User Friendly IoT Data Management Framework

CS523 Mid-Semester Report

Hyun Bin Lee, Aravind Sagar, Zicheng Li
University of Illinois at Urbana-Champaign
Urbana, United States
{lee559,asagar3,zli135}@illinois.edu

I. Introduction

Smaller and more power efficient processors enables almost everything in our life to collect information and access internet. This gave rise to the idea of Internet of Things (IoT).

IoT has seen a rapid growth in recent years, as benefits of a smarter world around us can be huge. On the other hand, rapid expansion of IoT devices also brings new technical and privacy related challenges related to them. Apart from the technical challenges like storage and processing of vast amounts of data, and building scalable communication channels, we also face a new era of privacy-related challenges arising from devices collecting a multitude of personal data. Assuring users of the privacy of their data is an essential step towards making IoT a commercial success [1].

The primary issues with currently available commercial IoT ecosystems is that third-party access control mechanisms of user data is centralized, and that permissioning is too coarse-grained. The first issue requires users to trust a single party with all their data, and at the same time introduces a single point of failure. The second issue provides more personal data than is required, to third parties. Several frameworks have been proposed to address these challenges [1], [2]. These systems aim to address the privacy concerns by putting the user directly in control of their data. Decentralizing access control mechanisms, fine-grained permissioning, and explicit user consent from the user to share just the right amount of data with third parties are common themes found in such frameworks.

While these frameworks can drastically improve the privacy of users in IoT systems, there are still gaps that need to be filled, in order to truly use these systems in real-world. One such gap is the daunting technicality of these systems. We cannot assume that a normal user can deal with cryptographic primitives, setting up servers, data aggregation and summarization, and a variety of other knowledge required to interact with such systems. This is the gap that we are trying to address in our work.

We propose building a user facing application and a corresponding server component that can help even non tech savvy users exploit privacy-aware, distributed, and cryptographically secure IoT systems. For the purposes of our implementation, we will assume that such an IoT system exists, and emulate the actions of that system. The app will help the users to visualize all the data owned by them, and manage fine-grained data access to third parties. Each user will have a dedicated instance of the server component, which will act as a privacy-mediator [1], that is, enforce the data access control policies.

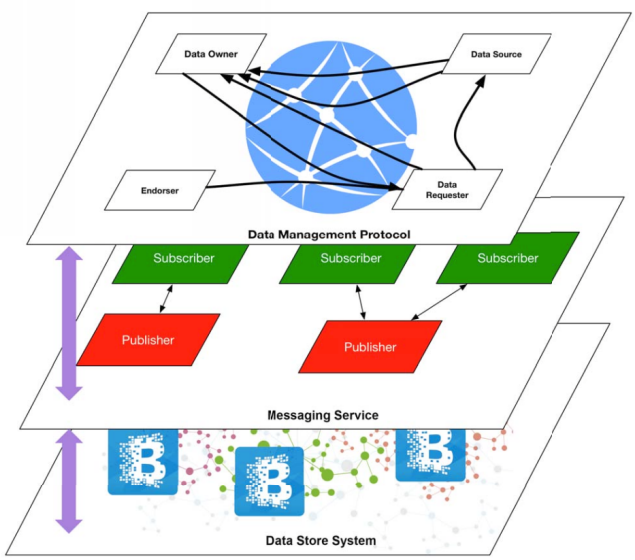

Fig. 1: Structure of the a privacy-aware IoT framework

Our main contributions include:

1) We aim to build a user friendly application that puts the user in control of their data, and lets them understand and allow just the right amount of data to be available for third parties.
2) For the server component, we aim to make it secure from external attacks, and also design it in such a way that users have the freedom to run their own instance of the server or get an instance of it in commercial servers.

II. Background

A. Internet of Things

The term Internet of Things (IoT) originated more than 15 years ago, when it was used to describe the work of the Auto-ID Labs at the Massachusetts Institute of Technology (MIT) on networked radio-frequency identification (RFID) infrastructures [9]. The definition has since expanded to beyond the scopr of RFID technologies. A lot of modern definitions have

been proposed, one of them being "a global infrastructure for the Information Society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [10].

The applications of IoT are diverse, ranging from smart industries to smart homes. In smart home area, thermostats, security systems, and energy management systems are particularly growing fast. In this paper, we focus on IoT technologies related to individual users, and these mostly fall under the smart home category, though the general principles are applicable in all areas of IoT.

Recent years have seen a rapid growth of smart devices available commercially. Ecosystems like Samsung SmartThings, Google Home and Apple Homekit are competing to become the primary player in personal IoT devices space, and for good reason - it is estimated that IoT could grow into a market worth $7.1 trillion by 2020 [11].

However, the rapid expansion of IoT market also means that many issues have been overlooked, and still need resolution. Some of the relevant issues are presented in 'Related Works'.

B. Privacy-Aware IoT Framework

IoT data security is crucial in protecting user privacy. Hashemi et al. [2] has proposed a framework to provide secure data manipulation in a scalable way.

Fig. 1 shows the overall structure of the framework [2]. It is composed of three components: a Data Store System that stores the transaction of data in a distributed way; a Massaging Service through which data owners, data sources and data requesters communicate; and a Data Management Protocol that defines the procedures of data access.

Our work focuses on the Data Management Protocol because the users do not directly interact with with the Messaging Service and the Data Store System. In the data management protocol, user data is stored in trusted entities, known as "data sources". IoT users are the "data owners". Entities who are trying to get access to user data are known as "data requesters".

The data owner is the only party who can grant data requesters the right, called "capability", to access its proprietary data. As the party who manages data, a data source need to always allow the owner to access its data, this is done by having the data source sending a access ticket to the data owner. The access ticket contains the data ID, the public key of the data owner, metadata and how the data can be accessed. To verify its identity, the data source also includes a copy of its public that was encrypted by the data owners private key in the message. The message is then encrypted with the private key of the data source.

When a data requester needs to access user data, it needs to contact the data owner to ask for the capability through the Messaging Service. The data requester needs to send to the user the Request ID, the its own public key and other conditions like access duration. Before the information reaches the owner, the data requester can increase the credibility of its request by finding third-party endorsers to endorse this request. The endorsers would sign the request in series with their private keys, and include their public key at the end.

If the data owner decides to grant the access to its data, it would send a package to the requester. Part of the package which contains the data ID, the data access path, the capability and the public key of the data source is signed with the private key of the data owner. In addition, the request ID and a special version of the owners public key is also included in the package. The entire package is then encrypted with the pubic key of the data requester.

After the requester gets the permission, it uses the capability it received from the data owner to access data on the data source. Before sending out the capability, the capability is encrypted with the private key of the data requester and the requesters public key is appended. Finally, the entire message is encrypted with the public key of the data source.

After data is accessed, the data source informs the data owner about the data access by sending a message containing the public key of the data source, the original data access ticket it sent to the data owner earlier, part of the capability that was signed by the data owners private key. This entire message is encrypted with the public key of the data source.

Finally, to ensure data transparency, all data transactions are recorded using BlockChains. After a successful data transaction, the data server broadcasts the transaction and the publishers update their BlockChain Record.

C. Related Works

Davies et al. points out that privacy concerns could be a major factor preventing wide-spread adoption of IoT devices [1]. Over-centralization of IoT systems is identified as a critical obstacle to eliminating concern over privacy. They propose a privacy mediator which sits in between IoT devices/data and outside world, and validates permissions according to the access control policies before any data is sent out. This model includes a trusted 'cloudlet', which could be a local hub or a trusted server at the edge of the cloud. They also emphasize some important characteristics that a privacy-aware IoT system should have, like exposing summarized data, user anonymity, control over inferred sensors, and ease of use. However, the model's limitation is that data storage has to happen in the sensor or the cloudlet - this could be limiting when the number of sensors increase. The model also does not discuss any framework for how to identify legitimate third party applications, and how to make sure that the request originates from the same application as it is claiming to be from.

Data mining and clustering of IoT related articles expose major problem areas in IoT [3]. App over-privilege, environment mistrust, LAN mistrust and weak authentication are identified as some of the problem areas that needs work. They also conclude that permissioning needs to be more fine-grained, and recommend better standards and widely-applicable systems.

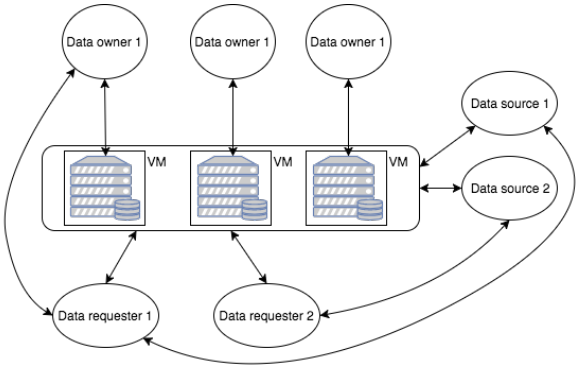

Fig. 2: Structure of the prototype framework

Security analysis of existing IoT systems show that serious vulnerabilities exist currently [14]. An analysis of Samsung Smartthings ecosystem in particular, shows that apps in the ecosystem are overpriveleged, and vulnerabilities like event spoofing can lead to privacy violation and even more serious consequences like loss of property. This goes even further in showing that a centralized proprietary system is probably not the right way to deal with IoT.

III. Problem and solution

As evident from the previous sections, effectively addressing privacy concerns about the data generated by IoT devices is essential for the success of IoT. There are some proposed architectures which tackles this problem using user-oriented and decentralized systems. However, there's a lack of a user-facing component in such systems, which can effectively make the complex architectures usable to everyone. Our work is aimed at tackling this problem, which is "to build a user-facing component for a privacy-aware, distributed, secure and user-centric IoT system (such as the blockchain based system described in [2]), which improves the user friendliness of the system while maintaining strong security and privacy guarantees."

To accomplish this, we build the user-facing component for the blockchain based IoT architecture described in "World of Empowered IoT Users [2]". Since that system is still in development, we concentrate on the user facing components of the system, and simulate some of the external components.

The proposed work consists of two components:

1) A smartphone app to manage IoT devices and data: Smartphones are becoming the preferred devices for internet access [4]. Hence we build a smartphone app with which users can manage their IoT devices and data. The functionality provided by the app includes:
   a) A centralized view of IoT generated data owned by them.
   b) A detailed view including the data source, access permissions and other details of this data.
   c) Permission request notifications from data requesters, and a way to accept or deny them.
   d) View, add, remove trusted parties.
   e) Provide user's (data owner's) public key to data requesters that provide personalized services to the user. This is not applicable for users providing anonymous data for purposes like analytics, where a different public key of the user is exposed to the data requesters.

   This is not a comprehensive list of the functionality, as we expect to add more as the project progresses. It also hides away some complexity from the users, like managing secure connection with the server described below, and managing user's private key etc. We also focus on the user-friendliness of the app, and expect to conduct user-research to find out what capabilities do users expect to find in such an app.
2) A trusted server which manages the access control: This is the component which actually manages access control and capability issuance in accordance to the user's wishes. This is similar to cloudlets described in [1], the key difference being that this server does not store data within it; it simply interacts with the data owner, data sources and data requesters in accordance to the protocol outlined in [2]. Interaction between server and data owner happens via the app described above. Interaction with other parties happen through a messaging layer, similar to the one outlined in [2]. The server ensures security in its communications using cryptographic primitives as outlined in the communication protocol.

   The server will also convert the technical specification of the data model and aggregation methods of each IoT device into user friendly texts, and uses these user friendly texts while pushing data access requests to the data owner's mobile app. We assume that the data model is agreed upon by the device and data sources. Creating required drivers for IoT devices is out of scope of this work.

   As we are focusing on user friendliness of the solution, and setting up a server may not be forte of everyone, a commercial implementation of our solution should ideally provide the users an option to automatically allocate a VM for the user in a commercial server. The users will of course have an option to use their own server instead. Another option is to have a 'hub' for each user, which will run the server process, but this solution may not be scalable for the user. We do not focus on this aspect, and instead focus on an implementation of the server which can be used for any of these purposes.

IV. Threat Model

We implement our solution based on Hashemi et al.'s [2] work for sharing IoT data objects. We particularly focus on their Data Management protocol to model communication between data requesters and data owners. Meanwhile, our work is not limited to this protocol as we can extend our framework to any equivalent multi-party cryptographic communication

protocol. In addition to four major parties involved in their protocol, we also add one additional party named Cloud Service Providers in our model. These providers are responsible for running our server-side implementation that enforces decisions made by users to regarding approvals/denials of data object requests.

1) **Data Owners**
   Our work primarily focuses on households with a small number of IoT devices. These individuals may interact with IoT devices using IoT applications written by IoT device manufacturers or other third-party programmers. Our framework does not trust these applications as they may leak sensitive data and violate Data Owner's privacy. Users grant or deny operations related with IoT data objects using our Android front-end application. Securing our application against Android malwares or compromised Android OS is out of the scope of our paper, but solutions that enforce Android application security and OS security [5] can be applied as orthogonal solutions to our work.
2) **Data Source**
   Although a Data Owner have a possession of data created by his Data Sources, the Data Owner may or may not manage Data Sources. We assume Data Sources are managed by trusted parties such that they honestly follow the data sharing protocol. On the other hand, enforcement of protocol on Data Sources is one of our future works.
3) **Data Requesters**
   We assume data requests coming from Data Requesters to be one of the major privacy threat vectors. Data Requesters query and find data objects using the Messaging service and request those data objects to Data Owners by following the Data Management protocol. Since we assumed Data sources to be honest, Data Requesters can only obtain data objects by following the protocol. Nonetheless, Data Owners may not fully understand the risks associated with approving or denying data requests. As Hashimi et al. has mentioned, the system they have proposed is user-centric such that user has full control of access control of data. In other words, Data Owners are solely responsible for granting access to all resources. This can be a very burdensome task such that it may not be scalable as we envision IoT devices to be more ubiquitous in the future. Thus, our solution primarily focuses on providing concise and accurate information for each data request and guiding Data Owners to make autonomous decisions.
4) **Endorsers**
   In our framework, trusted Endorsers are responsible for validating identity of Data Requesters and authenticity of Requesters' requests. Supplementary Information provided by Endorsers regarding the data requests may help user to make correct decisions but it may also
5) **Cloud Service Providers**
   Trusted third party Cloud Service Providers host back-end server applications for each Data Owner such that each front-end Android application has a corresponding back-end application. Each server application is isolated from another as each runs on a separate virtual machine. For our future work, we would like to have a stronger threat model and assume Cloud Service Providers to be untrusted as well. In such case, we could enforce security and privacy of our solution using Intel's SGX [12] enclaves. Data Owner trusts server-side code run in the enclave, and they can validate integrity of the code using measurements provided with a cryptographic hash. While, our solution provides all security guarantees that are enforced by the enclaves, attacks that target security limitations of the Intel SGX processors are out of scope in our paper. These include Cache timing side-channel attacks [6] and Denial of Service attacks. Meanwhile, we believe orthogonal approaches [7], [8] that mitigate such threats can be applied to our solution.

With guidances of our user-friendly user interface, our work protects Data Owners against data requests that may violate user privacy. We believe this is the most likely threat against ordinary IoT users as we envision complexity of IoT system to increase in the future. Our ultimate goal is to provide concise yet accurate information to users such that they make intended and autonomous decisions.

V. Challenges

There exists several challenges to implement our solution. We would to address these challenges in this section.

1) **Simulating Roles**
   Since our work is based on a theoretical model, we need to simulate each entity. During this simulation process, we may run into some implementation problems regarding details that are not explained in the literature. Thus, we need to make assumptions and augment the model in order to proceed our project. Furthermore, we may not have accurate performance benchmarks for our evaluation as we work on a simulated environment for testing our framework.
2) **Designing User Interface**
   Effective application design is an active area of research. Designing our front-end application would be the most challenging task for our project. We would like to explore Deka et al.'s [13] large repository of mobile app design to achieve our goal for the project. Davies et al. [1] discusses key challenges and concerns regarding how to design user policies that mediate between user and application. While they do not directly provide a clear-cut solution to the problem, authors provide a list of possible approaches to improve effectiveness of user policies.
3) **User Study for App Interface**
   The challenges related to conducting a user research for the mobile app is two fold.

(a) Overview of user's data (b) Details of a particular device (c) Navigation drawer (d) Notifications screen

Fig. 3: UI mockup of the user facing application

First, we are building an interface for which the backend is unfamiliar, and not available yet. This means that users need to know more background information about the interface for which we seek feedback. In particular, most of the IoT systems that exist now are centralized, and provides coarse-grained permissioning. By contrast, the system that we are targeting is distributed, user-centric, and fine grained in terms of third party data access permissions. Users need to know this to provide effective feedback about the interface.

Second, IoT is still in its infancy, and it's hard to find enthusiast users of IoT devices. We overcome this by treating smartphone sensors as basic IoT devices and building a story for user interaction with the application in terms of sensor data available from smartphones.

VI. Experiment

We started out by experimenting with the UI of the application. We are currently developing an Android app, and apps for other mobile platforms like iOS is left as future work.

We made an initial version of the UI mockups, and ran a user survey to iterate over the design. The current design is shown in "Fig. 3". These images show only some of the important screens in the app, and are not exhaustive. The capabilities of the app include having an overall view of the devices and data (Fig. 3a), seeing and editing details pertaining to a particular device (Fig. 3b), receiving and acting upon data requests (Fig. 3d), and viewing and managing trusted parties.

User research shows that the following design choices enhance the experience.

- Overall view of the devices and data is very helpful.
- Notifications screen is well done because users can see whether the requester is endorsed and act on the notification from the same screen.
- Users also feel more comfortable with IoT devices when a privacy-control mechanism like this app is present.

We also identified some major areas of concern.

- Users need more control over what data is stored. In particular, they might want to delete data between certain time periods.
- Users want to minimize the data shared with third parties, and hence each device should expose various spatial and temporal summarizations.
- Guarantee of anonymity is a must when sharing sensitive data with third parties, especially for analytics purposes.

References

[1] Davies, Nigel, et al. "Privacy mediators: Helping iot cross the chasm." Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications. ACM, 2016.

[2] Hashemi, Sayed Hadi, et al. "World of Empowered IoT Users." Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on. IEEE, 2016.

[3] Zhang, Nan, et al. "Understanding IoT Security Through the Data Crystal Ball: Where We Are Now and Where We Are Going to Be." arXiv preprint arXiv:1703.09809. arXiv, 2017

[4] (2016, Nov.) "Mobile and tablet internet usage exceeds desktop for first time worldwide." Statcounter global stats. [Online]. Available: http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide (Accessed Oct 24, 2017)

[5] Ahmed M. Azab, et al. 2014. "Hypervision Across Worlds: Real-time Kernel Protection from the ARM TrustZone Secure World." In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14). ACM, New York, NY, USA, 90-102. DOI: http://dx.doi.org/10.1145/2660267.2660350

[6] Wang, Wenhao, et al. “Leaky Cauldron on the Dark Land: Understanding Memory Side-Channel Hazards in SGX.” ACM Computer and Communications Security (CCS 17), October, 2017.
[7] Yan, Mengjia, et al. 2017. “Secure Hierarchy-Aware Cache Replacement Policy (SHARP): Defending Against Cache-Based Side Channel Atacks.” In Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA ’17). ACM, New York, NY, USA, 347-360. DOI: https://doi.org/10.1145/3079856.3080222
[8] Rane, Ashay, et al. “Raccoon: Closing digital side-channels through obfuscated execution.” In USENIX Security Symposium (2015).
[9] Atzori, Luigi, Antonio Iera, and Giacomo Morabito. “The internet of things: A survey.” Computer networks 54.15 (2010): 2787-2805.
[10] (2012, Jun) “Overview of Internet of Things.” ITU-T Recommendations. Available: http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060 (Accessed Oct 25, 2017)
[11] (2015, Jun) “Internet of Things Market to Reach $1.7 Trillion by 2020”. The Wall Street Journal. Available: https://blogs.wsj.com/cio/2015/06/02/internet-of-things-market-to-reach-1-7-trillion-by-2020-idc/ (Accessed Oct 25, 2017)
[12] Costan, Victor, et al. “Intel SGX Explained.” IACR Cryptology ePrint Archive 2016 (2016): 86.
[13] Deka, Biplab, et al. 2017. “Rico: A Mobile App Dataset for Building Data-Driven Design Applications.” In Proceedings of the 30th Annual Symposium on User Interface Software and Technology (UIST ’17).
[14] E. Fernandes, J. Jung and A. Prakash. “Security Analysis of Emerging Smart Home Applications.” 2016 IEEE Symposium on Security and Privacy (SP) (2016).