Privacy Protection In Smart Phones

M.Ikram Baig[1], G.Bharath Kumar[2], K. Lakshmi [3]
Dept of CSE
G.Pullaiah College of Engineering and Technology

Abstract- *While enjoying the benefit brought by various types of IOT devices, our personal data are revealed. Smart phones, as the typical pivot of IOT devices, use various types of applications, which collect our personal data. Intact, personal data revealed, as a latent hazard, is caused by the present design trend in industry, which is vast and vast. However, some people observe the side effect while we seem never bored by pursuing "smart" devices. In this paper, we introduce the data collection behavior, and demonstrate the motivations and reason posterior them. Thankfully, could computing with sufficient resources and exquisite services is a perfect way out. We initiate a mobile-cloud frame work to provide fine-grained permission authorization service for IOT devices, and present its performance by experimental results. Nevertheless, there are still more issues to be solved for the perfect IOT-cloud architecture. We list and summarize some obstacles and trends in the field of IOT-cloud.*

I. INTRODUCTION

In the past years the Internet of Things (IOT) has subtly impacts our daily life. All types of physical objects including groceries, buildings, vehicles etc., are connected and joined into a network with the support of all types of electronics, such as mobile devices, sensors and other wearable equipment. Everything is become smart and convenient for users. Meanwhile, due to the limited resources of IOT devices, cloud servers with minimum resources can be used to take charge of data processing and storage. We take first figure as an example to present three mail aspects of the effect of IOT and the cloud. For home and life style, IOT techniques can be used to implement electronic doors, automatic lighting and temperature control and indoor video monitoring. We do not need to carry a lot of keys for our houses, offices, cabinets, and cars. IN the field of transportation, every component in a vehicle is connected to the IOT to offer advanced driving assistance, navigation, and tire pressure of our cars, and we do not need to worry about getting lost when driving to some unfamiliar places. In the field of health care various types of wearable equipment monitor user physiological data connected to the IOT, and offer diverse health management suggestions. With the help of IOT techniques, we do not need to go to hospitals every month for physical examination, which saves a huge amount of time and improves accuracy. The most important characteristics and indispensable part of IOT is device, which are combined to the IOT and communicate with each other to build an intelligent modern ecological system. These devices play extremely important role, especially smart phones. According to a new pew research Center report, "technology device ownership: 2015", 68% of adults in the United States own a smart phone.

With the rapid development of IoT techniques, smart phones are not only communication equipment, but also health assistants, work secretaries, entertainment mates, and electronic IDs. Various types of smart phones are entering our routine lives with go-anywhere applications, which provide a wide array of enterprise, social, financial, and recreational services. The stream-lining of marketing, installation, and updating creates low barriers for mobile app developers to bring their products to the market, and even lower barriers for users to obtain and use apps. To enjoy the vivid functions and services offered by apps, users have to permit the authority for accessing local data to apps. However, these data may include our account numbers, email addresses, home addresses, photos, and other private information. It seems really convenient to store data in smartphones and further use them anywhere and anytime, but this kind of behavior brings about serious potential privacy hazard. We define a term, data over-collection, to describe the most frequently occurring and most serious privacy leakage behavior in smartphones. Apps collect users' data more than needed for the original function while within the permission scope, including tracking location, accessing photos, accessing address book, accessing calendar, tracking International Mobile Station Equipment Identity (IMEI) and Unique Device Identifier (UDID), and more. According to a report from Appthority, "App Reputation Report," 93 percent of iOS applications exhibit at least one type of data over-collection behavior, and 89 percent of Android applications have the similar problem.

II. DATA OVER COLLECTION-BEHAVIORS

Current mobile phone operating systems, such as iOS, Android, and Windows Phone, only provide coarse-grained permissions for regulating whether an app can access the data stored in smart phones. Meanwhile, few users actually notice and understand the permission agreement information shown during installation. Even knowing that an app may access their private information, few users choose to stop installing or to uninstall an app when it asks for permission authority. In fact, it is not users' responsibility to clearly know the permissions and cautiously manage the authority of apps. In this section, we choose some common data over-collection behaviors, analyze their inherent causes, and introduce the potential risks.

III. LOCATION

Users' location data can be used in various kinds of apps, including navigation, photo organization, social networking service (SNS), restaurant recommendation, weather, and travel. Normally, users are warned once an app intends to obtain location information, but they usually grant permission in order to use the functions or service offered by the app. The privileged apps always keep obtaining users' information data to offer location-based function or service accurately and quickly. However, these apps over-collect users' location data. From the report of Appthority, 50 percent of the top iOS free apps and 24 percent of the top iOS paid apps track users' locations. The iOS system offers a system service about location, named Frequent Locations, which is used to record the places users frequently visit. It is easy to disable this service, clear the record history, and stop it from running any time after initialization. However, users' location information is still collected by the operating system, and this information is just invisible and unavailable to users. Furthermore, this service meticulously records the amount of visits, the date, the time, and duration of staying. To offer such detailed information, this service must keep obtaining allocation information in the form of geographic location and time, since then the most frequent records are sorted by frequency. Much worse than iOS, from the report of Appthority, 82 percent of the top Android free apps and 49 percent of the top Android paid apps track user location. Due to the open developing and marketing environment of Android, it is extremely hard to restrict or prevent app developers to hide some data leakage codes into their Android apps and to put them into the market. Studied 1100 Android apps, and found that half of these apps exposed users' location information to third-party advertisement servers without requiring implicit or explicit user consent

IV. PHOTOS

Compared to traditional cell phones, one remark able function of current smart phones is taking photos. Smart phone users not only take photos for memories, but also for convenience. For example, taking photos of some slides instead of writing them down in notes is extremely convenient and time saving.

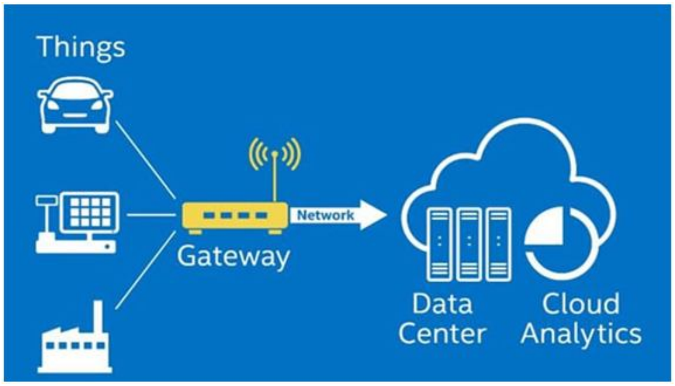

Figure 1. How IoT and the cloud influence our lives.

Meanwhile, with the increasing popularity of SNS, smart phone users form the habit of posting photos to show what they are doing via SNS apps. At restaurants, we can always see this universal situation. The first thing to do after waiters or waitresses serve the dishes is to take a photo, but not enjoy them at once. Besides SNS apps, there are some other kinds of apps that obtain users' photos as well, such as cloud storage, wallpaper, customized albums, and photo decorating. As a result, it is very easy for these apps to get permission to access albums and cameras from users. Users seemed to not really care about the accessing permission of their photos until the celebrity photo leakage scandal happen 2014. However, most responsibility was shirked to the cloud storage server. In fact, the origin culprit should be photo data over-collection behaviors of apps. By analyzing users' general purposes, we can see that they use these apps to deal with just several or parts of photos, not all of them. For example, I just want to post one photo via Facebook, and only want to authorize permission to access this one photo to my Facebook app. However, current smartphone operating systems, including iOS and Android, just offer coarse-grained permission authorization. The coarse-grained permissionauthorization only allows two modes of data accessing, which are all and none, and the users' authorization are always one-time operations. Once a user authorizes the permission of an app to access one photo, this app will hold this permission to the whole album forever. IOS gives users a way of escape, allowing users to manually disable the permission of an app to the album in Setting/Privacy/Photos. However, Android users have no way to disable the permission of some app other than uninstalling it. With current advanced techniques, most photos taken by smartphones are embedded with extra information about the location, time, device type, and more. As a result, the data over-collection behavior toward photos leaks not only users' photos, but also other private information. Users' photos reveal their daily lives. The exposure of photos not only infringes on users' rights such as profiles, but also may damage users' reputations. Even worse, some photos have fabulous value. Third party organizations can mine these photos for further commercially valuable information. This is the same as stealing assets from users. For example, one product designer is working on an innovator product, and he/she takes a photo of his/her design draft for recording. It will be a direct asset loss if the photo of a product design draft is obtained by some third party organizations and sold to someone else in the same domain as him/her.

V. ADDRESS BOOK

An address book is a traditional function provided by mobile phones, and this function is improved by adding more relevant information about contacts at the platform of smartphones. For convenience, smartphone users create new contacts when they make new friends, update existing contacts with email addresses, extra phone numbers, addresses, face portraits, and remarks. These advanced functions of address books really offer lots of benefits to users. They do not need to remember contact information about their friends or cooperators, which not only saves great time and efforts, but can also guarantee that there are no errors. The address book, usually a pre-loaded system app, provides a uniform interface for the apps running on its operating system to access users' address books. This uniform interface works as the bridge to connect apps and address books. Through this uniform interface, users can easily know which of their friends are using the same app and invite their friends to use this app. On the other hand, app developers are very willing to use this uniform interface, which can help to popularize their apps. As a result, apps can easily get permission to access address books from users. Similar to the operation on users' photos, mobile operating systems only provide coarse-grained permission authorization. It seems more reasonable for apps to access the whole address book, because users may want to check the status of all their friends about using some app. However, once an app gets permission to access a user's address book, it can keep obtaining contact data until the user manually disables the permission. For example, after we grant permission to access our contact data to a Facebook app, it always sends us a notification to invite more friends who do not have a Facebook account. This behavior shows that the Facebook app keeps obtaining our address book data. Facebook apps are not alone; there are various kinds of SNS apps, game apps, and commercial apps that have the same behavior of contact data over-collection. From the report of Appthority, 26 percent of the top iOS free apps and 8 percent of the top iOS paid apps access users' address books, and 30 percent of the top Android free apps and 14 percent of the top Android paid apps access users' address books. In fact, it is unnecessary to allow apps to access users' contact data all the time, and normally "share with friends" is just a one-time operation. The permission authorization should be flexible to suit this kind of one-time operation.

VI. IMEI/UDID

The IMEI and UDID make up the unique ID of one mobile phone, and cannot be deleted after manufacture. Similar to the idea of web cookies, the IMEI and UDID can be used to "remember" devices. From the report of Appthority, 88 percent of the top Android free apps and 65 percent of the top Android paid apps access IMEI/UDID, and 57 percent of the top iOS free apps and 28 percent of the top iOS paid apps access IMEI/UDID. This information is innocent itself, but combined with other kinds of information, the hazard becomes a juggernaut. IMEI/UDID works as the primary key in a relational database. It is the identification of all kinds of data, and can be used to integrate these data for one specific smartphone. In other words, all over-collected data can be labeled in the form of smartphones, which makes data more valuable for mining. As we all know, currently we do data mining research work based on anonymous data, even if they are recorded from real data. With the help of IMEI/UDID, users' behaviors can be correlated among multiple apps and matched to one unique device. Data over-collected by different apps can be integrated just via this ID, even though these data have nothing in common. Furthermore, IMEI/UDID can be used to build complete profiles with users' real data, including names, locations, accounts, and so on. For example, users 'locations, names, and account data are collected by different apps and bought by one third-party Company. This company can use the IMEI/UDID to create detailed profiles for in-depth views of users, which is undisputed privacy infringement. After discussing the data over-collection behaviors, we briefly analyze the motivations behind these behaviors. Loose Development Limitations. With the rapid popularity of smartphones, the number of mobile app developers keeps increasing. However, there are no established development limitations for app developers. In other words, app developers are permitted to implement any kinds of functions in their apps. Meanwhile, some app developers are not so familiar with app development, and they apply some open source libraries to achieve some functions of their apps. In the libraries, there are lots of code blocks implementing the obtaining of data functions. Without strict development limitations, it is hard to avoid the user using libraries with hidden data over-collection behaviors. Incomplete Censor Mechanisms. Apple hires about 100 employees to manually review iOS apps being published. However, they are only concerned with certain aspects, mainly focusing on the user interface (UI) and functions, such as crashes and bugs, broken links, advertisements, placeholder content, incomplete information, inaccurate descriptions, and repeated submission. In fact, the reviewers from App Store only check basic UIs and functions. For example, we submitted an iOS app recording users' operation functions to App Store for review. The result sent back showed that the reviewer just checked the home screen and some basic functions of our app, and most of our detailed functions were not checked. Worse, for Android, until April 2015, Google Play did not have manual censors for Android apps, which indicates that all Android apps could be on the market. Furthermore, data over-collection behaviors are much more complicated to detect than malware, because they happen with users' permission, and it is almost impossible to determine the exact amount of data needed for functionality. Apple hires about 100 employees to manually review iOS apps being published. However, they are only concerned with certain aspects, mainly focusing on the user interface (UI) and functions, such as crashes and bugs, broken links, advertisements, placeholder content, incomplete information, inaccurate descriptions, and repeated submission. Third-party companies or organizations are willing to buy users' data for commercial purposes. Third-party companies can be any kind of companies, even research organizations. As we know, the amount of customers is the most important parameter for the market share of one product. As a result, attracting potential customers to their products is the main reason for companies to obtain users' data from over-collection behaviors. Meanwhile, in this big data era, data are treasures. By analyzing users' behaviors, one company can accurately and quickly get the big picture of market trends. From the view of smartphones, the main reason for data over-collection is the defects of current mobile operating systems, including coarse-grained permission authorization, one-time permission authorization, and no different levels of privacy. Coarse-grained permission only provides two kinds of permission authorization: none or all. Once one app gets permission to access some kind of data from a user, it can obtain all of the same kind of data. One-time permission authorization indicates that once users authorize permission to an app, it can keep this permission and access to data. Furthermore, the permission authorization operations only occur once, whether accepted or rejected. For example, once you accept the first request of a Facebook app to access your album, a Facebook app can access your album without your permission again forever until you manually deny this permission. In addition, there are no different levels of users' data based on how private they are. In other words, current mobile operating systems treat users' data equally without discrimination. This kind of strategy is convenient for management and operation, but fails to protect users' private information.

VII. CLOUD-BASED SOLUTION

It is impossible to force app developers not to share users' data with advertisement networks and other third party organizations, and it is unreasonable to expect that all smartphone users can understand permissions clearly and protect their privacy carefully. In fact, the data over-collection behaviors of apps are created by us. We have improved the mobile phone from traditional communication equipment to advanced smartphones with various kinds of apps. As Albert Einstein said, "The significant problems we face cannot be solved at the same level of thinking we were at when we created them." To solve the mobile data privacy problem, we have to change our pattern of thinking, which makes everything bigger and bigger. We have to eradicate it in advance, but not deal with it in the aftermath. Meanwhile, current IoT devices have limited resources due to portability. These devices cannot undertake the due obligations of increasing requirements, including amount of storage, performance of calculation, availability, and other aspects. To solve the privacy issue of mobile data and to break though the resource limitations of IoT devices, a cloud-based solution is the best method. Thanks to cloud computing, we can offload a huge part, or even all, of the storage and calculation burdens to cloud servers at very low cost . Furthermore, after offloading the mobile data to cloud servers, we can use cloud computing techniques to provide fine-grained permission authorization. In addition, users are freed from complicated management of their data and permissions. Compared to users, cloud service providers are much more professional in assigning permissions to apps based on different privacy conditions. Several months ago, we proposed a mobile cloud framework for data privacy protection in smart cities. In this framework, as shown in Fig.2, we first separate mobile data into privacy levels from 1 to 3. The higher the value is, the more private the piece of data is. We assign privacy levels to users' data as follows. Location data: 3; photo data: 3 or 2; audio and video data: 1. Then we migrate location, photo, audio, and video data from smartphones to remote servers. Every time one app requests access permission for some kind of data, the access control service will determine whether to accept this request. Meanwhile, we design a grading system to evaluate our approach to data privacy. In this grading system, we use a formula to calculate the risk of apps using thetotal amount of data, the amount of over-collected data, and the privacy level of the data. The detailed working mechanism of this framework is shown. IoT devices send the request to access data to the cloud. The access control service receives the requests and validates the authorization. The en/decryption service encrypts and decrypts data before data is stored in storage and sent back to IoT devices. If we allow the access control service to authorize apps to access one specific type of data with the same privacy level, our approach can reduce over 2.5 times the privacy grade than original coarse-grained permission authorization. If we set the access control service to only allow apps to access the specific pieces of data they need, our approach can reduce over 35 times the privacy grade than the original

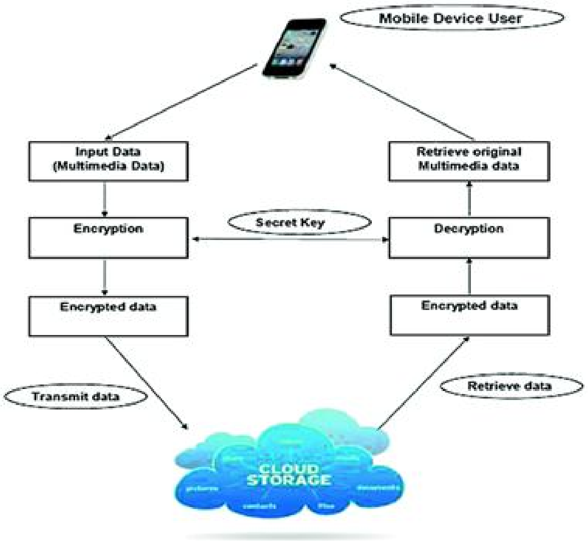

Figure 2. Mobile–cloud framework for data privacy protection.

VIII. OPEN RSEARCH ISSUES

Using cloud techniques to solve the mobile data privacy issue has not yet been fully studied. Some issues are even impossible to solve in current conditions. More research and effort are needed to completely achieve mobile data privacy protection. We list and analyze some thought-provoking and important research issues as follows.

IX. PRIVACY CLASSIFICATION

To provide fine-grained permission authorization, the first thing is to classify users' data based on their privacy. After separating data into different levels, we can use cloud storage service to store users' data in different servers or virtual machines. Without privacy classification, we have to separate all data into pieces and further check their detailed privacy parameters to determine the access control decision, which is really time consuming and tedious. As a result, privacy classification is not only a prerequisite of cloud-based fine-grained permission authorization, but also can make the access control strategy smooth and efficient. Once one app requests access permission to high-privacy-level data, it can be automatically denied. However, it is not easy to achieve the privacy classification of users' data. In the first step, we can separate data by types. For example, account information is much more private than photos; thus, they can be classified to different levels. Privacy classification by data types can be implemented without considering users' specific situations, because the types always have common characteristics and uses. Nevertheless, it is extremely hard to classify one kind of data into different privacy levels, because these data are user determined. For example, there are two photos in one user's album. One is a photo of a normal file that may be downloaded online, while the other one is a print-screen of a confidential document. It is obvious that these two photos should be assigned different privacy levels, but it is complicated for computers to tell the difference between these two photos. Machine learning and pattern recognition techniques seem to be suitable to solve this issue, but these methods have three deficiencies. The first one is that they must access the data in advance, which is alsoa hidden danger to users' privacy. The second one is that access control operations are dynamic, since users may keep creating, deleting, and updating their data. However, current machine learning and pattern recognition methods are not powerful enough to support such flexibility. The third one is that the rules of classification are closely related to users. Different users may have different attitudes toward the same piece of data. Thus, using these two kinds of methods to achieve classification still needs extra user operations, which is not very practical.

X. IOT NETWORKING

Without networking, the Internet of Things (IoT) is useless to some extent. IoT devices, including sensors, smartphones, and other gadgets, need cloud servers to store and analyze data due to their limited resources. However, current networking techniques are far from ideal. Besides traditional problems, such as bandwidth and stability, the security issue is the main enemy in the privacy protection field. Huber-connectivity threatens users' privacy in increasing ways. Hackers are given more opportunities and targets to attack, since almost everything is connected to the IoT. Using cloud resources is a wise method to offer fine-grained permission authorization by cutting off the ways of data over-collection behaviors of apps. However, we need to ensure the security of IoT networks to avoid attending to one while neglecting the other. Meanwhile, this issue is closely related to the advanced networking techniques, such as software-defined networking , fifth generation, and Internet 2, which make IoT networking research field complicated but profound.

XI. MOBILE DATA OFFLOADING METHODS

Currently, there is some research focusing on operations offloading from mobile devices to cloud servers.

Restricted by the communication cost, it is impractical to offload everything. Data offloading or migration seems much easier than operations, but in fact, it is much more complex. In mobile devices there are lots of internal or embedded data that are hidden and invisible unless rooted. For example, IMEI/UDID is one kind of embedded data of a device, which is similar to the medium access control address of the computer network card. Why do we need to offload mobile data to clouds? There are two reasons. The first one is that mobile or IoT devices will be totally released from the burdens of processing miscellaneous tasks that should not be in their charge. The second one is that this kind of framework is more suitable for both software developers and users. Developers do not need to develop various versions of their products for different platforms, and they do not need to upload every update to the Internet. Users never need to update their apps or software, and they can get rid of tedious data management. With the rapid development of service computing technology, the traditional client-server model has been replaced by the browser-server model, and this tide certainly will happen in the field of IoT. Mobile apps or software will disappear and be replaced by various kinds of services. The only thing needing to be installed in a device is a browser. All data are stored and processed in cloud servers, which can provide fine-grained access control and high-quality service.

XII. CONCLUSION

IoT technologies bring lots of convenience to our daily life. The smartphone is the pivot, and can be used to control various IoT devices. However, data over-collection behaviors are ubiquitous, due to the deficiencies of current mobile operating systems. They only provide coarse-grained permission authorizations and general privacy management. Cloud computing with sufficient resources and fine-grained access control service can be used to solve the data privacy issue. However, there are many technical challenges to implementing the IoT-cloud framework practically. After introducing a basic mobile-cloud framework we designed before, we analyze some thought-provoking research issues to protect users' mobile data.

REFERENCES

[1] S. Amini, J. Lindqvist, J. Hong, J. Lin, E. Toch, and N. Sadeh. Caché: Caching location-enhanced content to improve user privacy. In Proceedings of MobiSys '11, pages 197–210, New York, NY, USA, 2011. ACM.

[2] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In Proceedings of CCS '13, pages 901–914, New York, NY, USA, 2013. ACM.

[3] J. Ball. Angry birds and 'leaky' phone apps targeted by NSA and GCHQ for user data. http://www.theguardian.com /world/2014/jan/27/nsa-gchq-smartphone-app-angry-birdspersonal-data, January 2014.

[4] A. Bamis and A. Savvides. Lightweight extraction of frequent spatio-temporal activities from GPS traces. In Proceedings of RTSS '10, pages 281 –291. IEEE, December 2010.

[5] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan. Mockdroid: Trading privacy for application functionality on smartphones. In Proceedings of HotMobile '11, pages 49–54, New York, NY, USA, 2011. ACM.

[6] C. Bettini, X. Wang, and S. Jajodia. Protecting privacy against location-based personal identification. Secure Data Management, pages 185–199, 2005.

[7] T. Book, A. Pridgen, and D. S. Wallach. Longitudinal analysis of android ad library permissions. In Mobile Security Technologies (MoST '13), San Francisco, CA, May 2013.

[8] J. Brickell and V. Shmatikov. The cost of privacy: Destruction of data-mining utility in anonymized data publishing. In Proceedings of KDD '08, pages 70–78, New York, NY, USA, 2008. ACM.

[9] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the crowd: The privacy bounds of human mobility. Sci. Rep., 3, Mar 2013.

[10] W. Enck, P. Gilbert, 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. In Proceedings of OSDI '10, pages 1–6, Berkeley, CA, USA, 2010. USENIX Association. 249

[11] J. Freudiger, M. Manshaei, J.-P. Hubaux, and D. Parkes. Non-cooperative location privacy. IEEE TDSC, 10(2):84–98, March 2013.