A Holistic Framework for Enhancing Privacy Awareness

A. Alshehri1, N.L. Clarke1,2 & F. Li1,3
1Centre for Security, Communications & Network Research (CSCAN), Plymouth University, United Kingdom;
2Security Research Institute, Edith Cowan University, Western Australia
3School of Computing, University of Portsmouth, Portsmouth, United Kingdom
aziz.alshehri@plymouth.ac.uk

***Abstract*—Home users face increasingly higher risks of privacy loss and struggle with the difficult task of protecting large volumes of personal information. Most privacy research assumes that users have uniform privacy requirements. The main problem with this approach is that research has also shown that users have different privacy attitudes and expectations based upon a variety of factors, including (but not limited to) gender, age and education level. Privacy therefore can mean different things in different contexts, to different people at different times. For example, some uses are less concerned regarding the sharing and use of their location information while others will be very concerned. Therefore, it is important to factor these requirements in to a privacy-awareness model that can enhance user's awareness and make more informed decisions to reduce their specific degree of exposure. The quantity and range of sensitive information also requires approaches that give users back the control over their data. Therefore, prioritization of privacy-related information based on an individual user basis should be utilised to ensure relevant and timely notification about privacy-related information that is important to the user. This paper presents a critical analysis of the current state of the art and proposes a novel mobile-based architecture to provide users with effective and usable privacy protection.**

Keywords— Privacy in mobile computing; Context-aware privacy control.

I. INTRODUCTION

The number of Internet users has increased dramatically since the millennium, with the first billion users being reached in 2005, the second billion by 2010, and the third in 2014 [1]. The growth of users has also coincided with an increase in the range of devices that can access the Internet – with smartphones, smart watches, tablets, smart TVs, game consoles all being very popular. Indeed, 60% of UK adult online users use at least two devices every day and nearly 25% use three devices [2].

However, with the rapid growth of these devices, activities, services and information, the enormous amount of private and personal information that is stored has also increased. Therefore, users are becoming increasingly concerned about their personal information, how it is used, by whom and where it is stored [3]. For instance a Consumer Report found that 92% of British and U.S. Internet users are concerned about their privacy online [4]. When users became aware of online privacy issues, they were asked what made them most worried about their online privacy: 45% of British Internet users stated that it is personal information being shared between companies [4]. In addition, 89% of users avoided these companies because they believed that companies do not protect their privacy. It was also found that 76% of Internet users limited their online activity in the last 12 months due to these concerns [4]. This evidence indicates that users are sufficiently worried about their online privacy.

Users are also concerned about lack of control over their personal information as they are often unaware of what information an application collects about them [5]. Fifty percent of UK-based Internet users reported they wanted control over who has access to their personal information [4]. The lack of control over their personal information may decrease willingness to share sensitive information. For example, a study by Brandimarte et al. found when users who have control over their personal information, are willing to share more [6].

Due to users concerns about privacy protection, most mobile operating systems such as Android and iOS provide some privacy safeguards for users [7]. Despite these provisions, there are several usability issues related to the functionality and interface. For instance, Kelley et al. found that users struggle to understand the permissions in Android due to the lack of usability [7]. Furthermore, several studies have shown that privacy interfaces, whether for iOS or Android, did not provide users with sufficient information or control [8–10]. Therefore, the Federal Trade Commission suggested privacy controls need more improvement to protect users' privacy [11].

Focus has been given to the development of policies, procedures and tools that aid an end-user in managing and understanding their privacy-related information. However, these approaches assume that users can correctly configure all resulting settings and they have uniform privacy requirements. In reality, users do have different privacy concerns and requirements as they have heterogeneous privacy attitudes and expectations [12]. For example, some users consider personal information such as age, address and gender in their profile on a social network being more sensitive than others [13]. Furthermore, in practice it is unrealistic to assume homogeneous privacy requirements across a whole population [14].

Accordingly, there is a need to for an approach that considers individual requirements in a centralised and usable manner to meet users' needs. This paper, building upon an analysis of the current state of the art, will identify the requirements of such a privacy tool and propose a framework to realise it.
Section 2 presents an analysis background literature, followed by Section 3 presenting the problem statement. The proposed framework is presented in Section 4. The conclusions and future work are presented in Section 5.

II. BACKGROUND LITERATURE

Most of the prior studies focus upon how to develop useful techniques to detect and manage the leakage of sensitive personal information. Numerous techniques have been proposed to monitor personal information, for example, monitoring permissions, mobile network traffic and static and dynamic analysis. A total of 14 papers have been identified and categorised into three domains based on the purpose of each tool:

- Information flow analysers (4 papers):
- Finer grain privacy controls (8 papers):
- Privacy profiles (2 papers):

A. Information flow analysers:

Many tools have been developed in recent years that aim to analyse and to detect the personal information on mobile platforms. These tools analyse mobile apps regarding potential privacy breaches before they leave the system via untrusted apps. Some of the more prominent examples are, Taintdroid [15], AppIntent [16], Little BrothersWatching You, and PiOS

Taintdroid was designed to detect the sensitive data when it leaves the system via untrusted applications [15]. It was designed based on a dynamic approach which is executed whilst a program is in operation. The system can track the flow of data through four levels: variable, method, message, and file. Although, TaintDroid detects the sensitive data, the system assumes that users can correctly configure all the resulting settings. Therefore, this approach could impose an undue burden on the users. In addition, they do not examine the usability related to the interface displayed to users.

In comparison, [17] presented a solution that focuses on the user's awareness of privacy issues. The solution improves user's understanding of potential privacy leakages. It is built based upon the TaintDroid platform and helps users to know the frequency and destination of data being shared by an application. It also provides users interfaces in order to inform users about which privacy sensitive information leaves the phone. However, they do not provide users control over their personal information to allow them to specify which type of information that they prefer not to leave the phone.

Unlike previous studies that simply consider the transmission of private data, AppIntent determines if transmission is user intended or not because transmission of sensitive data in itself does not necessarily indicate privacy leakage [16]. AppIntent was designed to distinguish between user-intended data transmission from user unintended and develop an event-space constraint guided symbolic execution technique. This technique can reduce the event search space in symbolic execution for Android apps. Symbolic execution is used to determine the right input that makes the program to execute. AppIntent develops a new symbolic execution technique called event-space constraint guided symbolic execution for Android apps in order to avoid the possibly of the path explosion problem during symbolic execution. The researchers apply static analysis first to identify the possible execution paths that lead to the sensitive data transmission under analysis (such as sending SMS). Then they use these paths to generate event-space constraints. However, the evaluations should conduct with varying number of participants in order to identify the user acceptance of the system.

Another tool that aims to analyse programs for possible leaks of sensitive information from a mobile device to third party is PiOS. It detected privacy leaks related to device ID, location and phone number. Moreover, PiOS considered the address book, browser history, and photos. PiOS uses static analysis to detect data flows. They have analysed more than 1,400 iPhone apps and they found that a majority of apps leak the device ID, which can provide detailed information about the habits of a user. However, PSiOS does not provide user a fine grain control over their personal information.

B. Finer grain privacy controls:

A number of research prototypes have also offered used fine grain controls in order to prevent potential privacy leakages. For example, AppFence [18], TISSA [19], AntMonitor [20] and ProtectMyPrivacy [21]. There are several techniques were used to enhance information flow control for mobile.

AppFence uses replacing information approach in order to protect sensitive data [18]. AppFence provides users two privacy controls to protect sensitive resources: shadowing and blocking. Sometimes users do not want to provide application access to sensitive data. Therefore, AppFence sends shadow data instead of the actual data. For example, when application requires access to user's contacts, AppFence may provide application shadow data that contains no contact entries, contains only those genuine entries not considered sensitive by the user, or that contains shadow entries that are entirely fictional. The second approach for protecting sensitive data is blocking sensitive data from being exfiltrated off the device. AppFence uses TaintDroid information flow tracking to track the sensitive data and prevent information from transmissions of these data out of the device. However, the system does not alert users about how applications use data and whether they will exhibit side effects if privacy controls are applied. In order to know whether side effects impact user-desired functionality, it needs to consult users each time. In this case, the system may places a high level of burden on users.

The Taming Information Stealing Smartphone Applications (TISSA) provides user with a fine-grained control over disclosure of their personal information and consists of three main components [19]. TISSA was designed to protect four types of personal information: phone identity, location, contacts, and call log. The first one is the privacy setting content provider. It contains the current privacy settings for untrusted apps on the mobile device. It also provides users an interface in order to query the current privacy settings for an untrusted app (e.g., a location manager). In order to protect the personal information, TISSA provides users the empty or bogus options for personal information that may be requested by the app. The second component is the privacy-setting manager. It allows users to manage or update the privacy settings for installed apps. The third component contains content providers or services to regulate the access for four types of personal information: phone identity, location, contacts, and call log. For example, when an app requires access to private data, the system will query the privacy settings, and response to the requests according to the current privacy settings for the app. However, it is difficult for an average user to determine which type of permission is high or low risk for the app because he does not know the reason about permission requirements for individual apps. Additionally, the system does not assist the user to make the right choice in order to reduce the burden on mobile users.

In order to enhance privacy control, DROIDFORCE proposed another approach to enforce privacy controls based on a user's policy [22]. DROIDFORCE works at the application level. It targets apps with static data flow analysis to identify strategic policy enforcement points whether for a single application or for multiple applications at the same time. These policies may depend on data that available only at runtime. However, their policies allow or deny an activity, while do not provide users with finer control over the information. Additionally, the study does not show how users had could understand these policies to make informed decision.

PrivacyGuard [23] and AntMonitor [20] provide fine grained privacy control and provide ground truth mapping of packets to applications. They used an approach which analyses actual network traffic of Android using VPNService API to intercept traffic. This approach does not require root permissions and is portable to all devices with Android version 4.0 or later. The AntMonitor system consists of three components: an Android application, AnyClient, and two server applications, AntServer and LogServ. Whilst PrivacyGuard runs in its entirety on the local device. The purpose of the client-side analysis is to protect users in real time and provide fine grained privacy control. However, LogServer works as the central repository to store and analyse all network traffic data and does not have to analyse a large amount of live traffic compared to AntServer. To evaluate AntMonitor system, they recruited student volunteers to use AntClient on their phones. The system collects the packets of the applications that the volunteers selected and stores them at LogServer in order to check whether any of the installed applications are sending the personal data out to the Internet. They found that 44% and 66% of the users have applications that leak their International Mobile Equipment Identity (IMEI) and Android Device ID respectively. However, both PrivacyGuard and AntMonitor assume that average users can correctly specify their personal information to allow the system to detect them when they leave their mobiles. In this case, these solutions does not help user to overcome the burden associated with managing such a large number of data.

ProtectMyPrivacy (PMP) provides users fine grained privacy for each app in order to send the anonymized data instead of privacy sensitive information [21]. It detects privacy leaks on iOS Applications. The type of the data that PMP protects is unique device identifier, IMEI, Wi-Fi MAC address and Bluetooth MAC address. Another private data type that PMP protects is the user's address book. It includes names, addresses, phone numbers and emails because some apps upload these information to a server without user's permission. When the app wants to access to the private data, PMP allows the user to deny or allow the app to access private data in real time. Hence, PMP provides user two options to protect his address book: user can allow the app to access his address book or allow PMP to sends an alternative address book, filled with fictitious entries (names, emails and phone numbers). Additionally, they have developed a crowdsourcing system to help user to make informed decisions, which provides app specific privacy recommendations. However, the system just deals with mere access to private data, but does not address privacy once the data leaves the app. Moreover, the system does not provide each user personalized recommendations. Each user has its own privacy preferences. Therefore, it would be helpful to take account of user's profile when the system generates recommendations, in order to make a more personal recommendation.

Other studies have proposed approaches that do not rely on one operating system but can run on different mobile systems. Nadkarni and Enck [24] proposed Aquifer as a policy framework in modern operating systems such as Android, iOS, and Windows 8, which performs two types of restrictions that protect the entire User Interface (UI) workflow defining the user task and ensure only specific apps can export the data to the host. Aquifer provides each application a control over sensitive data therefore can contribute to the security restrictions. However, Aquifer does not show users the privacy policy of an application. Additionally, it does not provide users comprehensive tracking the sensitive data because just focus on the UI workflow that send data to another application.

Labyrinth [25] also supports both Android and iOS. Labyrinth a system to detect access to private data by using privacy enforcement system that automatically detects leakage of private data originating from standard and application-specific sources. Labyrinth contains a Packet Analyser that collects all the data application that sends to any remote Server. When the data is collected, Packet Analyser detects if private data has been sent in the clear. Then the system will terminate any unauthorized communication of private data in the clear between the client and the server via the proxy interface. Moreover, Labyrinth is equipped with an integrated Visual Configuration Framework in order to identify the private information that Labyrinth should protect. Therefore, when application access the private information via user input or through standard libraries, Labyrinth compared the data that is collected at run time by the Packet Analyser with the data that is collected by the instrumentation layer. If a match is found, a confidentiality warning is reported. Visually configuring, is directly atop the application's UI and it does not require operating-system instrumentation. In order to improve the usability of the security administration of a mobile application, the visual configuration provides users the type of data that may leak from application at run time. However, it is not feasible to notify users for each leak from application at run time. Constant notification for each leak from application may affect the user acceptance of the system. Additionally, configuring the privacy policy each time may places a high level of burden on users. It would be helpful to take account of users' choices, in order to make a more personal policy.

C. Privacy profiles:

A few studies have proposed modelling and predicting users' privacy preferences [26] [27]. Frank et al. cluster 188,389 Android apps and 27,029 Facebook apps to find patterns in permission requests [26] . They used a probabilistic method to extract permission request patterns from Android and Facebook apps. They identified over 30 common patterns of permission requests. However, they looked for permission request patterns in Android apps but they do not identify patterns in user privacy preferences.

Liu et al analysed the permissions users granted to mobile apps on Android and realized that the permission model is too complex and they can find patterns in permission requests [27]. They used machine learning clustering algorithms to split users into a small number of profiles based on their decisions to grant or deny apps the access to different permissions. Their result showed that it is possible to significantly reduce user burden while allowing users to better control their mobile app permissions. However, they do not elicit user's privacy preferences in a context where they are not just about the permissions requested by an app but also about type of information, app categories, data location, time of access data, the entity access to data, data usage and the level of data.

III. Discussion

The aforementioned studies aim to alert users by developing tools in order to emphasise privacy violating information on many applications. The majority of existing research has focussed upon the technical aspect to protect the privacy of users. They have shown that is possible to monitor sensitive information for users in real time. The majority of studies used a dynamic approach to monitor personal information for users. Whilst, a few studies used network approach to detect information leak in mobile as shown in the Table 1.

A number of research prototypes have not only monitored sensitive information for users but also provided user control over the personal information such as AntMonitor [20], ProtectMyPrivacy [21] and Labyrinth [25]. However, most studies have often assumed that users have uniform privacy requirements. Current research approaches to privacy are usually fundamentally static in nature. By contrast, personal information is dynamic because privacy preferences are diverse from time to time. For instance, some users are willing to share their locations for period of time with some groups such as close friends, family, Facebook friends and friends at work. Another example, some users may share their locations during certain hours of the day or days of the week. Therefore, there are a number of critical dimensions to these preferences, including time of day, day of week, and relevant groups.

Furthermore, a number of studies on the assessment of privacy risks assume that the value of each personal information is perceived similarly across all information. Few studies measure the users' privacy risk when they use applications. Some have used colour to inform user about the level of the risk. Other studies use the score to identify the level of the risk. However, it is difficult to quantify and measure privacy. First, the level of privacy differs from user to user. Second, users' preferences may change over time. In addition, the prior art has not presented a holistic assessment but rather focusing upon one aspect of privacy such as web or mobile applications. This makes privacy measurement more challenging.

TABLE I. A REVIEW FOR THE MONITORING TECHNIQUES AND PRIVACY CONTROL

N	**System**	**Monitoring Techniques**			**Privacy control**
		Network	***static***	***dynamic***	
1	TaintDroid			✓	
2	Little BrothersWatching You			✓	
3	AppFence		✓		✓
4	TISSA		✓		✓
5	AppIntent			✓	
6	DROIDFORCE			✓	✓
7	PrivacyGuard	✓			✓
8	AntMonitor	✓			✓
9	PiOS		✓		
10	PMP			✓	✓
11	Aquifer			✓	✓
12	Labyrinth	✓			✓

Moreover, some approaches allow users to know potential risk to their privacy and invite them to change their settings such as AntMonitor system [20]. However, it is difficult for average users to identify the level of the privacy – in particular, when the user changes privacy preferences from time to time. This could place a burden on users and could have a negative effect on initial adoption.

From the usability prospective, a few studies related to privacy focus on usability issues in order to increase the user awareness. The most prevalent method for privacy alerts is to inform users about potential single data leakage. This method does not consider long-term aspects such as frequency of access to the personal information. For example, some applications may access the location every hour for one week whilst users may not allow the application to access it in specific occasions for personal reasons. The evaluations of these studies do not conduct with varying groups of participants in order to identify the user acceptance of the system. In addition, the studies lack of statistical power because the total numbers of participants were small. None of these studies introducing a history view to allow users to know who has accessed their data when and at which degree of granularity. Current approaches present the same content of the interface for all users. Therefore, current approaches design static user interfaces. Some users may want to know more information about the potential privacy risks while others may not want to take the time to read the warnings in detail.

IV. SYSTEM ARCHITECTURE

It is envisaged that a novel architecture for user privacy will encompass the core functionality to allow for personalisation, adaptive interfaces and user feedback. Privacy preferences are diverse and cannot adequately be captured by one size-fits-all default settings because the level of privacy differs from user to user. Eventually, this needs to result in a privacy profile/configuration unique to each individual. However, understanding and adapting to an individual's specific preferences is challenging without overly burdening them at the initial setup. Therefore, in order to cater to different user preferences and expectations initially, user profiling could be utilised to cluster users into a smaller number of privacy profiles. Moreover, applying privacy profiles as default settings for initial interfaces could significantly reduce burden and frustration of the user. The system also will provide users with an adaptive and usable user interface in order to increase user awareness about potential privacy risks. As users are different and therefore have different needs from an interactive system. Adaptive interfaces can assist in providing personalisation and supporting flexibility. Figure 1 illustrates an architecture beginning with an initial interface that displays a series of questions related to users' demographic information, aiming at clustering users into a smaller number of privacy profiles. Then, the system will update personal privacy preferences of the user based on their interactions. Updating personal privacy preferences would enable the system to create individual privacy profiles in order to adapt an individual's specific preferences without overly burdening them. The system also provides user adaptive interface. In this case, the system could meet personal privacy preferences and personal visualisation.

The proposed architecture consists of a number of key components. An outline description of the components is provided below:

Monitor: the monitoring component is responsible to dynamically monitor the information related to individual user. In particular, when the information leaves apps. The monitor is real-time privacy monitoring.

Privacy Profile Refinement: the function of this component which is obtained a more care is to refine user profiles that were derived from system logs. It is responsible to update personal privacy preferences for induvial users based on users' interaction that stored in System Log. The Privacy Profile Refinement then takes the System Log as input and tries to match the information related to user with current user profile to observe if there is any change in order to update the user profile.

System Log: the main task of System Log is going to perform collecting a wider dataset of all information which is related to capture the user interaction with users' personal information.

Privacy Manager: this is the core component of the proposed system. The main function for privacy manager is to control the processing between each element. When the system chooses a privacy profiles that closely captured users' preferences from Profile storage, the Privacy Manager passes users' preferences to the Monitor in order to monitor these information. In order to update the personal privacy preferences for induvial users, the Privacy Manager will keep passing the privacy preferences from the Privacy Profile Refinement to The monitor.

User Privacy Profile: in order to assign user to the privacy profiles that most closely capture their privacy preferences, initial interface will display a series of questions related to users' demographic information such gender, age and education level when the user logs to the system first time. Based on the user's answers, the User Privacy Profile will determine the most closely privacy profile from the user profile storage.

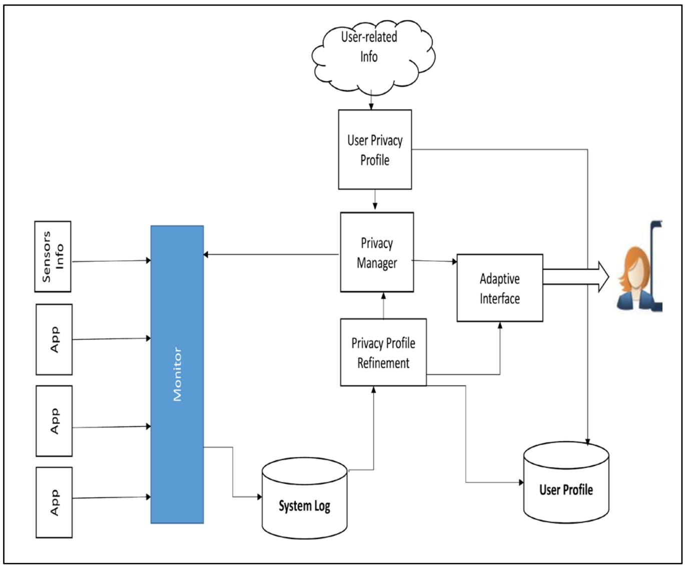

Figure 1 Architectural overview of adaptive user interface

Adaptive Interface: the main goal for Adaptive Interface is to notify the user about potential privacy risks based on the users' preferences. In this case, the presenting of information and notification will dynamically change, when the system updates users' preferences. For example, some users want to know more information about the potential privacy risks such as who had access to which data and when. In contrast, other users may not want to take the time to read the warnings in detail. Therefore, it would be useful to design an interface that is customized to the needs and desires of their specific users. Adaptive Interface aims to raising user awareness about privacy threats. Ultimately, this is the user-facing component of the system and therefore its design is of key importance.

V. Conclusion and future work

This paper reviewed the existing privacy awareness tools in order to critically evaluate the current state of the art. It was notable that few studies consider that the personal information is different from time to time and from person to person- taking a very generic approach to privacy.

It is evidence however that users do have different privacy concerns and requirements because users have heterogeneous privacy attitudes and expectations. Further research will build upon the architecture proposed and explore how privacy profiles can be developed, alongside adaptive interfaces that seek to optimise usability and convenience yet improve awareness.

References

[1] M. Taddicken, "The 'Privacy Paradox' in the Social Web: The Impact of Privacy Concerns, Individual Characteristics, and the Perceived Social Relevance on Different Forms of Self-Disclosure1," J. Comput. Commun., vol. 19, no. 2, pp. 248–273, 2014.

[2] R. Christopher, "More than 40% of online adults are multi-device users: stats Econsultancy," 2014. [Online]. Available: https://econsultancy.com/blog/64464-more-than-40-of-online-adults-are-multi-device-users-stats/. [Accessed: 10-Mar-2017].

[3] A. I. Anton, J. B. Earp, and J. D. Young, "How Internet Users ' Privacy Concerns Have Evolved," IEEE Priv. Secur., vol. 1936, no. February, pp. 21–27, 2010.

[4] TRUSTe, "2016 TRUSTe/NCSA Consumer Privacy Infographic - US Edition | TRUSTe," 2016. [Online]. Available: https://www.truste.com/resources/privacy-research/ncsa-consumer-privacy-index-us/. [Accessed: 11-Mar-2017].

[5] N. Hajli and X. Lin, "Exploring the Security of Information Sharing on Social Networking Sites: The Role of Perceived Control of Information," J. Bus. Ethics, vol. 133, no. 1, pp. 111–123, 2016.

[6] L. Brandimarte, A. Acquisti, and G. Loewenstein, "Misplaced Confidences: Privacy and the Control Paradox," Soc. Psychol. Personal. Sci., vol. 4, no. 3, pp. 340–347, 2012.

[7] P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall, "A conundrum of permissions: Installing applications on an android smartphone," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7398 LNCS, pp. 68–79, 2012.

[8] Felt, Ha, Egelman, Haney, Chin, and Wagner, "Android Permissions: User Attention, Comprehension, and Behavior," 2012.

[9] O. Kulyk, P. Gerber, M. El Hanafi, B. Reinheimer, K. Renaud, and M. Volkamer, "Encouraging privacy-aware smartphone app installation: Finding out what the technically-adept do," Proc. Usable Secur. Work., no. February, 2016.

[10] P. Gerber, M. Volkamer, D.- Darmstadt, and K. Renaud, "Usability versus Privacy instead of Usable Privacy [Google ' s balancing act between usability and privacy]," vol. 45, no. February, pp. 16–21, 2016.

[11] Federal Trade Commission, "Mobile privacy disclosures - Building trust through transparency," no. February, p. 29, 2013.

[12] M. Alaggan, S. Gambs, and A.-M. Kermarrec, "Heterogeneous Differential Privacy ´," Arxiv, pp. 1–14, 2015.

[13] M. Madden, A. Lenhart, S. Cortesi, A. Smith, and M. Beaton, "Teens , Social Media , and Privacy," 2013.

[14] Zhang, Nan, and W. Zhao, "Privacy-Preserving Data Mining," Secur. Priv. Trust Mod. Data Manag., pp. 151–166, 2007.

[15] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, A. N. Sheth, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones," ACM Trans. Comput. Syst., vol. 32, no. 2, p. 5, 2014.

[16] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang, "AppIntent: analyzing sensitive data transmission in android for privacy leakage detection," Proc. 2013 ACM SIGSAC Conf. Comput. Commun. Secur. - CCS '13, pp. 1043–1054, 2013.

[17] R. Balebako, J. Jung, W. Lu, L. F. Cranor, and C. Nguyen, "'Little Brothers Watching You': Raising Awareness of Data Leaks on Smartphones," SOUPS '13 Proc. Ninth Symp. Usable Priv. Secur., p. 12:1--12:11, 2013.

[18] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These Aren't the Droids You're Looking for: Retrofitting Android to Protect Data from Imperious Applications," Ccs, pp. 639–652, 2011.

[19] Y. Zhou, X. Zhang, X. Jiang, and V. W. Freeh, "Taming Information-Stealing Smartphone Applications (on Android)," 4th Int. Conf. Trust Trust. Comput., no. 2011, pp. 93–107, 2011.

[20] A. Le, U. C. Irvine, S. Langhoff, and A. Shuba, "AntMonitor : A System for Monitoring from Mobile Devices," no. 1, pp. 15–20, 2015.

[21] Y. Agarwal and M. Hall, "ProtectMyPrivacy : Detecting and Mitigating Privacy Leaks on iOS Devices Using Crowdsourcing Categories and Subject Descriptors," Proceeding 11th Annu. Int. Conf. Mob. Syst. Appl. Serv., vol. 6, no. September, pp. 97–110, 2012.

[22] S. Rasthofer, S. Arzt, E. Lovat, and E. Bodden, "D ROID F ORCE : Enforcing Complex , Data-Centric , System-Wide Policies in Android."

[23] Y. Song, "PrivacyGuard : A VPN-Based Approach to Detect Privacy Leakages on Android Devices by," pp. 15–26, 2015.

[24] A. Nadkarni and W. Enck, "Preventing accidental data disclosure in modern operating systems," Proc. 2013 ACM SIGSAC Conf. Comput. Commun. Secur. - CCS '13, pp. 1029–1042, 2013.

[25] M. Pistoia, O. Tripp, P. Centonze, and J. W. Ligman, "Labyrinth: Visually Configurable Data-Leakage Detection in Mobile Applications," Proc. - IEEE Int. Conf. Mob. Data Manag., vol. 1, pp. 279–286, 2015.

[26] M. Frank, B. Dong, A. P. Felt, and D. Song, "Mining permission request patterns from Android and Facebook applications," Proc. - IEEE Int. Conf. Data Mining, ICDM, pp. 870–875, 2012.

[27] B. Liu, J. Lin, and N. Sadeh, "Reconciling Mobile App Privacy and Usability on Smartphones: Could User Privacy Profiles Help?," 2013.

[28]