Position Paper: Towards Ubiquitous and Automated User Privacy Configuration

Song Liao[†], Jingwen Yan[‡], Yichen Liu[§], David Kotz[¶], Luyi Xing[§], Long Cheng[‡]
[†]Texas Tech University, song.liao@ttu.edu
[‡]Clemson University, {jingwey, lcheng2}@clemson.edu
[§]University of Illinois Urbana-Champaign, {yichen59, lxing2}@illinois.edu
[¶]Dartmouth College, david.f.kotz@dartmouth.edu

Abstract—**Mobile apps may collect, share, and analyze data from users. Although users can choose to decline apps' data collection behaviors through mobile permission systems or in-app settings, it is challenging and time-consuming for users to manually discover and correctly configure all the privacy settings for apps on their mobile phones. This issue also occurs in IoT apps, where users need to configure each device separately. Although they can manage some settings with platform apps (like Apple Home), many IoT devices expose device-specific settings within a device-specific app. In this position paper, we propose the PrivacyProfile, a framework that allows users to easily set their global privacy preferences and apply them to apps automatically. Users can indicate whether each of their privacy-related information can be collected, shared, and analyzed in their profile. Compatible apps then read the privacy profile and automatically configure their settings for users, *e.g.*, enabling data collection behaviors or disabling data sharing. This design enables users to easily configure their privacy preferences once, rather than having to manually open each app and locate the corresponding privacy settings.**

I. Introduction

Mobile applications (apps), *e.g.*, Android and iOS apps, provide convenience for people's daily lives. Meanwhile, these apps also collect various types of personal data from users for app functionalities, such as name, email, and location. Additionally, some apps may need to share user data with third parties, track users' behavior, or provide personalized advertisements. Users can manage such privacy-related behaviors through system-level permissions or in-app settings. Apps can request different permissions from mobile systems, some of which are relevant to user privacy, such as user location or contacts. Apps may also ask users to provide personal information in app settings, where users can input their information, *e.g.*, name and email. Typically, these apps provide users with the option to enable or disable data collection, sharing, tracking, and analytics.

However, manually discovering and correctly configuring these settings in different apps remains a significant challenge for users. Prior work [13] shows that one-third of apps place their privacy options at a deep or "hidden" location, making it difficult for users to find and configure privacy-related settings. Other work shows that on average, each user has over 80 apps installed on their phone [4]. Therefore, users need to manually open and click multiple times in each app to configure privacy settings, which can be time-consuming. This issue also occurs in IoT apps, particularly when users frequently interact with IoT devices in their homes nowadays. Since each IoT device may collect different types of data from users, users need to configure each device separately. Although users can manage some settings with platform apps, *e.g.*, Amazon Alexa, Google Home, and Apple Home, many IoT devices can only be set up and configured within the device-specific app. Therefore, users still need to review and configure privacy settings for IoT devices in different apps. As a result, users must spend significant time adjusting privacy settings, often without knowing all available settings or where to locate them. Overall, the lack of transparent and user-friendly privacy controls creates a significant burden for users and increases the risk of unintended data exposure.

In this work, we propose a framework named PrivacyProfile to enable ubiquitous and automated configurations of user privacy preferences. Each user can set up a unique privacy profile on their mobile phone that includes all privacy-related configurations, such as whether specific data can be collected, shared, and analyzed. After that, all the compatible apps can read this privacy profile and understand the user's privacy preferences. Then, apps can automatically configure their privacy settings for the user, saving the user's time. Our work does not aim to replace the existing mobile permission system or platform-level privacy protections. We position PrivacyProfile as a complementary and user-driven framework that allows users to specify high-level privacy preferences once and apply them consistently across apps.

II. Background

Privacy regulations such as the GDPR [2] and CCPA [1] emphasize users' rights to control how their personal data is collected and used. In practice, mobile platforms provide such user control primarily through system permissions (Section II-A) and in-app settings (Section II-B).

Workshop on Security and Privacy in Standardized IoT (SDIoTSec) 2026
23 February 2026, San Diego, CA, USA
ISBN 978-1-970672-01-5
https://dx.doi.org/10.14722/sdiotsec.2026.23043
www.ndss-symposium.org

A. Mobile Permission System

Current mobile systems, such as Android and iOS systems, provide a permissions system for users to control certain types of data (*e.g.*, Contact, Camera and Location) [6]. In Android and iOS systems, users can manage application permissions through system settings, typically via *Settings* → *Apps* → *[App Name]* → *Permissions*. Permissions are organized by type within one app, allowing users to view and control each permission type. Most permissions are presented as binary toggles (grant or revoke access). These controls operate at the application level, requiring users to configure permissions separately for each installed app.

Despite both Android and iOS offering system settings that make permission management more accessible to users, these two permission systems remain limited. First, the current permission system primarily focuses on controlling whether an application can access a certain type of data, rather than how the data will be used after access is granted. Once a permission is granted, users have limited visibility into or control over subsequent data practices, such as whether the collected data is stored, shared with third parties, or used for purposes beyond the app's core functionality. Second, users are required to manually manage permissions on a per-app basis, making it difficult to enforce consistent data preferences across multiple applications. This design further highlights the absence of a global setting that allows users to express unified permissions management across apps.

B. In-app Privacy Settings

Beyond system-level permissions, most mobile applications offer in-app privacy settings that enable users to control how their personal data is processed within the application. Unlike system permissions, which regulate access to a fixed and predefined set of permission types, in-app privacy settings are designed around an app's specific data practices and functional features. As a result, in-app privacy settings often provide more fine-grained and app-specific controls that reflect the app's unique functionality. For example, smart home IoT applications commonly allow users to configure whether data collected by in-home devices (*e.g.*, security cameras, microphones, or motion sensors) are continuously recorded or only triggered by specific events, or whether such data are stored locally on the device or uploaded to cloud servers. In addition, in-app privacy settings may also provide data-usage controls that are not available in system settings, *e.g.*, whether user data can be shared with third parties, used for targeted advertising, or used for analytics and personalization purposes. However, previous work has demonstrated the difficulties users face in manually configuring their privacy settings in apps, especially for hidden settings [10], [13], [19], [22]. For example, Chen *et al.* [13] found that nearly half of the examined privacy settings were hidden from direct user access, and 42.83% of these hidden settings were missed by at least one participant during user studies. This finding shows the difficulty and effort required for users to locate and configure each app's settings.

III. Security Model

Our approach aims to improve the ability of individual users to manage their data privacy, in the context of smart-home devices and applications. We assume that the user has all the necessary apps installed on their smartphone, and that the apps (collectively, and with the OS) include all the settings necessary for the user to express their preferences for data collection and use. Furthermore, we assume that the apps, smartphone OS, and back-end systems (such as cloud platforms operated by vendors of the phone, apps, and IoT devices) are honest in honoring the user's settings. Finally, we assume that the smartphone, IoT devices, and cloud systems, are all secure against compromise by any adversary. Thus, from the point of view of privacy and security, our work is focused on enhancing the *usability* of data-privacy management. Other methods, out of scope, are needed to assure the security of the software and hardware involved, and to enforce data-privacy settings. We focus on honest apps that are willing to adopt and follow our proposed mechanism. However, our framework cannot prevent malicious apps that ignore user privacy preferences. [1]

IV. System Design

A. Overview

Figure 1 shows the overview of our designed framework. We assume that users hope to control their data on mobile phones but are unwilling to manually configure privacy settings for each individual app. First, a user creates a privacy profile, which includes the user profile id and creation time. The privacy profile will be stored in OS-protected storage to protect its integrity. The user specifies their privacy preferences in the profile, such as whether they want each type of data to be collected, shared, or analyzed. Users can update their privacy profiles at any time. Second, the privacy profile is written in JSON format and stored in OS-protected storage to maintain its integrity, while also allowing other apps to access it. Later, when an app is installed, it asks the user for consent to access the privacy profile (but cannot edit it) and configures its own settings automatically. Apps periodically re-read the file and update their internal settings to match any changes in the profile. If the user allows apps to collect certain data from the smartphone's OS or sensors, the app still needs to request permission from the mobile OS to access such data, as PrivacyProfile complements rather than replaces the underlying permission system by specifying how the data should be handled after access is granted. Following such a framework, the user needs only to set their privacy profile in one place; thereafter, all apps will follow the preferences. The privacy profile doesn't store any user data but only specifies the rules to access user data. As a result, the PrivacyProfile mechanism doesn't introduce additional risks of direct user data leakage. After obtaining users' privacy preferences, apps still need to request system-level permissions and user approval to access specific data types. Overall, PrivacyProfile

[1] Addressing such behavior would require OS-level support to tag and control the flow of sensitive information within the app – and beyond, into cloud services. Such methods are beyond the scope of this work.

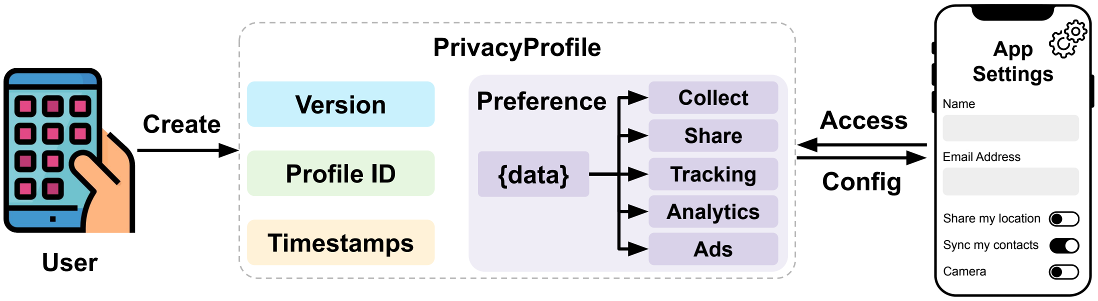

Figure 1: Overview of the framework.

can reduce the time and effort required for users to manage privacy preferences.

B. PrivacyProfile *Creation*

The PrivacyProfile creation process is designed to be simple for users while supporting flexible and fine-grained specification of privacy preferences. Users can create their privacy profile through a PrivacyProfile generator app, which automatically creates a default profile that sets maximum privacy. The PrivacyProfile generator app presents high-level privacy categories, *e.g.*, data collection, data sharing, tracking, analytics, and personalization, with human-readable descriptions. PrivacyProfile app will aggregate settings registered by each app on the phone, which are then organized and presented to the user. When a new app is installed, it registers its list of settings with PrivacyProfile, which then alerts the user to any new settings they need to review and update their profile.

Users can specify their privacy profiles through two approaches in the PrivacyProfile generator app. First, users can easily create a profile using a predefined template, which already includes a comprehensive list of privacy-relevant data types and configuration preferences (more details in Section IV-C). Users can directly edit it and select which data they allow or disallow to be collected and shared with third parties. Alternatively, we anticipate PrivacyProfile enabling users to express their desired preferences in natural language, which is then processed by large language models (LLMs) to automatically generate a list of privacy preferences. For example, users can simply say, "I don't want any of my contact information to be shared with others." The generator will produce the corresponding rules, such as: "Name/Email/Phone Number: collected: (empty), shared: deny". Users can view, approve or adjust the automatically generated preferences in the PrivacyProfile generator app.

PrivacyProfile Update: Users can update their privacy profiles at any time using the PrivacyProfile app, which displays the current configuration. Apps periodically re-read the file and update their internal settings to match any changes in the profile. Users can also choose to reset their profile, which cleans all the existing privacy preferences.

C. PrivacyProfile *Storage*

After users specify their choices, the PrivacyProfile app generates a structured and machine-readable privacy profile. To enable portability and developer accessibility, PrivacyProfile adopts a JSON-based schema. At a high level, the profile describes how an application is permitted to use a specific type of data (after obtaining OS-level permission to access that data, if necessary).

As shown in Listing 1, the top-level fields include the schema version, the user profile ID (an identifier for the profile), and the timestamps for creation and update. The preferences include a dictionary that maps a key, which is a data type, to a set of policy fields. For each key, PrivacyProfile encodes usage control policies such as:

- collect: whether an application is allowed to collect this data type at all (e.g., "allow", "deny").
- share: whether the collected data may be shared with third parties (e.g., "deny", "first_party_only", "anonymized_only").
- tracking: whether tracking or cross-app profiling is permitted.
- analytics_level: the allowed granularity and upload behavior for analytics and telemetry.
- personalized_ads: whether personalized advertisements are allowed.

Listing 1: Example PrivacyProfile schema

```
{
    "version": "1.0",
    "profile_id": "user-1",
    "created_at": "2025-01-10T12:34:56Z",
    "updated_at": "2025-02-01T09:12:00Z",
    "preferences":{
        "pii.email":{
            "collect":"deny",
            "share":"deny",
            "tracking":"deny",
            "analytics":"minimal",
            "ads":"off"
      },
        "pii.phone_number":{
            "collect":"allow",
            "share":"first_party_only",
            "tracking":"deny",
            "analytics":"minimal",
            "ads":"off"
      },
    ...
    }
}
```

Importantly, the privacy profile itself doesn't include any personal information (*e.g.*, email or phone number) but only defines the rules for how apps process user data. This separation enables PrivacyProfile to offer user-driven privacy semantics while maintaining compatibility with existing mobile platforms.

We anticipate PrivacyProfile will include common privacy behaviors across mobile applications, as related to common data types: Personally identifiable information (PII) (*e.g.*, name, email); Device identifiers (IMEI, MAC address); Behavioral data (*e.g.*, usage logs, interaction patterns); and sensitive data categories (*e.g.*, health data, location). In future work, drawing on previous data schema [8], [20], [23], we will generate a list of common data types and behaviors to use as a template for the PrivacyProfile app.

D. PrivacyProfile *Access Control*

When users download and install a new app, it needs to request access to the privacy profile. Users also need to manually approve the attempt, ensuring that no application can read the profile without user awareness and protect the integrity of PrivacyProfile. New applications can also "subscribe" to the privacy profile so that they can receive a message with a copy of the privacy profile at first, and then whenever it changes. In addition, PrivacyProfile provides a schema for apps so that they can "register" their settings with PrivacyProfile, some of which may be common and some of which may be device-specific. Then, PrivacyProfile extends its internal schema to incorporate the settings from new apps, and alerts (if necessary) the user to review and adjust the new settings. After that, it also alerts other apps that depend on the same settings.

After accessing the user's privacy profile, the application interprets the privacy preferences and compares them to the data that the app requires. The unified fields, including data collection, sharing, tracking, analytics, and advertising, allow each app to systematically map each preference to its own privacy settings. Then the app automatically configures its privacy settings based on the privacy preferences in the profile. For user-denied preferences, the app will automatically turn them off, such as turning off the data sharing of certain data or data tracking. For the data types that are provided by the mobile OS, such as user location, which the privacy profile allows an app to collect, the app needs to request permission from the mobile system to access the specific data content.

V. Discussion

Currently, the permission systems on mobile phones only regulate whether an app can access specific user data, without considering how applications will process, share, or track the collected data after users approve the permission. Our proposed framework advances the permission system by providing a unified, user-controlled specification of data handling preferences that apps should enforce after obtaining permissions. It allows users to globally set up **"how an app will handle the data if allowing accessing it"**, which is a higher-level and user-driven policy that complements the current permission system.

In our preliminary analysis, we manually analyzed 20 popular apps from Google Play, in which we identified 52 privacy-related preferences in app settings. This indicates that users need to manually open and configure an average of 2.6 privacy settings per app. Moreover, prior work [13] shows that many apps place their privacy options at a deep location, further increasing the time and effort required to locate and adjust these privacy settings. In contrast, with the help of our proposed PrivacyProfile, users only need to configure their privacy profiles once and apps will automatically apply the privacy preferences to their internal settings, significantly reducing the time and effort required for users to configure multiple apps.

Compared with general mobile apps, IoT apps have specific challenges, such as heterogeneous device capabilities, continuous sensor data, multi-user environments, device autonomy, and context-dependent data collection. In addition, privacy-relevant configurations can differ substantially across applications and devices in IoT ecosystems. Such challenges motivate additional design considerations in PrivacyProfile, which requires more flexible and fine-grained mechanisms to support consistent privacy control across diverse IoT devices and usage scenarios.

In the future, we plan to develop a user-friendly app that integrates our proposed mechanism and enables users to easily create, update, and manage their PrivacyProfile. We also plan to conduct a comprehensive user study to evaluate whether the system aligns with users' expectations and to identify any scenarios that may have been overlooked, such as dishonest apps or developers who are unwilling to adopt our proposed mechanism. In addition, we aim to collaborate with mobile app developers to assess the feasibility of integrating our proposed mechanisms into real-world development workflows.

VI. Related Work

Prior work on privacy control explores how users can explicitly express and enforce their privacy preferences across systems and platforms. Several works focus on privacy profile–based approaches, where users define reusable privacy preferences that can be automatically enforced, such as the Platform for Privacy Preferences (P3P) [3], [14], [18] which introduced machine-readable privacy preferences that enable users to specify their choices once and apply them across services. Subsequent research further investigates data-centric, attribute-based, and purpose-based privacy profiles to regulate access to IoT data and services [7], [9], [15]. Building on these ideas, PRIPRO [11] adopts profile-based designs in mobile and smart environments, which enables users to predefine context-aware privacy profiles that are enforced through system-level permission control. Other works extend privacy control to shared and multi-user settings by incorporating both individual and group privacy preferences [5], [12]. Finally, from a platform and regulatory perspective, privacy control is studied as a policy instrument that balances data integration and privacy costs through standardized or portable privacy preferences [24]. Complementary work further highlights practical challenges of enforcing privacy control in smart home and IoT environments, including highly variable device behaviors, context-dependent data sharing needs, and everyday misuse by household members [16], [17], [21].

VII. Conclusion

In this paper, we argue that existing mobile permission systems are insufficient to capture users' expectations about how their data should be handled after access is granted. We position the PrivacyProfile as a complementary and user-driven framework that enables users to specify high-level privacy preferences once and apply them consistently across apps. We don't aim to replace the current mobile permission systems or enforce compliance against malicious apps. Our approach focuses on reducing configuration burdens and ensures the ubiquitous and automated user privacy configuration, which allows users to globally set up "how an app will handle the data if accessing it". We hope this position encourages further research and standardized, automated, and user-centric privacy configurations across mobile and IoT ecosystems. For future work, we plan to conduct a comprehensive user study to evaluate whether the system aligns with users' expectations.

Acknowledgment

This work is partially supported by National Science Foundation (NSF) under award numbers 1955805, 2545822, 2523814, 2448205, 2239605, 2228616, and 2449402. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF. Any mention of specific companies or products does not imply any endorsement by the authors, by their employers, or by the NSF.

References

[1] California Consumer Privacy Act (CCPA). https://oag.ca.gov/privacy/ccpa.

[2] General Data Protection Regulation. https://gdpr-info.eu.

[3] The Platform for Privacy Preferences . https://www.w3.org/TR/P3P/.

[4] What the Tech? Too Many Apps. https://www.kaaltv.com/6-on-your-side/what-the-tech/what-the-tech-too-many-apps/.

[5] Khaled Alanezi and Shivakant Mishra. Incorporating individual and group privacy preferences in the internet of things. *Journal of Ambient Intelligence and Humanized Computing*, 13(4):1969–1984, 2022.

[6] Iman M Almomani and Aala Al Khayer. A comprehensive analysis of the android permissions system. *Ieee access*, 8:216671–216688, 2020.

[7] Morteza Amini and Farnaz Osanloo. Purpose-based privacy preserving access control for secure service provision and composition. *IEEE Transactions on Services Computing*, 12(4):604–620, 2016.

[8] Mehdi Bahrami and Mukesh Singhal. Cloudpdb: A light-weight data privacy schema for cloud-based databases. In *2016 International Conference on Computing, Networking and Communications (ICNC)*, pages 1–5. IEEE, 2016.

[9] Bruhadeshwar Bezawada, Kyle Haefner, and Indrakshi Ray. Securing home iot environments with attribute-based access control. In *Proceedings of the Third ACM Workshop on attribute-based access control*, pages 43–53, 2018.

[10] Amel Bourdoucen and Janne Lindqvist. Privacy of default apps in apple's mobile ecosystem. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–32, 2024.

[11] Jonas Cesconetto, Luís Augusto Silva, Fabricio Bortoluzzi, María Navarro-Cáceres, Cesar A. Zeferino, and Valderi RQ Leithardt. Pripro—privacy profiles: user profiling management for smart environments. *Electronics*, 9(9):1519, 2020.

[12] Antorweep Chakravorty, Tomasz Wlodarczyk, and Chunming Rong. Privacy preserving data analytics for smart homes. In *2013 IEEE Security and Privacy Workshops*, pages 23–27. IEEE, 2013.

[13] 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.

[14] Lorrie Cranor. *Web privacy with P3P*. " O'Reilly Media, Inc.", 2002.

[15] Johan Fernquist, Torbjörn Fängström, and Lisa Kaati. Iot data profiles: The routines of your life reveals who you are. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 61–67. IEEE, 2017.

[16] Weijia He, Kevin Bryson, Ricardo Calderon, Vijay Prakash, Nick Feamster, Danny Yuxing Huang, and Blase Ur. Can allowlists capture the variability of home iot device network behavior? In *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*, pages 114–138. IEEE, 2024.

[17] Weijia He, Nathan Reitinger, Atheer Almogbil, Yi-Shyuan Chiang, Timothy J Pierson, and David Kotz. Contextualizing interpersonal data sharing in smart homes. 2024.

[18] Marc Langheinrich. Privacy by design—principles of privacy-aware ubiquitous systems. In *International conference on ubiquitous computing*, pages 273–291. Springer, 2001.

[19] Jialiu Lin, Bin Liu, Norman Sadeh, and Jason I Hong. Modeling {Users'} mobile app privacy preferences: Restoring usability in a sea of permission settings. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 199–212, 2014.

[20] Siyuan Liu, Qiang Qu, Lei Chen, and Lionel M Ni. Smc: A practical schema for privacy-preserved data sharing over distributed data streams. *IEEE Transactions on Big Data*, 1(2):68–81, 2015.

[21] Phoebe Moh, Pubali Datta, Noel Warford, Adam Bates, Nathan Malkin, and Michelle L Mazurek. Characterizing everyday misuse of smart home devices. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2835–2849. IEEE, 2023.

[22] Kopo M Ramokapane, Anthony C Mazeli, and Awais Rashid. Skip, skip, skip, accept!!!: A study on the usability of smartphone manufacturer provided default features and user privacy. *arXiv preprint arXiv:2308.14593*, 2023.

[23] Monica Scannapieco, Ilya Figotin, Elisa Bertino, and Ahmed K Elmagarmid. Privacy preserving schema and data matching. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 653–664, 2007.

[24] Alessandro Sellitto. The regulatory trade-off: Data integration, privacy costs and welfare in digital platforms.