P. Kühtreiber, V. Pak, and D. Reinhardt. Replication: A Survey on Solutions to Support Developers in Privacy-Preserving IoT Development. Pervasive and Mobile Computing (PMC), 2022

A Survey on Solutions to Support Developers in Privacy-Preserving IoT Development

Patrick Kühtreiber[1,a], Viktoriya Pak[a], Delphine Reinhardt[a]

[a] *Computer Security and Privacy, Georg-August-Universität Göttingen, Göttingen, Germany*

Abstract

Internet-of-Things (IoT) devices are rising in popularity and their usefulness often stems from the amount of data they collect. Data regulations such as the European *General Data Protection Regulation (GDPR)* require software developers to do their due diligence when it comes to privacy, as they are required to adhere to certain principles such as *Privacy-by-Design (PbD)*. Due to the distributed and heterogeneous nature of IoT applications, privacy-preserving design is even more important in IoT environments. Studies have shown that developers are often not eager to implement privacy and generally do not see it as their duty or concern. However, developers are often left alone when it comes to engineering privacy in the realm of IoT. In this paper, we therefore survey which frameworks and tools have been developed for them, especially in the case of IoT. Our findings indicate that existing solutions are cumbersome to use, only work in certain scenarios, and are not enough to solve the privacy issues inherent in IoT development. Based on our analysis, we further propose future research directions.

Keywords: IoT Developer, Privacy Frameworks, Privacy-by-Design, IoT

[1] Goldschmidtstr. 7, 37077 Göttingen, Germany, Phone: +49 551 39-172028, Fax: +49 551 39-14403, E-Mail: kuehtreiber@cs.uni-goettingen.de

1. Introduction

IoT devices collect users' data almost constantly [1]. This makes privacy in the scope of IoT a key research topic [2]. However, this goal can only be reached when developers have both the ability and the awareness to provide privacy-preserving implementations of their IoT devices. Research has shown that it is possible to infer sensitive information from data collected from IoT devices. We are especially concerned about *Personally Identifiable Information (PII)*, i.e., any kind of data that can be used to identify a data subject and is thus considered sensitive.

For example, Conti et al. showed that it is possible to infer the user of a certain laptop by just monitoring its energy consumption [3]. Copos et al. further inferred whether people are at home, using smoke and carbon dioxide detectors [4]. Especially critical are findings from Hutton et al [5], which show that health data from self-tracking apps, such as calorie counters or pedometers, are also not secure in terms of privacy. In the context of smart homes, privacy is endangered by the interaction of poorly configured IoT devices [6]. Wearable devices and IoT devices with voice interfaces [7] can further reveal sensitive insights about the health of the users. In addition to the collection and inferences of sensitive data, it has been further shown that the security of IoT devices is often not guaranteed [8].

These examples hence show that providing privacy protection in the context of IoT is crucial, but still difficult to apply in practice. Different factors, such as lack of control and data inferences, can explain such difficulties [9]. In this paper, we focus on the human component. Contrary to approaches that focus mainly on the safe and secure operation of IoT environments, such

as, e.g., IoTSAN [10], IoTGuard [11], and Soteria [12], there exist relatively few methods that explicitly focus on the privacy aspect. Privacy threat modeling approaches, e.g., LINDDUN [13], have been developed. However, their main goal is not to guide IoT developers in privacy preserving development [14]. More formal methods to prove compliance to privacy laws have also been presented [15], but these methods are not user friendly enough.

Perera et al. identified five stakeholders that are responsible for data privacy in IoT [16]: device manufacturers, cloud providers, third-party developers, government, and consumers. In this survey, we focus on device manufacturers and third-party developers, as those groups face many obstacles when trying to adhere to privacy regulations [17]. We review existing tools, guidelines, and frameworks that aim at helping IoT developers to adhere to *Privacy-by-Design (PbD)* for protecting users' privacy. Our goal is to highlight possibilities for privacy-preserving IoT development by presenting general strategies in this domain. The selection of presented papers follows this goal as we only focus on general approaches for IoT developers. To the best of our knowledge no survey of this kind exists so far. Our contributions are as follows: (1) An overview over PbD, its implication for IoT development and associated problems, (2) analysis and comparison of existing solutions that are designed to help IoT developers in programming privacy-preserving IoT applications, (3) evaluation of their compliance with privacy principles and, thus, privacy law, and (4) identification and discussion of possible future research directions.

This paper is structured as follows. We provide background information in Sec. 2 and describe our methodology in Sec. 3. We compare and discuss existing privacy methods for IoT developers in Sec. 4 and Sec. 5, respectively. In Sec. 6, we suggest future research directions and conclude in Sec. 7.

2. Foundations

The goal of data protection is to prevent entities from misusing personal data [18]. To protect these data, different principles and frameworks have been proposed (Sec. 2.1) and put into practice (Sec. 2.2). However, IoT developers face challenges when implementing adequate solutions (Sec. 2.3) and applying PbD (Sec. 2.4).

2.1. Privacy principles

Privacy by Design. The idea of PbD introduced by Ann Cavoukian [19] is that the concept of privacy should be embedded within the whole process of software engineering and not be an afterthought. The risks resulting from ignoring privacy during the design stage of a software product are manifold including the lack of privacy controls in the final product and the additional amount of time and money spent to make it compliant to privacy regulations *after* the functional features of the application are already finished.

Privacy design principles have also been proposed by Hoepman [20]: Minimize, hide, separate, aggregate, inform, control, enforce, and demonstrate. Hoepman's principles are seen as most suitable for privacy preserving IoT development [21, 22], which is why we concentrate more on them in the remainder of this paper rather than Cavoukian's.

Minimization, which can be achieved via anonymization, pseudonymization, and privacy-preserving design choices, means that not more than the necessary amount of PII should be processed. *Hiding* data aims at, e.g., unlinkability of data (see Sec. 2.2), and can also be reached via anonymization and pseudonymization, as well as through encryption and mix networks. *Separation* deals with the problem of linking different PII data sources, which should be prevented by, e.g., storing and processing data

locally. *Aggregation* means that PII should be aggregated wherever possible in such a way that the data remains useful. It can be achieved via, e.g., k-anonymity [23]. The principle to *inform* calls for providing transparency of the data processing to the data subject, including data protection measures. These data subjects should then be able to *control* their data according to the information provided to them. The principle to *enforce* targets privacy policies, which should follow legal requirements and be enforced through technical and organizational measures. *Demonstrate* finally demands the data controllers to show that they are compliant with the aforementioned privacy policy. This can be achieved via, e.g., logs. These principles all address the planning and developing of IT systems, thus, need to be applied by the developers themselves. PbD has since become one of the pillars of privacy regulations, such as the GDPR.

GDPR. Some PbD principles can be found in Art. 25 of the GDPR, which states that data controllers should implement data protection measures, such as data minimization and pseudonymization in order to protect the data subjects' data (GDPR, Article 25). It is also required that "appropriate technical and organizational measures be taken" (GDPR, Recital 78). The recital mentions data minimization, pseudonymization, and transparency. It is also mentioned that the data controller should be able to "create and improve security features" (GDPR, Recital 78). Both these principles, as well as the GDPR, however neither enforce nor recommend any particular *Privacy Enhancing Technologies (PETs)*, whose choice is left to the developers, aiming at implementing these principles. The concrete implementation of those principles is left to developers, who sometimes simply do not know them (see Sec. 2.4), and even if they do, they are often not enabled to

Protection Goal	GDPR Article(s)	Hoepman's Principles
Unlinkability	5	Minimize, Hide, Separate, Aggregate
Intervenability	5, 12, 17, 18, 20, 22, 25, 33, 34	Control
Transparency	5, 24, 28, 30, 33, 35, 58	Inform, Demonstrate

Table 1: Classification of protection goals, GDPR articles, and Hoepman's principles

incorporate them into the design of new systems such as IoT devices.

2.2. From principles to practice

Protection goals. Following the privacy principles laid out in Sec. 2.1 arise certain protection goals. Next to the established security protection goals *confidentiality*, *integrity*, and *availability* there exist the three additional privacy goals: *Unlinkability*, defined in such a way that a data subject's PII cannot be linked to any other privacy relevant data, *intervenability*, defined as the possibility for the data subjects to interfere in the processing of their private data, and *transparency*, the property that any processing of personal data should be comprehensible and verifiable [18]. The German independent center for data protection ULD classified these privacy protection goals according to their respective GDPR articles [24]. In Tab. 1 we expand this mapping with Hoepman's privacy principles, which we have introduced in Sec. 2.1.

Privacy framework. The standard ISO 29100 defines a privacy framework for developers [25] which is intended to work on a very high level, independent of, e.g., programming language, platforms, or the size of the development team [26]. However, as for the GDPR, the support provided to the developers when applying the framework remains limited to general principles. Also, it does not focus in IoT developers or architectures.

Privacy engineering. The privacy principles (Sec. 2.1) are formulated at a very high-level, thus limiting the assistance provided to the developers. Privacy engineering translates them into a "developer friendly" language [27].

Privacy Patterns. To translate privacy principles to privacy engineering practices, so-called "privacy patterns" have been established [28]. These patterns follow the idea of software design patterns, i.e., to reuse established practices in software engineering. Their goal is to provide developers with standardized solutions to common privacy related problems [29].

2.3. Privacy challenges unique to IoT

In this section, we highlight certain risks that are connected to IoT devices and which are different to classical software development.

Amount of data. For example, a smart home consists of hundreds of sensors, which provide data constantly to the data controller [30]. The principle to minimize data is therefore especially challenging in this regard. Also, IoT devices are often interconnected. Thus, several data sources can be combined, which violates the principle to separate the data.

Resource constraints. The "things" of the IoT, e.g., smart light bulbs, smart locks, etc., have scarce resources such as energy, memory, and computing power, which makes complex encryption harder to implement [31]. This does not only impact security goals, but also affects the users' privacy, as a data leak compromises all the data, if they are not encrypted.

Heterogeneous nature of IoT. Due to the fast growth of IoT over the last couple of years, the IoT platforms and environments are very diverse. This lack of standardization not only decelerates future growth [32, 33] but also hinders the implementation of unified privacy protocols.

2.4. Developers' privacy perceptions

Several studies have been conducted and have highlighted the following issues when considering privacy and developers.

Lack of responsibility. Research in the context of privacy and software development have shown that developers do not see privacy as a problem they have to solve [34, 35, 36]. Privacy should hence be a joint goal of management and IT, as privacy still is an abstract term that remains unclear to developers, as well as where to put privacy in the development life cycle [35, 37]. Moreover, privacy is often seen as a byproduct of security [36, 38, 39].

Lack of privacy education. Additionally, most developers do not have any formal training regarding privacy [38]. Those who do have some education in this field are often required to mandatory training by certain certification constraints in their respective professions. Those who have no formal privacy education are only confronted with the topic when it pops up during their work. They often rely on their legal department (if their company is large enough to have one), or on friends and other third party sources [40], or decide for themselves according to their own personal opinions [41, 42]. That also leads to the problem, that developers often do not know which privacy technique fits which privacy requirements. This problem gets amplified because privacy techniques are hard to verify due to a lack of evaluation criteria [41] or key performance indicators [42] and that privacy terms are not clearly established and are thus not as well understood as, e.g., security terms [43]. Also, more advanced PETs such as homomorphic encryption [44], secure multi-party computing [45], or differential privacy [46] are seen as too difficult to use and adjust [47]. We have previously mentioned the GDPR, however, outside of Europe exist different regulations that have different

rules. It is arguably very hard for developers to keep track and thus to adhere to all of these regulations across continents [48] (see Sec. 4.1.2).

Privacy is not taken seriously. Other findings suggest that privacy is seen as outdated in the era of ubiquitous computing, privacy presumably cannot co-exist with knowledge creation, and that privacy is an obstacle which is unrealistic to implement [35, 43, 36]. However, privacy is also seen as a human right and also as means of making profit [35].

Current trends. To sum up, it has been shown that developers need assistance and guidelines in order to include privacy in the system design. Currently, developers try to find assistance regrading privacy not only in official documentations, but also in online forums, such as Stack Overflow [39, 49] or Reddit [50]. It has been shown that this could lead to less privacy-preserving coding practices [51]. Developers discuss privacy primarily if new privacy-related restrictions come up and, thus, usually in an unfavorable way [50]. Also, privacy tips on Stack Overflow focus only on some parts of privacy and can therefore only be used to a certain degree [49]. A current strategy are so-called "privacy champions", i.e., members of a software engineering team, who advocate and foster a privacy-preserving development culture [52]. This shows that developers need to be supported and encouraged on a larger scale in order to engineer privacy-preserving IoT applications.

3. Methodology

We conducted the literature study by using Google Scholar. As search string, we first used *IoT privacy, developers IoT privacy, and IoT privacy engineering* amongst other phrases. Next, we incorporated more refined

search strings and used terms that we identified as relevant in this field, e.g., *data flow, privacy architecture, IoT privacy framework*. We filtered relevant papers based on their abstracts and conclusions by the following criteria: (1) It is about IoT developers and focuses on privacy, (2) it is not an actual PbD implementation, but a framework or a tool targeted at IoT developers.

If a paper was classified as relevant, we performed a backwards search using the cited references and a forwards search (snowballing) via the "cited by" option in Google Scholar. Finally, we looked at other papers from authors who we identified as working in the field of IoT privacy and papers that were published on the author's personal or institutional websites. We repeated this process over time to include the newly published papers. Tab. 2 presents our search results. Papers are classified based on their nature into (1) frameworks and (2) data flow tools and their common underlying project.

4. Privacy methods for IoT developers

In this section, we discuss the context and the cope of the existing tools, frameworks, and guidelines listed in Tab. 2 and selected based on the methodology described in Sec. 3.

4.1. Guidelines and Frameworks

First we deal with papers that provide guidelines or a concrete framework for IoT developers to incorporate privacy in their IoT projects. The first steps towards IoT privacy frameworks have been taken by Kung et al. who already include an exhaustive list of topics and the privacy protection properties explained in Sec. 2.1 [66].

Category	Project	Author	Paper	Year
Guidelines and Frameworks	IoT privacy framework	Perera et al.	[22]	2016
			[14]	2020
			[53]	2021
		C. Perera	[54]	2017
	Combined privacy law framework	Aljeraisy et al.	[55]	2020
			[48]	2021
Data flow tools	FlowFence	Fernandes et al.	[56]	2016
	SAINT	Celik et al.	[57]	2018
	Databox	Crabtree et al.	[58]	2016
			[59]	2017
			[60]	2018
		Mortier et al.	[61]	2016
		Lodge et al.	[62]	2018
			[63]	2019
		Urquhart et al.	[64]	2019
	GDPR Controller	Rhahla et al.	[65]	2019

Table 2: Overview and categorization of papers.

4.1.1. IoT Privacy Framework

Building upon the same principles (and those of Hoepman [20]), Perera et al. developed a framework to support IoT software engineers in adapting those privacy principles [22]. The authors presented a step-by-step guide for IoT developers: (1) Identify data flows through the devices of the system, (2) Build an assessment table for each node with life cycle phases and the PbD guidelines, and (3) Go through the guidelines and assess the items using color codes. To use the guidelines, the developer now checks whether one of the 30 suggested privacy guidelines is addressed during the particular life cycle phase of each node. The authors summarized their privacy framework in a cheat sheet for IoT developers [54]. As such, it can be regarded as an important step towards the goal of incorporating PbD

into IoT but might be overwhelming for IoT developers who often lack the necessary training and skills to assess their applications from a privacy perspective as discussed in Sec. 2.4. However, the presented guidelines are an exhaustive, concrete, and holistic approach to tackle the problem of creating privacy-preserving IoT applications. This has been confirmed in [14], where the authors conducted an evaluation with a total of 26 participants. Both, novice and expert developers in their sample profited significantly from the use of the proposed guidelines as they were able to identify more privacy measures when using the guidelines [14]. However, the authors admit that their list of privacy guidelines is lengthy and that the participants might have experienced some fatigue. To address these shortcomings, the authors have proposed the concept of a tool that removes partially the burden for the developers by automating tasks [53]. The envisioned privacy tool should automate the inclusion of privacy protection techniques during the design of the IoT application, evaluate the privacy awareness of the system, and finally output a rating of these privacy measures and generates terms and conditions for the end users [53].

4.1.2. Combined Privacy Law Framework

Aljeraisy et al. further proposed a *Combined Privacy Law Framework (CPLF)* [48]. The CPLF is a combination and standardization of data regulation laws across the EU, North America, and Oceania. The resulting CPLF consists of 13 key principles (including Hoepman's principles) and 11 individual's rights [48]. The authors mapped privacy patterns to each of these key principles, which are in turn compared to other principles, such as those explained in Sec. 2.1, to show that all the different ideas regarding privacy principles are covered. Aljeraisy et al. released a technical report,

which includes the detailed mapping [55]. The way developers can now use this mapping is as follows: During the design of an IoT application, the developer can apply, e.g., privacy patterns to a certain action that takes place. For example, they can show the user *privacy icons* when data is collected to conform to the principle of *transparency* and to *the right to be informed* [55].

4.2. Data Flow Tools

We have seen that the frameworks and guidelines outlined in Sec. 4.1 are on a high level and lack the necessary technical implementations to directly help developers in complying with privacy goals. The general idea of data flow tools is to enable the developer to check and restrict data flows throughout the IoT infrastructure with a technical solution, such as a programming framework or an analysis tool, the developers can immediately use. Fig. 1 shows a simplified example of a data flow within, e.g., a smart home.

4.2.1. FlowFence—A sandbox for sensitive data

The first work discussed here—*FlowFence* [56]— targets developers of applications that use sensitive data from IoT sources such as smart homes and wearables. It is a Java framework that separates the computation of sensitive data from the rest of the application. That way, unintentional leaks of sensitive data (such as PII) can be prevented, as only allowed data flows are possible [56]. The operating principle is as follows: The developer can use FlowFence to build quarantined modules, which process all sensitive data of the application. Sensitive data cannot be accessed from anywhere outside of these modules. A core component of the architecture are the taint labels. These indicate the data source and the permitted flow of the

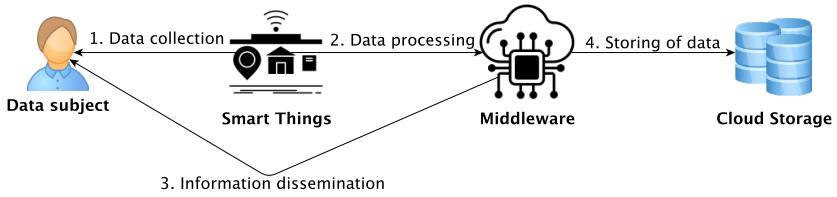

Figure 1: IoT data flow: (1) Smart things collect data from the data subject, (2) the data is processed via a middleware, and (3) the results are presented to the data subject. (4) Simultaneously the data gets stored, e.g., in the cloud for further analysis.

data. For example, they can state that the picture generated by the smart camera is not allowed to flow into the Internet. Taint labels are defined at the sensitive source's app policy and FlowFence takes care that all applications that use data from this source adhere to these labels. As a result, FlowFence would block requests that are not specified in the label [56]. The authors evaluated their solution by implementing an API and testing it on three IoT applications. Their results indicate that it takes a developer on average 1.7 days to incorporate FlowFence into their existing applications [56].

4.2.2. SainT

SainT (Static taint analysis Tool) is a tool to analyze the data flow of sensitive information in already developed IoT applications [57]. While originally designed for consumers of smart home IoT devices, it can also be useful for developers, however, it only works on the SmartThings [67] programming platform [57]. SainT is also available online [68]. SainT finds sensitive data flows by tracking taint sources to their respective outputs. Taint sources are device states (e.g., whether a smart lock is open or closed), device info (ID, model), location, user inputs (e.g., contact data for push

notifications via text message), and state variables (e.g. counter of a smart lock) [57]. SainT analyzes the IoT application and extracts a static taint analysis to report sensitive data flows by analyzing the source code. The authors state some limitations of this method, as some behaviors might lead to over-tainting [57]. Although the motivation for this project was to enable data subjects in having control over their data by providing transparency of the data flow, the presented tool also can be of use to IoT developers.

4.2.3. Databox

The *Databox model* was developed to support accountability towards personal data required by laws and regulations. The IoT Databox model was designed to help IoT consumers in building trust towards their IoT devices regarding the collected data [59, 61], however the scope also includes enabling developers to adhere to data protection laws [60]. The IoT Databox is a physical box with an accompanying IDE, which is a browser application available online [63, 69]. The general idea is that personal data should remain in a sandbox. It holds the personal data of all connected IoT devices, which would instead be stored in the cloud [64]. The goal is that developers are enabled to assess whether a *Data Protection Impact Assessment (DPIA)* is necessary in the current application design. Instead of taints, the authors use schemas to identify data types leaving the Databox, however, the principle is the same. The developers can use the IDE in order to build their IoT apps and publish them on the Databox app store. In doing so, they have to disclose all data that is being used as well as the respective purpose. When a user installs an app from this store, this policy is presented and the user can now give consent [62]. Furthermore, the IDE automatically detects sensitive data flow and visually prompts the developer to perform a DPIA [63].

The principle of tagging data in order to control and audit its data flow is similar to the previous approach FlowFence. However, the introduction of a physical object, which sits at the center of all personal data makes this approach less accessible for most developers (or end users). The main focus of the Databox project is to help developers in conforming to the GDPR during the design and development of IoT applications.

4.2.4. GDPR Controller

The goal of the GDPR Controller is to provide data subjects the possibility to restrict the data flow in an IoT environment [65]. The controller runs on top of IoT applications, identifies data flows, and presents them to the data subject in a graphical interface in which the data subject can consent to the respective data flows. The authors show the feasibility of the GDPR Controller with an e-health use case. The GDPR Controller gives developers a way to adhere to Hoepman's principle of *control*, however, it is questionable whether users wish to explicitly consent to every potentially personal data flow. Also it is not clear how this tool would work in practice.

5. Discussion

We next discuss the presented approaches based on their fulfilment of privacy principles, usefulness, and availability.

Privacy principles. Tab. 4 compares the methods according to Hoepman's privacy principles. The privacy framework and CPLF cover all the privacy principles laid out by Hoepman, as they used his principles explicitly to develop their frameworks. The data flow methods primarily cover the principles *minimize* and *enforce*. The flow of sensitive data is limited and

Projects	**Evaluation**	**Availability**	**Shortcomings**
IoT privacy framework	User study (26)	Cheat sheet [54]	Usability
CPLF	Case studies	Techn. report [70]	Usability
FlowFence	Micro Benchmarks, User study (1)	Online [71]	Correct privacy policies needed
SainT	Market study, IoT app analysis	Online [68]	Only works on SmartThings
Databox	—	Online [69]	Physical device needed
GDPR Controller	Case study	-	Limited usability

Table 3: Comparison of the presented projects.

partially prohibited, and the developers are forced to follow the privacy policies, which are in place. This is also a potential shortcoming, as the privacy concept hinges on correct and complete privacy policies from all involved parties. Databox is a larger project than the other data flow tools, however, its main advantage over the other two presented methods is, that it covers more privacy principles (see Tab.4). The GDPR Controller is the only data flow tool that covers the principle of *control*, however, it can only be used in combination with other approaches as it covers no other privacy principle.

	Minimize	Hide	Separate	Aggregate	Inform	Control	Enforce	Demonstrate
IoT privacy framework	✔	✔	✔	✔	✔	✔	✔	✔
CPLF	✔	✔	✔	✔	✔	✔	✔	✔
FlowFence	✔		✔				✔	
SainT	✔						✔	
Databox	✔		✔		✔		✔	✔
GDPR Controller						✔		

Table 4: Mapping of the presented methods to Hoepman's privacy principles (Sec 2.1).

Usability. The manual task of checking guidelines during every step of the design seems daunting and not very appealing to developers. Thus, the usability of the guidelines and frameworks is questionable, and, as the authors themselves note, an automation is necessary if IoT developers should be expected to adopt this method. The evaluation of the IoT privacy framework did not take the usability of the proposed method into account, however, it shows that their framework is exhaustive and helps developers in identifying privacy risks and choosing an appropriate countermeasure. There was no user evaluation of the CPLF, however, this work shows the connection between privacy patterns and privacy regulations. An automation of this process is thinkable in a way that an AI detects already applied privacy patterns and outputs possible non-compliance with data regulations.

As previously mentioned in Sec. 2.3, the volatile nature of IoT ecosystems makes automation difficult to achieve. Also, privacy patterns (see Sec. 2.2) do not yet capture the whole privacy picture and should therefore be extended, improved and categorized [53]. Finally, the challenge of "the human in the loop" remains, as ultimately the developer decides whether certain privacy requirements are sufficient or not. That means that the problem of privacy perceptions (see Sec. 2.4) still remains, even if certain processes would become automated [53]. In conclusion, a tool, which automates the application of privacy techniques in all layers of an IoT application, would be the goal of privacy engineering in the scope of IoT. Still, knowledge about privacy techniques would be required. The data flow tools would arguably be more useful for developers to restrict unwanted data flow, however, a huge burden lies on the publishers of the IoT application to use FlowFence in the intended way, i.e., to specify concrete policies. The idea of FlowFence is in line with the principle to *separate* sensitive data from the rest. Besides

the market study, the actual implementation and usage of FlowFence has been evaluated with one developer. SainT, on the other hand has only been evaluated on existing apps, as its target group is primarily IoT consumers. The idea of the GDPR Controller's seems to be usable for developers, however, as its main target are the data subjects who would have to consent to every possible data flow, the actual applicability seems to be limited.

Availability. The frameworks and guidelines provide exhaustive checklists and mapping tables for developers. FlowFence is available online [71], however, it seems that work on the project has been discontinued, as the last contribution was in November 2018. The major shortcoming of SainT is, that it only works on SmartThings devices, while Databox' deployment is more complicated, as it requires the installation of the physical Databox device. The GDPR Controller did not provide any code or software to use.

6. Future work / open research areas

Guidelines and frameworks suffer from the checklist-approach, which developers usually do not wish to partake in. Automated tool support is hence necessary [53]. Future research can tackle this problem by relying on tools that automatically assign relevant privacy techniques in the respective development step. It could, e.g., detect that PII is processed without being anonymized and prompt the developer to look into the issue. Additional help may however be necessary to solve it (see Sec. 2.4). A more advanced tool might even manage the anonymization itself in a context dependent way to not interfere with the actual task of the device, as anonymization is not practical at all times. Another possibility is to develop a method that compares already implemented privacy measures with current data protection laws

and hints at necessary adaptions. Both ideas might be feasible with current machine learning research. Data flow is an important aspect in this regard. However, current data flow tools rely on self reporting of certain data types. A method that automatically detects sensitive data flows might be feasible. Most privacy methods deal with basic techniques, such as anonymization and encryption. However, more sophisticated techniques, such as homomorphic encryption, secure multi-party computation, and differential privacy, should also be considered when designing an IoT privacy framework. Developers and other industry experts are not well versed in those advanced techniques [47], thus, research that enables developers to adapt these techniques is needed [72]. As previously mentioned, IoT developers get little assistance in choosing and configuring appropriate PETs. A technical solution that identifies privacy gaps and guides developers to suitable PETs would arguably reduce the privacy risks associated to the IoT devices' use of PII. Moreover, guidance for correct configuration of those PETs would also greatly reduce privacy and security risks [73]. Furthermore, usability evaluation of the presented techniques has been done only in two cases, and only one of those projects evaluated more than one developer. However, as techniques that support the implementation of PbD are only reasonable if they are used, extensive usability evaluations with IoT-developers, differing in experience, expertise, and sector should be conducted. This is something we plan to do in the scope of differential privacy for IoT [72].

7. Conclusion

Privacy in IoT is harder to accomplish than for normal web or mobile applications. Therefore, it is important to enable IoT developers to incor-

porate privacy techniques into their development life cycle, especially, as we have seen that developers do not feel responsible in engineering privacy and often lack the necessary skills. Very few approaches exist that deal explicitly with the challenges, that privacy-preserving IoT development hold. Many approaches focus on privacy in general or treat IoT privacy as a byproduct of IoT security. We have summarized and evaluated privacy guidelines and data flow tools for the IoT and analyzed their positive and negative aspects as well as the privacy goals that are met. Comprehensive privacy guidelines for developers cover all of the IoT layers and all privacy goals, however, they are difficult and tedious to implement. Automated tool support in this field is recommended and would increase the adoption of such frameworks and guidelines, however, currently they are only envisioned. Data flow tools are an automated way of tracking (sensitive) data through all the IoT layers, but rely on correct labelling of data or on user input. These kinds of tools are nevertheless helpful in providing developers some assistance in engineering privacy in the IoT. Future work should thus concentrate on as much automation as possible, as well as ensuring the usability of the solutions.

References

[1] B. Carminati, P. Colombo, E. Ferrari, G. Sagirlar, Enhancing User Control on Personal Data Usage in Internet of Things Ecosystems, in: IEEE International Conference on Services Computing (SCC), 2016.

[2] J. A. Stankovic, Research Directions for the Internet of Things, IEEE Internet of Things Journal (IoT-J) (2014).

[3] M. Conti, M. Nati, E. Rotundo, R. Spolaor, Mind the Plug! Laptop-User Recognition through Power Consumption, in: Proc. of the 2nd

ACM International Workshop on IoT Privacy, Trust, and Security (IoTPTS), 2016.

[4] B. Copos, K. Levitt, M. Bishop, J. Rowe, Is Anybody Home? Inferring Activity from Smart Home Network Traffic, in: IEEE Security and Privacy Workshops (SPW), 2016.

[5] L. Hutton, B. A. Price, R. Kelly, C. McCormick, A. K. Bandara, T. Hatzakis, M. Meadows, B. Nuseibeh, Assessing the Privacy of mHealth Apps for Self-Tracking: Heuristic Evaluation Approach, JMIR mHealth and uHealth (JMU) (2018).

[6] D. Geneiatakis, I. Kounelis, R. Neisse, I. Nai-Fovino, G. Steri, G. Baldini, Security and Privacy Issues for an IoT Based Smart Home, in: 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017.

[7] L. H. Acosta, D. Reinhardt, A Survey on Privacy Issues and Solutions for Voice-Controlled Digital Assistants, Pervasive and Mobile Computing (PMC) (2021).

[8] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, A. Zanella, IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices, IEEE Internet of Things Journal (IoT-J) (2019).

[9] P. P. Jayaraman, X. Yang, A. Yavari, D. Georgakopoulos, X. Yi, Privacy Preserving Internet of Things: From Privacy Techniques to a Blueprint Architecture and Efficient Implementation, Future Generation Computer Systems (FGCS) (2017).

[10] D. T. Nguyen, C. Song, Z. Qian, S. V. Krishnamurthy, E. J. Colbert, P. McDaniel, IoTSan: Fortifying the Safety of IoT Systems, in: Proc. of the 14th International Conference on emerging Networking EXperiments and Technologies, 2018.

[11] Z. B. Celik, G. Tan, P. D. McDaniel, IoTGuard: Dynamic Enforcement of Security and Safety Policy in Commodity IoT, in: NDSS, 2019.

[12] Z. B. Celik, P. McDaniel, G. Tan, Soteria: Automated {IoT} Safety and Security Analysis, in: Proc. of the 18th USENIX Annual Technical Conference (USENIX ATC 18), 2018.

[13] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, W. Joosen, A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements, Requirements Engineering (2011).

[14] C. Perera, M. Barhamgi, A. K. Bandara, M. Ajmal, B. Price, B. Nuseibeh, Designing Privacy-Aware Internet of Things Applications, Information Sciences (2020).

[15] F. Kammueller, Formal Modeling and Analysis of Data Protection for GDPR Compliance of IoT Healthcare Systems, in: Proc. of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018.

[16] C. Perera, R. Ranjan, L. Wang, S. U. Khan, A. Y. Zomaya, Big Data Privacy in the Internet of Things Era, IT Professional (2015).

[17] A. Pérez Fernández, G. Sindre, Mitigating the Impact on Users' Privacy Caused by Over Specifications in the Design of IoT Applications, Sensors (2019).

[18] M. Hansen, M. Jensen, M. Rost, Protection Goals for Privacy Engineering, in: Proc. of the IEEE Security and Privacy Workshops (SPW), 2015.

[19] A. Cavoukian, J. Polonetsky, C. Wolf, Smartprivacy for the Smart Grid: Embedding Privacy into the Design of Electricity Conservation, Identity in the Information Society (IDIS) (2010).

[20] J.-H. Hoepman, Privacy Design Strategies, in: Proc. of the 30th IFIP International Information Security Conference (SEC), 2014.

[21] C. Perera, R. Ranjan, L. Wang, End-to-End Privacy for Open Big Data Markets, IEEE Cloud Computing (2015).

[22] C. Perera, C. McCormick, A. K. Bandara, B. A. Price, B. Nuseibeh, Privacy-by-Design Framework for Assessing Internet of Things Applications and Platforms, in: Proc. of the 6th International Conference on the Internet of Things, 2016.

[23] L. Sweeney, k-anonymity: A model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (2002).

[24] Unabhängiges Landeszentrum für Datenschutz, The Standard Data Protection Model, Version 2.0, Technical Report, 2020.

[25] ISO/IEC 29100:2011, Information Technology – Security Techniques – Privacy Framework, ISO, Geneva, Switzerland, 2011.

[26] K. R. Boeckl, N. B. Lefkovitz, et al., NIST Privacy Framework: A Tool for Improving Privacy through Enterprise Risk Management, Version 1.0 (2020).

[27] S. Gürses, C. Troncoso, C. Diaz, Engineering Privacy by Design, Computers, Privacy & Data Protection (2011).

[28] N. Doty, M. Gupta, Privacy Design Patterns and Anti-Patterns, Trustbusters Workshop at the Symposium on Usable Privacy and Security (2013).

[29] Privacy Patterns, Online: https://privacypatterns.org/, (acc. 12/2021).

[30] T. T. Doan, R. Safavi-Naini, S. Li, S. Avizheh, P. W. Fong, Towards a Resilient Smart Home, in: Proc. of the 2018 workshop on IoT Security and Privacy (IoT S&P), 2018.

[31] D. Christin, A. Reinhardt, P. S. Mogre, R. Steinmetz, et al., Wireless Sensor Networks and the Internet of Things: Selected Challenges, Proc. of the 8th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (2009).

[32] V. Geetanjali, I. Subramanian, G. Kannan, S. B. Prathiba, G. Raja, IoTexpert: Interconnection, Interoperability and Integration of IoT Platforms, in: 11th International Conference on Advanced Computing (ICoAC), 2019.

[33] L. Lan, R. Shi, B. Wang, L. Zhang, An IoT Unified Access Platform for Heterogeneity Sensing Devices Based on Edge Computing, IEEE access (2019).

[34] S. Spiekermann, L. Cranor, Privacy Engineering, IEEE Transactions on Software Engineering (2009).

[35] S. Spiekermann, J. Korunovska, M. Langheinrich, Inside the Organization: Why Privacy and Security Engineering is a Challenge for Engineers, Proc. of the IEEE (2018).

[36] I. Hadar, T. Hasson, O. Ayalon, E. Toch, M. Birnhack, S. Sherman, A. Balissa, Privacy by designers: Software developers' privacy mindset, Empirical Software Engineering (2018).

[37] S. Spiekermann, The Challenges of Privacy by Design, Communications of the ACM (2012).

[38] M. Peixoto, D. Ferreira, M. Cavalcanti, C. Silva, J. Vilela, J. Araújo, T. Gorschek, On Understanding How Developers Perceive and Interpret Privacy Requirements Research Preview, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, 2020.

[39] M. Tahaei, K. Vaniea, N. Saphra, Understanding Privacy-Related Questions on Stack Overflow, in: Proc. of the CHI Conference on Human Factors in Computing Systems, 2020.

[40] R. Balebako, A. Marsh, J. Lin, J. I. Hong, L. F. Cranor, The Privacy and Security Behaviors of Smartphone App Developers, Workshop on Usable Security (USEC) (2014).

[41] A. Senarath, N. A. Arachchilage, Why Developers cannot Embed Privacy into Software Systems? An Empirical Investigation, in: Proc. of the 22nd International Conference on Evaluation and Assessment in Software Engineering (EASE), 2018.

[42] F. Bu, N. Wang, B. Jiang, H. Liang, "Privacy by Design" Implementation: Information System Engineers' Perspective, International Journal of Information Management (2020).

[43] K. Bednar, S. Spiekermann, M. Langheinrich, Engineering Privacy by Design: Are Engineers Ready to Live up to the Challenge?, The Information Society (2019).

[44] C. Gentry, Fully homomorphic encryption using ideal lattices, in: Proc. of the 41st annual ACM symposium on Theory of computing, 2009.

[45] A. C.-C. Yao, How to Generate and Exchange Secrets, in: 27th Annual Symposium on Foundations of Computer Science (SFCS), 1986.

[46] C. Dwork, Differential Privacy, in: Proc. of the 33rd International Colloquium on Automata, Languages and Programming (ICALP), 2006.

[47] N. Agrawal, R. Binns, M. Van Kleek, K. Laine, N. Shadbolt, Exploring Design and Governance Challenges in the Development of Privacy-Preserving Computation, in: Proc. of the CHI Conference on Human Factors in Computing Systems, 2021.

[48] A. Aljeraisy, M. Barati, O. Rana, C. Perera, Privacy Laws and Privacy by Design Schemes for the Internet of Things: A Developer's Perspective, ACM Computing Surveys (CSUR) (2021).

[49] M. Tahaei, T. Li, K. Vaniea, Understanding Privacy-Related Advice on Stack Overflow, Proceedings on Privacy Enhancing Technologies (PoPETS) (2022).

[50] T. Li, E. Louie, L. Dabbish, J. I. Hong, How Developers Talk About Personal Data and What It Means for User Privacy: A Case Study of a Developer Forum on Reddit, Proc. of the ACM on Human-Computer Interaction (2021).

[51] Y. Acar, M. Backes, S. Fahl, D. Kim, M. L. Mazurek, C. Stransky, You Get where You're Looking for: The Impact of Information Sources on Code Security, in: IEEE Symposium on Security and Privacy (SP), 2016.

[52] M. Tahaei, A. Frik, K. Vaniea, Privacy Champions in Software Teams: Understanding Their Motivations, Strategies, and Challenges, in: Proc. of the CHI Conference on Human Factors in Computing Systems, 2021.

[53] C. Perera, M. Barhamgi, M. Vecchio, Envisioning Tool Support for Designing Privacy-Aware Internet of Thing Applications, IEEE Internet of Things Magazine (2021).

[54] C. Perera, Privacy Guidelines for Internet of Things: A Cheat Sheet, Technical Report, 2017.

[55] A. Aljeraisy, M. Barati, O. Rana, C. Perera, Exploring the Relationships between Privacy by Design Schemes and Privacy Laws: A Comparative Analysis, 2020.

[56] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, A. Prakash, Flowfence: Practical Data Protection for Emerging IoT Application Frameworks, in: 25th USENIX Security symposium, 2016.

[57] Z. B. Celik, L. Babun, A. K. Sikder, H. Aksu, G. Tan, P. McDaniel, A. S. Uluagac, Sensitive Information Tracking in Commodity IoT, in: 27th USENIX Security Symposium, 2018.

[58] A. Crabtree, T. Lodge, J. Colley, C. Greenhalgh, R. Mortier, H. Haddadi, Enabling the New Economic Actor: Data Protection, the Digi-

tal Economy, and the Databox, Personal and Ubiquitous Computing (2016).

[59] A. Crabtree, T. Lodge, J. Colley, C. Greenghalgh, R. Mortier, Accountable Internet of Things? Outline of the IoT Databox model, in: IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017.

[60] A. Crabtree, T. Lodge, J. Colley, C. Greenhalgh, K. Glover, H. Haddadi, Y. Amar, R. Mortier, Q. Li, J. Moore, et al., Building Accountability into the Internet of Things: The IoT Databox Model, Journal of Reliable Intelligent Environments (2018).

[61] R. Mortier, J. Zhao, J. Crowcroft, L. Wang, Q. Li, H. Haddadi, Y. Amar, A. Crabtree, J. Colley, T. Lodge, et al., Personal Data Management with the Databox: What's inside the Box?, in: Proc. of the ACM Workshop on Cloud-Assisted Networking, 2016.

[62] T. Lodge, A. Crabtree, A. Brown, Developing GDPR compliant Apps for the edge, in: Data Privacy Management, Cryptocurrencies and Blockchain Technology, 2018.

[63] T. Lodge, A. Crabtree, Privacy Engineering for Domestic IoT: Enabling Due Diligence, Sensors (2019).

[64] L. Urquhart, T. Lodge, A. Crabtree, Demonstrably Doing Accountability in the Internet of Things, International Journal of Law and Information Technology (2019).

[65] M. Rhahla, T. Abdellatif, R. Attia, W. Berrayana, A GDPR Controller for IoT Systems: Application to e-Health, in: Proc. of the 28th

International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2019.

[66] A. Kung, F. Kargl, S. Suppan, J. Cuellar, H. C. Pöhls, A. Kapovits, N. N. McDonnell, Y. S. Martin, A Privacy Engineering Framework for the Internet of Things, in: Data Protection and Privacy:(In) visibilities and Infrastructures, 2017.

[67] Samsung SmartThings, Online: https://www.smartthings.com, (acc. 10/2021).

[68] SAINT, Online: http://saint-project.appspot.com, (acc. 10/2021).

[69] Databox, Online: https://github.com/me-box/databox/, (acc. 10/2021).

[70] A. Aljeraisy, O. Rana, C. Perera, A Systematic Analysis of Privacy Laws and Privacy by Design Schemes for the Internet of Things: A Developer's Perspective (2020).

[71] FlowFence, Online: https://github.com/earlence/FlowFence_Release, (acc. 10/2021).

[72] P. Kühtreiber, D. Reinhardt, Usable Differential Privacy for the Internet-of-Things, in: Proc. of the 19th IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), 2021.

[73] OWASP Top 10: Security Misconfiguration, Online: https://owasp.org/Top10/A05_2021-Security_Misconfiguration/, (acc. 01/2022).