Designing Privacy-aware Internet of Things Applications

CHARITH PERERA, Newcastle Univerisity
MAHMOUD BARHAMGI, Claude Bernard Lyon 1 University
AROSHA K. BANDARA, Open University
MUHAMMAD AJMAL, Newcastle Univerisity
BLAINE PRICE, Open University
BASHAR NUSEIBEH, Open University

arXiv:1703.03892v4 [cs.SE] 26 Apr 2018

Internet of Things (IoT) applications typically collect and analyse personal data that can be used to derive sensitive information about individuals. However, thus far, privacy concerns have not been explicitly considered in software engineering processes when designing IoT applications. In this paper, we explore how a Privacy-by-Design (PbD) framework, formulated as a set of guidelines, can help software engineers to design privacy-aware IoT applications. We studied the utility of our proposed PbD framework by studying how software engineers use it to design IoT applications. We also explore the challenges in using set of guidelines to influence the IoT applications design process. This paper also highlights the benefits of providing a framework that helps software engineers explicitly consider privacy for IoT applications and also surfaced a number of challenges associated with our approach. Our studies show that PbD framework significantly increase both novice and expert software engineers' ability to design privacy aware IoT applications.

CCS Concepts: • **Security and privacy**; • **Computer systems organization** → *Distributed architectures*; *Embedded and cyber-physical systems*; • **Software and its engineering** → Software architectures;

Additional Key Words and Phrases: Internet of Things, Software Engineering, Privacy by Design

ACM Reference format:
Charith Perera, Mahmoud Barhamgi, Arosha K. Bandara, Muhammad Ajmal, Blaine Price, and Bashar Nuseibeh. 2018. Designing Privacy-aware Internet of Things Applications. 1, 1, Article 1 (April 2018), 35 pages.
https://doi.org/0000001.0000001

1 INTRODUCTION

The Internet of Things (IoT) [39] is a network of physical objects or *'things'* that have computing, networking, sensing and actuation capabilities, enabling these objects to collect and exchange data. The design and development process for IoT applications is more complicated than that for desktop, mobile, or web applications. First, IoT applications require both software and hardware (e.g., sensors and actuators) to work together across multiple different type of nodes (e.g., micro- controllers, system-on-chips, mobile phones, miniaturized single-board computers, cloud platforms) with different capabilities under different conditions [37]. Secondly, IoT applications development requires different types of software engineers to work together (e.g., embedded, mobile, web, desktop). This complexity of different software engineering specialists collaborating to combine different types of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 Association for Computing Machinery.
XXXX-XXXX/2018/4-ART1 $15.00
https://doi.org/0000001.0000001

hardware and software is compounded by the lack of integrated development stacks that support the engineering of end to end IoT applications.

Typically, IoT applications collect and analyse personal data that can be used to derive sensitive information about individuals. However, thus far, privacy concerns have not been explicitly considered in software engineering processes when designing and developing IoT applications, partly due to a lack of Privacy-by-Design (PbD) methods for the IoT. Further, the engineering complexities explained above have forced software engineers to put most of their efforts towards addressing other challenges such as interoperability and modifiability, resulting in privacy concerns being largely overlooked. Additionally, lack of knowledge on tangible and intangible benefits of privacy practices have also contributed to the overlooking of privacy challenges [47].

We propose to address this issue by providing systematic guidance for software engineers towards building privacy-aware IoT applications. More specifically, we developed a set of privacy guidelines (set of heuristics). As we discuss later, guidelines, in general, have been efficiently and effectively used in many domains as a method of guiding both novice and domain expert.

Our research is motivated by the lack of privacy protection measures taken by IoT application designers and potential privacy violations that arise due to such inefficient designs. In our earlier work [38], we derived privacy guidelines by examining Hoepman's [20] eight design strategies and used them to ***'assess'*** privacy capabilities of IoT applications and platforms. In contrast, our objective in this paper is to explore how a PbD framework (a set of guidelines) can help software engineers to ***'design'*** privacy-aware IoT applications.

The primary contributions and the scope of this article can be listed as follows:

- We propose and evaluate how a set of privacy guidelines could be use to effectively improve the IoT application designs. Towards this, we proposed both guidelines as well as a methodology that explains how to apply the guidelines.
- Our methodology is uniquely designed to address IoT challenges such as heterogeneity and distributed nature. This is a significant different from excising privacy by design (PbD) frameworks (principles, design strategies, etc.).
- We gain insights on how set of guidelines could help software engineers to better design privacy aware IoT application by identifying and applying privacy protecting features into application designs.
- We also explore strengths and weaknesses of our approach as well as challenges in manual application design processes in general. We provide insights on how to address those weaknesses.

It is important to note that, in this paper, we do not claim our PbD framework is better than any previous work, neither we claim that applying set of privacy guidelines will eliminate all privacy risks. To best of our knowledge, this is one of the first PbD frameworks (which formulate as set of guidelines) that explicitly target IoT application design challenges. Our aim is to help software engineers to reduce as much privacy risks as possible at the design phase. We discuss more about our intentions later in Section 4.

The paper is structured as follows: Section 2 discusses common IoT architectures and their characteristics. It also briefly introduces the data life cycle phases and their importance when designing privacy into IoT applications. In Section 3, we present our motivation through three different use-cases. We have used these use-cases to evaluate the effectiveness and identify the challenges in designing privacy aware IoT applications. We briefly introduce the PbD framework in Section 4. In Section 5, we explain the research methodology and evaluate the effectiveness the PbD framework. We discuss findings and lessons learned in Section 6. Finally, Section 7 presents the related work and compares our PbD framework with existing approaches followed by the conclusion. In Section 8, we conclude the paper by highlighting future directions.

2 INTERNET OF THINGS SOFTWARE ARCHITECTURE

In this section, we briefly discuss how data flows in a typical IoT application. As illustrated in Figure 1, in IoT applications, data moves from sensing devices to gateway devices to the cloud infrastructure [37]. This is the most common architecture, also called the centralised architecture, used in IoT application development [42]. However, there are other types of architecture such as 1) collaborative, 2) connected intra-net of Things, and 3) distributed IoT [42]. Even for these other types of architectures, if we consider a flow of a single data item, we can observe a data flow analogous to that of the centralised architecture where data moves from edge devices to the cloud through different types of nodes. Therefore, while we use the centralised IoT architecture to explain our PbD approach in this paper, our approach is agnostic the choice of IoT architecture.

The centralised architectures typically consists of three components: 1) IoT devices, 2) Gateway devices, and 3) IoT cloud platforms (Figure 1). Each of these devices have different computational capabilities. They also have different types of access to energy sources from permanent to solar power to battery power. Further, depending on the availability of knowledge, each device may have limitations as to the type of data processing that can be done. A typical IoT application would integrate all these different types of devices with different capabilities. It is important to note that different types of privacy protecting measures can be taken on each of these different nodes based on their characteristics.

We divided the data life cycle into five phases in order to structure our discussion. Data life cycle phases play a major role in applying our PbD framework to design privacy aware applications in a systematic manner. Within each device (also called a node), data moves through five data life cycle phases: Consent and Data Acquisition [CDA], Data Preprocessing [DPP], Data Processing and Analysis [DPA], Data Storage [DS] and Data Dissemination [DD]. CDA phase comprises routing and data read activities by a given node. DPP describes any type of processing performed on raw data to prepare it for another processing procedure [40]. DPA is, broadly, the collection and manipulation of data items to produce meaningful information [16]. DD is the distribution or transmission of data to an external party.

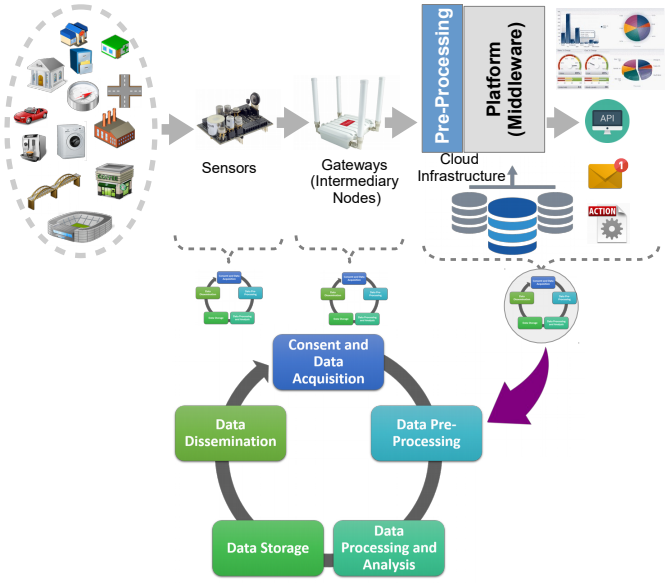

Fig. 1. Typical data flow in IoT Applications

We assume that all the data life cycle phases are present in all nodes in an IoT application to be utilised by engineers to protect user privacy. However, based on the decisions taken by software engineers, some data life cycle phases in some nodes may not be utilised. For example, a sensor node may utilise the DPP phase to average temperature data. Then, without using either the DPA and DS phases to analyse or store data (due to hardware and energy constraints) the sensor node may push the averaged data to the gateway node in the DD phase.

3 EXAMPLE IOT SCENARIOS

In this section, we present three use case scenarios, which we also use to evaluate the PbD framework as described in Section 5. Each scenario is presented from a problem owner's perspective, where each problem could be solved by developing an IoT application. More importantly, it should be noted that none of these scenarios explicitly highlight privacy requirements or challenges. They are primarily focused on explaining functional requirements at a high level. Later in Section 4, we explain how our PbD framework can be used by software engineers to extract additional information, from problem owners, that are crucial to design privacy aware IoT applications.

3.1 Use case 1: Rehabilitation and Recovery

Summary: Robert is a researcher who oversees a number of rehabilitation facilities around the country where patients with physical disabilities are treated and rehabilitated. Robert is interested in collecting and analysing data from sensors worn by patients while they engage in certain activities (e.g., walk using walker, walk using crutches, climbing stairs), in order to guide the patients' recovery processes in a more personalised manner. Robert has an application that is capable of analysing patient data and developing personalised rehabilitation plans. The application monitors the progress and alters the rehabilitation plans accordingly. There is a speciality nurse allocated for each patient in order to monitor the recovery progress and provide necessary advise when required.

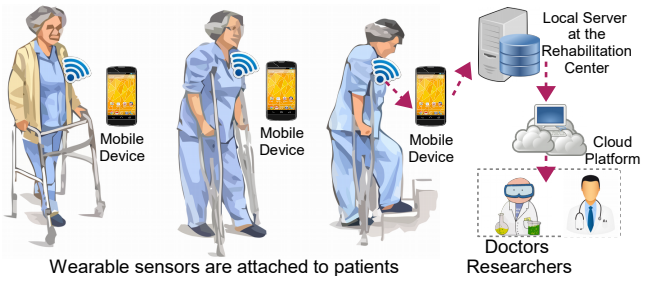

Fig. 2. IoT application to support rehabilitation

3.2 Use case 2: Health and Well-being

Summary: Michael works for the department of public health and well-being. He has been asked to develop a plan to improve the public health in his city by improving the infrastructure that supports exercise and recreational activities (e.g., parks and the paths that supports jogging, cycling, and place for bar exercise, etc.). In order to develop an efficient and effective plan, Michael needs to understand movements of people and several other aspects of their activities. Michael is planning recruit volunteers in order to gather data using sensors. Michael has an application that is capable of analysing different types of data and recommending possible improvements that need to be done. Michael only needs to collect data when the volunteers are within the park premises as illustrated in Figure 3

Fig. 3. City planning towards health and well-being

3.3 Use case 3: Amusement Park and Leisure

Summary: *TrueLeisure* is a company that operates different types of franchised entertainment attractions. Their amusement parks are located in United States, United Kingdom, and Australia. These amusement parks are fully owned and operated by franchisees. However, *TrueLeisure* continuously monitors and assesses the service qualities and several other aspects in each of the amusement parks. Jane is a data analyst overseeing the quality assessment tasks at *TrueLeisure*. She is responsible for continuously monitoring the service quality parameters. Waiting time is one of the key service quality parameters and is a key contributory factor to customer satisfaction. Local quality assessment teams continuously measure the crowd waiting time of each ride and attraction within their own amusement park. All the visitors use *TrueLeisure*'s theme park mobile app to buy tickets for attractions, further information, tour guide, maps, etc. Jane is interested in the big picture, i.e. she would like to measure the overall waiting time for each ride attraction by combining individual waiting times. Jane will report these measurements to *TrueLeisure* management to guide franchisees on future developments of their theme parks efficiently and effectively.

4 PRIVACY-BY-DESIGN GUIDELINES

In each of the example scenarios above, the software engineer would need to perform further analysis to extract explicit privacy requirements that could support the design of privacy enhancing features into the IoT applications that would be developed to delivery the required functionality. In this section we provide an overview of our PbD framework and explain how it could be used to design privacy into IoT applications. We also explain why guidelines are useful to help software engineers and where they fits in with similar other approaches such as principles, strategies, patterns and tactics.

4.1 Why Guidelines (or Heuristics or Check-lists)?

We primarily use the term guidelines as our intention is to guide the software engineers. In general, guideline aims to improve or maintain efficiency of a particular process based on to a set routine or sound practice. Guidelines may not mandatory to follow, but recommended. However, the term heuristics is also appropriate to identify our guidelines. Heuristics are techniques derived from past experiences by dealing similar problems. These techniques rely on using readily accessible, though loosely applicable, information to control problem solving in human beings, machines, and abstract issues [34]. Heuristics do not promise to produce perfect or optimal solutions. Finally, the term check-list is also appropriate to identify our guidelines. A check-list is a type of informational

aid used to reduce failure by compensating for potential limits of human memory and attention. Our guidelines also aim to reduce human errors by reducing knowledge requirements.

Sometimes, guidelines are considered as less useful approach due to their inherited characteristics such as lack of proof (for consistency or correctness), depending nature of the follower, lack of rigorous scientific method to extracting guidelines, lack of consistency in interpretation, and so on. Despite such weaknesses, guidelines are being used successfully in many domains. Following list showcases some examples where guidelines / heuristics / check-lists are used to address different challenges.

- Heuristics based usability design and evaluation is widely used in human computer interaction domain [31, 32].
- The Information commissioner's office, UK's independent authority set up to uphold information rights in the public interest, use check lists to guide businesses to prepare themselves for GDPR [23].
- Surgical Safety Check-list developed for the World Health Organization by Dr. Atul Gawande has been able to reduce mortality by 23% and all complications by 40% [19]. Airplane pilots rely upon check-lists to ensure that both routine procedures and emergency responses are handled appropriately [17].

Above usages and successes have given us confidence to formulate our framework based on guidelines. Guidelines reduce the thinking time of the follower. For example, the guidelines we formulated are not something that ordinary software engineers cannot think by themselves. However, if they try to come up guidelines by themselves, it may take time. In software development processes time is money. Due to time-cost pressure, software engineers may not have enough time to think by themselves. Therefore, our guidelines could come handy to many engineers. They can just go thought the guidelines one by one and check whether they can apply them. Our node-by-node deign methodology also simplifies the complex IoT application designs. Guidelines also provide meaningful ways to divide workload among engineers (e.g., each engineer may focus / specialise on addressing few guidelines) and can be used a common knowledge base to discuss about their application designs with colleagues. Guidelines make design process comparatively less tiring for engineers as it reduces intensive thinking and knowledge requirements. Guidelines also allow engineers to pause and resume conveniently and keep track of design changes. We acknowledge that guidelines are not perfect, neither they intend to be. However, evidence suggests that guidelines can help to improve effectiveness and efficiency in many occasion. In this paper, we explore the usage of guidelines in the context of IoT application designs.

4.2 Where Guidelines Fit in?

Let us now introduce few different terms that are frequently used in the privacy research: *principle*, *strategies*, *patterns*, and *tactics*. As showing in Figure 4, Principles can be considered as high level (more abstract or less concrete) ideas. In contrast, tactics are low level instructions (less abstract or more concrete). Strategies, guidelines and patterns sits in between. This does not mean one type is better or worse than other. Each of these layers have their own strengthens and weaknesses. Bottom layer tactics provide specific solutions to specific problems where as top layer principles provides insights on umbrella direction for us to explore further and solve problem by ourselves. However, we acknowledge that boundaries of these layers may not stand strong where some principles may interpret as strategies and wise-versa.

Principle: A principle is a concept or value that is a guide for behaviour or evaluation. Typically, they are very abstract but shows a direction to follow. Ten Privacy Principles of Personal Information Protection and Electronic Documents Act (PIPEDA) [30] and Seven Foundation Privacy by Design principles by Information & Privacy Commissioner, Canada [10] can be identified as an example.

Strategies: In contrast to principles / ideas, strategies are focused on achieving something. A design strategy describes a fundamental approach to achieve a certain design goal. Therefore, strategies are more specific

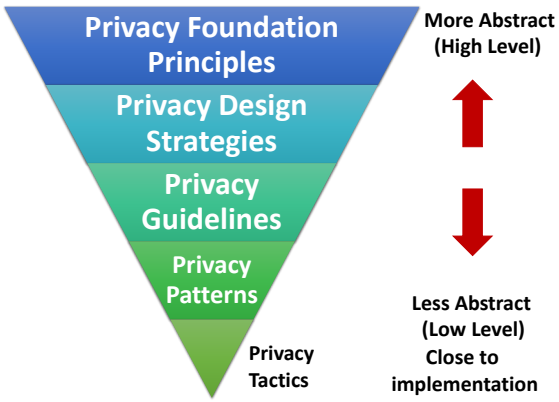

Fig. 4. From high level principles to low level tactics: Rubinstein and Good [44] correctly argues that making a specification or requirement for software design is to make it concrete, specific, and preferably associated with a metric. These layered approach aims to achieve this in a systematic way.

in terms of what it wants to achieve. Hoepman's [20] seven privacy design strategies can be identified as an example.

Guidelines: Our guidelines breaks down strategies into lower level concrete set of instructions.

Patterns: Design patterns are useful for making design decisions about the organisation of a software system. A design pattern *"provides a scheme for refining the subsystems or components of a software system, or the relationships between them. It describes a commonly recurring structure of communicating components that solves a general design problem within a particular context."* [6]. Patterns solve a specific problem but are neutral or have weaknesses with respect to other qualities. In contrast, there is also a term called *'anti-patterns'*. In software engineering, an anti-pattern is a pattern that may be commonly used but is ineffective or counter productive in practice [5].

Tactics: Patterns are built from tactics (e.g., *if a pattern is a molecule, a tactic is an atom*) [3]. In other terms, patterns package multiple tactics together to solve a specific problem. Tactics help to fine tune patterns and typically they address specific quality attributes and trade-off decisions. Each tactic may have pros and cons. New tactics can be introduced to a existing set in order to address existing weaknesses. However, such introductions could introduce new issues or weaknesses as well. Ideally, we may try different tactics until eventually the side-effects of each tactic become small enough to ignore.

It important to note that top three layers (principles, strategies, guidelines) are primarily focused bottom up approach. Typically, we adopt principles, strategies, or guidelines because they suggest good practices and historically or logically proved to reduce privacy risks. Typically, they are blanket solution aim to eliminate multiple privacy issues at a time (without addressing them individually). In contract, patterns and tactics focus on problem solving. This is more like a top down approach where we try to find solutions to specific privacy problems.

Let us explain these layers using an example. This example also highlights the fact that boundaries of these layers can be quite weak at times. **[Principle]** *"Proactive not Reactive; Preventative not Remedial"* is one of the principles proposed by Information & Privacy Commissioner, Canada [10]. The official explanation is *"The Privacy by Design (PbD) approach is characterized by proactive rather than reactive measures. It anticipates and prevents privacy invasive events before they happen. PbD does not wait for privacy risks to materialize, nor does it offer*

remedies for resolving privacy infractions once they have occurred âĹŠ it aims to prevent them from occurring. In short, Privacy by Design comes before-the-fact, not after.".

By examining this principle, we may come up with a strategy called *'Minimise'*. **[Strategy]** Hoepman [20] describes *'Minimise'* strategy *"as limiting usage as much as possible by excluding, selecting, stripping, or destroying any storage, collection,retention or operation on personal data, within the constraints of the agreed upon purposes"*. Hoepman's minimise strategy can be identified as a way to follow *'proactive principle'* (i.e. minimise the amount of data collected is a proactive measure to avoid or reduce potential privacy violations).

We can further break down the minimise strategy into guidelines (Refer Table 1). **[Guideline]** One minimise guidelines is *"Minimise data raw data intake"*. We further describe this guideline as *"Whenever possible, IoT applications should reduce the amount of raw data intake. Raw data could lead to secondary usage and privacy violation. Therefore, IoT applications should consider converting (or transforming) raw data into secondary context data."*

Privacy pattern can be identified as a low-level design that aims to solve a specific privacy challenge. The relationship between guidelines and patterns may be quite weak as, in most instances, patterns can stand by themselves problem solving techniques. However, still, privacy patterns can be identified as low level designs that help to implement guidelines. **[Patterns]** By continuing the example, we can extract a pattern[1] called *'On-line Activity Detector'*. This pattern *extracts orientation (e.g. sitting, standing, walking) by processing accelerometer data and store only the results (i.e. secondary context) and deletes the raw accelerometer data.*

[Tactics] *'On-line Activity Detector'* pattern may compose tactics such as *'average'* and *'periodic delete'* . *'Average'* tactic may be used to prepare accelerometer data for activity detection. *'Periodic delete'* tactic may be used to delete data after detection. In some designs *'Periodic delete'* may be replced with a *'In-memory processing'* tactic which aims to perform the activity detection in memory.

4.3 Targeted Audience

We developed our PbD framework to be used by wide range of users. However, we do not aim to provide any guarantee (i.e., we do not guarantee that an IoT application designed using our guidelines is privacy proof). This is not our intention. We wanted to provide an instrument for engineers to use so they can take certain steps to make their designs 'better' in terms of privacy awareness. We believe software engineer will, at least, be able to apply one privacy guidelines into their design which they would not do otherwise. Mostly, we wanted to help and guide individuals and small teams who do not have time, or resources to invest to hire or consultancy privacy experts or service. Completely ignoring privacy issues could cost such small teams a lot it long run as they grow. Later re-factoring is always costly in any software development process. Therefore, our guidelines will help small entrepreneurial teams, IoT hackers, hobbyists, etc. to embed privacy protecting features built into their IoT application designs at initial stages without consulting privacy experts. It is important to note that our guidelines do not intend to replace privacy experts and consultants. However, software engineers using our guidelines may reduce the workload need to be done by privacy experts.

4.4 Overview of the Framework

In our earlier work [38], our literature search led to determine that Hoepman's [20] is the most appropriate starting point for developing a more detailed PbD framework for IoT. The guidelines were compiled by using the structured-case research method [7], a research metod typically used for building theory in information systems research. Detailed explanation on each of the guidelines and reasoning behind the extraction of each guideline are presented in [38]. However, these guidelines are not fool-proof recommendations that can be used naively. Each IoT application is different in term of their objectives, implementations, execution, etc.

[1]Detailed discussions about patterns and tactics are out of the scope this paper.

Table 1. Privacy-by-Design Framework

Guideline	DA	DPP	DPA	DS	DD	Capability	Minimise	Hide	Separate	Aggregate	Inform	Control	Enforce	Demonstrate	Privacy Risks
1-Minimise data acquisition	✓	✓					✓	✓							⊗ ⊖
2-Minimise number of data sources	✓						✓								⊗ ⊖
3-Minimise raw data intake	✓	✓					✓			✓					⊗ ⊖
4-Minimise knowledge discovery			✓				✓								⊗
5-Minimise data storage				✓			✓								⊗ ⊖
6-Minimise data retention period				✓			✓	✓							⊗ ⊖
7-Hidden data routing	✓				✓			✓							⊖
8-Data anonymisation	✓	✓	✓		✓			✓							⊗ ⊖
9-Encrypted data communication	✓				✓			✓							⊖
10-Encrypted data processing		✓	✓					✓							⊖
11-Encrypted data storage								✓							⊖
12-Reduce data granularity	✓	✓	✓		✓		✓	✓							⊗
13-Query answering					✓		✓	✓							⊗
14-Repeated query blocking					✓		✓	✓							⊗
15-Distributed data processing			✓						✓						⊗ ⊖
16-Distributed data storage				✓					✓						⊗ ⊖
17-Knowledge discovery based aggregation	✓	✓	✓	✓	✓					✓					⊗
18-Geography based aggregation	✓	✓	✓	✓	✓					✓					⊗
19-Chain aggregation	✓	✓	✓	✓	✓					✓					⊗
20-Time-Period based aggregation	✓	✓	✓	✓	✓					✓					⊗
21-Category based aggregation	✓	✓	✓	✓	✓					✓					⊗
22-Information Disclosure	✓	✓	✓	✓	✓						✓			✓	⊗
23-Control	✓	✓	✓	✓	✓							✓	✓		⊗
24-Logging	✓	✓	✓	✓	✓									✓	⊗ ⊖
25-Auditing														✓	
26-Open Source														✓	
27-Data Flow Diagrams														✓	
28-Certification														✓	
29-Standardisation														✓	
30-Compliance														✓	⊗ ⊖

Risk Types: Secondary Usage (⊗), Unauthorised Access (⊖)

We developed these guidelines to act as a framework to support software engineers, so they can adopt our guidelines into their IoT applications in a customised manner. For example, certain applications will require aggregation of data from different sources to discover a certain new knowledge (i.e. new pieces of information). We do not discourage such approaches as long as data is acquired through proper consent acquisition processes.

However, IoT applications, at all times, should take all possible measures to achieve their goals with a minimum amount of data. This means that out of eight privacy design strategies proposed by Hoepman [20], minimisation is the most important strategy.

The relationship between Hoepman's [20] design strategies and our guidelines are presented in Table 1. At a high-level, we have identified two major privacy risks, namely, secondary usage (⊗) and unauthorized access (⊖) that would arise as consequences of not following the guidelines. Secondary usage refers to the use collected data for purposes that were not initially consented to by the data owners [27], which can lead to privacy violations. Unauthorised access is when someone gains access to data without proper authorisation during any phase of the data life cycle. We will use the above symbols to denote which threat is relevant to each guideline. In Table 1, privacy guidelines are colour coded based on the primary privacy design strategy that they are belong to. However, it is important to note that some guidelines may belong to multiple design strategies. For example, **(Guidelines 6)** *minimise data retention period* can primarily be identified as a minimise strategy, but it can also be classified as a hide strategy as it reduces the period for which data is visible.

4.5 Use of Privacy-by-Design Framework

One of the primary objectives of the proposed PbD framework is to help software engineers to ask the right questions regarding privacy protection when designing IoT applications and their architectures. These guidelines provide them with a framework to start thinking about privacy and direct them to incorporate privacy features into IoT application designs. A piece of software is designed to solve a problem. Sometimes, a problem may be identified by a person who is affected by the problem (e.g., Robert, Michael or Jane in our motivating scenarios). At other times, a third party company may identify a generic problem that affects many other people (e.g., Enterprise resource planning solutions). This type of software engineering is common in the IoT domain as well. Some IoT solutions are generic middleware platforms that can be used to build end to end applications. Others are complete IoT applications that aim to solve a specific problem [36, 37].

However, problem owners mainly focus on the requirements that would help to solve their problem [3], ignoring privacy considerations. Therefore, privacy requirements are largely overlooked when designing software architectures for IoT applications. The PbD framework allows both problem owners and software engineers to sit together and discuss the problem and incorporate privacy protecting measures into IoT application designs.

In section 3, we presented three use case scenarios. For each scenario, we have a problem owner's expectation and a brief set of requirements. There are no explicit reference to privacy protecting measures. We assume, additional information can only be gathered through questioning the problem owners and domain experts. In the user study, we simulated such discussions between the problem owners (i.e, represented by ourselves, the researchers) and the software engineers (i.e., represented by the study participants). Our hypothesis was that the PbD framework helps software engineers to ask questions from both problem owners and domain experts in order to extract detailed requirements that could be used to design privacy into IoT applications.

Let us revisit the scenario presented in section 3.1 in order to demonstrate how to use the PbD framework to extract privacy requirements towards designing a privacy-aware IoT application.

Guideline 1 leads software engineers to ask the question: what type of data is required to achieve the Robert's objective? In our scenario the problem owner responds as follows:

Robert collects data using wearable sensor kits. The collected data types are pulse, oxygen in blood (SPO2), airflow (breathing), body temperature, electrocardiogram (ECG), glucometer, galvanic skin response (GSR-sweating), blood pressure (sphygmomanometer), patient activity (accelerometer) and muscle / eletromyography sensor (EMG). Accelerometer are used to derive patient activity. In addition to the sensor data, weather information such as temperature, humidity are also important for the Robert's research. Patients' mobile phones GPS sensors and weather APIs are used to collect such information. The data collection sampling rate is expected to be 30 seconds. Data is only

required to be collected when patients are performing either one of the monitored activities (i.e. walking with walker or crutches, or climbing stairs).

Based on this information the software engineer can decide not acquire any other types of data and also design appropriate sampling rate controls into the application. This will have the effect of minimising data acquisition and reducing the risk of both secondary usage and unauthorised access to private data.

In a similar fashion, guidelines 3, 5, 20 and 21 would lead a software engineer to ask questions such as: what type of data is required in raw format and what type of information can be aggregated in order to reduce privacy risks?. As a result, the following information may be gathered.

Robert requires oxygen in blood (SPO2), airflow (breathing), body temperature data types in raw format and need to be accurate. The data collection sampling rate is expected to be five seconds. In contrast, other data items can be aggregated into averaged values (e.g., aggregated over two minutes).

Similar guidelines based questioning can be used to extract privacy requirements which the software engineers can use to systematically design privacy aware IoT applications. Due to space limitations, we don't detail all the questions that could be asked in relation to the scenario. Instead, below we provide the information that could be acquired using our PbD approach by annotating a detailed description of the scenario with references to the relevant PbD guidelines at the end of each statement.

*The sensor kit is expected to push data to the patient's mobile phone using Bluetooth. The mobile phone pushes data to the rehabilitation centre's local server using Wi-Fi. The local server pushes data to the cloud IoT platform. Patients come to the rehabilitation centre 3 days a week in order to perform the tasks assigned to them. Another 3 days they perform the task at their homes. The smart phone is expected to push data to the local server at the end of each day (**Guideline 6**). However, if the patients perform their tasks at home, data need to be kept stored on the mobile until the next time they visit the rehabilitation centre (**Guideline 6**). The speciality nurses monitor the progress and advice the patients on weekly basis. The speciality nurses' responsibility is to make sure that the patient are performing the tasks as assigned by the recommendation system and assists patients if they have any difficulties in following the assigned tasks and schedules. Robert is required to analyse data every six months in order to understand the how to improve the rehabilitation processes in a personalized manner (**Guideline 6**). For long term data analysis purposes, Robert's application stores data after averaging over five minute (**Guideline 6**).*

Table 2. Relevant Privacy Requirements for Each Scenario

Guideline (↓) Use Case Number (→)	1	2	3
1-Minimize data acquisition	✓	✓	✓
2-Minimize number of data sources	–	✓	–
3-Minimize raw data intake	✓	✓	✓
5-Minimize data storage	✓	✓	✓
6-Minimize data retention period	✓	✓	✓
7-Hidden data routing	✓	✓	✓
8-Data anonymization	✓	✓	✓
9-Encrypted data communication	✓	✓	✓
11-Encrypted data storage	✓	✓	✓
12-Reduce data granularity	✓	✓	✓
15-Distributed data processing	✓	✓	✓
16-Distributed data storage	✓	✓	✓
18-Geography based aggregation	–	–	✓
20-Time-Period based aggregation	✓	✓	✓
21-Category based aggregation	✓	✓	✓
	13	14	14

*Robert's application requires average over five minute when patients are performing their tasks (**Guideline 20**). Patient data can be anonymized (**Guideline 8**). Data storages in moth mobile device, local server and Robert's cloud server should store data in encrypted form (**Guideline 11**). End-to-end encryption can be used to secure the data communication (**Guideline 9**). Robert does not require exact location of the patient where they may have performed the activities. The requirement is to acquire the weather parameters such as temperature, humidity, etc. Therefore, location data can be abstracted without affecting the accuracy of the data (**Guideline 12**). In this IoT application, data processing and storing happens in three different locations (nodes), namely, 1) patient phone, 2) local server, and 3) Robert's cloud server (**Guideline 15 and 16**).*

The above example illustrates how our PbD guidelines could be used to extract additional information regarding a use case which enables software engineers to design appropriate privacy enhancing features into their IoT applications. In order to evaluate the effectiveness of our approach, we developed similar detailed requirement descriptions for each of the use case scenarios, which we have omitted here due to space limitations. It is important to note that not all privacy guidelines are relevant to all IoT applications. In Table 2, we summarise which privacy guidelines are relevant to each scenario.

5 EVALUATION

This section explains how we conducted the evaluations and justifies our research methodology. More specifically, we conducted two studies as follows:

(1) **Study 1 (Primary)**: [Interview based] This was our primary study in which we tested our primary hypotheses. It was administered by a researcher. In this study, we focused on both quantitative (for hypothesis testing) and qualitative data. In brief, our primary objective was to answer the question *'Can the proposed PbD framework guide novice / expert software engineers to design IoT applications that are more privacy-aware than they would do otherwise?'*. Additionally, we explored engineers' (privacy) mindset towards each guideline, there usefulness, and applicability towards different IoT use case scenarios.
(2) **Study 2 (Secondary)**: [On-line activity based] This was a self administered on-line study. In this study, we explored engineers' (privacy) mindset towards each guideline, there usefulness, and applicability towards different IoT use case scenarios. In contrast to study 1, here we used a anonymised, informal, and relaxed methodology (i.e., self administered on-line activity and given 3 days to complete). We used this study to strengthen our findings in Study 1 as well as to reach theoretical saturation. Theoretical saturation is the phase of qualitative data analysis in which the researcher has continued sampling and analysing data until no (or very minimum) new data appear [26]. In this study, we mainly focused on qualitative data (though we present some quantitative aspects).

For each study, first, we will explain the aims of this study. Next, we explain and justify the participant recruitment strategy and sample size before describing the procedures followed at each step of the study. We partially inspired by the LINDDUN [13]'s evaluation methodology. Similar to LINDDUN, we also followed a use case based evaluation techniques.

5.1 Study 1 (Primary) - Interview-based

5.1.1 Purpose. The purpose of this work is to explore how our PbD framework can help software engineers to design privacy-aware IoT applications. Through user studies, using quantitative and qualitative data analysis, we aimed to answer following three questions that explore the effectiveness of the proposed PbD framework. We discuss these questions later in this section.

- Can the proposed PbD framework guide novice software engineers to design IoT applications that are more privacy-aware than they would do otherwise?

- Can the proposed PbD framework guide expert software engineers to design IoT applications that are more privacy-aware than they would do otherwise?
- Out of novice and expert software engineers, who would benefit most from the proposed PbD framework? or in other words, does the software engineering expertise matter when it comes to incorporating privacy protection features into IoT application designs?

In the first two questions above, we consider the design of a IoT application to be more privacy-aware if it considers a greater number of privacy concerns to incorporate appropriate privacy protecting features. We measure this in terms of the number of privacy guidelines considered by the study participants when designing the example IoT applications.

5.1.2 Recruitment and Remuneration. In total, we recruited 10 participants for the study of which five were novice software engineers and five were expert software engineers. A participant was classified as a novice if they had less than three years of experience (full time) in a software engineering role (design or development). Participants with more than three years of experience (design or development), were considered to be experts. We adopted an opportunistic sampling technique and recruitment was from the staff and student populations at The Open University and the University of Surrey. No criteria other than software engineering experience was considered when recruiting. We collected demographic information such as age, highest education qualification, and the number of years in a software engineering role. Each participant was compensated with shopping vouchers valued at GBP 20. There were no failure criteria as long as the participant attend the data collection session of the study. The study design was reviewed and approved by the Human Research Ethics Committee of The Open University. Table 3 summarises the demographic information about the participants. We have labelled them E1-E5 (Expert) and N1-N5 (Novice) and consider them as independent cases during the qualitative analysis process.

5.1.3 Procedure. All the data collection sessions were carried out as 1-to-1 lab-based observational studies [45]. The principal investigator (PI) acted as the facilitator as well as the observer during each of the sessions. The duration for each session was 1.5 hours. At the beginning of the each session, participants were given the consent from to sign off and brief demographic information were collected. We audio recorded all the discussions between the participants and the PI for qualitative analysis purposes. Next, participants were given a instruction sheet, as shown in Figure 5, that comprised a set of example notations that could be used to illustrate the design of the IoT applications. Participants were reassured that adherence to the notation was not essential.

Table 3. Demographics of Study 1 (Primary Study)

ID	Age	Highest Qualification (ICT)	Years of Experience	Area of Experience
E1 (Male)	20-29	MSc	4 (Expert)	Desktop, Mobile, Web
E2 (Female)	30-39	PG(Diploma)	8 (Expert)	Mobile, web, system integration
E3 (Female)	30-39	MSc	8 (Expert)	Embedded, Textile Design, wearable
E4 (Male)	40-49	BSc	10 (Expert)	Data Science
E5 (Male)	20-29	BSc	6.5 (Expert)	Desktop, Mobile, Web
N1 (Male)	30-39	PhD	3 (Novice)	Signal Processing
N2 (Male)	30-39	MSc	2.5 (Novice)	Desktop
N3 (Male)	20-29	BSc	3 (Novice)	Desktop
N4 (Male)	30-39	MSc	1 (Novice)	Desktop
N5 (Male)	30-39	MSc	3 (Novice)	Web

We divided the rest of the study into three rounds, which we call **Round 1 (NoPrivacy):** *IoT application design without any guidance to consider privacy or reference to the PbD guidelines*, **Round 2 (WithPrivacy):** *IoT application design without privacy guidelines but guidance to consider privacy*, **Round 3 (WithPbDGuidelines):** *IoT applications design with privacy guidelines*. However, this segmentation was only used to structure the discussions and observations and none of them were formally acknowledged or identified during the interviews.

Round 1: It is important to note that we only informed the participants that this is an IoT application design study, without making any reference to privacy. By doing so, we expected them to be unbiased and follow the steps they would have naturally followed in designing an IoT application. We gave them separate A4 sheets to draw their IoT application designs with respect to each use case. They were briefed about the notations they could use, but we did not restrict them to any particular notation as long as their designs are understandable and clearly annotated.

Next, the participants and they were asked to design IoT applications to satisfy the requirements of each the scenarios presented in Section 3. Initially the participants worked from the summary descriptions provided in this paper but the PI was prepared to provide more detailed information, similar to that presented in Section 4.5 if the participant explicitly asked any related questions. We designed the study to simulate a conversation between a software engineer and a problem owner where the engineer is trying to elaborate the requirements and design the architecture of the IoT application.

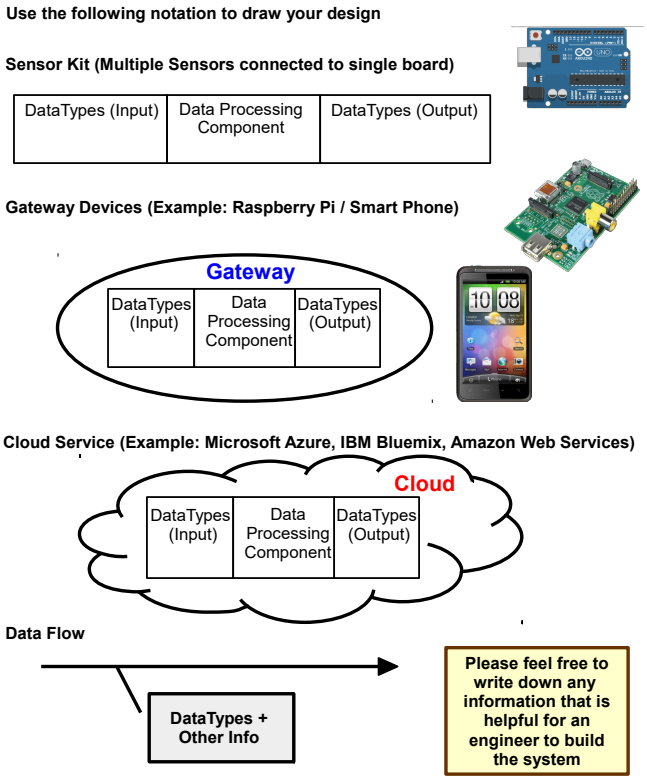

Fig. 5. Notations to be used in IoT application Design

We encouraged participants to ask as many question as possible about the case studies and application requirements. This means that participants could have asked any question regarding privacy requirements if they wanted to. Some of the commonly asked questions are discussed later in this paper. We gave them 50 minutes to complete the IoT application designs for the three use cases provided. However, the time limit was given as a guidance and we did not enforce it. The actual time of each study was varied based the the amount taken by the participants on each phase. So the actual total time varied between 1 hour and 15 minutes to 2 hours. We always allowed each participant to naturally progress through their designs without rushing them through each phase. After the designs were completed, we asked the participants to explain their designs and briefly justify their design decisions.

Round 2: Next, we gave participants a ten minute introduction on privacy. In order to achieve consistency, accuracy, and a well recognised description of privacy and related challenges, we selected two videos[2] [3] from You Tube produced and published by *Privacy International* (www.privacyinternational.org). The objective of showing these videos to each participant was to provoke them to think about privacy and help them to recall their past experiences and knowledge of dealing with privacy issues. This was intended to help them with the next task. It is important to note that we did not provide any additional material on privacy at this stage.

Next, we asked the participants to refine their previous IoT application designs further to protect user privacy. Similar to the previous round, questions were welcomed. We gave the participants 20 minutes to refine the IoT application designs for the three use cases provided. For Round 2, they wrote in a different colour to round 1, which enabled us to distinguish the design activities from each round clearly. After the revisions were made, we asked the participants to explain their revised designs and how they improved the privacy protection.

Round 3: Finally, we gave participants an introduction on the PbD guidelines and how to use them. We asked the participants to refine their previous IoT application designs to protects user privacy. Similar to the previous

[2]What Is Privacy? (youtube.com/watch?v=zsboDBMq6vo)
[3]Big Data (youtube.com/watch?v=HOoKhnvoYkU)

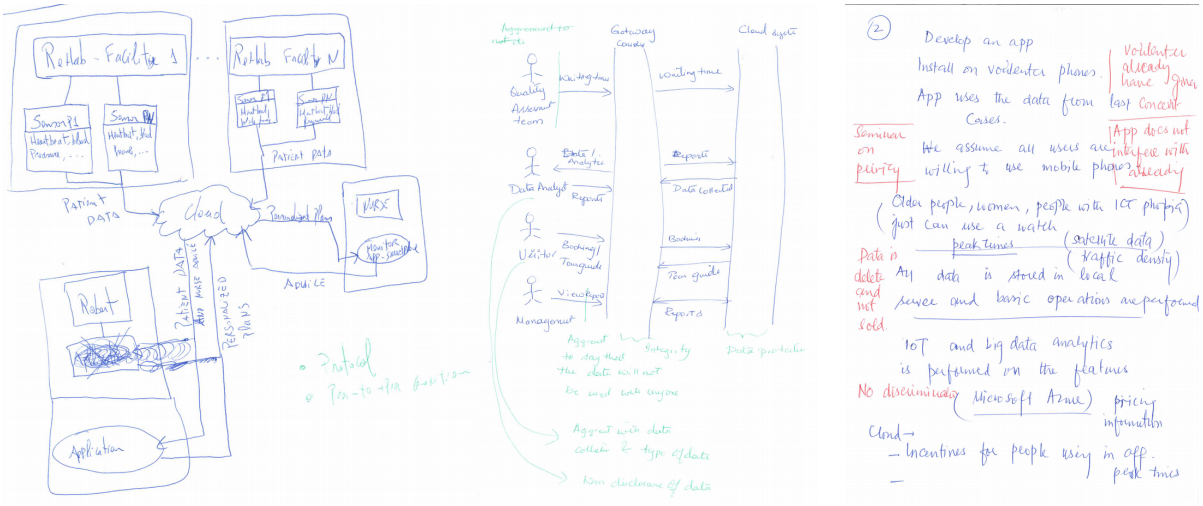

Fig. 6. Sample IoT application designs that illustrate a variety of approaches used by participants to express their high-level designs. In addition to block diagram notations based on the examples we provided, participants used sequence diagrams, pictorial diagrams and detailed text descriptions as illustrated above.

round, questions were welcomed. We gave the participants 20 minutes to enhance the privacy features of their IoT application designs for the three use cases provided. After the revisions were made, we asked the participants to explain their revised designs and how they improve the privacy protection. Once completed, we collected the IoT application designs produced by the participant. Some sample application designs produced by participants are presented in Figure 6.

5.2 Study 2 (Secondary) - On-line Activity-based

5.2.1 Purpose. The study 1 was conducted by a researcher in interview form. Therefore, participants may have compelled to think and perform harder during the study. On the other hand, at times, we failed to convince (about the relevance and the importance of a guideline) the engineers to apply certain guidelines into a given IoT application scenario. In real world situations, these PbD guidelines are expected to be used by engineers without supervision (or assistance). By taking these factors into consideration, in this study, we aimed to explore engineers' mindset towards Pbd guidelines. More specifically, we explored what software engineers think about each guideline and their reasoning and decision behind applying them. It is important to note that we have gathered similar information in Study 1 (Round 3) as well. We will compare these results in Section 6. We used study 2 to strengthen the findings in Study 1 as well as to reach theoretical saturation [26].

5.2.2 Recruitment. In total, we recruited 17 participants (one participant dropped out and end up with 16) where they completed 32 IoT use case scenarios. This survey was conducted in French University and participants were Master students (i.e., convenient sampling). No compensation were given to the participants. Based on the lessons we learned from study 1, we did not consider expert level as factor in this study. Demographic summary is presented in Figure 7.

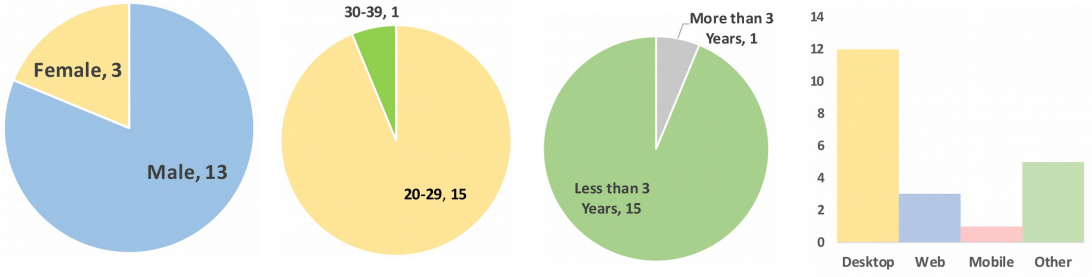

Fig. 7. Demographics of Study 2 (Secondary Study)

5.2.3 Procedure. This study was organised as a on-line survey. Each participant was given three days to complete the survey based activity. We used the same three use cases as study 1. We formulated the study into two logical rounds (in contrast, to the three rounds in study 1): *1) without privacy guidelines*, and *2) with privacy guideline*. Each participant were given two on-line surveys to compete. Each survey has two rounds as follows:

Round 1: A use case scenario is presented to each participant (same as study 1). Then, we asked the question *"What kind of privacy protecting measures that you might incorporate into the IoT application design?"*. We also recommended the participants to sketch a data flow diagram saying *"Even though it is not required, it might be useful for you to sketch a data flow diagram to understand how you might want design the IoT application"*.

Round 2: In this round, we presented different PbD guidelines, one by one, and asked the participants to answer appropriately. (*"Please read the above privacy guideline. Do you think whether this guideline can be applied during the IoT application design process. If 'Yes'; please briefly explain how you might apply this guideline. If 'No': Please explain why this guideline cannot be applied").*

6 FINDINGS, DISCUSSION AND LESSONS LEARNED

In this work, we followed the multimethod-multistrand method [49]. More specifically, we used two data collection method (i.e., interviews and on-line activity) and collected multiple types of data (i.e., IoT application designs [*drawings*]), participants views [*audio*], participants ability of identifying privacy preserving measures [*numeric*]). In this section, first, we analyse and discuss the results quantitatively. Our aim is to address the three questions presented earlier in Section 5.1 with the help of data collected through the users studies (study 1). Later, we discuss the results of both study 1 and 2 qualitatively in order to understand software engineers' approach towards designing privacy-aware IoT applications. Please note that, unless otherwise we explicitly refer to study 2 (secondary), we are implicitly referring to the study 1 (primary).

6.1 Quantitative Analysis (Exploring Effectiveness)

As shown in Table 2, we expected each participant to identify a maximum of 41 privacy protecting measures (Use-case 1: 12 measures, Use-case 2: 14 measures, Use-case 3: 14 measures). The participants may identify these privacy measures either using their experience, common sense, or using the PbD guidelines. In total, we collected 410 data points (41 measures x 10 participants). We have presented a snapshot view of the data gathered using two heat-maps in Figure 8 where the results for novice and expert software engineers are presented separately.

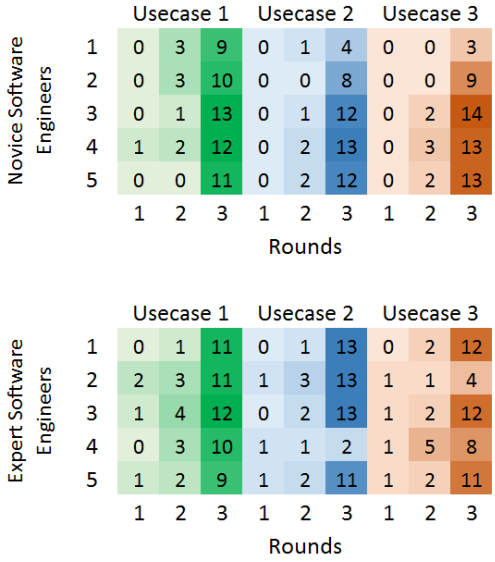

Fig. 8. The three use-cases are marked using three separate colours. The x-axis denotes how much privacy protecting measures have been identified in each round (higher the number of privacy requirements identified, darker the shading is). The y-axis denotes the participant ID.

The heat-maps clearly show that both novice and expert software engineers were able to identify a greater number of privacy protecting measures by using the PbD guidelines than they would do otherwise. In Figure 9, we illustrate how the mean of the *'number of privacy measures'* identified, by both novice and experts software engineers, changes at different privacy knowledge levels. The average number of privacy measures identified, in Round 1, by novices is 0.2 and experts is 2.2. Similarly, the average number of privacy measures identified,

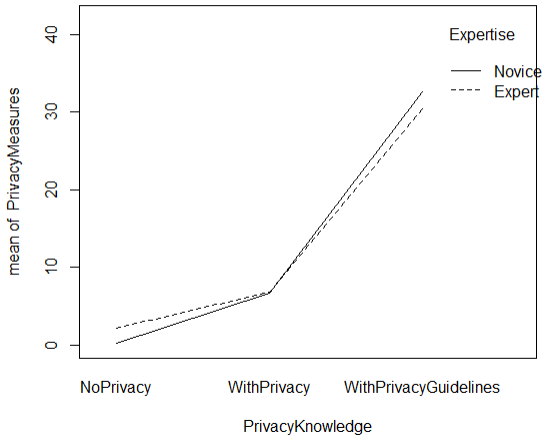

Fig. 9. Number of privacy measures identified in each round

in Round 2, by novices is 6.6 and experts is 6.8. Further, the average number of privacy measures identified, in Round 3, by novices is 32.6 and experts is 30.4.

Next, we ran statistical tests (i.e., ANOVA[4]) and found out that there is a significant difference between the number of privacy measures identified with and without the PbD guidelines (within=PrivacyKnowledge (ANOVA $p = 2.099781e\text{-}09$; $p < 0.05$)). Further, our results show that the expertise of the software engineers (novice vs. expert) has no significant effect on the ability of identifying privacy protecting measures (between=Expertise (ANOVA $p = 6.897806e\text{-}01$; $p < 0.05$)) .

Figure 10 illustrates which privacy guidelines have been identified in each round by the participants. It is also important to note that PbD guideline 2 and 18 were only relevant in one of the use case scenarios (out of three)

[4]statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide.php

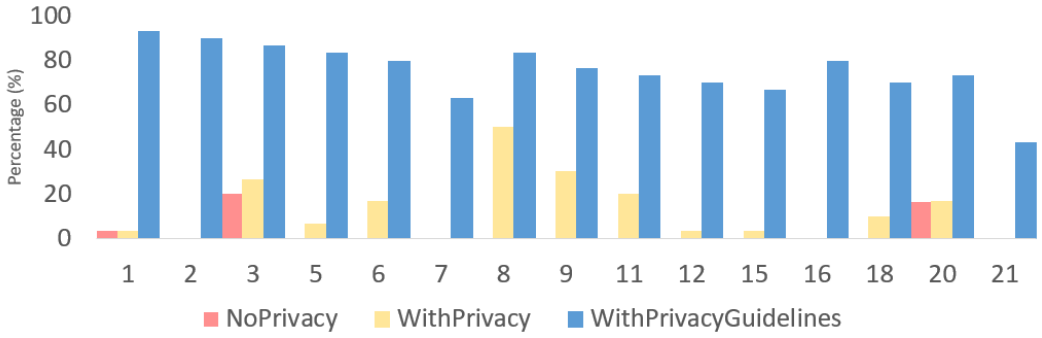

Fig. 10. Privacy guidelines identified in each round: the x-axis denotes privacy guidelines by number and each colour represents the three rounds. The y-axis denotes the frequency which participants identified a given privacy guideline. Legend for both Figure 10 and Figure 11: 1-Minimize data acquisition, 2-Minimize number of data sources, 3-Minimize raw data intake, 5-Minimize data storage, 6-Minimize data retention period, 7-Hidden data routing, 8-Data anonymization, 9-Encrypted data communication, 11-Encrypted data storage, 12-Reduce data granularity, 15-Distributed data processing, 16-Distributed data storage, 18-Geography based aggregation, 20-Time-Period based aggregation, 21-Category based aggregation.

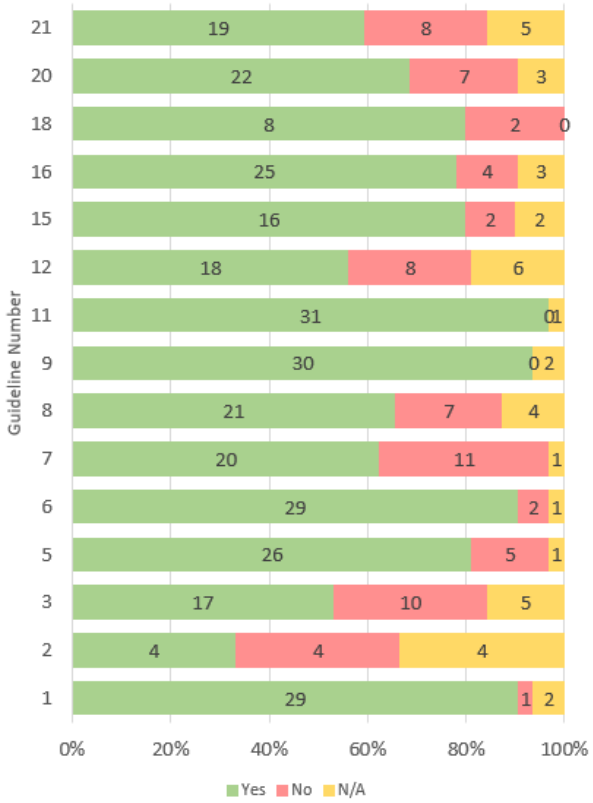

Fig. 11. Participants' view on whether a given guideline can be applied or not to the given IoT use case scenarios (Results of Study 2) Legend: Yes = participant agrees that a given guideline can be applied; No = participant refuses to apply a given guideline; N/A = participant did not clearly specify whether the guideline is applicable or not.

which justifies its unusually low identification rate in Figure 8. To avoid any confusions, we have presented the x-axis of the Figure 10 as a percentage. Comparatively, more participants have identified PbD guideline 3 (Minimise raw data intake) and 20 (Time period based aggregation) in Round 1. However, our discussions revealed that participants integrated these features into their designs to meet functional requirements of the scenarios rather than due to a consideration of privacy. In Round 2, after we explicitly asked them to improve the privacy awareness of their IoT application designs, participants primarily identified guidelines 8 (data anonymisation), 9 (encrypted data communication), and 11 (encrypted data storage). In Round 3, there was no noticeable different in the guidelines identified by the participants.

Results of round 3 of study 1 and round 2 of study 2 is fairly comparable. Results from both study 1 (Figure 10) and study 2 (Figure 11) show that participants mostly understand and agree with the usage of encryption (communication and storage) and data minimization very well. However, we can observe higher refusal / disagreement rate in study 2. We discuss this phenomenon further in Section 6.2.

In total, we expected participants to identify a maximum of 410 privacy preserving measures that they could take in order to improve the privacy awareness of the three given IoT application scenarios. They identified 308 privacy preserving measures with the help of the PbD guidelines. Therefore, the success rate is 75.12%. As

shown in Figure 9, this results is significantly better than *'without PbD guidelines'*. Based on our discussions with the participants, we identified two main reasons why sometimes they failed to apply a given guideline into their application designs: 1) unique IoT application designs eliminates the necessity of applying certain privacy preserving measures and 2) the lack of time. It is important to note that the PbD guidelines can only be applied to protect user privacy in certain application design contexts. Some participants designed their IoT applications in such away that certain PbD guidelines have no role to play. We discuss one such example in the next section.

6.2 Qualitative Analysis and Lessons Learned

We followed Miles [29] framework to conduct the qualitative analysis. Further, for data reduction phase, we use Richards [41] three tier coding technique (i.e., descriptive coding, topic coding, and analytic coding). The thematic areas we found by analysing the data from both study 1 and 2 are as follows:

(1) Challenges of the Methodology and Opportunities.
(2) Challenges Towards Adoption in Real-world.
(3) Software engineers need to develop *Privacy Mindset*.
(4) Privacy guidelines provide cues to follow up for software engineers to explore beyond their expertise
(5) Knowledge limitations and gaps could lead to weaker privacy designs.
(6) Convincing software engineers to apply PbD guidelines could be difficult.
(7) From Guidelines to Patterns: Different types of advice could be useful for software engineers to solve different privacy problems.
(8) Guidelines should be better explained.
(9) Guidelines are important and provides interesting ideas towards designing privacy aware applications.
(10) Post hoc rationalisation and software engineers felt guilty for not pro-actively taking measures to protect user privacy.
(11) Acquisition of user consent should not be used to counter poor privacy design choices.
(12) Lack of consistency and sometimes engineers need a push.
(13) Software engineers' IoT applications designs are influenced by their own expertise.
(14) Privacy should not be treated like a secondary objective when designing IoT applications.
(15) Time is a unique type of data that has direct impact on privacy.
(16) Some privacy issues can be eliminated by using alternative technologies.
(17) Software engineers consider authentication and encryption as the only ways to protect privacy.
(18) Over thinking and applications could lead to unnecessary complexities.

6.2.1 ***Challenges of the Methodology and Opportunities***. As shown in Section 3, we formulated our study based on IoT use case scenarios. During the design of this study, we had to make the decision on the level of details that we need to provide in each use case scenario. Our aim was to provoke the participants thought process. Therefore, we decided to keep the scenario as brief as possible. However, we did not want to make the participants completely blank and wanted to give them sufficient context information to start their thought process. By doing this, we expected participants to face difficulties in designing the IoT applications without our (i.e., interviewer) help. Therefore, we expected them to ask lot of questions about the scenario and design requirements. Further, we always informed the participants that we are happy to provide any information that is necessary to design the application and strongly encourage to ask questions. Further, we intentionally embedded vague and questionable statements in each scenarios to encourage participants to ask questions. Some sample sentences are as follows.

[Extract from Usecase 2] *However, TrueLeisure continuously monitors and assesses the service qualities and several other aspects in each of the amusement parks.*

[Extract from Usecase 3] *In order to develop an efficient and effective plan, Michael needs to understand movements of people and several other aspects of their activities.*

Segments such as *'several other aspects'* and *'understand movements of people'* forced participants to ask questions such as *'What will be movements of people?'* [E1] and *'What would be several other aspects? That's kind of too broad.'* [E1]. Further, in use case three, we asked the participant to focus on capturing *'waiting time'*. However, participant E1 challenged this by saying *'Just the waiting time might not be enough'*. As expected these leads successfully initiated natural conversation between the interviewer and the participant. However, non of these discussions grew into privacy requirement gathering. Participants questions were primarily lead towards functional and technological requirements. This was not a complete surprise as this kind of mindset is the challenge we are trying to address. Hence, it strengthen our argument of the necessity of developing privacy mindset.

It is important to note that out of ten participants (30 designs) only one participant explicitly discussed the privacy requirement during one of the round 1 designs. For example, participant [E1] said *'Thinking about issues as privacy, for example, I would just be interested to know how many are there and not who is there. By that I could, for example, use the signal of the mobile phone and identify how many mobile phones are there. Then I can kind of understand the movement.'* when designing for the use case 2. However, this seems to be a one of incident, though our end goals is to see more of this type of proactive privacy thinking at early stages of the design process.

The total duration wa about 1.5 hrs. We should admit that we asked participants perform substantial task during the task. Even though we did not hear any direct complaints about the workload or duration, we felt, at time, participants got tired. However, we do not consider this tiring has made any impact on our final results. On the other hand, we also need understand , in real world scenario engineers would get tired. Going through privacy guidelines and decided when, whether, or how to apply them is a significant and tiring task, specially when there number guidelines are increased. On the other hand, if we try to reduce the number of guidelines, this will lead to increase the abstractness of vagueness of each guideline (e.g., Ann Cavoukian [10]). In that situation engineer may get tired by thinking and translating principles into actionable guidelines by themselves. In either way it is hard to avoid this tiring process on applying privacy awareness in into IoT application without building assisting design tools.

6.2.2 ***Potential challenges towards adoption in real-world.*** We can observe higher refusal / disagreement rate in study 2 (round 3) compared to study 1 (round 3). We attribute this difference to number of factors:

- Self-administered nature of the study: Therefore, they are relaxed and do not feel necessity (or pressure) to agree with the guidelines and motivate them to express their view freely.
- Absence of supervision: Lack of overseeing process may have also lead to lack of focus and perform.

However, study 2 is much closer to real world situations where software engineers have to use the proposed PbD guidelines by themselves. Therefore, tooling support will be really essential to assist software engineers to improve their application designs. Automated tools will help to overcome the above two factors.

6.2.3 ***Software engineers need to develop a 'Privacy Mindset'.*** Software engineers usually have trained their mind to think about software designs from a business view point. This is understandable as software engineering projects are typically starts with business requirement conducted by business analysts. For example, participant [N1] recognised the importance of anonymising and deleting the data with regards to scenario 1 in round 2. However, he was reluctant and refused to apply the same ideas into scenario 3 saying *'I mean I can see a whole bunch of scenarios where they would want to pitch different kinds of deals to these individuals. That's why I'm saying it's very unlikely that they would adopt any sort of privacy enhancement measure, to get rid of or de-anonymise that data. Yes, just realistically I don't see that happening in that use-case'*. N1's mindset is the reality mindset we are dealing with and it would take lot of effort to change this mindset. Guidelines put effort in this

direction. However, we will require more effort such as sophisticated tools that can alter a given design (e.g., DFD diagram) and force engineers to think privacy as a first class citizen.

We also observed that some participants think about privacy very lightly. For example, participant [E4] was not much interested in thinking about privacy from certain aspects such data minimization saying that *'So, that information would, in theory, it might be possible to infer from the raw data, but in practice that could be quite tricky (Laughter).'*. It is important to understand that not all engineers understand the challenges and risks in privacy. For example, participant would not have said this if he new about state of the art (e.g., accelerometer data from and smart wrist band can be used to identify ATM pin numbers[5]).

Another aspect of the mindset problem is blind assumption towards some domains. For example, one of the use case scenarios we used was on healthcare domain. Software engineers seems to have a predefined mindset regarding certain domains. Participant [E2] correctly raised the concern that *'how much is that going to impact the health plan, or the rehabilitation plan for that. If they don't have access to those, to be able to link it to the medical records. Is that going to impact the patient, if the plan is kept separate from their own doctor?'*, when thinking about applying data minimization guideline. However, more often engineers tend to assume that doctors need the most accurate and data with highest possible granularity to make decisions, though it is not necessarily in all cases. For example, participant [E3] was reluctant to apply privacy preserving measures thinking that her actions would jeopardize the medical outcomes and mentioned (in round 2) and mentioned *'This one is quite challenging, this is the medical one, because obviously we need to use that data in order for the nurses to improve the experience in some way. So I do not know'*. We observed similar remarks in study 2 as well. For example, one participant refused to apply category-based aggregation saying *'It's interesting, but it lacks precision in a medical context'*.

6.2.4 ***Privacy guidelines provide cues to follow up for software engineers to explore beyond their expertise***. The IoT application development process requires different types of expertise to come together to work efficiently. This is a fundamental difference between traditional web, mobile, desktop, embedded and IoT. Therefore, designing privacy aware applications could be challenging specially when the designer do not have certain types of expertise (e.g., networking, embedded design). For example, participant [E1] highlighted his lack of expertise saying *'Yes. The main problem is the cloud itself because as the data will be going through the cloud, the data will be available for attackers or someone like that. A way just so it might be or to take a look into which cloud service we are using. The protocols and this kind of stuff because this will be really, really important. Yes. It's not my speciality, this area'*. Privacy guidelines can effectively educate and inform intelligent, but non specialists engineers and designers. This is an important step towards developing privacy mindset.

As a side effect, engineers may also learn to identify and respect different design requirements impose by their colleagues who are looking at a given design from a different speciality point of view (e.g., networking). Further, guidelines can also force non speciality engineers to look for speciality assistance as necessary to design better privacy aware applications. Without guidance, non speciality engineers may not know where or when to seek assistance. We heard similar expression few time such as *'The hidden data routing, I had not actually heard of that before, I think that is quite exciting. I think, yes, that would be good to do.'* [E3]. In another instance, participant [E2] mentioned that *'The distributed data processing, I had not thought about at all to be honest I do not think but yes, I think it could definitely apply to all of these in some way. I am not sure how because I do not work in networks, or do this kind of stuff but I think that it would be good'* . These expressions convince us that, guidelines play more than the guidance role, but can effectively play the educating role.

6.2.5 ***Knowledge limitations and gaps could lead to weaker privacy designs***. Previously, we discussed the challenge of engineers not having certain expertise. We observed slightly different type of cases where the participants incorrectly believe what they new was correct. For example, participant [N1] mentioned that *'This is*

[5]http://uk.pcmag.com/smartwatches-1/82816/news/how-smartwatch-sensors-can-reveal-your-atm-pin

volunteers, it said, so I'm assuming that at the very start of this data collection, you would start off by collecting no data about the individual. Yes, so as far as you are aware, it's just somebody. So that should be fine for that.'. However, this is not correct. Even though we might not gather personal data initially, it could be possible to track the volunteers, if the communication is not secured through encryption.

In another case, participant [E1] mentioned that *'I guess it is not necessary to encrypt and anonymise the data.'*. However, this is not true. Encryption and anonymisation techniques are designed for two different purposes. Ideal approach is to do both instead of picking one. These techniques act as two lines of defences. Encryption makes the data unreadable without authorization. However, if somehow malicious party achieve to decrypt data, still if the data is anonymised, it makes it much harder for malicious party as they now need to de-anonymise as well. We found similar cases in study 2 as well. For example, one participant mentioned that *'Distributed data could not be necessary if all data is strongly encrypted'*. In reality, distributed storage and encrypted storage are two different independent guidelines that can be applied together.

Based on these above cases, it is clear that knowledge limitations of software engineers could lead to weak IoT application designs. Most reliable approach to address this challenges is to develop automated design tools to help the software design process.

6.2.6 ***Convincing software engineers to apply PbD guidelines could be difficult.*** We realised that, at times, convincing a software engineer to apply a particular privacy guidelines is difficult. For example, participant [N5] refused to apply 'categories based aggregation guidelines', even though we were successfully able to explain it to him saying *'Yes, I understood, but I don't think that we need the categories based aggregation, we don't need.'*. This means we need to do more to make these guidelines more useful, but also make sure we do not push them to over think as we discuss in section 6.2.18. One of the ways to address this issue could be developing privacy patterns. Patterns are more concrete and able to convince the usage in a given context more strongly than guidelines. We observed similar difficulties in study 2 as well. For example, one participant has refused to apply data minimisation guideline my mentioning *'No, we need precise data that we can treat , to be able to understand them'*.

6.2.7 ***From Guidelines to Patterns: Different types of advice could be useful for software engineers to solve different privacy problems.*** We realise that sometime, engineers' thinking process is just wrong. For example, participant [E1] said that *'In this case, he needs to know which one person it is. It's important because the personal is a person then I can't anonymise or blow it. Yes. Because this one is really a personal thing, so I think the main problem is the cloud.'*. However, this is not correct. In use case 1, personal information can be replaced by an ID (for example, without using the real name). However, in this particular instance, our engineer concluded that personal data has to be retained. However, this kind of problem can be address by developing patterns. As we discussed earlier, patterns are solutions for common design problems. What we discuss here is a common problem that is not some thing unique to use case 1. Guidelines do have limitations on how much concrete or specific can they be as they are developed with the expectation of apply for wide range of circumstances. However, pattern on the other hand are ideal to be used to address this kind of problem. In this case, indirectly we are solving a problem of knowledge gaps of engineers.

In a slightly different case, participant [N4] refused to apply 'minimise raw data intake guideline' saying that *'I think it was not considered in scenario three, where I said that we will be sending the video feed to the Cloud. That can actually give the information regarding a particular user at that particular place'*. Then we asked the question *'Is it necessary to send the entire video?, Is it sufficient to send, maybe you extract some pieces of that?'*. Then the participant realised the applicability of this guidelines and mentioned that *'Yes, instead of just sending the- because I was using the video feed- in the beginning I was using the video feed to calculate the queue times in the parking. So instead of sending just a complete video, you can just send the number plate information, if it can be done at the module at the camera. So you don't need to send that, because that will violate the personal space and privacy.'*. This

situation is somewhat difficult to handle by guidelines alone. Guidelines are designed to be broad than patterns and difficult to provide concrete examples. We believe that this kind of challenges can also be better managed through privacy patterns.

Let us consider the following extract from participant [E4]. *'So, âĂŸMinimised data acquisition for study one.' Actually, this was an interesting one, that I would say we didn't really think about at the beginning, because one of the ways in which I have failed to minimise data acquisition is continuous data collection, (Laughter) which we did have a reason for that, which was that it might be difficult for the user to have to switch- remember to switch it on and off.'*. This problem could have easily avoided by programming the mobile devices to automatically on and off the data collection based on the context (e.g., when doing the exercise). However, sometime we all may run out ideas and need little help. Privacy patterns and automated tool could comes in handy to address this type of challenges.

6.2.8 ***Guidelines should be better explained.*** We also had few instances where participants struggled to understand the differences between some guidelines. For example, participant [E1] asked *'What's the difference between the reduced data granularity and the data that I am getting? The one, the âĂIJMinimise data acquisition.'*. Such difficulties can easily be addressed by providing an example. Further, we also had some disagreements with some guidelines. For example, participant [N1] mentioned that *'Yes, that's what I'm guessing. Can you call it distributed processing?'*. The root cause of this problem is that, most engineers think distributed processing is all about processing at different clouds or servers. However, hierarchical data processing is also comes under distributed processing (e.g., some processing happens within micros-controller and the further processing happens in the gateways and final processing happens in the cloud). We observed similar remarks in study 2 as well. Participants have mentioned in several times that they do not understand certain guidelines or how they can be helpful. Figure 11 clearly illustrates this issue. However, this type of confusions and weaknesses (of PbD guidelines) can be easily addressed by providing examples. Automation tools could also help to address this issue.

6.2.9 ***Guidelines are important and provides interesting ideas towards designing privacy aware IoT applications.*** Over the course of the study, at number of times, participants clearly and sincerely express that guidelines are useful. For example, related to 'minimise data retention period guideline' participant [E1] mentioned that *'This, I haven't thought about it and this is very important. Very important.'*. We also had number of instanced that our guidelines were successfully changed the mind of the participants. For example, [N5] admitted the importance aggregating data saying *'So now I think if we collect the GPS data of that user, we need to aggregate the data by showing the GPS. The time periods by each aggregation, yes, I think this is quite an important thing because before that I did not think at all about that, but now I think instead of storing the raw data or the real time data, we just store the data in a certain amount of time, like, an hour or per days or per week, per month'*.

6.2.10 ***Post hoc rationalisation and software engineers felt guilty for not pro-actively taking measures to protect user privacy.*** We also observed post hoc rationalisation from most of the participants. After we showed the PbD guidelines, most of the participants felt the responsibility of addressing privacy issues in their IoT application designs. We observe their feeling more likely to some kind of guilty feeling. Most of them not only followed the guidelines and successfully improved their designs, but also claimed that they thought about certain privacy considerations before we showed them the guidelines, even though their designs did not show any evidence of this. This behaviour suggests that software engineers are well aware of the importance of privacy issues, though they do not make any effort to address them until an external impetus (e.g., we as researchers in this case or it could be pressure from laws and regulations, or explicit client demands) that explicitly encourage them to do so. When we explicitly encouraged them to address privacy issues, most of the participants felt the need of defending themselves and claim that they thought about privacy before. This post hoc rationalisation behaviour justifies the importance of developing a *Privacy Mindset* among software engineers. We observed three

different types of responses: (1) complete lies or made up answers where participant says that they have thought about a certain guideline, but they have not mentioned it in their designs on the paper or no evidence to suggest that they thought about it (e.g., *'So, I think I did consider the minimising the data that has been recorded'[E2]*), (2) reluctantly acknowledged that they haven't thought about it (e.g., *'So, seven, I had sort of considered that, but need to make it more explicit'*[E2]), (3) reluctantly acknowledged with some guilty (expressed in facial expressions and tone) (e.g., *'It is tricky actually because when you are thinking about stuff you are like I am kind of understand it, but I was not really thinking that at the time. [Laughter]. So maybe actually the walking one should be N/A as well actually'* [E3]).

6.2.11 ***Acquisition of user consent should not be used to counter poor privacy design choices.*** We also noticed the notion of using 'consent forms' as a way to overcome or by pass privacy challenges. In other word, engineers may design sloppy or poor application designs (in terms of privacy awareness) by using consent forms as an excuse. For example, participant [E2] mentioned that *'Okay. So, the first user case. Assuming that all the patients were part of the trial that the researcher is doing, and had already signed up to allowing the data to be tracked.'*. Further, she mentioned that *'The second one, as I said, these were volunteers, so, under the assumption that they've been signed up and made fully aware that this is going to track their movements'*. However, such data collection approach is not allowed in new GDPR regulations [14] where all the data collection and retention activities need to be justified and documented. We made similar observations in study 2 as well. One of the participants have mentioned that *'once analyse are made, data should be destroy. However, the user may want to access to his old data to know his evolution. So i think it's not possible to destroy them, unless the user asks for'*. Ideally, there has to be a properly justified reason in order to store data. Therefore, storing data until user explicitly asks to delete is a weak design choice, specially in the context of GDPR.

6.2.12 ***Lack of consistency and sometimes engineers need a push.*** We also noticed that some participants struggled to maintain the consistences. For example, participant [E1] suggested to use secure protocols for communication with regards to use case 1 even before seeing our privacy guidelines. However, he did not suggest to use secure protocols for scenario two and three. Later, he did the suggestion after seeing our guideline saying *'Yes. This would help with one, but with two and three, I haven't thought about it. Yes. I guess they are important to the use case two and use case three. That I haven't thought it but yes. It is really good to think about it'*. This issue is quite normal in many other domain. Maintaining consistency without any assistance is difficult. In medical field, check-lists are being developed to guide the surgical procedures. This is due to the fact that, even highly skilled doctors and medical staff struggled to always maintain the consistence without any assistance (i.e., reference points) [19]. We further discussed this issue in Section 4.1.

Additionally, we noticed that guidelines can also act as stimulating agent to help engineers to act upon things they already know. For example, [E3] admitted that she knows about the data retention very well. However, she did not apply them in the design and said that *'Yes, it is kind of in line with this one here. So, I kind of had inadvertently had thought about but probably not a mega amount. I am also the kind of person who would collect all the data and then decide to do what afterwards. [Laughter]. I am the typical scatty artist like that. With the retention period, I mean I know that it is something that you obviously need to think about, but to be honest I had not really thought about it before even this. I know from my own studies that I need to do that but when I was reading this I was not thinking, "Oh yes, I should only keep it for a little bit." I guess you would delete it after you sort of put it into a secondary context* . This means that there is actual gap between knowledge and actions. Guidelines can be used to bring gap between the tow which others engineers may do not want push themselves forward.

Further, guidelines can be used to eliminate the challenge of 'cold start' (i.e., start thinking about something without any assistance or structure). Therefore, guidelines could speed up the process. For example, we had one participant who could not identify any privacy measure in phase 2 by himself. Even though this is one off case out of ten, it is fair to assume this is not an isolated case depending on our sampling size. Participant [N3]

mentioned that *'About the privacy control, I don't have that much of knowledge about the privacy control.'* and then vaguely mention about using policies to govern the data management process. Therefore, privacy guidelines a re significantly useful in guiding this kind of engineers.

Rarely, we noticed that participants ask direct privacy related question in round 1. However, such questions are usually come from business requirement gathering line of thinking, not necessarily from privacy mindset. For example, participant [E4] asked *'So, could you just give me an example of kind of sensor that we might have or an example of the sort of data that you might be collecting from one of your patients?'*. However, the ideal question we would like them to ask is *'what would be the minimum data set that you need collect in order to achieve the task at hand?'*.

6.2.13 ***Software engineers' IoT applications designs are influenced by their own expertise***. Design and development of IoT applications require different types of expertise to come together and merge into single design. These designs are influenced by the expertise of the engineers. For example, engineer who is familiar with wireless network communication may look at a design from communication point of view. For example, participant [N1] implicitly thought about data minimization from networking point of view *'Are you gathering a lot of data, meaning you will not be able to transmit it over a wireless network? Or is it sort of a very low-bitrate data that you can collect on the cloud and analyse later? I'm wondering if you need to do any data processing at all?'*. It is important to note that engineers may implicitly apply certain guidelines without thinking about privacy, instead thinking about challenges in their own expert areas as shown above. Such decisions should not consider as privacy protecting measures taken by engineers.

Having expertise (or confidence) could also help engineers to make more concrete design decisions. For example, participant [N2] based on his own expertise mentioned that *'In this activity, we don't need very specialised data. I think two sensors are enough, the gyroscope. I have written the gyroscope and the heart-rate monitor. That actually tells us a lot.'* . In this extract, our participant, implicitly focus on minimise data course guidelines. In this context, our participant is confidant that particular data types are sufficient address the challenge at hand. This a contrast view we saw in Section 6.2.14, where the participant mentioned her willing to gather data *'just in case'*. More technical knowledge and expertise of the technology could lead to change the mindset from gathering all data to gathering sufficient data.

One of the ways to address this challenges is to create IoT knowledge bases. As IoT application development is going main stream and more individual and small teams are expected to enter into engineering force. Therefore, it would be useful to develop usable tools that can inform engineers specially regarding *'what can be achieved by different types of data'*. For example, what can be understood by analysing accelerometer data? what can be understood by temperature data or the questions would be, what are the different ways to detect human presence in a certain locations. Depending on the expertise of the expert you ask the above listed questions, the answer could be varied greatly. Different IoT application designs that achieve the same overall goal may have different consequences in terms of cost, accuracy, replicability, privacy awareness and so on. We propose to develop an IoT knowledge base where anyone can search answers to above mention questions. Such platform should be a crowd-source platform where different experts get to submit their experiences and also provide facilities to critique by each others solutions. Such resource would help us as a community to collectively push towards not only to achieve the desired objectives, but also to achieve them in a privacy aware manner.

Privacy guidelines can also be used to justify or contrast other design decisions. For example, a decision to collect less data in order to save bandwidth can be strengthen by the arguments brought in by the data minimization privacy guideline. Such triangulated decision will have much better chance in surviving in design reviews by multiple parties who have different expertise. Other way round arguments may also useful in making strong decision decision. For example, it would be much credible to put emphasis on the secondary benefits of data minimization guidelines, when possible, as it could be seen as not only a privacy protection measures but

also cost saving measure for the company in long run. However, the challenge is to combine privacy guidelines with secondary benefits. Above discussed knowledge bases could be very useful to address this challenge.

6.2.14 ***Privacy should not be treated like a secondary objective when designing IoT applications.*** Our study showed that software engineers do not consider privacy as a first class citizen in their IoT application designs. This justifies our decision to develop a PbD framework to guide the thought process of software engineers. During our user studies, participants candidly expressed their wish to collect as much data as possible (e.g., Participants [E2] said *'As a developer, you just wants all of the data'*). We believe that this mindset of collecting as much data as possible needs to be changed towards a *Privacy Mindset* where only the most essential data items are gathered and processed. We explained the privacy risks of gathering non-essential data in Section 4..

Another participant signalled that it is ok to collect data without any control saying that *'If it's completely anonymised, and it's just business data about who's come and come out.'* [E2]. This mindset is wrong and create additional problems such as resource wastage (e.g., for storage, data cleaning, data processing etc.). Further, anonymising is a risk mitigation approach, not a risk elimination approach. Anonymisation also could lead to privacy violations due to un-lawful de-anonymisation approaches. We heard similar views with regards to data storage as well.

6.2.15 ***Time is a unique type of data that has direct impact on privacy.*** Time is a unique piece of data that commonly gathered in IoT applications. Time can provide significant information when it is attached to a different data types (e.g., location). Time can also be used to aggregate data in certain ways to significantly reduce privacy risks. In IoT world, sensors collect data mostly with respect to time (i.e., frequency of collecting data). Typically, higher frequency means higher the granularity of the data captured, therefore higher the information that can be derived. However, the required frequency depends on the tasks at hand. Therefore, IoT applications need make particular attention the frequency of collecting data as it not only a contributing factor to privacy violations in long run but also make impact on energy consumption, storage and data communication.

6.2.16 ***Some privacy issues can be eliminated by using alternative technologies.*** An important aspect of IoT application design is the choice of the right sensors and technologies to collect data. We realised that these choices also have a direct impact on the privacy. In relation to Scenario 2 (section 3.2), one of our participants [E4] used stationary sensors that do not capture any personally identifiable information to collect the necessary data (e.g., pressure sensors deployed in the ground, motion sensors, infra-red sensors, and so on). Sensor technologies have their own strengths and weaknesses. Similarly, privacy risks also varies depending on the technology used. However, the decision on which technology to use is based on the exact application, cost associated, and how much privacy risks willing to take. For example, deploying pressure sensors on different paths of a given park would eliminate the necessity of hiring volunteers with wearable sensor kits and associated privacy risks. However, deploying such sensing technology in real world could be much more challenging, in terms of cost, time, and effort, than distributing number of sensor kits among volunteers. On the other hand, stationary sensors would eliminate the hassle of recruiting volunteers, managing them, and their sensor kits. The lesson is that privacy risks can also be reduced by selecting certain types of sensing technologies given that they are feasible to be used in a given IoT application.

6.2.17 ***Software engineers consider authentication and encryption as the only ways to protect privacy.*** It is also important to note that three participants identified authentication as a measure of protecting user privacy. However, in our PbD framework, we considered authentication as a security measure than a privacy protection measure. Further, three participants highlighted the importance of acquiring consent from data owners before collecting data. They also pointed out the importance of giving control to the data owners so they can decide on which data to share. Both consent acquisition (information disclosure - guidelines 22) and control (guidelines 23) appeared in our PbD framework even though we did not use them in the user study. Study 2

(round 1) also highlighted the same issue. As shown in Figure 12, most common privacy protection measures identified are authentication and encryption.

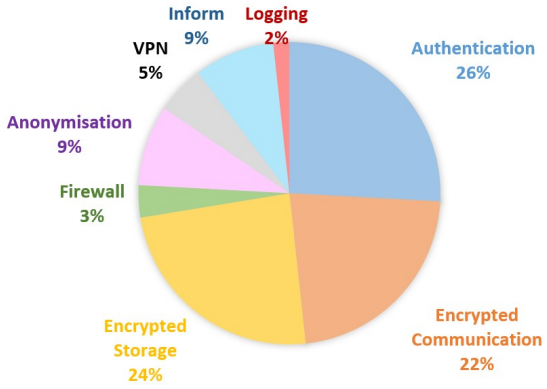

Fig. 12. Common privacy protecting measures suggested by participants in Study 2 (Round 1 - Before seeing the PbD Guidelines)

6.2.18 ***Over thinking and applications could lead to unnecessary complexities.*** We noticed that sometimes, using guidelines could be tricky and engineers may apply them in an over-thinking manner. Privacy guidelines are designed to guide the thinking process. They are not mandatory steps that someone should to follow blindly. Effectiveness need to be thought through before using them. For example, participant [N5] mentioned that *'Distributed data processing, I did not think about this before reading the guidelines. For scenarios two and three, we can distribute the data for processing. We send them to different Clouds, first of all with scenario three, like for attraction, like, we send the data for each attraction to different Cloud servers'*. Even though distributed processing is applicable in the scenario, it is not really a effective approach for scenario 2. Attempting to employ multi cloud processing as a way to apply distributed processing in scenario 2 could lead to unnecessary complexity and higher costs with little contribution to privacy protection. Therefore, it is important to assess each context carefully before applying a particular guideline.

6.3 Limitations

Although all the participants were able to understand our proposed guidelines, it was apparent that familiarity is key to applying them in a given IoT application design in a short period of time. For our study, we printed the PbD guidelines on plain A4 sheets as a list. However, the experience of our study participants highlighted that this type of printed list is difficult to follow and can be more time consuming to use. We believe that approaches such as *Privacy Ideation Cards* [28] and KnowCards[6] would be more effective by allowing users to quickly familiarise themselves with the guidelines. In particular, using a colour coded, iconographic approach to represent the guidelines could improve the users recall them and thus lead to faster application of guidelines with less frustration.

An additional limitation of this work is that we did not consider the adaptive nature of privacy. While some decisions about implementing privacy preserving measures can be taken at design-time, IoT applications are by nature unpredictable. As a result, the ability to adapt is an important feature in IoT applications. Ideally, IoT

[6]know-cards.myshopify.com

applications should be able to compose built-in privacy preserving techniques into a run-time configuration, that maximises the privacy protection level while maintaining the overall utility of the application.

7 RELATED WORK

Our objective is to explore ways in which we can help software engineers to efficiently and effectively design privacy aware IoT applications. Towards this, in this paper, we developed a privacy guideline based methodology. There are a number of existing frameworks that have been proposed to help elicit privacy requirements and to design privacy capabilities into systems. Privacy principles, privacy strategies, privacy patterns are been developed to support software engineering processes. It is important to note that non of these approaches explicitly focus on IoT domain or IoT application development processes.

Spiekermann [47] has identified number of challenges in privacy by design approach. Spiekermann identified PbD as *"an engineering and strategic management approach that commits to selectively and sustainably minimize information systems' privacy risks through technical and governance controls."*. Privacy is a vague concept without a rigid definition. Therefore, at times, it is difficult to measure the effectiveness or efficiency of privacy protection techniques. Further, distinguishing privacy from security is vital in order to develop methodologies to address privacy challenges. Spiekermann [47] also highlight the problem of not having widely agreed methodology for systematic engineering of privacy into systems. This justifies our attempt to develop a methodology to incorporate privacy protecting measures into IoT application designs.

Primarily, there are two approaches to incorporate privacy measures into a system design. First approach (let us call *'Threat-based'*) explicitly examines a given system design to identify privacy threats and address them. LINDDUN [13], which we discuss later in this section, can be considered as an example. Privacy Impact Assessment (PIA) [51] is also an example for this approach. Second approach (let us call *'Blanket-based'*) suggests to apply series of privacy protecting measures into a given design without explicitly considering specific privacy threats. The expectation it to apply set of blanket measures aiming to improve the overall privacy awareness of the design. Our proposed methodology follows this approach. Some other examples are privacy principles [10], and privacy strategies [20]. Both *'Threat based'* and *'Blanket-based'* approach have their own strengths and weaknesses. Due to unique characteristics of each approach, an hybrid approach may potentially create better system designs.

Threat-based This approach eliminates specific threats that a system might have. Therefore, it is a mission oriented approach where it force system designers to deeply think about specific threats. On the down side, systems may struggled to handle threats that the designers haven't though about during design time. Deep thinking process would take longer time and complexities could lead to poor threat analysis.

Blanket-based This approach is some what simpler and less error prone due to the absence of a threat analysis process. However, the same reason could lead to weak privacy design due to not handling specific threats unique to a given system. On the other hand, this approach has more chance to handle unprepared privacy risks at run time due to less dependence on threat identification processes. Therefore, highly dynamic systems may benefit from this approach.

Principles, Strategies, and Guidelines: The original PbD is a framework proposed by Ann Cavoukian [10], the former Information and Privacy Commissioner of Ontario, Canada. This framework identifies seven foundation principles by which privacy sensitive application should be developed. These are: (1) proactive not reactive; preventative not remedial, (2) privacy as the default setting, (3) privacy embedded into design, (4) full functionality positive-sum, not zero-sum, (5) end-to-end security-full life-cycle protection, (6) visibility and transparency- keep it open, and (7) respect for user privacy, keep it user-centric. Cavoukian and Jonas [12] has extended Cavoukian's [10] seven privacy principle by proposing seven more specific guidelines to build PbD systems to manage big data, namely, (1) full attribution, (2) data tethering, (3) analytics on anonymized

Table 4. Summary of PbD Evaluation Methodologies

Area	Descriptions of Evaluation the approach
Garde-Perik [50]	This work explores relative importance of complying with privacy related guidelines in the context of a Health Monitoring System. A total of 50 participants were given a text scenario describing a health care system. This system does not adhere to any of the OECD guidelines. Participants were then provided with potential âĂŸfixes' to the system, each of which would make it comply with one specific OECD guideline. The guidelines were presented in pair where participants needed to pick which guidelines is most important.
Iachello et al. [22]	This work had developed a mobile application to conduct user studies in order to extract privacy guidelines. Those guidelines are then used to develop a second mobile application to evaluate and critique the proposed guidelines. Specific guidelines are presented later in this section.
Bellotti and Sellen [4]	This work has proposed a framework for design for privacy in ubiquitous computing environments. They have proposed eleven criteria to evaluate a given design as presented later in this section. They take each criteria and evaluate it against their sample design.
LINDDUN [13]	LINDDUN is a threat modelling technique that supports the elicitation of privacy threats during the early stages of the software development life-cycle. Three groups have been involved in the evaluation process (total of 8 individuals) where they were asked to create a DFD diagram for a given high level scenario description (two groups focused on a e-health system and one group focused on a smart grid system) and use it to elicit the privacy threats using the LINDDUN framework. Group discussions were used to gather the participants experience. They analysed both the results the participants documented in their reports (discovered threats), as well as the opinions of the participants with regard to their hands-on experience. • *Correctness:* On average, how many threats uncovered by the participants are correct (true positives vs false positives)? • *Completeness:* How many threats are undetected by the participants (false negatives)? • *Productivity:* How many valid threats are identified by the participants in a given time frame? • *Ease of use:* Did the participants perceive the methodology as easy to learn and apply? In order to explore any flows in the LINDDUN method, they have asked a panel of three privacy experts to perform an independent threat analysis of a smart grid system using their own expertise. They have measured the reliability by comparing expert designs with earlier designs. • *Reliability:* Is LINDDUN missing any important threats?
Rubinstein and Good [44]	Based on a review of the technical literature, this work has derived a small number of relevant principles and illustrates them by reference to ten recent privacy incidents involving Google and Facebook.

data, (4) tamper-resistant audit logs, (5) false negative favouring methods, (6) self-correcting false positives and (7) information transfer accounting. The ISO 29100 Privacy framework [24] has proposed eleven design principles, namely, (1) consent and choice, (2) purpose legitimacy and specification, (3) collection limitation, (4)

data minimisation, (5) use, retention and disclosure limitation, (6) accuracy and quality, (7) openness, transparency and notice, (8) individual participation and access, (9) accountability, (10) information security, and (11) privacy compliance. Wright and Raab [53] has proposed to extend these ISO guidelines by adding 9 more guidelines, namely, (12) right to dignity, i.e., freedom from infringements upon the person or her reputation, (13) right to be let alone (privacy of the home, etc.), (14) right to anonymity,including the right to express one's views anonymously, (15) right to autonomy, to freedom of thought and action, without being surveilled, (16) right to individuality and uniqueness of identity, (17) right to assemble or associate with others without being surveilled, (18) right to confidentiality and secrecy of communications, (19) right to travel (in physical or cyber space) without being tracked, and (20) people should not have to pay in order to exercise their rights of privacy (subject to any justifiable exceptions), nor be denied goods or services or offered them on a less preferential basis.

The Fair Information Practice Principles (FIPPs) [8] proposed by the United States Federal Trade Commission is also formulated as set of guidelines, namely, (1) notice / awareness, (2) choice / consent, (3) access / participation, (4) integrity / security, and (5) enforcement / redress. Organisation for Economic Cooperation and Development (OECD) [33, 52] has also proposed similar privacy guidelines, namely, (1) notice, (2) purpose, (3) consent, (4) security, (5) disclosure, (6) access, and (7) accountability. Historically, OECD guidelines are considered as a successful milestone [52] where it laid the foundation for both subsequent Data Protection Directive (95/46/EC) and General Data Protection Regulation (GDPR) [14]. Rost and Bock [43] have identified six data protection goals, namely, (1) availability, (2) integrity, (3) confidentiality, (4) transparency, (5) unlinkability, and (6) ability to intervene. Fisk et al. [15] have proposed three privacy principles, namely, (1) least disclosure [internal disclosure, privacy balance, inquiry-specific release], (2) qualitative evaluation [legal constraints, technical limitations], and (3) forward progress.

Building on the ideas of engineering privacy by architecture vs. privacy-by-policy presented by Spiekerman and Cranor [48], Hoepman [20] proposes an approach that identifies eight specific privacy design strategies: minimise, hide, separate, aggregate, inform, control, enforce, and demonstrate. This is in contrast to other approaches that we considered. In a similar vein, Singh et al. [46] has proposed 20 security consideration (somewhat similar to guidelines) for IoT, namely, (1) secure communications, (2) access controls for iot-cloud, (3) identifying sensitive data, (4) cloud architectures: public, private, or hybrid?, (5) in-cloud data protection, (6) in-cloud data sharing, (7) encryption by *'things'*, (8) data combination, (9) identifying *'things'*, (10) identifying the provider, (11) increase in load, (12) logging at large scale, (13) malicious 'things'-protection of provider, (14) malicious 'things'-protection of others, (15) certification of cloud service providers, (16) trustworthiness of cloud services, (17) demonstrating compliance using audit, (18) responsibility for composite services, (19) compliance with data location regulations, and (20) impact of cloud decentralization on security.

Frameworks: LINDDUN [13] is a privacy threat analysis framework that uses data flow diagrams (DFD) to identify privacy threats. LINDDUN focuses on eliminating set of pre-identified privacy threats using a systematic review of data flow diagrams. It consists of six specific methodological steps: (1) define the DFD, (2) map privacy threats to DFD elements, (3) identify threat scenarios, (4) prioritize threats, (5) elicit mitigation strategies, and (6) select corresponding privacy enhancing technologies. However, both LINDDUN and Hoepman's framework are not aimed at the IoT domain. Further, they not prescriptive enough in guiding software engineers. Bellotti and Sellen [4] have proposed a framework for design for privacy in ubiquitous computing environments. They argue that systems must be explicitly designed to provide feedback and control about (1) capture [when and what information collected], (2) construction [what happens to information], (3) accessibility [which people and what software have access to information], and (4) purposes [why data is being collected]. They also propose eleven criteria to evaluate a given design, namely, (1) trustworthiness, (2) appropriate timing, (3) perceptibility, (4) unobtrusiveness, (5) minimal intrusiveness, (6) fail-safety, (7) flexibility, (8) low effort, (9) meaningfulness, (10) learnability, (11) low cost. In contrast, the STRIDE [21] framework was developed to help software engineers consider security threats, is an example framework that has been successfully used to build secure software

systems by industry. It suggests six different threat categories: (1) spoofing of user identity, (2) tampering, (3) repudiation, (4) information disclosure (privacy breach or data leak), (5) denial of service, and (6) elevation of privilege. However, its focus is mostly on security than privacy concerns.

Patterns and Anti-Patterns: Both patterns and anti-patterns are important and relevant to our work. However, due to space limitation, we do not review pattern literature in detail. Some important information on privacy pattern can be found in [1, 2].

Design Aids: In similar direction, Luger et al. [28] aims to understand how to make emerging European data protection regulations more accessible to general public using a series of privacy ideation cards. They have extracted 40 design principles by examining EU General Data Protection Regulation 2012 Com Final 11 [14]. These high level principles are proposed for computer systems in general but not prescriptive enough to be adopted by IT professionals for designing and developing IoT applications. In addition to using descriptions to explain guidelines, Zevenbergen et al. [54] have produced set of questions to explicitly guide the designers' mind towards following the guidelines. Inspired by their approach, we also formulated our proposed guidelines in question based format [35].

Domain Specific: Privacy guidelines can also be domain focused or contextual as well. Iachello et al. [22] has proposed privacy guidelines for social location disclosure applications and services. Their proposed guidelines (quite specific) are (1) don't start with automation, (2) flexible replies, (3) support denial, (4) support deception, (5) support simple evasion, (6) start with person-to-person communication, (7) status/away messages, (8) operators: avoid handling user data, (9) power relationships, (10) user characterization, (11) account for long learning curve, and (12) account for specific circumstances. Gritzalis et al. [18] has proposed 36 guidelines, formulated as counter measure, to address common privacy risks in healthcare domain. guidelines are extracted through a use case analysis and a risk assessment. Langheinrich [25] has develops six principles for guiding system design, based on a set of fair information practices common in most privacy legislation in use today: notice, choice and consent, proximity and locality, anonymity and pseudonymity, security, and access and recourse. Langheinrich discusses these generic principle in the context of ubiquitous computing in detail. It is important to note that, due to abstract nature, privacy principles can be interpreted in different ways related to different contexts. Therefore, both privacy principles as well as different interpretations are both important. Cavoukian [9] has proposed several privacy guidelines to serve as privacy *'best practices'* guidance for organizations when designing and operating Radio-Frequency Identification (RFID) information technologies and systems. Proposed guidelines are (1) accountability, (2) identifying purposes, (3) consent, (4) limiting collection, (5) limiting use, disclosure and retention, (6) accuracy, (7) safeguards, (8) openness, (9) individual access, and (10) challenging compliance. Zevenbergen et al. [54] has proposed specific set of guidelines to measure mobile connectivity in a ethical way. The aim of their guidelines is to help network researchers navigate the challenges of preserving the privacy of data subjects, publishing and disseminating datasets, while adhering to and advancing good scientific practice.

Cavoukian [11] argues the important of empowering software engineers to develop and adopt privacy best practices. We believe that providing methodologies, tools, and techniques is part of the empowerment process.

7.1 Privacy Guidelines in GDPR Context

General Data Protection Regulation (GDPR) is a regulation made by *European Parliament and Council* which aims to regulate how personal data of EU citizens should be handled by any entities within or outside EU. GDPR aims primarily to give control back to citizens and residents over their personal data. This regulation is expected to be implemented in May 2018. Even though our PbD framework is not design specifically to address GDPR, we would like to briefly highlight our framework in the context GDPR.

Part of the GDPR regulation is organised as principles which are quite similar to the principles we discussed in this paper. Example principle is listed below.

- *"adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed ('data minimisation');"*

Our privacy guidelines (especially the ones that target minimization) will help to implement this principle. It would be useful to develop more concrete guidelines, patterns and tactics to address each of the principles proposed in GDPR.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we explored how a Privacy-by-Design (PbD) framework, formulated as a set of guidelines, can help software engineers to design privacy-aware IoT applications. We evaluated the effectiveness of the proposed PbD framework through a use cases based observational study where the participants were asked to design IoT applications to satisfy three given use cases. However, it is important to note that our objective is not to position our proposed framework as the de-facto PbD framework. Instead, our objective is to show that a set of guidelines can significantly assist the software engineers to design better privacy aware IoT applications. According to our findings, the proposed PbD guidelines framework has significantly improved the privacy awareness of the IoT applications designed by both novice and expert users. Further, results show that software engineering expertise does not matter significantly when it comes to incorporating privacy protection features into IoT application designs.

In the future, we will conduct research to develop a set of privacy tactics and patterns that are less abstract than guidelines. Such tactics and patterns will help software engineers to tackle specific privacy design challenges in IoT domain. At the moment, privacy guidelines are presented to the software engineers in plain text organised into a list. Though it is usable, in the future, we will explore how we can use human computer interaction (HCI) techniques to make these PbD guidelines more user friendly and accessible to the software engineers. HCI techniques will help software engineers to efficiently and effectively browse and find relevant privacy guidelines, patterns and tactics in a given IoT application design context.

In long term, we aim to change the way that the engineering community looks at privacy challenges. Privacy challenges are often considered to be time consuming and difficult to address and require significant expertise. Therefore, ideally, we need to develop new techniques that automatically address privacy challenges in the IoT application design process while letting engineers focus solely on other design challenges (e.g. interoperability, efficiency). More specifically, such automated tools and techniques will not only transform application designs into privacy aware application designs, but also validate and verify them. Such automated tools and techniques will save millions of engineer hours which engineers would otherwise need to spend developing privacy expertise (also called privacy mindset) and applying it. Privacy guidelines (similar to ours), patterns, and, tactics will formulate the underlying knowledge base that is required by automated tools and techniques.

ACKNOWLEDGMENTS

We acknowledge the financial support of European Research Council Advanced Grant 291652 (ASAP) and the funding from EPSRC award number DERC EP/M023001/1 (Digital Economy Research Centre).

REFERENCES

[1] [n. d.]. Privacy Patterns. ([n. d.]). https://privacypatterns.org/

[2] [n. d.]. privacypatterns.eu - collecting patterns for better privacy. ([n. d.]). https://privacypatterns.eu

[3] Len Bass, Paul Clements, and Rick Kazman. 2012. *Software Architecture in Practice* (3 edition ed.). Addison-Wesley Professional, Upper Saddle River, NJ.

[4] Victoria Bellotti and Abigail Sellen. 1993. Design for Privacy in Ubiquitous Computing Environments. In *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13-17 September 1993, Milan, Italy ECSCW '93*. 77–92. https://doi.org/10.1007/978-94-011-2094-4_6 arXiv:arXiv:1011.1669v3

[5] David Budgen. 2003. *Software design.* Addison-Wesley. 468 pages.

[6] F Bushmann, Regine Meunier, and Hans Rohnert. 1996. Pattern-oriented software architecture: A system of patterns. *John Wiley&Sons* 1 (1996), 476. https://doi.org/10.1192/bjp.108.452.101

[7] Jennie M Carroll and P A Swatman. 2000. Structured-case: a methodological framework for building theory in information systems research. *European Journal of Information Systems* 9, 4 (2000), 235–242. https://doi.org/10.1057/palgrave/ejis/3000374

[8] Fred H Cate. 2006. The Failure of Fair Information Practice Principles. In *Consumer Protection in the Age of the 'Information Economy'*. 341–377.

[9] Ann Cavoukian. 2006. *Privacy Guidelines for RFID Information Systems.* Technical Report. Information and Privacy Commissioner of Ontario, Ontario. 1–10 pages.

[10] Ann Cavoukian. 2009. *Privacy by Design The 7 Foundational Principles Implementation and Mapping of Fair Information Practices.* Technical Report. https://www.iab.org/wp-content/IAB-uploads/2011/03/fred

[11] Ann Cavoukian. 2012. Operationalizing Privacy by Design. *Commun. ACM* 55, 9 (2012), 7—-7. http://doi.acm.org/10.1145/2330667.2330669

[12] Ann Cavoukian and Jeff Jonas. 2012. *Privacy by Design in the Age of Big Data.* Technical Report. Information and Privacy Commissioner, Ontario, Canada. 1–17 pages.

[13] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. 2011. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering* 16, 1 (2011), 3–32.

[14] European Commission. 2016. General Data Protection Regulation (GDPR). *Official Journal of the European Union* (2016).

[15] Gina Fisk, Calvin Ardi, Neale Pickett, John Heidemann, Mike Fisk, and Christos Papadopoulos. 2015. Privacy principles for sharing cyber security data. In *Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015.* 193–197. https://doi.org/10.1109/SPW.2015.23

[16] Carl S French. 1996. *Data Processing and Information Technology.* Cengage Learning Business Press.

[17] Atul. Gawande. 2011. *The checklist manifesto : how to get things right.* 215 pages.

[18] Stefanos Gritzalis, Costas Lambrinoudakis, Dimitrios Lekkas, and Spyros Deftereos. 2005. Technical guidelines for enhancing privacy and data protection in modern electronic medical environments. *IEEE Transactions on Information Technology in Biomedicine* 9, 3 (2005), 413–423. https://doi.org/10.1109/TITB.2005.847498

[19] Alex B. Haynes, Thomas G. Weiser, William R. Berry, Stuart R. Lipsitz, Abdel-Hadi S. Breizat, E. Patchen Dellinger, Teodoro Herbosa, Sudhir Joseph, Pascience L. Kibatala, Marie Carmela M. Lapitan, Alan F. Merry, Krishna Moorthy, Richard K. Reznick, Bryce Taylor, and Atul A. Gawande. 2009. A Surgical Safety Checklist to Reduce Morbidity and Mortality in a Global Population. *New England Journal of Medicine* 360, 5 (jan 2009), 491–499. https://doi.org/10.1056/NEJMsa0810119

[20] Jaap-Henk Hoepman. 2014. Privacy Design Strategies. In *ICT Systems Security and Privacy Protection*, Nora Cuppens-Boulahia, Frédéric Cuppens, Sushil Jajodia, Anas Abou El Kalam, and Thierry Sans (Eds.). IFIP Advances in Information and Communication Technology, Vol. 428. Springer Berlin Heidelberg, 446–459.

[21] Michael Howard and Steve Lipner. 2006. *The security development lifecycle: SDL, a process for developing demonstrably more secure software.* Microsoft Press.

[22] Giovanni Iachello, Ian Smith, Sunny Consolvo, Mike Chen, and Gregory D. Abowd. 2005. Developing privacy guidelines for social location disclosure applications and services. In *Proceedings of the 2005 symposium on Usable privacy and security - SOUPS '05.* 65–76. https://doi.org/10.1145/1073001.1073008

[23] Information Commissioner's Office. [n. d.]. *The Guide to Data Protection.* Technical Report. 1–131 pages. https://ico.org.uk/media/for-organisations/guide-to-data-protection-2-4.pdf

[24] ISO/IEC 29100. 2011. *Information technology Security techniques Privacy framework.* Technical Report.

[25] Marc Langheinrich. 2001. Privacy by design-principles of privacy-aware ubiquitous systems. *Ubicomp 2001: Ubiquitous Computing* (2001), 273–291. https://doi.org/10.1007/3-540-45427-6_23 arXiv:9780201398298

[26] Michael Lewis-Beck, Alan Bryman, and Tim Futing Liao. 2004. *The SAGE Encyclopedia of Social Science Research Methods.* Sage Publications, Inc., 2455 Teller Road, Thousand Oaks California 91320 United States of America. https://doi.org/10.4135/9781412950589

[27] William Lowrance. 2003. Learning from experience: privacy and the secondary use of data in health research. *Journal of Health Services Research & Policy* 8, suppl 1 (2003), 2–7.

[28] Ewa Luger, Lachlan Urquhart, Tom Rodden, and Michael Golembewski. 2015. Playing the Legal Card: Using Ideation Cards to Raise Data Protection Issues within the Design Process. *Proceedings of the ACM CHI'15 Conference on Human Factors in Computing Systems* 1 (2015), 457–466.

[29] Matthew B. Miles, A. M. Huberman, and Johnny. Saldaña. 2013. *Qualitative data analysis : a methods sourcebook.* 381 pages.

[30] Minister of Justice. 2015. *Personal Information Protection and Electronic Documents Act.* Technical Report. http://laws-lois.justice.gc.ca/eng/acts/P-8.6/FullText.html

[31] Rolf Molich and Jakob Nielsen. 1990. Improving a human-computer dialogue. *Commun. ACM* 33, 3 (mar 1990), 338–348. https://doi.org/10.1145/77481.77486

[32] Jakob Nielsen and Rolf Molich. 1990. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90.* ACM Press, New York, New York, USA, 249–256. https://doi.org/10.1145/97243.97281

[33] Danie E. Oleary. 1995. Some Privacy Issues in Knowledge Discovery: The OECD Personal Privacy Guidelines. *IEEE Expert-Intelligent Systems and their Applications* 10, 2 (1995), 48–59. https://doi.org/10.1109/64.395352

[34] Judea Pearl. 1984. *Heuristics : intelligent search strategies for computer problem solving*. Addison-Wesley Pub. Co. 382 pages. https://dl.acm.org/citation.cfm?id=525

[35] Charith Perera. 2017. *Privacy Guidelines for Internet of Things: A Cheat Sheet*. Technical Report. arXiv:1708.05261 http://arxiv.org/abs/1708.05261

[36] Charith Perera, Chi Harold Liu, and Srimal Jayawardena. 2014. A Survey on Internet of Things From Industrial Market Perspective. *IEEE Access* 2 (2014), 1660–1679. https://doi.org/10.1109/ACCESS.2015.2389854

[37] Charith Perera, Chi Harold Liu, and Srimal Jayawardena. 2015. The Emerging Internet of Things Marketplace from an Industrial Perspective: A Survey. *IEEE Transactions on Emerging Topics in Computing* 3, 4 (2015), 585–598.

[38] C. Perera, C. McCormick, A.K. Bandara, B.A. Price, and B. Nuseibeh. 2016. Privacy-by-design framework for assessing internet of things applications and platforms. In *ACM International Conference Proceeding Series*, Vol. 07-09-Nove. https://doi.org/10.1145/2991561.2991566

[39] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. Context Aware Computing for The Internet of Things: A Survey. *Communications Surveys Tutorials, IEEE* 16, 1 (2014), 414 – 454.

[40] Dorian Pyle. 1999. *Data preparation for data mining*. Morgan Kaufmann Publishers, San Francisco, Calif.

[41] Lyn Richards. 2014. *Handling qualitative data : a practical guide*. 236 pages.

[42] Rodrigo Roman, Jianying Zhou, and Javier Lopez. 2013. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks* 57, 10 (2013), 2266–2279.

[43] Martin Rost and Kirsten Bock. 2011. Privacy by Design and the New Protection Goals. *DuD, January* November 2009 (2011), 1–9. https://www.european-privacy-seal.eu/AppFile/GetFile/ca6cdc46-d4dd-477d-9172-48ed5f54a99c

[44] Ira S. Rubinstein and Nathaniel Good. 2013. Privacy by Design: A Counterfactual Analysis of Google and Facebook Privacy Incidents. *Berkeley Technology Law Journal* 28, 2 (2013), 1333–1413. https://doi.org/10.2139/ssrn.2128146 arXiv:arXiv:1011.1669v3

[45] Helen Sharp, Yvonne Rogers, and Jenny Preece. 2015. *Interaction Design: Beyond Human-Computer Interaction*. 584 pages. https://doi.org/10.1162/leon.2005.38.5.401

[46] Jatinder Singh, Thomas Pasquier, Jean Bacon, Hajoon Ko, and David Eyers. 2016. Twenty Security Considerations for Cloud-Supported Internet of Things. *IEEE Internet of Things Journal* 3, 3 (2016), 269–284. https://doi.org/10.1109/JIOT.2015.2460333 arXiv:1207.0203

[47] Sarah Spiekermann. 2012. The challenges of privacy by design. *Commun. ACM* 55, 7 (2012), 38. https://doi.org/10.1145/2209249.2209263

[48] S. Spiekermann and L.F. Cranor. 2009. Engineering Privacy. *IEEE Transactions on Software Engineering* 35, 1 (jan 2009), 67–82.

[49] Abbas. Tashakkori and Charles. Teddlie. 2010. *Sage handbook of mixed methods in social and behavioral research*. SAGE Publications. 893 pages.

[50] Evelien van de Garde-Perik, Panos Markopoulos, and Boris de Ruyter. 2006. On the relative importance of privacy guidelines for ambient health care. In *Proceedings of the 4th Nordic conference on Human-computer interaction changing roles - NordiCHI '06*. 377–380. https://doi.org/10.1145/1182475.1182516

[51] David Wright and Paul De Hert. 2012. *Privacy impact assessment*. 1–523 pages. https://doi.org/10.1007/978-94-007-2543-0

[52] David Wright, Paul De Hert, and Serge Gutwirth. 2011. Are the OECD guidelines at 30 showing their age? *Commun. ACM* 54, 2 (2011), 119. https://doi.org/10.1145/1897816.1897848

[53] David Wright and Charles Raab. 2014. Privacy principles, risks and harms. *International Review of Law, Computers and Technology* 28, 3 (2014), 277–298. https://doi.org/10.1080/13600869.2014.913874

[54] Bendert Zevenbergen, Ian Brown, Joss Wright, and David Erdos. 2013. *Ethical Privacy Guidelines for Mobile Connectivity Measurements*. Technical Report. Oxford Internet Institute, University of Oxford, Oxford. 1–41 pages.