online].

[20] J. Pack, "Demand for 010 numbers will peak in 2032... It will decrease later due to population decline," https://ddaily.co.kr/page/view/2024100410494899037, 2024, [Online].

[21] D. Hardt, "The OAuth 2.0 Authorization Framework," https://www.rfc-editor.org/info/rfc6749, 2012, [Online].

[22] Y. Zhang, Y. Yang, and Z. Lin, "Don't leak your keys: Understanding, measuring, and exploiting the appsecret leaks in mini-programs," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2023.

[23] P. Siriwardena and P. Siriwardena, "OAuth 2.0 Token Binding," *Advanced API Security: OAuth 2.0 and Beyond*, pp. 243–255, 2020.

[24] Kakao, "Kakao Login," https://developers.kakao.com/docs/latest/en/kakaologin/rest-api#request-token-response-body, 2024, [Online].

[25] M. B. Jones and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage," https://www.rfc-editor.org/info/rfc6750, 2012, [Online].

[26] J. Gregory, "National Public Data breach publishes private data of 2.9B US citizens," https://www.ibm.com/think/news/national-public-data-breach-publishes-private-data-billions-us-citizens, 2024, [Online].

[27] T. Bui, S. Rao, M. Antikainen, and T. Aura, "XSS vulnerabilities in cloud-application add-ons," in *Proceedings of the ACM Asia Conference on Computer and Communications Security*, 2020.

[28] K. Drakonakis, S. Ioannidis, and J. Polakis, "The Cookie Hunter: Automated Black-box Auditing for Web Authentication and Authorization Flaws," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[29] Simple Design Ltd., "Android Auto Clicker," https://play.google.com/store/apps/details?id=autoclicker.clicker.autoclickerapp.autoclickerforgames, 2024, [Online].

[30] Talkatone, "Talkatone," https://www.talkatone.com/, 2024, [Online].

[31] TextMe, "TextMe," https://www.go-text.me/, 2024, [Online].

[32] SMS@MAN, "SMS@MAN," https://sms-man.com/, 2024, [Online].

[33] daisySMS, "daisysms," https://daisysms.com/, 2024, [Online].

[34] J. Gu, "Monthly telecommunications bill of 100 won... Altel phone cost-effective rate plan on the rise again," https://www.fnnews.com/news/202410021827451181, 2024, [Online].

[35] J. Kim, K. Kim, J. Cho, H. Kim, and S. Schrittwieser, "Hello, Facebook! Here Is the Stalkers' Paradise!: Design and Analysis of Enumeration Attack Using Phone Numbers on Facebook," in *International Conference on Information Security Practice and Experience*. Springer, 2017, pp. 663–677.

[36] Telegram, "tdlib/td," https://github.com/tdlib/td, 2024, [Online].

[37] ——, "Telegram Database Library," https://core.telegram.org/tdlib, 2024, [Online].

[38] KiwiTalk, "Kiwitalk," https://github.com/KiwiTalk/KiwiTalk, 2024, [Online].

[39] National Institute of Korean Language, "Vocabulary list for learning Korean," https://www.korean.go.kr/front/etcData/etcDataView.do?mn_id=46&etc_seq=71, 2003, [Online].

[40] Oxford, "The Oxford 3000 from the Oxford Advanced American Dictionary," https://www.oxfordlearnersdictionaries.com/wordlist/american_english/oxford3000/, 2019, [Online].

[41] D. Hardt, A. Parecki, and T. Lodderstedt, "The OAuth 2.1 Authorization Framework," https://datatracker.ietf.org/doc/html/draft-ietf-oauth-v2-1-11, 2024, [Online].

[42] T. Lodderstedt, J. Bradley, A. Labunets, and D. Fett, "OAuth 2.0 Security Best Current Practice," https://datatracker.ietf.org/doc/html/draft-ietf-oauth-security-topics, 2024, [Online].

[43] Lexa, "Fake GPS location," https://play.google.com/store/apps/details?id=com.lexa.fakegps, 2022, [Online].

[44] Tinder, "Tinder Supported Platforms and Devices," https://tinder.com/, 2025, [Online].

[45] S. E. Kayce Basques, "Sensors: Emulate device sensors," https://developer.chrome.com/docs/devtools/sensors, 2020, [Online].

[46] R. Heaton, "How Tinder keeps your exact location (a bit) private," https://robertheaton.com/2018/07/09/how-tinder-keeps-your-location-a-bit-private/, 2018, [Online].

[47] M. Carman and K.-K. R. Choo, "Tinder me softly–How safe are you really on tinder?" in *Proceedings of the International Conference on Security and Privacy in Communication Networks*. Springer, 2017.

[48] E. Kim, K. Park, H. Kim, and J. Song, "Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk," *computers & security*, vol. 52, pp. 267–275, 2015.

[49] ——, "I've Got Your Number: Harvesting users' personal data via contacts sync for the KakaoTalk messenger," in *Proceedings of the International Workshop on Information Security Applications*, 2015.

[50] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," *arXiv preprint arXiv:1905.00641*, 2019.

[51] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[52] S. I. Serengil and A. Ozpinar, "A Benchmark of Facial Recognition Pipelines and Co-Usability Performances of Modules," *Bilisim Teknolojileri Dergisi*, vol. 17, no. 2, pp. 95–107, 2024. [Online]. Available: https://dergipark.org.tr/en/pub/gazibtd/issue/84331/1399077

[53] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[54] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," University of Massachusetts, Amherst, Tech. Rep., 2007.

[55] Interpark Triple, "Interpark," https://interpark.com/, 2024, [Online].

[56] M. Li, H. Zhu, Z. Gao, S. Chen, L. Yu, S. Hu, and K. Ren, "All your location are belong to us: Breaking mobile social networks for automated user location tracking," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2014.

[57] S. Gupta, P. Gupta, M. Ahamad, and P. Kumaraguru, "Exploiting phone numbers and cross-application features in targeted mobile attacks," in *Proceedings of the Workshop on Security and Privacy in Smartphones and Mobile Devices*, 2016.

[58] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the ACM SIGCOMM conference on Internet measurement*, 2007.

[59] J. B. Walther, B. Van Der Heide, S.-Y. Kim, D. Westerman, and S. T. Tong, "The role of friends' appearance and behavior on evaluations of individuals on Facebook: Are we known by the company we keep?" *Human communication research*, vol. 34, no. 1, pp. 28–49, 2008.

[60] A. M. Gonal, L. Umadevi, and P. Sreedevi, "A comparative study on friends on social network sites (SNS) and friends in reality of adolescents," *International Journal of Pure and Applied Bioscience*, vol. 6, no. 5, pp. 625–630, 2018.

[61] F. Meng, H. Sun, J. Xie, C. Wang, J. Wu, and Y. Hu, "Preference for number of friends in online social networks," *Future Internet*, vol. 13, no. 9, p. 236, 2021.

[62] T. G. Kolda, A. Pinar, T. Plantenga, C. Seshadhri, and C. Task, "Counting triangles in massive graphs with MapReduce," *SIAM Journal on Scientific Computing*, vol. 36, no. 5, pp. S48–S77, 2014.

[63] K. Dhondt, V. Le Pochat, Y. Dimova, W. Joosen, and S. Volckaert, "Swipe Left for Identity Theft: An Analysis of User Data Privacy Risks on Location-based Dating Apps," in *Proceedings of the USENIX Security Symposium*, 2024.

[64] I. Polakis, G. Argyros, T. Petsios, S. Sivakorn, and A. D. Keromytis, "Where's wally? precise user discovery attacks in location proximity services," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[65] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2013.

[66] B. Campbell, J. Bradley, N. Sakimura, and T. Lodderstedt, "OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens," https://www.rfc-editor.org/info/rfc8705, 2020, [Online].

[67] D. Fett, B. Campbell, J. Bradley, T. Lodderstedt, M. B. Jones, and D. Waite, "OAuth 2.0 Demonstrating Proof of Possession (DPoP)," https://www.rfc-editor.org/info/rfc9449, 2023, [Online].

[68] S.-T. Sun and K. Beznosov, "The devil is in the (implementation) details: an empirical analysis of OAuth SSO systems," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2012.

[69] S. Farooqi, F. Zaffar, N. Leontiadis, and Z. Shafiq, "Measuring and mitigating oauth access token abuse by collusion networks," in *Proceedings of the Internet Measurement Conference*, 2017.

[70] M. Caprolu, S. Sciancalepore, A. Grigorov, V. Kolev, and G. Oligeri, "Watch Nearby! Privacy Analysis of the People Nearby Service of Telegram," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2024.

[71] R. Wightman, H. Touvron, and H. Jégou, "Resnet strikes back: An improved training procedure in timm," *arXiv preprint arXiv:2110.00476*, 2021.

[72] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

[73] PyTorch Image Models (timm), "Model card for mobilenetv3_large_100.ra_in1k," https://huggingface.co/timm/mobilenetv3_large_100.ra_in1k, 2023, [Online].

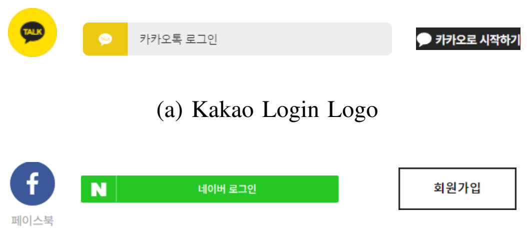
[74] G. G. Simpson, "Mammals and the nature of continents," *American Journal of Science*, vol. 241, no. 1, pp. 1–31, 1943.

XI. Appendix

A. Crawler

For efficient crawling, we implemented a crawler leveraging a logo classifier. The classifier is based on a pre-trained MobileNetV3 model [71]–[73] to extract embedding vectors from images. These embeddings are then classified using a multi-layer perceptron (MLP) network. The MLP network was trained on the logo image dataset that consists of 4,000 logo images collected from 2,000 websites. Figure 10 show the collected images of HTML elements with logo images.

To evaluate the performance of this classifier, we tested it on a dataset comprising 500 KakaoTalk login logo images and 500 non-KakaoTalk login logo images. The results, shown in Figure 11, demonstrate the overall performance. The ROC curve indicates that the classifier achieved an AUC of 0.996, reflecting a near-perfect ability to distinguish between KakaoTalk login logos and non-KakaoTalk login logos.



(a) Kakao Login Logo

(b) Non-Kakao Login Logo

Fig. 10: Logo image examples

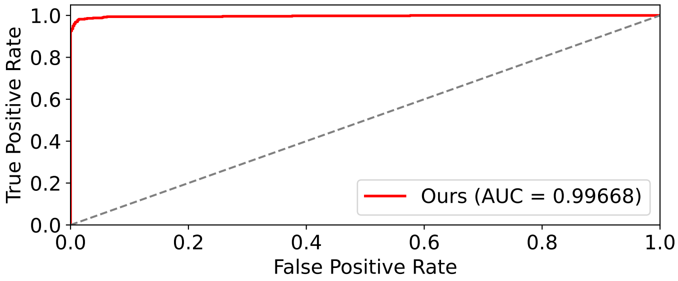


Fig. 11: Classifier ROC curve

B. Messaging app users on multiple platforms

Our proposed chaining attack for revealing user anonymity (§V-A) exploits a commonly observed usage pattern: users often seek anonymity on one platform while disclosing their real identities on another (i.e., KakaoTalk).

To quantify the extent of cross-platform account overlap, we analyzed user intersections among different messaging platforms. Specifically, we randomly sampled 1,000 phone numbers and added them to the contact lists of Telegram, WhatsApp, KakaoTalk, and Signal, respectively. For each platform, we measured (1) the number of registered users and (2) the number of users who were simultaneously registered on other platforms. Based on these metrics, we computed the overlap coefficient for each platform pair, defined as in the Equation 2.

$$Overlap(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)} \tag{2}$$

We used the overlap coefficient, also known as the Szymkiewicz-Simpson coefficient [74], to measure the similarity between two user sets because the number of registered users varies across messaging platforms (e.g., 720 users on KakaoTalk and 88 users on Telegram).

Figure 12 illustrates the overlap coefficient values between pairs of messaging platforms in the lower triangular matrix form. KakaoTalk, having the largest user base in S. Korea, shows a high degree of user intersection with all other platforms. Conversely, Telegram demonstrates low user intersection with other platforms, excluding KakaoTalk.

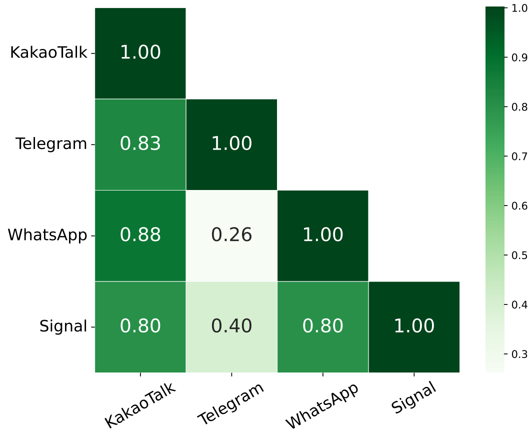


Fig. 12: Overlap coefficient Matrix

C. Trajectory tracking

We conducted an additional trajectory tracking experiment in City B (distinct from the City A setting) under challenging conditions. The target was located in a crowded subway station, approximately 150 km away from the attacker. The target's location was inferred at 30-minute intervals by controlling the attacker's Tinder account. We successfully estimated three target locations with errors of 280 m, 410 m, and 530 m, respectively. Figure 13 illustrates these results. This experiment demonstrates that our attack remains effective even under adverse conditions and across different urban environments.

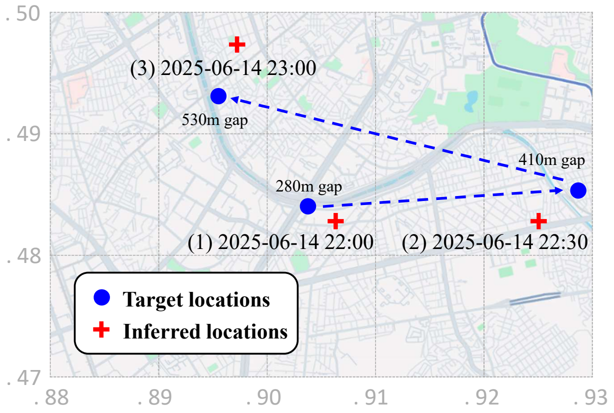


Fig. 13: Trajectory tracking under challenging conditions