I Can Tell Your Secrets: Inferring Privacy Attributes from Mini-app Interaction History in Super-apps

Yifeng Cai[1], Ziqi Zhang[2], Mengyu Yao[1], Junlin Liu[1], Xiaoke Zhao[3], Xinyi Fu[3], Ruoyu Li[3], Zhe Li[3], Xiangqun Chen[1], Yao Guo[1], and Ding Li[1]

[1]*MOE Key Lab of HCST (PKU), School of Computer Science, Peking University*
[2]*Department of Computer Science, University of Illinois Urbana-Champaign*
[3]*Ant Group*

Abstract

Super-apps have emerged as comprehensive platforms integrating various mini-apps to provide diverse services. While super-apps offer convenience and enriched functionality, they can introduce new privacy risks. This paper reveals a new privacy leakage source in super-apps: mini-app interaction history, including mini-app usage history (Mini-H) and operation history (Op-H). Mini-H refers to the history of mini-apps accessed by users, such as their frequency and categories. Op-H captures user interactions within mini-apps, including button clicks, bar drags, and image views. Super-apps can naturally collect these data without instrumentation due to the web-based feature of mini-apps. We identify these data types as novel and unexplored privacy risks through a literature review of 30 papers and an empirical analysis of 31 super-apps. We design a mini-app interaction history-oriented inference attack (THEFT), to exploit this new vulnerability. Using THEFT, the insider threats within the low-privilege business department of the super-app vendor acting as the adversary can achieve more than 95.5% accuracy in inferring privacy attributes of over 16.1% of users. THEFT only requires a small training dataset of 200 users from public breached databases on the Internet. We also engage with super-app vendors and a standards association to increase industry awareness and commitment to protect this data. Our contributions are significant in identifying overlooked privacy risks, demonstrating the effectiveness of a new attack, and influencing industry practices toward better privacy protection in the super-app ecosystem.

1 Introduction

Super-apps. With the advances in mobile computing, apps are becoming more and more powerful. Super-apps, such as AliPay [1], WeChat [8], and Careem [2], are comprehensive and one-stop applications that allow users to access various mini-apps easily. Mini-apps are similar to native apps, enabling super-apps to establish an ecosystem like Google Play and the Apple App Store. This design enriches super-apps' functionalities and offers great convenience to mobile users. For example, the food delivery service provider Ele.me [4] has greatly benefited from this paradigm: merchants can use mini-apps to sell foods through AliPay and WeChat directly. Users can browse products, share interests on social networks, and make purchases without leaving the super-app. Therefore, super-apps have significantly facilitated daily activities (e.g., transactions and transportation) from the built-in or third-party mini-apps. As shown in Figure 1 (a), on the homepages of super-apps, the mini-apps are crucial components and can be easily accessed (highlighted in the yellow box in Figure 1 (a)).

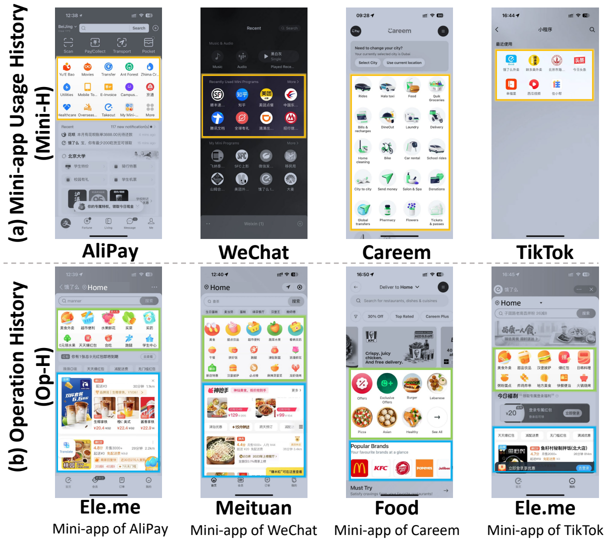

Figure 1: Samples of mini-app interaction history.

Blur Definition in Regulations. Similar to normal mobile apps, the data collection and storage of super-app is regulated by laws, such as the General Data Protection Regulation (GDPR) and the Personal Information Protection Law (PIPL) [12, 34, 35, 43]. The laws require mobile apps to provide users (data subjects) with transparent information on the collection and processing of personal data [9]. The regulations also require super-app vendors to store the data safely and prevent them from transferring the collected data to other parties without the consent of the users [21]. However, laws are general. In practice, *what kind of data should be protected depends on the academic community's and industry's research and standard agreement*.

New Leakage Source. This paper focuses on an underexplored aspect of data privacy in super-apps: *the mini-app interaction history*. The interactions include various user activities, including clicking buttons, and recently-accessed and preferred mini-apps. A department in the super-app vendor is motivated to use such data to infer user privacy to improve their service [70]. These interaction data are dangerous because *the super-app vendor can naturally collect such data from the released version of super-app due to the design philosophy of mini-apps.* The vendors do not need to modify the app (e.g., instrumentation) to obtain the interaction history data. This is because existing mini-app services are provided online and based on cloud servers according to the development guidance of popular super-apps [3, 5, 6]. Specifically, we study two typical types of mini-app interaction history: **the Mini-app Usage History (Mini-H)** and **the Operation History (Op-H)**. **Mini-H** includes the mini-apps that the user engaged with over a period of time, which offer clues about user preferences because the frequently accessed mini-app categories can reveal privacy attributes. When a user opens a super-app, the super-app displays the frequently accessed mini-apps on the main screen to provide a better user experience. The list of frequently used mini-apps is obtained by sending an HTTP request to the super-app vendor's server, so it is naturally obtained by the super-app provider. **Op-H** includes user interactions within the mini-app, such as button clicks, bar drags, and image views. Op-H can expose privacy attributes as well. For example, the interaction speed can indicate the user's age information: younger users can navigate and switch interfaces swiftly, whereas older users exhibit slower operation speeds and more repetitive clicks to locate desired information. Because all mini-apps are web-based applications, once the user clicks a button, super-apps need to send the name and ID of the clicked button to the server to request the mini-app's subsequent feedback [11]. Thus, the super-app vendor can naturally access Op-H from the request logs on the server.

People's Unawareness. We first study how much attention people (academic researchers and industrial practitioners) pay to this new privacy leakage. To perform a comprehensive study, we surveyed 30 papers published in top-tier conferences from 2019 and 31 real-world super-apps. We found that, to concretize laws like GDPR and PIPL, the community has converged on an extensive list of concrete data items considered sensitive. These include personal information and attributes (e.g., name, gender, age) and device information (e.g., IMEI, MAC address). The large number of protected data types shows the efforts of the community to preserve user privacy. However, we observed that *neither academia nor industry has fully recognized the seriousness of the leakage from mini-app interaction history*, let alone the laws and regulations. The literature review shows that none of the existing academic papers has discussed the privacy issues of the mini-app interaction history. The empirical study of super-apps reveals a parallel lack of awareness in the industry. Despite the widespread use of these apps, our analysis indicates that only one app recognized the potential privacy risks associated with Mini-H. This oversight is particularly concerning because Mini-H and Op-H can leak many types of user privacy attributes according to our study.

New Attack. To show the real threat to user privacy from the mini-app interaction history data, we designed a novel attack called THEFT, *mini-app interacTion History-oriEnted inFerence aTtack*. THEFT leverages Deep Neural Networks (DNNs) to infer users' privacy attributes from their mini-app interaction history. Unlike prior work [61] that relies on comprehensive OS-level data (e.g., cellular tower IDs and all HTTP requests), THEFT only needs mini-app interaction history, which is naturally collected by super-apps through their cloud-based services and deemed non-sensitive. This low requirement significantly lowers the bar for insider threats [54] as the adversaries to obtain minimal labeled data (e.g., from privacy breaches [42, 47, 59]), match mobile phone numbers, and train DNN models. Once trained, these models can infer sensitive privacy attributes of users at scale, posing severe threats to user privacy.

Results. We conduct a large-scale experiment on the internal data of AliPay, a real-world super-app with more than 1.3 billion users. Experimental results show that THEFT can achieve more than 95.5% accuracy in inferring privacy attributes of over 16.1% of more than 219K users with the training data from only 200 users, making THEFT a practical attack. Our attacks in AliPay itself mean the possible privacy leakage for more than 200 million users. This highlights the substantial risk of mini-app interaction history data. The finding demonstrates the practical implications of our threat model: current privacy measurements in super-apps are insufficient against advanced inference attacks.

Generalization. Although we mainly use AliPay as a representative super-app to study the privacy leakage of Mini-H and Op-H, our findings are generalizable to other super-apps. This is because super-apps share the same *interface design and code structure* across different platforms. All mini-apps use JavaScript and WebView to easily replicate in different super-apps [40, 63, 76]. Therefore, for different super-apps, mini-apps of different functionalities often have similar user interfaces and features. Figure 1 (b) displays four different food delivery mini-apps from four different super-apps. Green boxes highlight different subcategories that the mini-app can provide, while blue boxes show the list of stores. The content in both the green and blue boxes shares similar designs across super-apps. The server's interaction history data collected

Table 1: Focused topic and concerned privacy of the papers published in top-tier conferences from 2019 to 2024.

Paper	Conference	Focused Topic	Concerned Privacy
Dong et al. [20]	Sec 24	SDK collected hardware identifiers	Pers./Devi.Info.
Pan et al. [56]	Sec 24	Online automated privacy policy generators	Contacts, Location, Pers/Devi.Info., Microphone, Camera, Sensors
Khandelwal et al. [32]	Sec 24	Data safety sections	Location, Pers./Devi.Info.
Klein et al. [33]	CCS 23	Unlawful data processing in web apps	Contacts, Pers.Info.
Xiang et al. [71]	CCS 23	Completeness of privacy policy	Location, Pers.Info.
Wang et al. [65]	CCS 23	Hidden APIs in super-apps	Contacts, Location, Pers./Devi.Info.
Zhang et al. [78]	CCS 23	Leaked master key of mini-apps	Pers.Info.
Ferreira et al. [23]	SP 23	Pers.Info. data compliance in web apps	Location, Pers/Devi.Info.
Zhang et al. [77]	SP 23	Face verification system in apps	Camera
Xiao et al. [72]	Sec 23	Compliance of Apple privacy labels	Location, Pers./Devi.Info.
Nan et al. [50]	Sec 23	IoT collected data	Devi.Info., Sensors
Wang et al. [64]	Sec 23	APIs in cross platform super-apps	Location, Devi.Info., Microphone, Camera
Koch et al. [34]	Sec 23	Privacy consent dialogs	Location, Devi.Info.
Lyons et al. [45]	Sec 23	Logged personal data in Android	Location, Pers./Devi.Info.
Meng et al. [48]	NDSS 23	User-unresettable identifiers in Android	Devi.Info.
Jordan et al. [31]	NDSS 23	Verifiable accountless consumer requests	Contacts, Location, Pers./Devi.Info., Microphone, Camera, Sensors
Nguyen et al. [52]	CCS 22	Notice of third-Party tracking in apps	Pers./Devi.Info.
Li et al. [39]	CCS 22	Cross-user personal data over-delivery in apps	Pers.Info.
Wang et al. [67]	CCS 22	Location-based service in apps	Location
Yang et al. [73]	CCS 22	Cross mini-app request forgery	Location, Pers/Devi.Info., Microphone, Camera
Young et al. [74]	Sec 22	Policy violation of voice assistant apps	Microphone, Camera
Balash et al. [10]	Sec 22	Third-party app access for Google account	Pers.Info.
Zhang et al. [76]	Sec 22	APIs identity confusion in mini-apps	Pers.Info., Contacts
Diamantaris et al. [18]	CCS 21	Misuse sensors in apps	Location, Pers.Info., Microphone, Camera, Sensors
Bui et al. [13]	CCS 21	Consistency of data-usage purposes in apps	Location, Pers./Devi.Info.
Haney et al. [26]	Sec 21	Privacy implications	Microphone, Camera
Nguyen et al. [51]	Sec 21	Personal data compliance in apps	Location, Pers./Devi.Info.
Lu et al. [44]	CCS 20	Security risks of API flaws in mini-apps	Location, Pers/Devi.Info., Microphone, Camera
Zuo et al. [79]	SP 19	Cloud APIs in apps	Pers.Info.
Chen et al. [16]	SP 19	Hidden privacy settings in apps	Pers.Info.

from these mini-apps are also similar. Therefore, our attack and findings could apply to any super-apps.

Industrial Feedback. We notified 31 super-app vendors and one standards association about the privacy risks of mini-app interaction history. Four vendors agreed to modify the privacy statement of their apps, and the standards association committed to strengthening data protection. Furthermore, these vendors and the association also provided insightful feedback. This feedback ranged from acknowledgments of previously overlooked privacy risks to commitments to enhance data protection measurements. Specifically, developers intended to revise privacy policies and terms on potential dangers and recognize Mini-H and Op-H as private data. These responses highlight a growing awareness and proactive stance towards user data privacy in the industry. Overall, the feedback from the industry shows that the mini-app interaction history-oriented inference attack is practical in the real world.

We summarize our contributions as follows.

• We identify the unprotected user mini-app interaction history as a novel underexplored privacy risk in super-apps. We systematically reveal new privacy threats that have been overlooked in academia (through 30 top-tier papers) and industrial practice (through 31 real-world super-apps).

• We design a mini-app interaction history-oriented inference attack (THEFT), to exploit this privacy risk and demonstrate that the attack can effectively and accurately infer the users' privacy attributes.

• We provide our findings to the super-app vendors to bring their attention to this new privacy risk. We receive valuable feedback acknowledging our contribution to enhancing privacy measurements and the super-app ecosystem.

2 A Blind Spot in Mini-App Ecosystem

In this section, we present the motivation of this paper: people's unawareness of the security risk of mini-app interaction history. Contrary to the vast usage of mini-apps, we found that very few people/companies pay attention to the security risk. To comprehensively evaluate people's consciousness, we perform a thorough literature review in the academic community and an empirical study of industrial practice. Our goal is to present the attitude of both researchers and practitioners.

2.1 Perspective from Academia Community

Methodology. To cover the mainstream opinions of the academic community, we survey the papers published in top-tier security conferences which are included in CSRankings, including USENIX Security (Sec), IEEE S&P (SP), ACM CCS (CCS), and NDSS. We survey the papers published from 2019 to 2024 because 2019 is the emergence of super-apps [79]. Our literature review consists of three steps. First, we reviewed the 4,020 accepted papers from the four conferences and selected 43 papers whose titles relate to the privacy of mobile apps or devices. Then, we read the selected paper's abstract and introduction sections to filter out 13 papers that did not focus on privacy protection. Finally, we read the full text of the 30 selected papers to summarize the specific types of privacy data that the paper studied. To ensure reliability, four authors (two academic and two industry researchers) independently participated in the above processes, achieving a 99.52% agreement rate, which indicates a high inter-rater reliability (IRR).

Result. Over the past few years, there has been a significant focus on privacy protection. We display the papers and detailed privacy data types in Table 1. We find that *none* of the existing papers have studied privacy issues of the Mini-H and Op-H in super-apps. The privacy content that existing papers have focused on includes contacts, location, personal information (Pers.Info.), device information (Devi.Info.), and data generated by cameras, microphones, and sensors. Specifically, 22 papers concern personal information, such as phone numbers, ID numbers, gender, and age. 17 papers examine the protection of device information such as IMEI and MAC addresses, and 15 papers focus on system-level location information. Furthermore, increasing studies focus on broader aspects of privacy security. Nine and eight papers study the potential privacy leaks from camera and microphone data, respectively. Four papers study the security risks of sensor data, as the adversary can use these data to infer user privacy attributes. These studies underscore the multifaceted nature of privacy concerns on mobile devices and reflect ongoing efforts within the academic community to address these evolving challenges. Nevertheless, existing literature lacks a rigorous study on the potential privacy leakage of Mini-H and Op-H.

2.2 Practice in Industrial Companies

Methodology. To comprehensively analyze industrial companies' attitudes toward the types of private data, we study the most popular super-apps in the Google Play and Apple App Store. Our study includes two steps. First, we identified 31 different super-apps from recent research [64, 73, 76]. Second, we manually reviewed the privacy policies and terms of each super-app and summarized the types of privacy data in the privacy policies and terms. We regard the data types explicitly mentioned in the privacy policies and terms as the data the super-apps are concerned with. Consistent with the methodology in Section 2.1, the four authors independently analyzed the privacy policies and terms, achieving a high IRR with a 99.93% agreement rate.

Result. Table 2 displays the fine-grained categorization of private data that appear in the 31 super-apps' privacy policies and terms. We display twelve different types of privacy data, including location, contacts, camera, gallery, microphone, calendar, device information (Devi.), personal information (Pers.), payment details, search history, Mini-H and Op-H.

We found that the industry does not pay enough attention to Mini-H and Op-H. Among all super-apps, only one super-app (WeChat) mentions that it collects Mini-H in the privacy policies and terms. Mini-H is crucial for understanding users' behavior and preferences, as it includes the mini-apps the user prefers. However, most industrial companies do not consider Mini-H a primary privacy concern due to the lack of research on its privacy leakage. From a company's perspective, Mini-H consists of structural elements, like the types of mini-apps. Intuitively, these elements are predetermined and do not involve user input data. Similarly, companies consider Op-H non-invasive because it only captures limited structured behavioral data without explicitly requiring personal input.

Conversely, super-apps often have clear notifications on other common privacy data. Personal information and location data are commonly included in privacy policies and terms, highlighting their importance for app functionality. Device information is also frequently collected for security and optimization. Payment data helps the super-app to understand user preferences. Additionally, 22, 17, and 20 super-apps claim access to cameras, galleries, and microphones. Contacts and calendar data are collected by 20 and 10 social-based super-apps, respectively. Search history is widely used in 19 super-apps, raising attention among users for its sensitive content.

3 Threat Model

Given people's unawareness of the security risk of mini-app interaction history, the next part of this paper will reveal the dangers. In this section, we will first introduce the threat model in which the interaction data can be used to threaten users' privacy.

Scenario. We consider a leading global super-app vendor that consists of multiple departments, each with distinct roles and responsibilities. In this scenario, there are two types of departments: the high-privilege business department (D_{high_priv}), such as payment and risk control, and the low-privilege business department (D_{low_priv}), such as lifestyle or third party-collaborators. Both D_{high_priv} and D_{low_priv} strictly comply with privacy protection measures to ensure service security and regulatory compliance. D_{high_priv} has access to sensitive data, such as personal financial information, while D_{low_priv} can only access data explicitly classified as non-sensitive by the super-app vendor and should know minimal user privacy.

In this scenario, the insider threats [54] within the D_{low_priv} may act as the adversary. These insiders cannot access private data but can access mini-app interaction history because it is considered non-sensitive. For instance, employees in advertising or loan departments cannot directly access sensitive user attributes due to data-isolation policies. However, they can access Mini-H and Op-H to infer those attributes indirectly. Their motivation may include selling the inferred data for profit and gaining a competitive advantage by enhancing tar-

Table 2: The summary of the collected data types claimed in the privacy policies and terms of 31 super-apps.

No.	Super-app	Location	Contacts	Camera	Gallery	Microphone	Calendar	Devi.	Pers.	Payment	Search	Mini-H	Op-H
01	AliPay	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗
02	Taobao	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗
03	UC Browser	✓	✗	✓	✓	✗	✗	✓	✓	✗	✓	✗	✗
04	Gaode	✓	✓	✗	✗	✓	✗	✓	✓	✓	✓	✗	✗
05	DingTalk	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
06	Youku	✓	✗	✓	✗	✓	✗	✓	✓	✓	✓	✗	✗
07	WeChat	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✓	✗
08	WeCom	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
09	QQ	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗
10	QQ Music	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓	✗	✗
11	Tencent Video	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗
12	TikTok	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗
13	Toutiao	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗
14	Feishu	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✗	✗
15	Meituan	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗
16	Dianping	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗
17	Baidu	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
18	Baidu Map	✓	✓	✗	✗	✓	✗	✓	✓	✓	✓	✗	✗
19	iQiYi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
20	PinDuoDuo	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗
21	XiaoHongShu	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗
22	KuaiShou	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗
23	NetEase Cloud Music	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
24	JingDong	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
25	Suning	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗
26	Bilibili	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
27	Grab	✓	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗
28	Paytm	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗
29	Go-Jek	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗
30	UnionPay	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗
31	Air Asia	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
	Total	**31**	**20**	**22**	**17**	**20**	**10**	**29**	**30**	**25**	**19**	**1**	**0**

geted advertising or differential pricing strategy. Even when their mini-app does not hold explicit data (e.g., a user's assets status), relevant privacy attributes of unlabeled users can still be inferred by analyzing broader interaction patterns. As an example, identifying whether a user owns property or a vehicle helps loan department insiders tailor services more effectively. Meanwhile, since users rarely access the loan department's mini-app, the department cannot gather enough privacy attributes directly. Nonetheless, these attributes can still be inferred by analyzing the mini-app interaction history.

Adversary's Goal. The adversary's goal is to *accurately predict the privacy attributes of as many users as possible from the non-sensitive data*. The goal consists of twofold. First, the adversary aims to identify a user subset whose privacy attributes are strongly correlated with Mini-H and Op-H. These users are believed to have sufficient mini-app interaction history data to infer their privacy attributes. For example, one user might frequently use the fueling mini-apps, which could indicate that he/she owns a vehicle. For other users, if the Mini-H and Op-H cannot reveal their privacy attributes, the adversary may classify them as unknown rather than forcibly assign labels. This can avoid blind and inaccurate predictions. Second, for users in the subset, the adversary aims to infer their privacy attributes as accurately as possible. This is because privacy attributes are valuable but scarce. With such attributes, adversaries can design unfair market strategies (e.g., targeted advertising, price discrimination, or denial of services) for certain user groups. For example, they might use financial status to implement differential pricing that targets wealthier users with higher service fees. Meanwhile, all super-app users are potential victims because many users do not provide such information due to privacy concerns. For instance, according to AliPay's data, only 6.2%, 5.8%, and 6.9% users provide their marital status, property ownership, and vehicle ownership, respectively.

Note that this goal is realistic and dangerous because super-app vendors often have a large user base (over one billion). Even if the adversary can only infer the privacy of a small portion of users, the number of threatened users is significant. Although the super-app vendor strives to control access by D_{low_priv}, as super-app functionality expands, data initially considered privacy-irrelevant may become more privacy-revealing, leading to new risks.

Adversary's Ability. Unlike prior work [61] that requests a vast number of OS-level data, we assume the adversary can acquire only Mini-H and Op-H from the vendor's server because existing super-apps (e.g., Alipay, WeChat, and Tiktok) are web-based and hosted on the servers [3, 5, 6], the vendor naturally needs Mini-H and Op-H to provide the desired app functionality. In practice, the privilege level of different data is determined by regulations, industrial practices, or research papers. Since Mini-H and Op-H are not identified as sensitive in these above sources, as discussed in Section 2, they are not identified as sensitive and can be accessed by D_{low_priv}.

Due to regulation and performance constraints, we assume the attacker cannot modify the released versions of the super-apps on users' devices, such as instrumentation or probing, to collect user information on the device side.

We also assume that the adversary can collect the privacy attributes of a small number of users from a publicly available data breach. These breaches are widely available on the Internet [42,47,59], which contains phone numbers, personal identification numbers, and other sensitive privacy attributes. The adversary can match their user data to the samples in breach databases (e.g., by matching phone numbers [17,59]) to identify the privacy attributes of a few hundred users. The adversary can then collect Mini-H and Op-H of these users to train the model to infer the privacy of other users.

4 THEFT: Mini-app Interaction History-Oriented Inference Attack

To reveal the new security risk of mini-app, we design a new attack, THEFT, *mini-app interacTion History-oriEnted inFerence aTtack*. This attack is proof of the potential security issue of the interaction data. THEFT uses both Mini-H and Op-H to infer the privacy attributes of users. We will first illustrate the overall pipeline, followed by a detailed presentation of each component.

4.1 Overview

Generally, the THEFT employs a DNN model to infer user privacy. The inputs of the DNN model are Mini-H and Op-H, and the model outputs are the predicted attribute label and a confidence score, which is used to filter out the high-confidence victim subset. The confidence score needs to approximate the real accuracy of the prediction. For example, if a model predicts the user to be a female with a confidence level above 0.9, then the accuracy of this prediction should be more than 90%. Using a predefined threshold, the adversary can select high-confident samples to identify a vulnerable victim subset and provide label predictions while labeling other users with lower confidence as unknown.

The attack process consists of three stages, as shown in Figure 2. The first step is the attack model training, where the adversary uses the Mini-H and Op-H data of a relatively small set of leaked users with one-hot privacy attribute labels to train a DNN model. The second step is model confidence calibration, where the adversary uses a calibration technique to ensure that the confidence score can faithfully reflect the accuracy of the prediction. The last step is online inference, where the adversary uses the trained model to predict the privacy attributes of unlabeled users from Mini-H and Op-H.

Table 3: 28 categories of mini-apps.

Code	Category	Code	Category
1	Education	15	Finance
2	Entertainment	16	Food and Drink
3	House and Home	17	Health and Fitness
4	Lifestyle	18	Art and Design
5	Maps, Navigation, and Taxi	19	Books
6	Music and Audio	20	Comics
7	Parenting	21	Communication
8	Shopping	22	Medical
9	Auto and Vehicles	23	News
10	Beauty	24	Photo
11	Business	25	Productivity
12	Dating	26	Sports
13	Social	27	Weather
14	Travel and Local	28	Event

4.2 Attack Model Training

In the attack model training stage, THEFT trains a model that can better learn the features of mini-app interaction history data and thus can accurately predict the privacy attributes of users. The training data are the mini-app interaction history (Mini-H and Op-H), the training labels are collected privacy attributes from data breaches. We select the transformer-based [62] model as the default architecture of THEFT. The transformer model comprises 12 encoders, each equipped with 12 bidirectional self-attention heads, resulting in a total of 110 million parameters. In Section 5.4, we also compare our transformer architecture with another two architectures to demonstrate the superiority of our choice. For each type of privacy attribute, we modified the output dimension of the last fully connected layer to the number of labels.

Model Input. The model input consists of two parts, Mini-H (denoted as x_m) and Op-H (denoted as x_o). x_m contains the list of the latest N mini-apps accessed by a user. For the i-th mini-app in the list, we record 1) the unique id m^i_{id} of the mini-app (maintained by the AliPay backend), 2) the mini-app category code m^i_c, and 3) the number of access times m^i_f over the last 30 days. For the category code of mini-apps, we classify the mini-apps into 28 types following prior literature [39]. Table 3 displays all the categories. Therefore, x_m is a tensor of the shape of $(M,3)$. Meanwhile, x_o contains button IDs that the user clicked over the past M timestamps, where each timestamp represents a 500-millisecond interval. We convert the original click logs with timestamps from AliPay into this format, to unify the input format for our deep learning model. Within each interval, if the user clicks a button in the mini-app, we record its ID. Otherwise, we record 0 to indicate that the user does not click any button. These button IDs are globally unique in the super-app, with each ID corresponding to a specific function and remaining consistent across all mini-apps. The input of the model is the fusion of x_m and x_o. To fuse the two types of data, we empirically set $N = M$ and concatenate x_m and x_o into a single input tensor for the model.

Model Output. The output is the predicted label for a specific type of privacy attribute. In this paper, we focus on seven

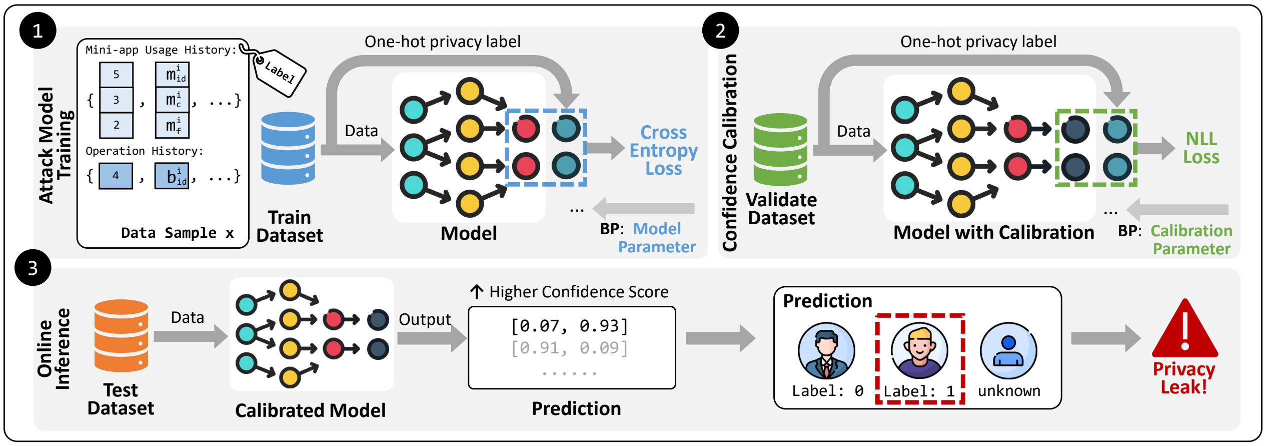

Figure 2: The pipeline of THEFT consists of three steps: attack model training, confidence calibration, and online inference.

privacy attributes: gender, location, age, property ownership, vehicle ownership, marital status, and parental status. All these types of privacy are maintained by AliPay and are considered important privacy information in the internal company. For each privacy attribute, we regard it as a classification task. For five privacy attributes, the inference task is a binary classification task. The five privacy attributes are gender, whose labels are male and female, and the ownership of property and vehicles, marital and parental status, whose labels are yes and no. For location, we give three labels: Tier-1 cities, Tier-2 cities, and Tier-3 cities, making this task a ternary classification task. Specifically, Tier-1 cities include municipalities, well-developed provincial capitals, and economic centers; Tier-2 cities encompass other provincial capitals and major cities; and Tier-3 cities comprise ordinary cities and regions primarily composed of towns and villages. Regarding age, we follow prior research and assign four labels: Under 18, 18∼39, 40∼65, and Above 65 [58].

4.3 Model Confidence Calibration

Necessity. Recall that the goal of THEFT is to identify a subset of users whose privacy attributes can be inferred accurately. However, the original output confidence of the model cannot faithfully reflect the inference accuracy. This is because the original model is optimized by the cross-entropy (CE) loss. The CE loss will lead to overconfidence because it greedily maximizes the confidence of the predicted label. However, for samples in which the model is not confident, we do not want to give a high confidence score to the predicted label because it will lead to a high false positive rate. In other words, when the attack model generates a confidence score of 0.9 for the prediction of user privacy, it should also ensure that the prediction accuracy is above 0.9. Therefore, by selecting predictions with high confidence scores, we can identify the subset of users from which we can accurately infer their privacy attributes.

Temperature Calibration. We adopt model calibration techniques [49, 66] to achieve our goal. The general idea of model calibration is to append an extra trainable calibration module after DNN's output layer. The calibration module takes DNN's output (overconfident prediction) as input and generates a new confidence score. The calibration module is fine-tuned on a calibration dataset to minimize the gap between the generated confidence score and the real accuracy. Our design chooses the temperature calibration [49, 69] because it performs best in our evaluation. In Section 5.5, we compare our temperature calibration with two other calibration techniques to demonstrate the superiority of our choice.

Specifically, given a trained attack model, we replace its softmax function with a calibrated softmax function, which is shown in Equation 1. The calibrated softmax adds a trainable parameter, temperature t, to the softmax function. Then we fine-tune t by minimizing the negative log-likelihood loss (NLL loss) in the validation dataset. Note that we fixed the parameters of the attack model during the calibration phase and only updated t. The loss function choice and the validation dataset's use are consistent with prior work [36, 69].

$$C(x) = \frac{\max_{i \in K} \exp(o_i/t)}{\sum_{i \in K} \exp(o_i/t)} \tag{1}$$

4.4 Online Inference

For inference, we use a pre-defined threshold t_d as the confidence bar to determine whether our model can confidently infer the privacy attribute of a user. For the input of a user x, the output confidence score of the calibrated module is $C(x)$. THEFT returns "unknown" if $C(x)$ is smaller than t_d. Otherwise, THEFT will generate the corresponding attribute label for the given user.

5 Evaluation

In this section, we comprehensively evaluate the inference performance of THEFT. We first describe the experimental setup in Section 5.1. In Section 5.2, we display the effectiveness of THEFT, particularly the high inference accuracy on the high-confidence samples. After that, in Section 5.3, we provide insight and in-depth analysis on the success of THEFT for each of the privacy attributes. In Section 5.4, we prove the effectiveness of our model architecture selection. In Section 5.5, we conduct ablation studies to demonstrate the generalizability of our approach and the recommended parameter settings.

Ethical Disclaimer. Since our evaluation involves collecting privacy attributes and mini-app interaction history, we obtained approval from the IRB of AliPay. For details, please refer to the Ethics Considerations Section.

5.1 Experiment Setup

Volunteer Selection. We first select 5% of daily active users of AliPay as volunteer candidates and send invitations to them. To guarantee the quality of the data, we asked the internal engineers of AliPay to confirm that the candidates have been active in AliPay for more than 7 days at the time of invitation. In each invitation, we provided a clear consent notice on what information is collected and stated that the data were collected only for research purposes. The candidates who accepted the invitation become our participants. In total, we selected 38,351,206 candidates, of which 288,895 users agreed to share their data.

For our participants, after their consent, we pushed the AliPay-dev version to their devices, replacing the original version. The AliPay-dev version is the same as the original version. The only difference is that it displays an additional user agreement when the user first opens the app. This agreement notifies the users that he/she is involved in our experiment and displays detailed information about our experiments. Note that we do not add extra instrumentation or probes in AliPay-dev. We store Mini-H and Op-H in an encrypted database to protect the privacy of volunteers.

Data Collection. We collect data during the whole process of using AliPay. Specifically, the collection phase starts when the user opens the app and ends when the user exits AliPay or switches to another app. We collect both Mini-H and Op-H during this phase. We collected data for 50 days and obtained 10,987,662 data samples from 288,895 users.

We empirically set the length N to 200 for each mini-app interaction history data sample. Specifically, for Mini-H, we record the Top 200 recently used mini-apps. We confirmed with the internal engineers of AliPay that 200 mini-apps are sufficient to represent the user's recent behavioral patterns, as the average number of mini-apps used by AliPay users is 117.8. For Op-H, we convert the original logs in AliPay with timestamps into 0.5-second intervals over a 100-second window, resulting in a 200-dimensional vector per data sample. We empirically determined these values based on AliPay's report that the average time of user interactions is 89.2 seconds. A period of 100 seconds is sufficient to cover most user interactions. The internal engineers of AliPay also confirmed that the 500-millisecond interval is the minimum for user interactions, and most interactions are shorter than 100 seconds. Recent literature also supports the setting of these values [30, 53]. Therefore, the dimension of each mini-app interaction history data sample is 200×4.

Building the Ground Truth. We strictly adhered to IRB requirements (illustrated in the Ethics Considerations Section) to build ground truth privacy attributes as training labels. For the location attribute, we obtained user consent to access the data of the system location service. For other attributes, we first ask users to provide their information. Then we sent authorization requests to acquire user consent to access users' data in AliPay's server, thereby obtaining gender, age, as well as property and vehicle information registered under their names to confirm user-provided labels. For users who are unwilling to share all their privacy attributes or the shared attributes do not match the data in AliPay's server, we removed their data in our evaluation. In summary, we construct the dataset with 1,099,130 data samples from 219,826 users.

Unbiasedness of Collected Dataset. We further demonstrate that the collected dataset closely aligns with real-world population distributions. We analyzed label distribution for various privacy attributes: 22.8% users are from tier-1 cities, 43.4% from tier-2, and 33.8% from tier-3. The gender distribution is almost equal: 50.3% male and 49.7% female. For age, 17.5% are under 18, 38.0% are between 18-40, 30.9% are 40-65, and 13.6% are above 65. Moreover, 48.6% users own property, 21.0% users own vehicles, 69.7% are married, and 53.6% have children. To verify the unbiasedness of our data, we conducted a chi-square test comparing these distributions with data from China's National Bureau of Statistics [7]. The test confirmed no significant statistical difference, indicating that our dataset is indeed representative of the wider population.

Experiment Protocol. We split the collected data into three sets: a training set, a validation set, and a test set. To demonstrate that THEFT can use a very small dataset as the training set to infer the privacy attributes of large-scale users, we set the size of training and validation sets to a small number (less than 0.1% of the test set). Specifically, the training set includes 200 users, and the validation set includes another 200 users. We collect five data samples from each user. Thus, the total number of training and validation sets is 1,000. The test set contains 1,097,130 data samples from the other 219,426 users. Furthermore, we repeated the evaluation ten times with random data partitioning to provide a reliable assessment. For all comparisons in our evaluation, we run the hypothesis test [15] to ensure the significance of our observation.

Implementation Details We conducted our experiment on a server with an Intel Xeon E5-2678 v3 CPU (48 cores), 128GB RAM, and 2 NVIDIA RTX 3090 GPUs. The server OS is Ubuntu 20.04 LTS. We implemented our code base using Python 3.8 and PyTorch 2.0. All models are trained for 100 epochs. We monitor the loss in the validation set and ensure that the models are converged. We set the discriminator threshold (t_d) to 0.9, meaning we identify data samples with a confidence score greater than 90.0% as a high-confidence subset, according to the literature [37,68]. We utilize Adam as the optimizer, and the learning rate is set to 1e-3.

5.2 Attack Effectiveness

In this section, we will first introduce the general effectiveness of THEFT. Then, we will show that THEFT can effectively select a subset of users and accurately infer their privacy attributes. Finally, we will illustrate that output confidence of our model is aligned with the true accuracy of the prediction, which is an essential indicator of the attack's effectiveness.

General Effectiveness. In Figure 3 (a) we show the overall inference accuracy across all inference samples for each privacy attribute label. We ran each experiment 10 times and reported the average accuracy. We manually checked the standard error and confirmed that for all labels, the standard error is below 0.7%. Therefore, we omit the standard error in the figure. In Figure 3 (a), we also mark the accuracy of the random guess baseline with a dashed line. The average inference accuracy of THEFT is 65.2%, which is 48.1% higher than the baseline. In particular, for the age of users, THEFT is 2.9× higher than the baseline. We also performed a chi-square test to confirm the superiority of THEFT. Our null hypothesis is that the performance of the model is equal to random guessing, and the p value is 0.02. Therefore, we reject the null hypothesis ($p < 0.05$) and conclude that THEFT is significantly better than the baseline.

Performance on High-Confident Subset. In Figure 3 (b)-(h), we split the model confidence score into intervals of 10%. Each subfigure represents the results of one type of privacy. For each confidence interval, we show the proportion of samples within this interval (P_{int}, represented in light blue bars) and the correctly predicted samples within that interval (P_{conf}, represented in dark blue bars). Note that the more dark blue bars cover the light blue bars, the more samples of P_{int} are correctly predicted (a higher attack performance). We also show the prediction accuracy within that interval ($acc_{int} = P_{conf}/P_{int} \times 100\%$, represented by the red line). For each subfigure, the y-axis on the left represents the prediction accuracy (acc_{int}), and the right y-axis represents the proportion of samples within the interval (P_{int} and P_{conf}).

For samples predicted by our model with high confidence scores, the attack performance is *near-perfect*. Setting the confidence threshold as 90%, we can on average identify 16.1% samples, and the accuracy of the inference is 95.5%. For the bar at the index 90-100 in Figure 3 (b)-(h), the dark blue bars cover almost entirely the light blue bars. which means that the proportion of correctly predicted samples (P_{conf}) is very close to all samples in this interval (P_{int}). This implies that our attacks are very likely to be successful in 15.4% of all data samples.

We can also observe that high attack performance is consistent across all types of privacy. For six out of seven types of privacy, we can identify more than 10% users with an accuracy of more than 90%. For gender, we can identify 11.0% data samples with an accuracy of 97.7% by setting the confidence threshold to 90%. For location and age, we can identify 12.0% and 23.9% data samples with an accuracy of 97.3% and 98.6%, respectively. For property ownership, we can identify 4.9% data samples with an accuracy of 99.0%. For vehicle ownership, we can identify 14.7% data samples with an accuracy of 91.9%. For marital status, we can identify 19.9% data samples with an accuracy of 93.5%. For parental status, we can identify 25.8% data samples with an accuracy of 91.3%.

Besides, our analysis shows that the distribution of users with high prediction accuracy is demographically representative of the overall collected dataset, indicating no bias toward specific groups. We perform a chi-square test to check consistency. The results show that the distributions of high-confidence data (p-value = 0.87) and other data (p-value = 0.96) align with the distribution of the entire test dataset, further confirming the absence of bias.

Correlation between Confidence and Accuracy. From the red lines in Figure 3 (b)-(h), we can also observe that *the confidence score produced by our model is positively correlated with accuracy*. We calculate the Pearson's correlation coefficient between confidence and accuracy, which is 0.992 on average. It implies that for a given data sample, the higher the confidence score, the more likely the prediction is to be correct. This is an important indicator of the effectiveness of the attack. It means that we can effectively select the subset of users by model confidence, and for the selected users, we can accurately infer their privacy attributes.

5.3 Insights From Data

To further understand the reasons for the privacy leakage, we conduct a thorough investigation into the users that the model has high confidence scores to identity the sources of leakage for each attribute. We found that the users' privacy attributes relate to the preferred mini-app types and how they operate smartphones. Specifically, we mainly study three metrics: two metrics are related to Mini-H: the number of mini-apps (#Mini-app) and the access frequencies (#Access) of specific mini-app types, and one metric relates to Op-H: the number of button clicks (#Click) in each Op-H sample.

Gender. We display #Mini-app, #Access of representative mini-apps in Figure 4 (a) and (b). In Figure 4 (c), we plot the user distribution w.r.t. #Click. For all figures, the red bars represent female users, and the blue bars represent male users. For Mini-H, we found that female users prefer `Shopping` and `Beauty` mini-apps. Female users averagely use 6.8 and 5.9 kinds of mini-apps for `Shopping` and `Beauty`, with a frequency of 14.2 and 10.9 times. Conversely, male users only

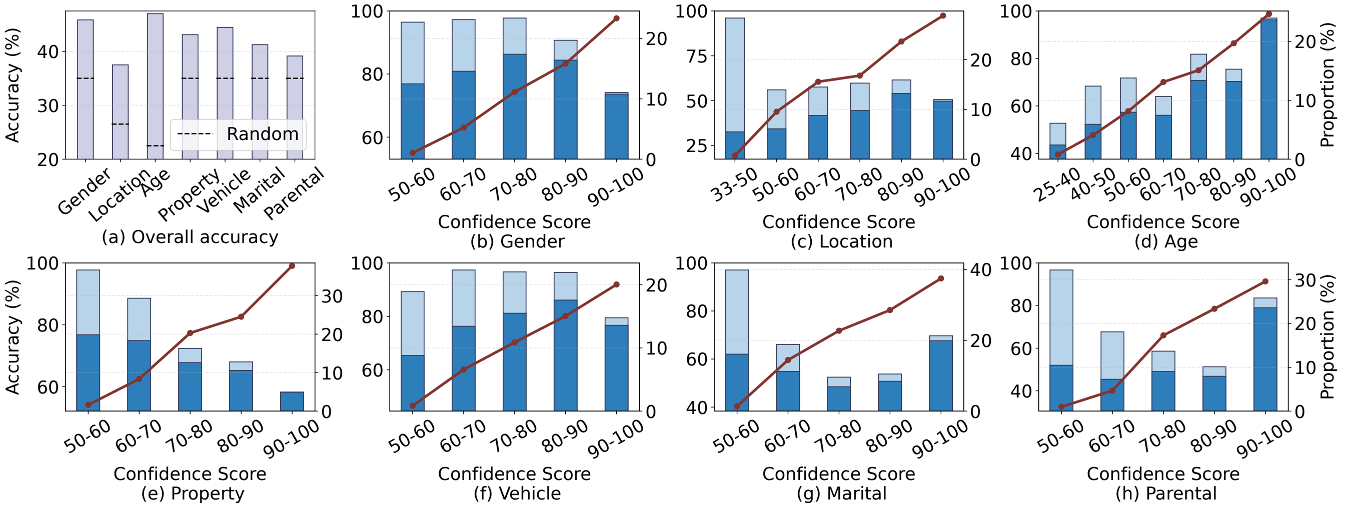

Figure 3: Attack effectiveness of THEFT. (a) Overall inference accuracy for each type of privacy. (b)-(h) For each privacy type and confidence interval, the proportion of users predicted to be within the interval (P_{int}), the proportion of correctly predicted users (P_{conf}), and inference accuracy ($acc_{int} = P_{conf}/P_{int} \times 100\%$).

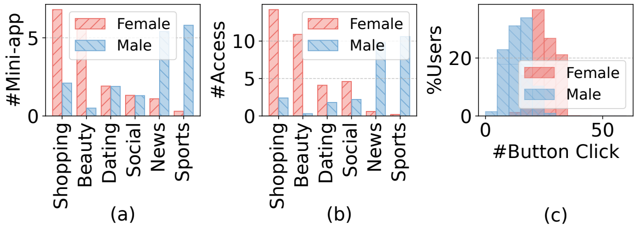

Figure 4: Detailed analysis on the privacy of *gender*.

access 2.1 Shopping mini-apps for 2.4 times, and 0.5 Beauty mini-apps for 0.3 times. We found that male users prefer News (5.4 mini-apps for 9.8 times) and Sports (5.8 mini-apps for 10.6 times). Conversely, female users only access 0.7 mini-apps of these two categories for 0.4 times. For Op-H, female users usually operate more frequently than male users, confirmed by a recent study [60]. Figure 4 (c) shows that male's #Click ranges from 10 to 20, while female's #Click ranges from 20 to 35. This observation aligns with recent research showing that females tend to use apps for a longer time than males [60].

Location. Mini-H and Op-H can reflect the economic and population status of the city where users live, thus leaking the location attribute. To display this result, we first show the statistical observation of Mini-H, then map the mini-app access frequencies to the users' geographical locations in China, and display the statistics of #Click.

For Mini-H, both #Mini-app and #Access correlate with users' location. Users in Tier-1, Tier-2, and Tier-3 cities averagely utilize 27.9, 23.6, and 17.3 types of mini-apps. For #Access, Users in Tier-1, Tier-2, and Tier-3 cities use mini-apps for 36.3, 20.9, and 13.1 times, respectively. To further verify our observation, we map #Access to the users' geographical locations in China (by province) in Figure 5 (a). The darker color represents a higher #Access. Red and yellow dots represent the Tier-1 and Tier-2 cities. We plot the Heihe-Tengchong Line [25] with a dashed red line. Heihe-Tengchong Line is a famous geographical line in China representing population density. The line's southeast side has 94% of China's population and represents a higher economic level. The northwest side only has 6% of the population and represents a lower economic level. From Figure 5 (a), we can see that the southeast (lower right in the figure) area of the line is darker than the northwest (upper left) area, which means #Access is positively correlated with the economic level. We think it is because people in more developed cities rely more on digital services daily. For example, people in Tier-1 cities are more likely to use mini-apps to order food, buy tickets, and pay utility fees.

For Op-H, we observe a remarkable difference in #Click between users from different locations. Specifically, we focus on the button of password-free payment [41]. This button lets users execute small-amount transactions (below 200 RMB, approximately 30 USD) without inputting passwords. This button is designed to improve user experience and avoid frequently entering passwords. We found that users in the more developed areas have higher #Click on password-free payment. For users from Tier-3, Tier-2, and Tier-1 cities, #Click on password-free payment is 0.1, 0.7, and 1.1, respectively. We also observe that this statistical result is specific to the password-free payment button. We do not observe this phenomenon on other buttons with similar functions, such as the payment button.

Age. The good performance of THEFT for age is because people of different ages display different patterns on the types of mini-apps and how they interact with mini-apps.

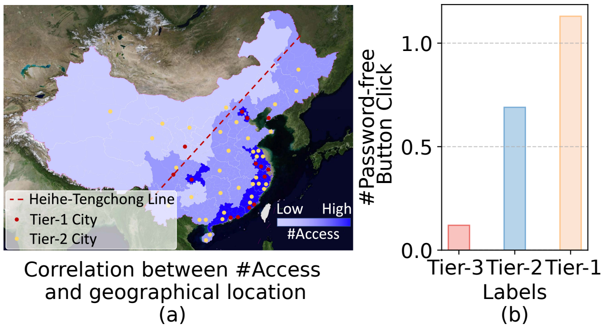

Figure 5: Detailed analysis on the privacy of *location*.

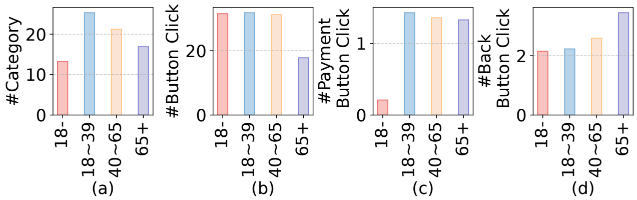

Figure 6: Detailed analysis on the privacy of *age*.

For Mini-H, we plot the number of accessed mini-app categories (#Category) in Figure 6 (a). Minors (Under 18) and the elderly (Above 65) access a relatively smaller #Category (13.2 and 16.9) than other age groups (above 21.0). Notably, minors rarely access mini-apps of Finance because regulations restrict financial services to minors. Similarly, elderly users rarely access mini-apps of Dating and Comics, as older people are not the target users of these mini-apps.

For Op-H, we display #Click on any buttons (#Button Click), #Click on the Payment button (#Payment Button Click), and #Click on the Back button (#Back Button Click) of each age group in Figure 6 (b)-(d). For #Button Click, elderly users have the lowest value (average 17.8) compared to other groups (31.5). This is because the elderly are less familiar with using super-apps and tend to react more slowly to the response. For #Payment Button Click, minors rarely click this button (average 0.2 clicks) than other groups (1.4 clicks) because minors are not allowed to make payments without the consent of their guardians. For #Back Button Click, the elderly have the highest value (3.5) compared to other groups (2.3) because the elderly usually cannot accurately select the desired button and thus need to click the Back button to return to the previous page and try again.

Property. Table 4 shows representative mini-apps, the services provided by each mini-app, and the access frequencies of users with and without property. Users without any property rarely access mini-apps that can pay household bills (e.g., Utilities, State Grid, and CSGrid; frequency below 0.1). Contrarily, users with property access these mini-apps more frequently (an average of 1.7 times). This is because these users must pay the household bills themselves. On the other side, users without property prefer mini-apps about house renting (e.g., Ziroom, Anjuke, and Renting). The access frequency (1.3) is higher than users with property (0.0). This is because users without property are likelier to rent a house.

Table 4: Comparison for users with and without property.

Mini-app	**Prodived**	**#Access**	
Name	**Services**	**W/o Prop.**	**W/ Prop.**
Utilities	Household bills payment	0.1	2.6
State Grid	Electricity bill payment	0.0	1.3
CSGrid	Electricity bill payment	0.0	1.1
Ziroom	Housing rental	1.2	0.0
Anjuke	Housing rental	1.1	0.0
Renting	Housing rental	1.6	0.0

Table 5: Comparison for users with and without vehicles.

Mini-app	**Prodived**	**#Access**	
Name	**Services**	**W/o Veh.**	**W/ Veh.**
12123	Traffic police platform	0.2	1.5
Sinopec	Fueling	0.0	3.1
CNPC	Fueling	0.0	2.7
Shell	Fueling	0.0	2.3
Car Life	Vehicle insurance, highway toll	0.0	2.8
Halo	Shared bicycle	12.5	2.6
Didi	Taxi	8.1	3.3
Caocao	Taxi	6.3	1.3
Transport	Public transportation	13.9	2.4
Outgoing	Taxi, public transportation	16.8	3.1

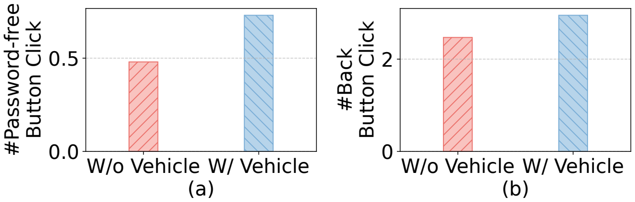

Figure 7: Detailed analysis on the privacy of *vehicle*.

Vehicle. Table 5 shows representative mini-apps that may expose vehicle ownership, the services provided by each mini-app, and the access frequencies of users with and without vehicles. Users without vehicles rarely access mini-apps for traffic police, fueling, and vehicle insurance (e.g., 12123, Sinopec, and Car Life; frequency lower than 0.2). Contrarily, vehicle users access these mini-apps more frequently (average 2.5 times). Users without vehicles prefer public transportation mini-apps (e.g., Transport, Didi and Caocao). In Table 5, #Access for these users (11.5) is significantly higher than vehicle users (2.5). This is because users without vehicles rely on public transportation or a taxi to travel, while users with vehicles can drive their cars.

For Op-H, we show #Click on two buttons in Figure 7: Back and Password-free Payment. The blue and red bars represent the users with and without vehicles. Users with vehicles have a high #Clicks on both buttons, higher than users without vehicle by an average of 38.55%. This may be because users with vehicles are more likely to use super-apps

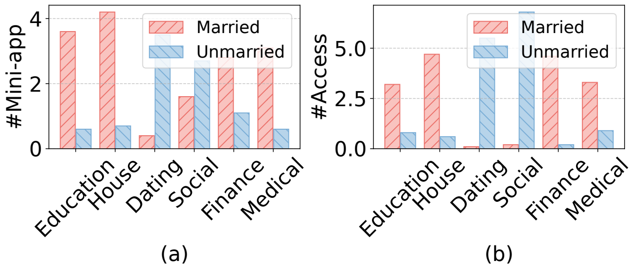

Figure 8: Detailed analysis on the privacy of *marital*.

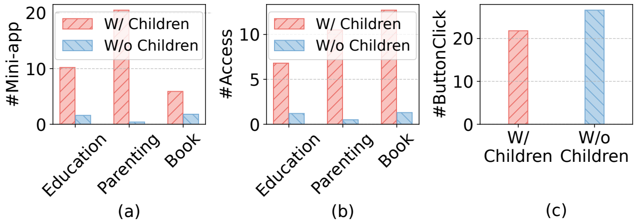

Figure 9: Detailed analysis on the privacy of *parental*.

while driving [55], making password-free payments more convenient and more frequently using the `Back` button to return to the previous page.

Marital. We display #Mini-app and #Access for representative mini-app categories in Figure 8. The married users prefer mini-apps about `Education`, `House and home`, `Finance`, and `Medical`. For these types in Figure 8, the red bars (married users) are significantly higher than the blue bars (unmarried users) by 4.7× for #Mini-app and 6.3× for #Access. This means married users focus more on family aspects of life. Conversely, unmarried users prefer mini-apps about `Dating` and `Social`, with 1.5× higher #Mini-app and 82.0× higher #Access than married users.

Parental. We display #Mini-app and #Access for representative mini-app categories in Figure 9 (a) and (b), and show #ButtonClick in Figure 9 (c). For Mini-H, users with children prefer the mini-apps about `Education`, `Parenting`, and `Books`. In Figure 9 (a) and (b), the #Mini-app and #Access of users with children (red bars) are higher than those without children (blue bars) by 9.6× and 10.0×, respectively. We believe the reason is that parents tend to pay more attention to their children's education. For Op-H, users with children have a lower #ButtonClick than users without children. In Figure 9 (c), the #ButtonClick of parents (red bar) is 18.0% lower than childless users (blue bar).

5.4 Architecture and Calibration Comparison

In this section, we study different choices of model architecture and confidence calibration techniques. For the model architecture, we select three representative architectures: a CNN-based model, an RNN-based model, and our Transformer-based model. The CNN-based model is a ResNet model with 48 convolutional layers [27, 38]. The RNN-based model has 8 LSTM layers following the default architecture of prior literature [28, 57]. All architectures have been demonstrated to be effective to extract high-dimensional information from time series data [19, 29, 46, 75]. For calibration, we compare the temperature calibration with two representative techniques: vector calibration and matrix calibration [24].

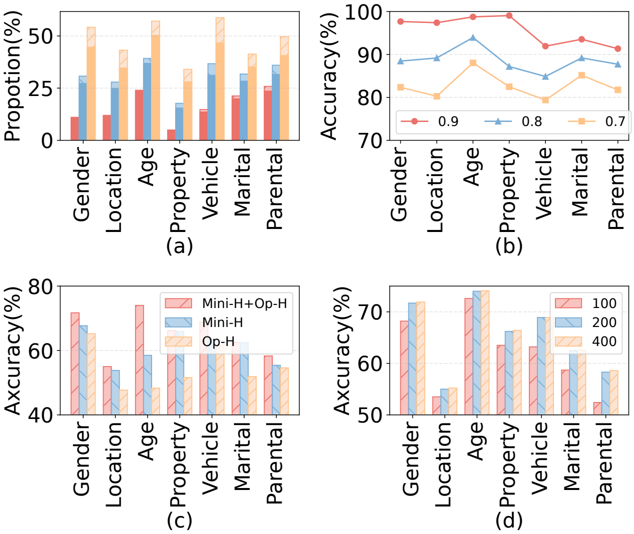

Figure 10: Comparision of different settings.

Evaluation Metrics. We use four different metrics to evaluate the performance: the proportion of high-confidence data (PHC), precision, recall, and F1-score. Given a confidence threshold, the PHC measures the proportion of data samples that the model can confidently infer. A higher PHC indicates a higher attack performance. Precision, recall, and F1-score are three widely used metrics to evaluate classification models. We use these three metrics to comprehensively evaluate the model's correctness on the PHC data samples.

Results. We display the averaged results in Table 6, in which the values are averaged across the seven labels of privacy attributes. First, for all settings, the PHC is above 8.8%. Precision, recall, and F1-score are all above 89.6%. The proposed attack is general to different architectures and calibration techniques. Second, among all the settings, the Transformer-based model with temperature calibration achieves the best performance. The PHC is 16.1%, and the precision, recall, and F1-score are all above 95.4%. This means that this setting can correctly infer the privacy of the most number of users with the highest accuracy.

5.5 Ablation Study

In this section, we compare the results of different settings used in THEFT.

Threshold. First, we assessed the impact of the threshold settings. A higher threshold means we only attack samples with higher confidence scores, implying a smaller victim subset

Table 6: Comparison between model architectures and calibration techniques. For each metric, we report the average value across seven privacy attributes.

Model	CNN-based			RNN-based			Transformer-based		
Calibration	Temperature	Vector	Matrix	Temperature	Vector	Matrix	Temperature	Vector	Matrix
PHC	14.6%	11.8%	15.9%	10.0%	7.9%	9.8%	**16.1%**	15.6%	8.8%
Precision	91.9%	91.3%	90.4%	91.1%	92.2%	91.4%	**95.5%**	94.4%	95.3%
Recall	91.8%	90.6%	89.8%	90.2%	90.5%	89.6%	**95.4%**	94.1%	94.4%
F1-score	91.8%	90.8%	89.9%	90.5%	91.1%	90.1%	**95.4%**	94.2%	94.8%

and higher accuracy. Specifically, we compared the proportion of identified samples and their accuracy when setting the threshold at 0.9, 0.8, and 0.7. In Figure 10 (a), light-colored bars represent the proportion of samples that can be identified, and dark-colored bars represent the proportion of accurately inferred samples. We can observe that a lower threshold encompasses more samples, but the accuracy is relatively decreased. The results for the three threshold settings are displayed in Figure 10 (b), where a 0.9 threshold achieves an average accuracy of 95.7%, while 0.8 and 0.7 only reach average accuracies of 88.7% and 82.8% respectively. These findings demonstrate that higher confidence scores align with higher inference accuracy, thereby validating our model calibration technique. In this paper, we therefore adopt a threshold of 0.9 to maximize our attack performance.

Input Data. THEFT's input data is a concatenation of Mini-H and Op-H. To further demonstrate that both data types are privacy-related, we trained models using Mini-H and Op-H separately and reported the results in Figure 10 (c). Only using Mini-H or Op-H achieves an accuracy of 61.1% and 54.4%, respectively. Although the values are lower than THEFT's concatenation solution (65.2%), they are higher than the random guess baseline by an average of 44.0%. Thus, both Mini-H and Op-H are effective for THEFT.

Data Length. In THEFT, we empirically set $N = 200$. In this section, we compare it with $N = 100$ and $N = 400$. As shown in Figure 10 (d), using $N = 400$ only provides a marginal improvement of 0.2%, while $N = 100$ decreases accuracy by 5.4%. Therefore, our setting is an effective choice.

6 Industry Feedback

To help improve the super-app ecosystem, we notified our findings super-app developers and the privacy standards expert teams of the IEEE Standards Association (IEEE-SA) in China. We summarize their feedback and critical lessons as follows.

6.1 Notification Process

Methodology. Following prior literature, we contacted super-app developers via email addresses extracted from Apple's App Store submissions and super-apps' official websites. In our emails and online communications, we briefly explained potential leakage risks and presented our results. We also posed three crucial questions to the developers and experts: (1) whether they were aware of the privacy implications of the data collected under existing legal regulations, (2) whether they were aware that such information could infer privacy, and (3) whether they had any remedial plans or suggestions for addressing these privacy risks.

Results. We contacted the vendors of all 31 super-apps in Table 2 and the standards association. We conducted two rounds of communication. The first round was completed by October 31, 2023. A second round was initiated for those who did not reply in the first round and was completed by March 31, 2024. Among the 31 vendors, eight acknowledged our report, and four acknowledged and provided feedback on our questions. Most of the unresponsive vendors either didn't reply or stated that they would forward the report to relevant teams, after which no further response was received. We speculate that the lack of response from some vendors could be due to competitive concerns and the protection of business secrets. We also engaged the privacy standards department, and they acknowledged our report. Notably, during our communication with company developers, they showed significant interest in our study and frequently asked for details about our attack.

6.2 Feedback

Four super-app vendors and the standards association have committed to updating their privacy policies. This update aims to alert users about the recorded mini-app interaction history data during usage and the potential privacy connections. Moreover, we received the following feedback from the developers and experts:

Feedback 1: The developers acknowledged that "*We will fix the notification of potential risks of the mini-app interaction history data in our user terms.*" They further stated that "*The identified privacy risks indeed exist. Previously, there was a subconscious belief that collecting mini-app interaction history data does not violate regulations due to the absence of studies on the privacy leaks they might cause.*"

Feedback 2: The developers stated: "*We are working in progress to protect the mini-app interaction history as other sensitive data.*" Furthermore, they admitted that they had overlooked the privacy issue of mini-app interaction history by saying: "*We usually follow the latest research papers to pro-*

tect users' privacy. However, to our knowledge, there is no paper revealing the privacy risks of this data. Thus, we were not aware of the privacy issue."

Feedback 3: The developers promised to "*improve protection of mini-app interaction history by reducing the time to store such data.*" However, developers also mentioned the trade-offs between cost and privacy protection: "*The computation budget to protect privacy is limited. There is a balance between privacy protection and delivering satisfactory services. The user data generated by super-apps is vast, and the cost to protect all data is immeasurable.*"

Feedback 4: The developers agreed to improve their user term in collecting mini-app interaction history by acknowledging: "*Before this study, we considered mini-app interaction history as privacy insensitive because we only simply reviewed the data but didn't conduct in-depth research on it.*"

Feedback 5: The experts from the standards association noted: "*During our communications with super-app companies, their developers agreed to add mini-app interaction history into the user terms. The association will also standardize the collection and use of such data in the future.*"

7 Discussions

Lessons Learned. We identified several key lessons to enhance privacy in super-apps. First, developers must rectify all existing misconceptions regarding the boundaries of privacy security. This includes thoroughly re-examining their services to identify and safeguard potentially vulnerable mini-app interaction history. Second, developers must ensure user data privacy rather than merely reacting to privacy breaches. From a regulatory perspective, standards associations should strive to preemptively address new security vulnerabilities instead of waiting to act after breaches occur.

Potential Interaction Changes. One potential concern is that the user agreement could introduce bias, influencing volunteers' interaction patterns with the mini-apps. However, we believe this bias has a negligible impact due to the large volume of data, which helps mitigate individual biases. Nevertheless, we acknowledge that measuring bias is challenging since we cannot access ground-truth data (e.g., usage history outside the experimental period or from non-participants).

Performance of Attacks. Note that even though our focus is not on all users, the implications of accurate predictions on a high-confidence subset are profound. This subset, albeit a fraction of the overall users, includes many people due to the colossal scale of super-app users. For instance, for a super-app with over a billion users, accurately inferring privacy attributes of even 1% users will impact 10 million individuals. This magnitude is substantial and raises serious concerns on personal privacy. Our research highlights this issue and draws attention to robust privacy protection measures, even when only a subset of users is identified.

Privacy Statement Revision. In the feedback, five super-app vendors promised to revise their user terms. This marks a milestone in super-app privacy, as these industry giants acknowledge and address the privacy implications of mini-app interaction history. By updating the privacy statements, these vendors have demonstrated a commitment to enhancing user privacy and set a precedent for others. The willingness of these vendors to adapt and improve their privacy statements in light of our research is also a strong testament to the generalizability of our findings.

Generalizability. A key aspect of our research is its broad generalizability, not just to AliPay but to various super-apps as well. The unified interface design and operational paradigms of super-apps and mini-apps lead to the generality of Mini-H and Op-H. Therefore, the privacy vulnerabilities we identified from the mini-app interaction history are generalizable findings.

Defense Techniques. Common defense methods include differential privacy or trusted hardware. However, applying differential privacy to mini-app interaction history could lead to information loss and mislead user interactions with other operations. Utilizing trusted hardware like SGX [22] significantly increases costs, which is unsustainable for super-apps with hundreds of millions of active users. Therefore, there is a need to discover more effective defense methods.

8 Related Work

Previous research has witnessed a significant focus on privacy protection in the mobile area. These works highlight the intricate privacy dimensions specific to mobile apps. Specifically, these studies offer valuable perspectives on several key areas: they explore the privacy challenges in super-app ecosystems [16, 20, 26, 64, 65, 73, 76, 78], investigate methods through which data might be stealthily exfiltrated or mishandled [18, 34, 50], assess the alignment between the mobile app and regulations [10, 13, 32, 39, 48, 51, 56, 71–74, 79], and propose strategies to enhance privacy compliance and transparency within apps [14, 23, 31, 33, 52, 67]. These studies emphasize the complexity of privacy issues, covering personal and device information, location, camera, microphone, etc. These investigations reflect the community's commitment to tackling mobile app privacy and security challenges.

However, an unexplored area remains regarding the potential privacy risks in the mini-app interaction history within super-app ecosystems. Unlike location or microphone data, these data do not typically trigger system-level permission prompts, making them less visible and hence less concerned. Meanwhile, they can reveal extensive insights into user privacy attributes. In this paper, we close a critical gap in understanding how seemingly non-sensitive mini-app interaction history can pose novel risks, underscoring the need for more nuanced privacy protections for the super-app ecosystems.

9 Conclusion

This paper reveals a new super-app privacy vulnerability: the mini-app interaction history. We design a new attack, THEFT, that can achieve more than 95.5% accuracy in inferring privacy attributes of over 16.1% of users with less than 0.1% training data. We also highlight a significant oversight in the academic community and industry practitioners on protecting mini-app interaction history. Our findings have also raised awareness and proactive measurements among super-app vendors and standards associations.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback. Ding Li is the corresponding author. This work was partly supported by the National Science and Technology Major Project of China (2022ZD0119103) and the CCF-AFSG Research Fund (RF20220006).

Ethics Considerations

We carefully reviewed the conference's ethical guidelines, submission instructions, and ethics documents. Our research was conducted ethically and responsibly, with IRB approval from AliPay obtained before starting the study.

We conducted our study transparently, assessing and mitigating risks for all stakeholders. For volunteer participants, we secured informed consent, anonymized their data, and ensured secure handling to prevent unauthorized access. A dedicated team of internal engineers, under strict confidentiality agreements, managed data extraction and anonymization, providing researchers only with anonymized, non-identifiable datasets.

For AliPay as a company and its employees, there was a potential risk of reputational damage or exploitation of identified vulnerabilities by insiders. We proactively engaged with relevant departments to address these vulnerabilities and provided recommendations for improved security practices, aiming to minimize any negative impact on the company.

Considering the broader super-app user base, we recognized that our findings might reveal vulnerabilities that could be exploited if not properly addressed. To mitigate this risk, we disclosed our findings to AliPay's security team and other major super-app vendors. We received positive responses, and several vendors have committed to addressing these issues, thereby enhancing the security of all users.

AliPay's IRB Overview. AliPay, an international company with over a billion users, has an IRB that complies with ethical standards like the Common Rule and ISO 27701 for privacy management. This independent committee includes legal experts, ethicists, and external professionals unaffiliated with the company, ensuring unbiased and thorough reviews. The IRB evaluates all research involving human subjects to uphold the highest standards of informed consent, confidentiality, and data protection.

Participant Data Protection within AliPay. We protected participant data both externally and within AliPay by restricting access to a dedicated team of internal engineers bound by strict confidentiality agreements. This team handled data extraction and anonymization, with all activities logged and monitored. Engineers were prohibited from using the data for any purposes beyond this study. Researchers received only anonymized datasets with pseudonyms replacing unique identifiers, ensuring no access to personally identifiable information. Additionally, the engineers did not participate in research analysis, maintaining a clear separation of roles and further safeguarding participant privacy.

User Consent and Data Collection. Since our evaluation involves the collection of privacy attributes, we obtained IRB approval from AliPay before collecting any data. The IRB requires us to obtain volunteers' consent prior to data collection, and all procedures were conducted by internal engineers at AliPay under IRB supervision.

For volunteer selection, internal engineers sent invitations to active users using automated systems that only checked login timestamps, ensuring no sensitive information was accessed. This procedure was approved by the IRB as low-risk and compliant with data protection policies. Once volunteers were selected, the engineers assisted us with the following steps:

- Informed Consent: We informed the volunteers about the data we would collect, its intended use, and the measures taken to protect it. We assured them that the data would not be used for commercial purposes or shared with third parties.
- Consent Acquisition: We obtained consent from users to access their mini-app interaction history and privacy attributes. For adult users, we obtained their consent directly. For minors, who typically use authorized accounts, we obtained consent from both the users and their parents.
- Attributes Verification: During data verification, internal engineers matched participants' provided attributes with internal databases(e.g., databases from the loan and insurance departments) in a secure and controlled environment. The matching process was automated to minimize human exposure to sensitive data.
- Data Collection: We used AliPay-dev to notify volunteers of their participation, after which internal engineers extracted their mini-app interaction history from AliPay's servers

Note that the internal engineers anonymized all unique user identifiers before being provided to us. The data was securely stored in an encrypted database and was carefully deleted under IRB supervision after the evaluation.

Data Access. All code and anonymized data were accessible only to the authors. The NDA approval process involved submitting a data access request to AliPay's IRB, which reviewed the request for compliance with data protection policies. An agreement was then signed by the authors and authorized personnel, and upon NDA approval, data access permissions were granted. Overall, all data access and handling complied with AliPay's data protection policies and relevant laws, ensuring participant privacy and data security throughout the research.

Open Science

We recognize the importance of open science and are committed to supporting the research community. However, due to AliPay's IRB and business regulations, and strict user privacy concerns, all code and data are processed on supervised servers. Therefore, we cannot publicly release certain artifacts, including code and comprehensive datasets. Meanwhile, we have provided detailed descriptions of our work, including design, methods, input formats, configurations, etc. In addition, we are committed to helping colleagues who wish to replicate or fully understand our work. We encourage contacting us to discuss potential collaborations or access materials under an NDA and with approval from AliPay.

References

[1] Alipay. `https://en.wikipedia.org/wiki/Alipay`, 2024.

[2] Careem. `https://en.wikipedia.org/wiki/Careem`, 2024.

[3] Development guide of the framework in WeChat mini-program. `https://developers.weixin.qq.com/miniprogram/en/dev/framework/`, 2024.

[4] Ele.me. `https://en.wikipedia.org/wiki/Ele.me`, 2024.

[5] Explaination of Alipay OpenAPI (Chinese). `https://open.alipay.com/api#jsapi`, 2024.

[6] Introduction of the mini-app in Tiktok/ByteDance. `https://developer.open-douyin.com/docs/resource/zh-CN/mini-app/develop/overview/introduction`, 2024.

[7] National Bureau of Statistics of China. `https://www.stats.gov.cn/english/`, 2024.

[8] WeChat. `https://en.wikipedia.org/wiki/WeChat`, 2024.

[9] Fatemeh Alizadeh, Timo Jakobi, Alexander Boden, Gunnar Stevens, and Jens Boldt. Gdpr reality check-claiming and investigating personally identifiable data from companies. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 120–129. IEEE, 2020.

[10] David G Balash, Xiaoyuan Wu, Miles Grant, Irwin Reyes, and Adam J Aviv. Security and privacy perceptions of {Third-Party} application access for google accounts. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3397–3414, 2022.

[11] Supraja Baskaran, Lianying Zhao, Mohammad Mannan, and Amr Youssef. Measuring the leakage and exploitability of authentication secrets in super-apps: The wechat case. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID '23, page 727–743, New York, NY, USA, 2023. Association for Computing Machinery.

[12] Nataliia Bielova, Laura Litvine, Anysia Nguyen, Mariam Chammat, Vincent Toubiana, and Estelle Harry. The effect of design patterns on (present and future) cookie consent decisions. In *USENIX Security Symposium. USENIX Association. Accepted for publication*, 2024.

[13] Duc Bui, Yuan Yao, Kang G Shin, Jong-Min Choi, and Junbum Shin. Consistency analysis of data-usage purposes in mobile apps. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2824–2843, 2021.

[14] Yifeng Cai, Ziqi Zhang, Jiaping Gui, Bingyan Liu, Xiaoke Zhao, Ruoyu Li, Zhe Li, and Ding Li. {FAMOS}: Robust privacy-preserving authentication on payment apps via federated multi-modal contrastive learning. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 289–306, 2024.

[15] Yanzuo Chen, Yuanyuan Yuan, and Shuai Wang. Obsan: An out-of-bound sanitizer to harden dnn executables. In *NDSS*, 2023.

[16] Yi Chen, Mingming Zha, Nan Zhang, Dandan Xu, Qianqian Zhao, Xuan Feng, Kan Yuan, Fnu Suya, Yuan Tian, Kai Chen, et al. Demystifying hidden privacy settings in mobile apps. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 570–586. IEEE, 2019.

[17] Long Cheng, Fang Liu, and Danfeng Yao. Enterprise data breach: causes, challenges, prevention, and future directions. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(5):e1211, 2017.

[18] Michalis Diamantaris, Serafeim Moustakas, Lichao Sun, Sotiris Ioannidis, and Jason Polakis. This sneaky piggy went to the android ad market: Misusing mobile sensors for stealthy data exfiltration. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1065–1081, 2021.

[19] Daizong Ding, Mi Zhang, Xudong Pan, Min Yang, and Xiangnan He. Modeling extreme events in time series prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1114–1122, 2019.

[20] Zikan Dong, Tianming Liu, Jiapeng Deng, Haoyu Wang, Li Li, Minghui Yang, Meng Wang, Guosheng Xu, and Guoai Xu. Exploring covert third-party identifiers through external storage in the android new era.

[21] Xiaolin Du, Zhemin Yang, Jiapeng Lin, Yinzhi Cao, and Min Yang. Withdrawing is believing? detecting inconsistencies between withdrawal choices and third-party data collections in mobile apps. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 14–14. IEEE Computer Society, 2023.

[22] Erhu Feng, Xu Lu, Dong Du, Bicheng Yang, Xueqiang Jiang, Yubin Xia, Binyu Zang, and Haibo Chen. Scalable memory protection in the PENGLAI enclave. In *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*, pages 275–294. USENIX Association, July 2021.

[23] Mafalda Ferreira, Tiago Brito, José Fragoso Santos, and Nuno Santos. Rulekeeper: Gdpr-aware personal data compliance for web frameworks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2817–2834. IEEE, 2023.

[24] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017.

[25] Huadong Guo, Lizhe Wang, Fang Chen, and Dong Liang. Scientific big data and digital earth. *Chinese science bulletin*, 59:5066–5073, 2014.

[26] Julie M Haney, Yasemin Acar, and Susanne Furman. " it's the company, the government, you and i": User perceptions of responsibility for smart home privacy and security. In *USENIX Security Symposium*, pages 411–428, 2021.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[29] Shenda Hong, Yanbo Xu, Alind Khare, Satria Priambada, Kevin Maher, Alaa Aljiffry, Jimeng Sun, and Alexey Tumanov. Holmes: Health online model ensemble serving for deep learning models in intensive care units. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1614–1624, 2020.

[30] Chakajkla Jesdabodi and Walid Maalej. Understanding usage states on mobile devices. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 1221–1225, New York, NY, USA, 2015. Association for Computing Machinery.

[31] Scott Jordan, Yoshimichi Nakatsuka, Ercan Ozturk, Andrew Paverd, and Gene Tsudik. Viceroy: Gdpr-/ccpa-compliant enforcement of verifiable accountless consumer requests. *arXiv preprint arXiv:2105.06942*, 2021.

[32] Rishabh Khandelwal, Asmit Nayak, Paul Chung, and Kassem Fawaz. Unpacking privacy labels: A measurement and developer perspective on google's data safety section. *arXiv preprint arXiv:2306.08111*, 2023.

[33] David Klein, Benny Rolle, Thomas Barber, Manuel Karl, and Martin Johns. General data protection runtime: Enforcing transparent gdpr compliance for existing applications. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 3343–3357, 2023.

[34] Simon Koch, Benjamin Altpeter, and Martin Johns. The {OK} is not enough: A large scale study of consent dialogs in smartphone applications. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5467–5484, 2023.

[35] Konrad Kollnig, Lu Zhang, Jun Zhao, and Nigel Shadbolt. Before and after china's new data laws: Privacy in apps. *arXiv preprint arXiv:2302.13585*, 2023.

[36] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2796–2804. PMLR, 10–15 Jul 2018.

[37] Fabian Kuppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 326–327, 2020.

[38] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[39] Shuai Li, Zhemin Yang, Nan Hua, Peng Liu, Xiaohan Zhang, Guangliang Yang, and Min Yang. Collect responsibly but deliver arbitrarily? a study on cross-user privacy leakage in mobile apps. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1887–1900, 2022.

[40] Wei Li, Borui Yang, Hangyu Ye, Liyao Xiang, Qingxiao Tao, Xinbing Wang, and Chenghu Zhou. Minitracker: Large-scale sensitive information tracking in mini apps. *IEEE Transactions on Dependable and Secure Computing*, 2023.

[41] Rongbing Liu. The role of alipay in china. *Nijmegen, Radboud University, Nijmegen, The Netherlands*, 2015.

[42] Yang Liu, Armin Sarabi, Jing Zhang, Parinaz Naghizadeh, Manish Karir, Michael Bailey, and Mingyan Liu. Cloudy with a chance of breach: Forecasting cyber security incidents. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 1009–1024, 2015.

[43] Chaoyi Lu, Baojun Liu, Yiming Zhang, Zhou Li, Fenglu Zhang, Haixin Duan, Ying Liu, Joann Qiongna Chen, Jinjin Liang, Zaifeng Zhang, et al. From whois to whowas: A large-scale measurement study of domain registration privacy under the gdpr. In *NDSS*, 2021.

[44] Haoran Lu, Luyi Xing, Yue Xiao, Yifan Zhang, Xiaojing Liao, XiaoFeng Wang, and Xueqiang Wang. Demystifying resource management risks in emerging mobile app-in-app ecosystems. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications Security*, pages 569–585, 2020.

[45] Allan Lyons, Julien Gamba, Austin Shawaga, Joel Reardon, Juan Tapiador, Serge Egelman, and Narseo Vallina-Rodríguez. Log:{It's} big,{It's} heavy,{It's} filled with personal data! measuring the logging of sensitive information in the android ecosystem. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2115–2132, 2023.

[46] Haoyu Ma, Jianwen Tian, Debin Gao, and Chunfu Jia. On the effectiveness of using graphics interrupt as a side channel for user behavior snooping. *IEEE Transactions on Dependable and Secure Computing*, 19(5):3257–3270, 2021.

[47] Peter Mayer, Yixin Zou, Florian Schaub, and Adam J Aviv. " now i'm a bit {angry:}" individuals' awareness, perception, and responses to data breaches that affected them. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 393–410, 2021.

[48] Mark Huasong Meng, Qing Zhang, Guangshuai Xia, Yuwei Zheng, Yanjun Zhang, Guangdong Bai, Zhi Liu, Sin G Teo, and Jin Song Dong. Post-gdpr threat hunting on android phones: dissecting os-level safeguards of user-unresettable identifiers. In *The Network and Distributed System Security Symposium (NDSS)*, 2023.

[49] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[50] Yuhong Nan, Xueqiang Wang, Luyi Xing, Xiaojing Liao, Ruoyu Wu, Jianliang Wu, Yifan Zhang, and XiaoFeng Wang. Are you spying on me? large-scale analysis on iot data exposure through companion apps. 2023.

[51] Trung Tin Nguyen, Michael Backes, Ninja Marnau, and Ben Stock. Share first, ask later (or never?) studying violations of {GDPR's} explicit consent in android apps. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3667–3684, 2021.

[52] Trung Tin Nguyen, Michael Backes, and Ben Stock. Freely given consent? studying consent notice of third-party tracking and its violations of gdpr in android apps. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2369–2383, 2022.

[53] Naoto Nishida, Kaori Ikematsu, Junichi Sato, Shota Yamanaka, and Kota Tsubouchi. Single-tap latency reduction with single- or double- tap prediction. *Proc. ACM Hum.-Comput. Interact.*, 7(MHCI), sep 2023.

[54] Jason R.C. Nurse, Oliver Buckley, Philip A. Legg, Michael Goldsmith, Sadie Creese, Gordon R.T. Wright, and Monica Whitty. Understanding insider threat: A framework for characterising attacks. In *2014 IEEE Security and Privacy Workshops*, pages 214–228, 2014.

[55] Oscar Oviedo-Trespalacios, Verity Truelove, and Mark King. "it is frustrating to not have control even though i know it's not legal!": A mixed-methods investigation on applications to prevent mobile phone use while driving. *Accident Analysis & Prevention*, 137:105412, 2020.

[56] Shidong Pan, Dawen Zhang, Mark Staples, Zhenchang Xing, Jieshan Chen, Xiwei Xu, and Thong Hoang. Is it a trap? a large-scale empirical study and comprehensive assessment of online automated privacy policy generators for mobile apps.

[57] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*, 2017.

[58] Orr Shauly, Gregory Stone, and Daniel Gould. The public's perception of the severity and global impact at the start of the sars-cov-2 pandemic: a crowdsourcing-based cross-sectional analysis. *Journal of Medical Internet Research*, 22(11):e19768, 2020.

[59] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, et al. Data breaches, phishing, or malware? understanding the risks of stolen credentials. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 1421–1434, 2017.

[60] Yuan Tian, Ke Zhou, and Dan Pelleg. What and how long: Prediction of mobile app engagement. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–38, 2021.

[61] Zhen Tu, Runtong Li, Yong Li, Gang Wang, Di Wu, Pan Hui, Li Su, and Depeng Jin. Your apps give you away: distinguishing mobile users by their app usage fingerprints. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):1–23, 2018.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[63] Chao Wang, Ronny Ko, Yue Zhang, Yuqing Yang, and Zhiqiang Lin. Taintmini: Detecting flow of sensitive data in mini-programs with static taint analysis. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 932–944. IEEE, 2023.

[64] Chao Wang, Yue Zhang, and Zhiqiang Lin. One size does not fit all: Uncovering and exploiting cross platform discrepant apis in wechat. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, USA, 2023. USENIX Association.

[65] Chao Wang, Yue Zhang, and Zhiqiang Lin. Uncovering and exploiting hidden apis in mobile super apps. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 2471–2485, New York, NY, USA, 2023. Association for Computing Machinery.

[66] Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820, 2021.

[67] Han Wang, Hanbin Hong, Li Xiong, Zhan Qin, and Yuan Hong. L-srr: Local differential privacy for location-based services with staircase randomized response. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2809–2823, 2022.

[68] Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. *Advances in Neural Information Processing Systems*, 34:23768–23779, 2021.

[69] Yiding Wang, Kai Chen, Haisheng Tan, and Kun Guo. Tabi: An efficient multi-level inference system for large language models. In *Proceedings of the Eighteenth European Conference on Computer Systems*, pages 233–248, 2023.

[70] Fan Wu, Yunhui Long, Ce Zhang, and Bo Li. Linkteller: Recovering private edges from graph neural networks via influence analysis. In *2022 ieee symposium on security and privacy (sp)*, pages 2005–2024. IEEE, 2022.

[71] Anhao Xiang, Weiping Pei, and Chuan Yue. Policychecker: Analyzing the gdpr completeness of mobile apps' privacy policies. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 3373–3387, New York, NY, USA, 2023. Association for Computing Machinery.

[72] Yue Xiao, Zhengyi Li, Yue Qin, Xiaolong Bai, Jiale Guan, Xiaojing Liao, and Luyi Xing. Lalaine: Measuring and characterizing {Non-Compliance} of apple privacy labels. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 1091–1108, 2023.

[73] Yuqing Yang, Yue Zhang, and Zhiqiang Lin. Cross miniapp request forgery: Root causes, attacks, and vulnerability detection. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3079–3092, 2022.

[74] Jeffrey Young, Song Liao, Long Cheng, Hongxin Hu, and Huixing Deng. {SkillDetective}: Automated {Policy-Violation} detection of voice assistant applications in the wild. In *31st USENIX Security Symposium (USENIX Security 22)*, 2022.

[75] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

[76] Lei Zhang, Zhibo Zhang, Ancong Liu, Yinzhi Cao, Xiaohan Zhang, Yanjun Chen, Yuan Zhang, Guangliang Yang, and Min Yang. Identity confusion in {WebView-based} mobile app-in-app ecosystems. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1597–1613, 2022.

[77] Xiaohan Zhang, Haoqi Ye, Ziqi Huang, Xiao Ye, Yinzhi Cao, Yuan Zhang, and Min Yang. Understanding the (in) security of cross-side face verification systems in mobile apps: a system perspective. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 934–950. IEEE, 2023.

[78] Yue Zhang, Yuqing Yang, and Zhiqiang Lin. Don't leak your keys: Understanding, measuring, and exploiting the appsecret leaks in mini-programs. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 2411–2425, New York, NY, USA, 2023. Association for Computing Machinery.

[79] Chaoshun Zuo, Zhiqiang Lin, and Yinqian Zhang. Why does your data leak? uncovering the data leakage in cloud from mobile apps. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1296–1310. IEEE, 2019.