Privacy in Internet of Things: from Principles to Technologies

Chao Li, *Student Member, IEEE,* and Balaji Palanisamy, *Member, IEEE*

***Abstract*—Ubiquitous deployment of low-cost smart devices and widespread use of high-speed wireless networks have led to the rapid development of the Internet of Things (IoT). IoT embraces countless physical objects that have not been involved in the traditional Internet and enables their interaction and cooperation to provide a wide range of IoT applications. Many services in the IoT may require a comprehensive understanding and analysis of data collected through a large number of physical devices that challenges both personal information privacy and the development of IoT. Information privacy in IoT is a broad and complex concept as its understanding and perception differ among individuals and its enforcement requires efforts from both legislation as well as technologies. In this paper, we review the state-of-the-art principles of privacy laws, the architectures for IoT and the representative privacy enhancing technologies (PETs). We analyze how legal principles can be supported through a careful implementation of privacy enhancing technologies (PETs) at various layers of a layered IoT architecture model to meet the privacy requirements of the individuals interacting with IoT systems. We demonstrate how privacy legislation maps to privacy principles which in turn drives the design of necessary privacy enhancing technologies to be employed in the IoT architecture stack.**

***Index Terms*—Internet of Things, privacy, privacy by design, privacy enhancing technologies, PET, privacy laws, GDPR**

I. Introduction

UBIQUITOUS deployment of low-cost smart devices and widespread use of high-speed wireless networks have led to the rapid development of Internet of Things (IoT). IoT embraces countless physical objects embedded with Radio Frequency Identification (RFID) tags, sensors and actuators that have not been involved in the traditional Internet and enables their interaction and cooperation through both traditional as well as IoT-specific communication protocols [1], [2]. Gartner [3] estimates that around 20.4 billion 'things' will be connected by the year 2020. These pervasive and heterogeneous devices that interact with the physical and digital worlds have the potential to significantly enhance the quality of life for individuals interacting with the IoT. With smart home and wearable devices, users obtain seamless and customized services from digital housekeepers, doctors and fitness instructors [4]. Smart building and smart city applications provide an increased awareness of the surroundings and offer greater convenience and benefits to the users [5], [6].

Many services offered by IoT may require a comprehensive understanding of user interests and preferences, behavior patterns and thinking models. For instance, in the Christmas special episode of the British series 'Black Mirror', the soul of a woman is copied to serve as the controller of her smart home, which can wake up the woman with her favorite music and breakfast as the copy knows her as no one else can [7]. Such a digital copy, which could be hard to create in the traditional Internet, is relatively easier to be generated in the IoT era. While some individuals prefer the convenience of the services, some others may be concerned about their personal data being shared [8]. In 2013, the IEEE Internet of Things survey showed that 46% of respondents consider privacy concerns as the biggest challenge for IoT adoption [9]. Large scale data collection in the IoT poses significant privacy challenges and may hamper the further development and adoption by privacy-conscious individuals [10].

Information privacy is a broad and complex notion as its understanding and perception differ among individuals and its enforcement requires efforts from both legislation and technologies [5], [11]. Privacy laws help to enforce compliance and accountability of privacy protection and make privacy protection a necessity for every service provider [11]. Privacy enhancing technologies (PETs) on the other hand support the underlying principles guided by privacy laws that enable privacy protection strategies to be implemented in engineering [12], [13]. In this paper, we study the privacy protection problem in IoT through a comprehensive review by jointly considering three key dimensions, namely the state-of-the-art principles of privacy laws, architectures for the IoT system and representative privacy enhancing technologies (PETs). Based on an extensive analysis along these three dimensions, we show that IoT privacy protection requires significant support from both privacy enhancing technologies (PETs) and their enforcement through privacy legislation. We analyze how legal principles can be supported through a careful implementation of various privacy enhancing technologies (PETs) at various layers of a layered IoT architecture model to meet the privacy requirements of the individuals interacting with IoT systems. Our study is focused on providing a broader understanding of the state-of-the-art principles in privacy legislation associated with the design of relevant privacy enhancing technologies (PETs) and on demonstrating how privacy legislation maps to privacy principles which in turn drives the design of necessary privacy enhancing technologies to be employed in the IoT architecture stack.

We organize the paper in the following manner. In Section II, we analyze the principles of privacy laws and present the privacy-by-design strategies that can adopt the general principles to engineering practice. In Section III, we introduce the IoT system using a layered reference architecture and

Chao Li and Balaji Palanisamy are with the School of Computing and Information, University of Pittsburgh.
E-mail: chl205@pitt.edu, bpalan@pitt.edu

describe the functionalities and enabling technologies of each layer. We discuss how privacy-by-design strategies can be integrated into the reference architecture. In Section IV to Section VI, we introduce the state-of-the-art privacy enhancing technologies (PETs), analyze their suitability for privacy-by-design strategies and discuss the pros and cons of their use and implementation in each IoT layer. In Section VII, we discuss privacy issues in IoT applications. Finally, we present the related work in Section VIII and conclude in Section IX.

II. Privacy

Privacy is a complex and a subjective notion as its understanding and perception differ among individuals. In this section, we review the definitions of privacy in the past, introduce the privacy laws and analyze the state-of-the-art privacy legislation. We then introduce the privacy-by-design (PbD) strategies that facilitate the design of privacy-preserving systems satisfying the legal principles.

A. Definition

As far back as the thirteenth century, when the eavesdroppers were claimed to be guilty, the notion of media privacy had come into being [14]. Then, with the technical and social development, the notion of privacy successively shifted to territorial (eighteenth century), communication (1930s), and bodily privacy (1940s) [11]. Finally, in the 1960s, it was the rise of electronic data processing that brought into being the notion of information privacy (or data privacy) that has achieved lasting prominence until now. In 1890, Warren and Brandeis defined privacy as 'the right to be let alone' in their famous article 'The Right to Privacy' [15]. After that, many privacy definitions have been emerging unceasingly, but the one proposed by Alan Westin in his book 'Privacy and Freedom' has become the base of several modern data privacy principles and law [11]. Westin defined privacy as 'the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others' [16], which mainly emphasized the control of the data subjects over their data. The authors in [10] argued that Westin's definition was too general for the IoT area and they proposed a more focused one that defines the IoT privacy as the threefold guarantee including 'awareness of privacy risks imposed by smart things and services surrounding the data subject; individual control over the collection and processing of personal information by the surrounding smart things; awareness and control of subsequent use and dissemination of personal information by those entities to any entity outside the subjects personal control sphere'.

B. Legislation

Privacy laws form a critical foundation in the design of any privacy-preserving system. As the cornerstone of most modern privacy laws and policies, the Fair Information Practices (FIPs) are a set of internationally recognized practices to protect individual information privacy [17]. The code of FIPs was born out of a report from the Department of Health, Education & Welfare (HEW) [18] in 1973 and then adopted by the US Privacy Act of 1974, the most famous privacy legislation in the early stage. The original HEW FIPs consist of five principles that can be summarized as [18]:

1. No secret systems of personal data.
2. Ability for individuals to find out what is in the record, and how it is used.
3. Ability for individuals to prevent secondary use.
4. Ability to correct or amend records.
5. Data must be secure from misuse.

However, as a federal law, the US Privacy Act of 1974 only works with the federal government. There is no general information privacy legislation that covers all states and areas [19]. As a result, the FIPs always act as the guideline of the various privacy laws and regulations ranging from different organizations (e.g., Stanford University [20], Department of Homeland Security [21]) to different areas (e.g., HIPAA [22], COPPA [23]).

In 1980, based on the core HEW FIPs, the Organization for Economic Cooperation and Development (OECD) adopted the Guidelines on the Protection of Personal Privacy and Transborder Flows of Personal Data [24]. It is considered a historical milestone as it represented the first internationally-agreed upon privacy protection [19]. The eight principles extended from the five basic FIPs have been the foundation of most EU privacy laws later. They can be summarized as:

1. **Collection Limitation**: Collection should be lawful, fair and with knowledge or consent of the data subject.
2. **Data Quality**: Personal data should be purpose-relevant, accurate, complete and kept up-to-date.
3. **Purpose Specification**: Purposes should be specified earlier than collection and complied with.
4. **Use Limitation**: Personal data should not be disclosed, made available or used for non-specified purposes.
5. **Security Safeguards**: Personal data should be protected by reasonable security safeguards.
6. **Openness**: There should be a general policy of openness about developments, practices and policies with respect to personal data.
7. **Individual Participation**: An individual should have the right to access his data, be timely informed on data collection, be given disputable reason for denied lawful request and challenge his data to have the data erased, rectified, completed or amended.
8. **Accountability**: A data controller should be accountable for complying with measures which give effect to the principles stated above.

Although the OECD guidelines achieved worldwide recognition, it was nonbinding. It was not until 1995 that the EU passed Directive 95/46/EC [25] and the OECD guidelines were incorporated into an influential privacy law for the first time. Unlike the US, the EU dedicated to enforcing the omnibus privacy laws to comprehensively protect individual data in its

member countries through not only the principles, but the restriction on the data transference with non-EU countries, which in turn has influenced the development of the privacy laws in the non-EU countries and the appearance of the data exchange regulations such as the Safe Harbor [26] and its later replacement, the EU-US Privacy Shield framework [27]. Recently, as the successor of Directive 95/46/EC, the General Data Protection Regulation (GDPR) [28] was adopted by the EU in 2016 and it has come into force in 2018. In the GDPR, most principles are covered by the Article 5, including 'lawfulness, fairness and transparency', 'purpose limitation', 'data minimization', 'accuracy', 'storage limitation', 'integrity and confidentiality' and 'accountability'. Its key changes in terms of the principles, compared with the Directive 95/46/EC, include six aspects [28], [29]:

- **Consent**: The GDPR is more strict with consents. A consent should be graspable, distinguishable and easy to be withdrawn.
- **Breach Notification**: The GDPR makes the breach notification mandatory. The notification should be sent within 72 hours after being aware of the breach.
- **Right to Access**: The first right mentioned in the OECD 'individual participation' principle is strengthened in the GDPR.
- **Right to be Forgotten**: In the Article 17, data is required to be erased when the personal data are no longer necessary in relation to the purposes or the consent is withdrawn.
- **Data Portability**: In the Article 20, a data subject has the right to receive his uploaded data in a machine-readable format and transmit it to another data controller.
- **Privacy by Design**: The privacy by design is finally integrated into the privacy legal framework. As claimed in the Article 25, 'the controller shall, both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organizational measures', which asks the privacy to be taken into account at the design stage, rather than as an add-on function.

C. Privacy by design

The notion of privacy by design (PbD) [5], [13], [14], namely embedding the privacy measures and privacy enhancing technologies (PETs) directly into the design of software or system, is not new. As early as 2001, Langheinrich [14] proposed six principles to guide the PbD in the ubiquitous systems, including *notice*, *choice and consent*, *proximity and locality*, *anonymity and pseudonymity*, *security*, and *access and recourse*. However, the PbD has never been extensively used in engineering. The main reason for its rare adoption is that most engineers either neglect the importance of privacy or refuse their responsibility on it [34], [35].

A privacy law may also need support from technologies. In IoT, the enforcement of each principle in a privacy law may need to be supported by a set of technologies (e.g., PETs) in one or multiple layers. Here, the principles in the laws are usually described with very general and broad terms [5] that makes it hard for engineers to properly implement them in the system design. Also, the availability of so many technologies makes the engineers' job of mapping technologies to principles difficult. Therefore, we need the PbD to take the role as an adaptation layer between laws and technologies to translate legal principles to more engineer-friendly principles that can facilitate the system design.

In [34], Spiekermann and Cranor divided the technologies into two types of approaches to enable privacy in engineering, namely '*privacy-by-architecture*' and '*privacy-by-policy*'. The privacy-by-architecture approaches can protect higher-level privacy through technologies enabling data minimization and local processing but is rarely adopted because of the lack of legal enforcement at that time and its conflict with the business interests. In contrast, the privacy-by-policy approaches protect only the bottom-line privacy through technologies supporting the notice and choice principles when the privacy-by-architecture technologies are not implemented. The authors argued that the privacy-by-policy technologies become less important when rigorous minimization has been guaranteed by the privacy-by-architecture technologies. Based on the two approaches, in 2014, Hoepman [13] proposed eight privacy design strategies, including four data-oriented strategies and four process-oriented strategies that roughly match the privacy-by-architecture and privacy-by-policy classification [12], [13]:

Data-oriented strategies:

1. **Minimize**: The amount of processed personal data should be restricted to the minimal amount possible.
2. **Hide**: Any personal data, and their interrelationships, should be hidden from plain view.
3. **Separate**: Personal data should be processed in a distributed fashion, in separate compartments whenever possible.
4. **Aggregate**: Personal data should be processed at the highest level of aggregation and with the least possible detail in which it is (still) useful.

Process-oriented strategies:

1. **Inform**: Data subjects should be adequately informed whenever personal data is processed.
2. **Control**: Data subjects should be provided agency over the processing of their personal data.
3. **Enforce**: A privacy policy compatible with legal requirements should be in place and should be enforced.
4. **Demonstrate**: Be able to demonstrate compliance with the privacy policy and any applicable legal requirements.

These strategies proposed by Hoepman not only inherit and develop the two engineering privacy approaches proposed by Spiekermann and Cranor, but also support the legal principles and PbD enforcement of the GDPR [13]. As a good combination point between legal principles and privacy enhancing technologies(PETs), these privacy design strategies have been widely accepted by recent work on privacy to fill the gap between legislation and engineering [5], [12], [36]. Therefore, we also adopt the eight privacy design strategies in this paper and study their relevant IoT layers (Section III-B) and enabling PETs (Section IV to Section VI) in the context of IoT privacy.

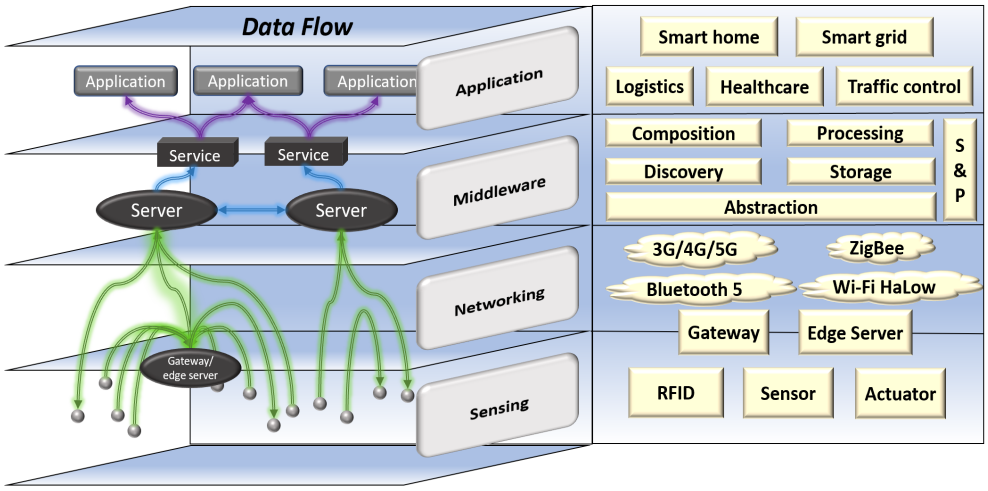


Fig. 1: IoT architecture and data flow [30], [31], [32], [33]

III. Privacy Protection in a Layered IoT Architecture

Several existing Internet-of-Things systems are designed using a layered architecture [30], [31], [32], [33]. In an IoT system, data is usually collected by end devices, transmitted through communication networks, processed by local/remote servers and finally provided to various applications. Thus, private data as it flows through multiple layers of the architecture stack, needs privacy protection at all layers. Here, implementing proper privacy design strategies based on the roles of the layers in the lifecycle of the data is important. Otherwise, techniques implemented at a specific layer may become either insufficient (privacy is breached at other layers) or redundant (privacy has been protected by techniques implemented at other layers). In this section, we introduce the reference IoT architecture adopted in this study and present the IoT privacy protection framework that shows how to integrate the privacy design strategies in the layered IoT architecture.

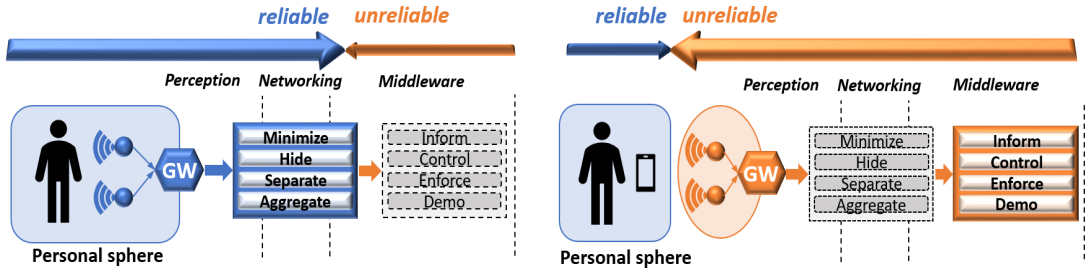
A. Reference IoT architecture

In general, the number of layers proposed for the architecture of IoT varies considerably. After reviewing a number of existing IoT architectures, we adopt a four-layer architecture as the reference IoT architecture in this paper, which consists of perception layer, networking layer, middleware layer and application layer. The adoption of the four-layer reference architecture in our study has two key benefits. First, the importance of each layer in the four-layer architecture has been recognized by most existing architectures as the four-layer architecture allows a comprehensive view of privacy in IoT. As shown in Table I, several existing architectures [33], [32], [31], [41], [42], [43] include all the four layers, either as separate layers or integrated layers. Second, as the four-layer architecture is the most fine-grained model among all the candidate architectures, it allows a detailed and fine-grained analysis of privacy protection at different layers and avoids possible lack of differentiation when the layers are not distinct as in [33], [32], [31], [41], [42].

TABLE I: IoT architecture comparison ($\surd$ contained as a separate layer, ○ merged with other layers, × not contained)

Source	Perception	Networking	Middleware	Application
IEEE [37]	$\surd$	$\surd$	×	$\surd$
M2M [33]	$\surd$	○	○	○
oneM2M [30]	×	$\surd$	$\surd$	$\surd$
CASAGRAS [32]	$\surd$	$\surd$	○	○
Cisco [31]	$\surd$	$\surd$	○	○
Soma *et cl.* [38]	$\surd$	×	$\surd$	$\surd$
Addo *et cl.* [39]	$\surd$	$\surd$	$\surd$	×
Funke *et cl.* [40]	$\surd$	$\surd$	$\surd$	×
Sun *et cl.* [41]	$\surd$	$\surd$	○	○
Perera *et cl.* [42]	$\surd$	$\surd$	○	○
Dabbagh *et cl.* [43]	$\surd$	$\surd$	$\surd$	$\surd$
Botta *et cl.* [44]	$\surd$	×	$\surd$	$\surd$

As the lowest layer of the architecture (Fig. 1), perception layer works as the base of entire Internet of Things. It bridges the gap between physical world and digital world by making innumerable physical entities identifiable (e.g., RFIDs [46]), perceptible (e.g., sensors [1]) and controllable (e.g., actuators [47]) to enable deep interaction between physical and digital worlds [37], [48]. The networking layer plays a pivotal role to link the perception layer and middleware layer so that sensed data and corresponding commands can be seamlessly transmitted between the two layers. Unlike the traditional Internet, the vast number of heterogeneous power-limited devices in the perception layer and the various application scenarios in the application layer create a vital need for communication technologies that support low energy consumption, low latency, high data rate and high capacity. Main techniques supporting IoT networking layer include ZigBee [49], Bluetooth 5 [50], Wi-Fi HaLow [51] and 5th generation mobile networks [52]. The middleware layer works as the 'brain' of IoT to process the numerous data received from lower layers. To cope with the interoperability of the heterogeneous



(a) Active collection (b) Passive collection

Fig. 2: IoT privacy protection framework [10], [13], [34], [45]

physical devices [53], [54], the *device abstraction* component semantically describes the resources with a consistent language such as the eXtensible Markup Language (XML), Resource Description Framework (RDF) or Web Ontology Language (OWL) [55], [56], [57]. Based on that, resources are made discoverable through the *resource discovery* component by using Semantic Annotations for WSDL and XML Schema (SAWSDL) [56] or simply key words [58]. Then, if needed, multiple resources can be composed through the *composition* component [1], [59] to enhance their functionality. After that, received data could be stored (*storage* component) in either cloud or databases and kept available to be queried. Different computational and analytical units can be combined to form the *processing* component. Here, security of data, namely their confidentiality, integrity, availability, and non-repudiation [60] need to be well protected. If data can make its owner either identified or identifiable, privacy enhancement technologies (PETs) are necessary to protect privacy so that privacy principles required by the laws can be satisfied. As the highest layer of the architecture, the application layer contains various IoT applications that have been widely studied in past literature [1], [4]. Depending on the scenarios that the private data is collected, different applications may encounter different privacy issues.

B. IoT privacy protection framework

In this section, we study the integration of privacy design strategies in the layered architecture by introducing an IoT privacy protection framework. From the viewpoint of the IoT architecture, data is collected by devices at perception layer and transmitted to middleware layer through networking layer, which makes data move away from the control of data subjects [8], [10]. We can apply the notion of personal sphere [10], [45] to assist interpretation. A personal sphere consists of a set of personal devices and a gateway, both trusted by data subjects. In some cases, gateway can be assisted by a more powerful trusted third party (TTP). Data collected by these personal devices has to be passed to the gateway and/or the TTP to be processed before being transmitted to the data controllers that data subjects distrust. Such a personal sphere is quite important to implement the four data-oriented privacy design strategies because it offers a reliable platform to make the raw data minimized, hidden, separated and aggregated. As pointed out by Spiekermann and Cranor [34], once sensitive information in data has been adequately constrained through PETs such as homomorphic encryption [61] and k-anonymization [62], the privacy-by-policy approaches, namely the four process-oriented strategies become less important. In IoT, such a personal sphere can be created when data is actively collected, as shown in Fig. 2(a). For example, smart appliances and home router form an indoor personal sphere while wearable devices and smartphones compose an outdoor personal sphere. The trusted local gateway and/or remote TTP is a critical element in the system, which allows data subjects to launch proper PETs to process data with the four data-oriented strategies.

Due to the invisibility of numerous IoT devices at perception layer, personal data may be sensed by untrusted devices outside the personal sphere and the data subjects may be completely unaware of the collection [37], [39], [63]. Such a passive collection makes data subjects lose control over their personal data at the earliest stage and provides no trusted platform to implement the four data-oriented privacy design strategies, as shown in Fig. 2(b). It is therefore the four process-oriented strategies that can play a more important role by promoting the power of data subjects when raw data is obtained by data controllers [13], [34]. Specifically, the *inform* strategy and *control* strategy enhance the interaction between data subjects and their data while the *enforce* strategy and *demonstrate* strategy force data controllers to comply with privacy policy and further require the compliance to be verifiable. As it is the remote data controllers that should offer proper PETs to support the four process-oriented strategies, these system-level strategies are primarily implemented at middleware layer, with the assistance of other layers. It is worth mentioning that we do not mean active collection only needs data-oriented strategies and passive collection only requires process-oriented strategies. In both cases, all the strategies are required to jointly work to support the legal principles. For example, although the single *minimize* strategy is hard to be fulfilled in the passive collection, its implementation can be enforced and verified by process-oriented strategies.

In the next three sections, we present and evaluate PETs implemented at the perception layer, networking layer and middleware layer respectively.

IV. Privacy at Perception Layer

We evaluate and compare the anonymization-based PETs and perturbation-based PETs that help to implement the *Min-*

imize and *Aggregate* strategies and present the encryption-based PETs that implements the *Hide* strategy. These PETs are primarily implemented at the perception layer (e.g., local personal gateway, trusted edge server), but they can also be implemented at the middleware layer using a trusted third party (TTP). It is worth noting that the *Separate* strategy is naturally achieved by local processing in the perception layer.

A. Anonymization and perturbation

Both anonymization [64] and perturbation [65] techniques can fulfill the *Minimize* and *Aggregate* strategies by reducing the released information and increasing the granularity. The main difference between them is that the results of the anonymization are generalized while the results of the perturbation are with noises. In this section, we evaluate the representative anonymization and perturbation privacy paradigms, namely the k-anonymity [62] and differential privacy [66] respectively, in terms of their practicability in IoT by analyzing their performance under the following IoT specific challenges [5]:

- Large data volume: The gateways may control thousands of sensors that collect massive data.
- Streaming data processing: In some real-time IoT applications (e.g., traffic control), data may be seamlessly collected to form streams.
- Lightweight computation: Since the gateways (e.g., router and phone) are still resource-constrained devices, algorithms are expected to have low complexity.
- Decentralized computation: In the IoT applications such as smart grid, the personal data may be collected by untrusted entities. Decentralization data aggregation may be employed under such scenarios.
- Composability: The privacy should still be guaranteed after the data uploaded to the middleware layer is combined with other data.
- Personalization: For most personal service IoT applications, each customer has different privacy understanding and requirements and there is a natural need for personalized solutions.

1) Anonymization: The traditional privacy-persevering data publication (PPDP) schemes typically involve four entities, namely data subject, data curator, data user and data attacker [67]. The data curator collects data from data subjects, processes the collected data and releases the privacy-preserving dataset to the data users. Usually, the collected data related to a data subject can be classified into four categories, namely explicit identifiers (e.g., names, SSN), quasi-identifiers (e.g., age, gender), sensitive attributes and non-sensitive attributes [62]. In IoT, unlike the traditional Internet that requires all the records to be typed in, the identifiers are usually input through RFID tags and cameras. For example, vehicles can be identified by E-ZPass [68] through RFID and individuals can be identified through RFID-enabled smart cards in shopping malls [69]. The sensitive and non-sensitive attributes are usually collected by sensors.

As a candidate PPDP approach, anonymization aims to cut off the connection between each record and its corresponding data subject so that the sensitive attributes cannot be linked with specific individuals [70]. Obviously, the explicit identifiers should be removed before publication for the privacy purpose. However, in 2000, Sweeney found that 87% of US citizens can be uniquely re-identified by combining three quasi-identifiers, namely [ZIP, gender, date of birth] [64]. This linking attack has motivated the researchers to devise stronger anonymization paradigms including k-anonymity [62], l-diversity [71] and t-closeness [72], where k-anonymity [62] requires each quasi-identifier group to appear at least k times in the dataset. We next discuss the use of anonymization in the context of IoT:

Large volume: The performance of anonymization algorithms may be affected by the dimensions of both rows and columns in the table, so the anonymization scheme is expected to be scalable for datasets with millions of records and multi-dimensional attributes. For the former, spatial indexing has been proved to be a good solution to handle numerous records in a dataset [73], [74]. One attribute can be efficiently k-anonymized through B^+ tree indexing and the R^+ tree indexing can be implemented to effectively generate non-overlapping partitions for tables with 100 million records and nine attributes [74]. However, as analyzed by [75], the k-anonymity algorithms may work well for tables with a small number of attributes (e.g., 10) but not the ones with a large number of attributes (e.g., 50). The increasing number of attributes makes the number of combinations of dimensions exponentially increased and results in unacceptable information loss. Therefore, how to enhance the utility of k-anonymized datasets with a large number of attributes is still an open issue for future research. An anonymization method for the sparse high-dimensional binary dataset with low information loss was proposed in [76], but there were no effective schemes for non-binary datasets.

Streaming data: There have been several strategies to anonymize data streams [77], [78]. In CASTLE [77], a set of clusters of tuples are maintained and each incoming tuple in a stream is grouped into a cluster and generalized to the same level of other tuples in the cluster. Each tuple maintains a delay constraint δ and must be sent out before the deadline to make the processing real-time. At the end of δ, if the cluster containing that tuple has at least k members, all the tuples within it can be released. Otherwise, a cluster satisfying the k requirement can be generated through a merge and split technique for the tuple and the information loss during the process can be minimized. In SKY [78], a top-down specialization tree is maintained and each incoming tuple is mapped to one node in the tree based on its attributes. Each node can be a work node or a candidate node depending on whether there have been at least k tuples generalized and output from it. If the incoming node is mapped to a work node, it can be directly generalized and released. Otherwise, it has to wait for other arriving tuples at the node during the time δ or be generalized and released through the parent node at the end of δ.

Lightweight: It has been proved that the optimal k-anonymity aiming to anonymize a table with minimum suppressed cells is NP-hard even when the attribute values are ternary [79], [80], [81]. The complexity of approximate algorithms for

k-Anonymity has been reduced from $O(k \log k)$ [79] to $O(k)$ [80] and later to $O(\log k)$ [81].

Collaboration: Anonymization techniques can be implemented in a distributed manner. That is, multiple entities can locally anonymize their own table to make the integrated table k-anonymous without revealing any additional information during the process. Several SMC protocols have been proposed to solve this problem [82], [83]. In [82], a top-down specialization scheme was proposed to support joint anonymization between two parties. Specifically, the two parties first generalize their local table to the root. Then, in each iteration, they find the local specialization maximizing the ratio between information gain and privacy loss (IGPL) over the local table. The party with a higher IGPL wins the game in this iteration, applies its local specialization over its local table and then instructs the grouping in the table of the other party. For the same objective, a scheme based on cryptography was proposed in [83]. Each party locally generalizes the local table and then jointly determines whether the integrated table is k-anonymous. If not, each party then generalizes its local table to the next layer and repeats the two steps.

Composability: As shown in [5], the k-anonymity does not offer composability. That is, two k-anonymous datasets cannot guarantee their joint dataset is k'-anonymous ($k' > 1$). Because of this, the integration of multiple k-anonymous datasets in the middleware layer can be a significant challenge.

Personalization: Most anonymization algorithms assume that all the record owners have same privacy preference. Therefore, less-anonymization can put privacy in risk but over-anonymization increases the information loss. To solve this, Xiao et al. [84] organize the sensitive attributes in a top-down taxonomy tree and allow each record owner to indicate a guarding node. That is, the sensitive attribute of a specific record owner should be generalized to at least the guarding node in the taxonomy tree and the adversary has little opportunity to link the record owner with the child nodes of the guarding node that carry fine-grained information. Their algorithm first runs common k-anonymity algorithms over the quasi-identifiers and then generalizes the sensitive attribute through the taxonomy tree based on the claimed guarding nodes. Recently, Xu et al. [85] argued that the generalization of sensitive attributes results in information loss and they allow the record owners to claim the expected value of k. Their algorithm first achieves k_{min}-anonymity over the entire dataset, where k_{min} is the minimum expected k value, namely the most strict privacy requirement. Then, based on the data structure called d-dimensional quasi-attribute generalization lattice, some quasi-attributes can be merged to match the lower values of k expected by some record owners.

2) Differential privacy: Differential privacy is a classical privacy definition [66] that makes very conservative assumptions about the adversary's background knowledge and bounds the allowable error in a quantified manner. In general, differential privacy is designed to protect a single individual's privacy by considering adjacent data sets which differ only in one record. Before presenting the formal definition of ϵ-differential privacy, we first define the notion of adjacent datasets in the context of differential privacy. A data set D can be considered as a subset of records from the universe U, represented by $D \in \mathbb{N}^{|U|}$, where $\mathbb{N}$ stands for the non-negative set and D_i is the number of element i in $\mathbb{N}$. For example, if $U = \{a, b, c\}$, $D = \{a, b, c\}$ can be represented as $\{1, 1, 1\}$ as it contains each element of U once. Similarly, $D' = \{a, c\}$ can be represented as $\{1, 0, 1\}$ as it does not contain b. Based on this representation, it is appropriate to use l_1 distance (Manhattan distance) to measure the distance between data sets.

DEFINITION 1 (DATA SET DISTANCE)**.** *The l_1 distance between two data sets D_1 and D_2 is defined as $||D_1 - D_2||_1$, which is calculated by:*

$$||D_1 - D_2||_1 = \sum_{i=1}^{|U|} |D_{1_i} - D_{2_i}|$$

The manhattan distance between the datasets leads us the notion of adjacent data sets as follows.

DEFINITION 2 (ADJACENT DATA SET)**.** *Two data sets D_1, D_2 are adjacent data sets of each other if $||D_1 - D_2||_1 = 1$.*

Based on the notion of adjacent datasets defined above, differential privacy can be defined formally as follows. In general, ϵ-differential privacy is designed to protect the privacy between adjacent data sets which differ only in one record.

DEFINITION 3 (DIFFERENTIAL PRIVACY [66])**.** *A randomized algorithm $\mathcal{A}$ guarantees ϵ-differential privacy if for all adjacent datasets D_1 and D_2 differing by at most one record, and for all possible results $\mathcal{S} \subseteq Range(\mathcal{A})$,*

$$Pr[\mathcal{A}(D_1) = \mathcal{S}] \leq e^{\epsilon} \times Pr[\mathcal{A}(D_2) = \mathcal{S}]$$

where the probability space is over the randomness of $\mathcal{A}$.

Many randomized algorithms have been proposed to guarantee differential privacy, such as the Laplace Mechanism[66], the Gaussian Mechanism[86] and the Exponential Mechanism[87]. Given a data set D, a function f and the budget ϵ, the Laplace Mechanism first calculates the actual $f(D)$ and then perturbs this true answer by adding a noise[66]. The noise is calculated based on a Laplace random variable, with the variance $\lambda = \triangle f/\epsilon$, where $\triangle f$ is the l_1 sensitivity. We next analyze differential privacy in terms of the challenges in the context of IoT:

Large volume: The large volume of data is naturally not a problem for differential privacy as the perturbation is usually implemented over the statistical value of the collected data.

Streaming data: There have been many works on applying differential privacy over streaming data since 2010 [88], [89]. The data stream was assumed to be a bitstream, where each bit can be either 1 or 0 representing if an event was happening or not at each timestamp. Mechanisms were proposed to protect either the event-level or user-level differential privacy, depending on whether a single event or all the events related to a single user can be hidden by the injected noise. The early works focused on event-level privacy. In [88], a counter was set to report the accumulated 1s in the data stream at each timestamp and each update value can be added with a $Lap(\frac{1}{\epsilon})$ noise to guarantee the differential privacy. Furthermore, for a sparse stream with few 1s, an update can be set to happen

TABLE II: Evaluation of k-anonymization and differential privacy (○ Good ◐ Not enough ● Poor)

PET	Large volume	Streaming data	Lightweight	Collaboration	Composability	Personalization
k-anonymization	◐	○	○	○	●	◐
Differential privacy	○	○	◐	○	○	◐

only after the number of 1s has been accumulated over a threshold. Later in [90], the noise error was reduced through using a binary tree data structure. Specifically, the nodes in the binary tree, except the leaf nodes, represent the sums of sections of consecutive bits in the stream and the Laplace noises were added to these nodes, instead of the leaf nodes. This scheme can effectively reduce the noise error from $O(T)$ to $O((\log T)^{1.5})$, where T denotes the number of timestamps, namely the length of the stream. In [91], the user-level privacy was supported and the noise error in this work was suppressed through sampling.

Lightweight: The complexity of differential privacy algorithms is quite variable on a case-by-case manner. If both the sensitivity and budget allocation are fixed, the complexity can be very low, as only one value is required to be sampled from a random distribution with fixed variance. However, in the cases that the sensitivity or budget allocation has to be calculated on the fly, the complexity will increase.

Collaboration: Differential privacy for data aggregation is usually guaranteed by noises added through Laplace mechanism [66]. A simple solution for this is to make the data aggregator directly aggregate the raw data received from data subjects and then add noise to it. However, in some scenarios such as smart metering, the aggregator (electricity supplier) may be untrusted [92] and may require the data subjects (smart meters) to locally add noise to perturb its raw data and then send the perturbed data to the aggregator so that the raw data is protected from the aggregator and the aggregated noise automatically satisfies the Laplace Mechanism. This distributed implementation of Laplace Mechanism, also known as Distributed Perturbation Laplace Algorithm (DLPA), has recently received attention from privacy researchers. The base of DLPA is the infinite divisibility feature of Laplace distribution [93] that allows the noise sampled from Laplace distribution (central noise) to be the sum of n other random variables (local noises). The local noise can still follow the Laplace distribution [94]. However, since a Laplace distributed random variable can be simulated by two gamma distributed random variables and four normal distributed random variables, the local noise can also follow the gamma distribution [92] or Gaussian distribution [95]. In [94], the three schemes were compared and the Laplace distributed local noise was shown to be more efficient in terms of local noise generation.

Composability: Differential privacy offers strong composability:

Theorem 1 (COMPOSITION THEOREM [86])**.** *Let $\mathcal{A}_i$ be ϵ_i-differential private algorithms applied to independent datasets D_i for $i \in [1, k]$. Then their combination $\mathcal{A}_{\sum_{i=1}^{k}}$ is $max(\epsilon_i)$-differential private.*

In the middleware layer, multiple independent differentially private outputs can be combined and their integration still satisfies differential privacy. Differential privacy also satisfies the post-processing theorem, which further enhances its flexibility in the middleware layer.

Theorem 2 (POST-PROCESSING [86])**.** *Let $\mathcal{A}$ be a ϵ-differentially private algorithm and g be an arbitrary function. Then $g(\mathcal{A})$ is also ϵ-differentially private.*

Personalization: In traditional differential privacy, the parameter ϵ is usually set globally for all the record owners. Recently, several works try to make it personalized. In [96], two solutions were proposed, based on sampling and Exponential Mechanism respectively. The first approach non-uniformly samples the records from the dataset with the inclusion probabilities related to the preferred privacy preferences (values of ϵ). For each record, if the expected ϵ is smaller than a threshold t, it may only be selected with a probability related to the ϵ. Otherwise, the record will be selected. Then, any t-differentially private mechanism can be applied to the sampled dataset. Their second approach is inspired by the Exponential Mechanism. Unlike the traditional Exponential Mechanism, to take personalization into account, the probability of each possible output values is computed based on the personalized privacy preferences (values of ϵ).

3) Anonymization vs. Differential privacy: To sum up, as shown in Table II, both the techniques have similar features in terms of their support for streaming data, collaboration and personalization. Anonymization techniques are difficult to scale for datasets with many attributes while the complexity of differential privacy algorithms varies case by case. It is the composability feature that makes differential privacy a clear winner. Due to lack of composability, the operability and utility of the data protected by the k-anonymization paradigm are significantly constrained in the middleware layer.

B. Encryption

Encryption techniques are not only the fundamental building block of security, but also the foundation of a large number of PETs in privacy. With respect to the eight privacy design strategies, encryption is the most direct supporter of the *Hide* strategy, which also satisfies the 'security safeguards' requirement of privacy laws. Therefore, in terms of IoT privacy, the role of encryption is twofold. On one hand, the commonly used cryptographic primitives, such as AES [97] and RSA [98], protect the security of every IoT layer so that the adversaries are prevented from easily compromising the confidentiality and integrity of data in IoT devices. From this perspective, the personal data is confined to a safe zone without being disclosed to unknown parties, thus also protecting the privacy of the data subject as the control over the data is enhanced.

On the other hand, in IoT, the middleware may not be trusted or trustworthy but it is an indispensable stakeholder in most IoT applications. Hence, PETs such as homomorphic encryption [61], searchable encryption [99] and SMC [100] are required to make the middleware work without accessing the private information. Here, lightweight cryptography that can support encryption over devices with low capacity becomes a critical element in protecting IoT privacy. In this section, to comprehensively review the current state of work in this area, we first go through the real capacity of various types of IoT devices in the perception layer and evaluate the implementation of commonly used cryptographic primitives over them to see when and where lightweight cryptography is required. Then, we review the candidate lightweight solutions in each area of cryptography and present the NIST General Design Considerations [101]. Finally, we discuss the PETs aiming to blind the middleware and their performance over IoT devices.

The capacity of IoT devices: The types of IoT edge devices in the perception layer range from resource-rich devices such as computers and smartphones to resource-constrained devices such as embedded systems, RFID and sensors. For the resource-rich devices, the traditional cryptographic primitives work well for the encryption tasks. Thus, the lightweight cryptography techniques are mainly required by the resource-constrained devices that can not support traditional cryptographic primitives. This also requires the resource-rich devices in the middleware layer to adopt them in order to decrypt the data encrypted using lightweight cryptography techniques. Most IoT embedded systems and intelligent systems are enabled by the 8-bit, 16-bit or 32-bit microcontrollers (MCUs) with highly restricted random-access memory (RAM) as low as 64 bytes RAM (e.g., NXP RS08KA, $0.399) [102]. The RFID and sensor devices are usually more cost-sensitive and they employ the use of application specific integrated circuit (ASIC) [103]. Therefore, in hardware, the price of these devices is proportional to the area of ASIC in silicon, measured by the gate equivalents (GE), namely the ratio between the area of ASIC and the area of a two-input NAND gate [104]. The implementation of lightweight cryptography techniques over such devices has to meet several stringent conditions, including under 2000 GE to achieve low-cost, under 50 cycles for obtaining low-latency and less than $10\frac{\mu W}{MHz}$ average power usage for meeting low-energy requirements [103].

Traditional cryptographic primitives over constrained devices: Most traditional commonly-used cryptographic primitives face severe challenges in the constrained environment. The AES-128 [97] may be the most suitable lightweight block cipher because of its low number of rounds and small key size. In [2], AES-128 was tested over several MCUs and smart cards and achieved $1.58ms$ execution time and 0.6kB RAM consumption over the MSP microcontrollers. The results show that AES works well for most MCUs, but not the ones with ultra-low RAM (e.g., NXP RS08KA). In terms of hash functions, the SHA-2 is acceptable to implement the cryptographic schemes requiring a few hash functions over the MSP microcontrollers with tens to hundreds of milliseconds execution time and 0.1kB RAM. However, as illustrated by Ideguchi et al. [105], the SHA-3 candidates cannot be supported by the low-cost 8-bit microcontrollers with 64 byte RAM. In the NIST competition, the lowest number of GE required by the SHA-3 is still 9200 [106]. Also, both the RSA [98] for asymmetric encryption and elliptic curve point multiplication for ECDH and ECDSA schemes were found to be too high-cost for even the MSP microcontrollers [2].

Attribute-Based Encryption in IoT: Attribute-Based Encryption (ABE) [107] is a promising mechanism to implement fine-grained access control over encrypted data. With ABE, an access policy can be enforced during data encryption, which only allows authorized users with the desired attributes (e.g., age, gender) to decrypt the data. Depending on whether the access policy is associated with the key or ciphertext, Key-Policy ABE (KP-ABE) [108] and Ciphertext-Policy ABE (CP-ABE) [109] were proposed, respectively. Although ABE looks like the desired approach to secure data communication and storage in IoT with flexible access control, its implementation in IoT may encounter three main challenges. First, current IoT applications only need IoT devices to encrypt data using public keys and hence, key management may not be a significant issue. However, future autonomous IoT devices would require direct device-to-device communication with each other requiring different secret keys from the attribute authority (AA) based on their attributes to decrypt data. In such cases, the AA may become a bottleneck for issuing secret keys and we will need techniques to distribute secret keys in a scalable and efficient manner. Potential solutions for this include Hierarchical ABE (HABE) [110] and decentralizing multi-authority ABE (DMA-ABE) [111]. In short, the HABE scheme manages the workflow in a hierarchical structure with each domain authority serving a set of domain users, whereas the DMA-ABE scheme decentralizes the single centralized AA to multiple AAs. Second, when an access policy needs to be updated, due to the limited storage space of IoT devices, the re-encryption of the data based on the new policy is hard to be operated locally. A solution for this has been proposed by Huang *et al.* [112], which designs a set of policy updating algorithms that allow the re-encryption to be operated at untrusted remote servers without breaching the privacy of the encrypted data. The third and perhaps the greatest challenge is the issue of limited resources in IoT devices. It has been demonstrated that most classical CP-ABE schemes can hardly fit the smartphone devices and IoT devices such as Intel Edison board and Raspberry Pi [113], [114], [115]. To solve this, the most common approach is to outsource the most consuming operations of ABE to powerful nodes in the network [116], [117]. In case that such powerful nodes are not available, Yao *et al.* [118] proposed a lightweight no-pairing ECC-based ABE scheme to reduce the power consumption.

Lightweight cryptographic candidates: As can be seen, most traditional cryptographic primitives are not applicable over resource-constrained devices. Hence, IoT privacy creates a critical need for lightweight cryptographic solutions. A non-exhaustive list of lightweight cryptographic candidates can be found in [119]. The design of lightweight block ciphers, based on the classification in [119], consists of the Substitution-Permutation Networks (SPN) family and Feistel Networks family. The SPN-based schemes usually apply the S-boxes

and P-boxes to perform confusion and diffusion respectively and can be roughly divided into three categories, namely the AES-like schemes (e.g., KLEIN [120]), schemes with Bit-Sliced S-Boxes (e.g., PRIDE [121]) and other schemes (e.g., PRESENT [122]). The schemes based on the Feistel Networks split the input block into two sides, permute one with the other and then swap them. They can be designed to only use modular Addition, Rotation and XOR (e.g., RC5 [123]) or not (e.g., DESLX [124]). These lightweight schemes usually apply smaller block sizes lower than 128 bits as AES or simpler rounds without S-boxes or with smaller S-boxes to reduce the resource requirements [101]. The lightweight hash functions are designed based on either the Merkle-Damgård or P-Sponge and T-Sponge. The existing lightweight hash functions such as PHOTON [125] and SPONGENT [126] have already been able to achieve under 2000 GE with $0.18\mu m$ technology for 128 digest size. In terms of lightweight stream ciphers, the Grain [127], MICKEY [128] and Trivium [129] have stood out since 2008. In addition, recently, the NIST published its report on lightweight cryptography [101] and recommended the General Design Considerations for the future design:

1. **Security strength**: The security strength should be at least 112 bits.
2. **Flexibility**: Algorithms should be executable over an assortment of platforms and should be configurable on a single platform.
3. **Low overhead for multiple functions**: Multiple functions (such as encryption and decryption) should share the same logic.
4. **Ciphertext expansion**: The size of the ciphertext should not be significantly longer than the plaintext.
5. **Side channel and fault attacks**: Algorithms should be resilient to the side channel and fault attacks.
6. **Limits on the number of plaintext-ciphertext pairs**: The number of plaintext/ciphertext pairs processed should be limited by an upper bound.
7. **Related-key attacks**: Algorithms should be resilient to the related-key attacks, where the relationship between multiple unknown keys is used by the adversary.

Middleware-blinding PETs in IoT: The homomorphic encryption [61], as the most fundamental building block of the Middleware-blinding PETs, is a suite of cryptographic techniques that enable the decrypted results of computations over the ciphertext to match the results of computation over the plaintext. Its characteristics make it the best solution for outsourcing private data to untrusted parties to get their service without compromising privacy, which refers to blinding the middleware in IoT domain. Homomorphic encryption was proposed as early as 1978 but it was not until the year 2009 that the first plausible solution of the fully homomorphic encryption was proposed by Craig Gentry [61]. Unlike the partially homomorphic cryptosystems such as the ones based on Paillier cryptosystem [130] that support a small number of operations, the fully homomorphic encryption can enable both addition and multiplication operations over ciphertexts and therefore arbitrary computations. However, although the efficiency of the fully homomorphic encryption has been significantly improved, it is still too time-consuming for most applications. Therefore, in many cases, the partially homomorphic cryptosystems are still the preferred solution. IoT can benefit a lot from the homomorphic encryption [131] as well as the secure multi-party computation (SMC) schemes in the context of service discovery, data retrieval, data sharing and data outsourcing. Although most of the applications interact closely with the middleware layer, the encryption of private data is usually implemented in the perception layer. In [2], the Paillier's partially homomorphic scheme was tested and the results showed that the scheme is still heavy for the resource-constrained devices.

V. Privacy at Networking Layer

In this section, we discuss the secure communication and anonymous communication in the networking layer that support the *Hide* and *Minimize* strategies respectively.

A. Secure communication

In the traditional Internet with TCP/IP stack, the communication is usually secured by either IPsec [132] in the network layer or TLS [133] in the transport layer. In the context of IoT, due to numerous devices with constrained power, the protocol stack has to be adapted to support the transmission of IPv6 over IEEE 802.15.4 PHY and MAC, which is enabled by the adoption of 6LoWPAN [134] as an adaptation layer between them. A reference IoT protocol stack is shown in Fig. 3, which is mainly based on the IETF LLN protocol stack [135]. Above the network layer, TCP and UDP in the transport layer support different IoT application layer protocols, such as Message Queue Telemetry Transport (MQTT) [136] and Constrained Application Protocol (CoAP) [137], respectively. In terms of security, as pointed out by RFC 4944 [138] and other literature [139], [140], the AES-based security modes provided by the IEEE 802.15.4 that can support confidentiality, data authenticity and integrity, have some shortcomings. That is, the IEEE 802.15.4 only provides hop-by-hop security that requires all nodes in the path to be trusted without host authentication and key management. It may be acceptable for isolated WSNs, but not for the Internet-integrated WSNs when the messages have to travel over an IP network. Therefore, security mechanisms are required to be implemented in the higher layers to provide end-to-end security. Like the traditional Internet, the potential options include the IPsec in the network layer and the TLS/DTLS in the transport layer, where TLS and Datagram TLS (DTLS) [141] support TCP and UDP, respectively. The TLS/DTLS solution is the default security option of most common IoT application protocols. For example, the MQTT Version 3.1.1 [136] claimed that it should be the implementer's responsibility to handle security issues and then recommended the TLS and registered TSP port 8883 for MQTT TLS communication. In contrast, the CoAP is secured by DTLS as it transmits messages over the unreliable but simpler UDP [137]. The various security modes allow the devices to have either a list of pre-shared symmetric keys or a pair of asymmetric keys with or without

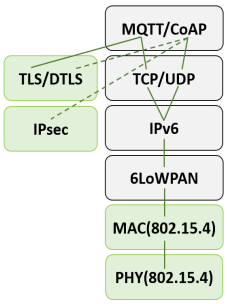


Fig. 3: Reference IoT protocol stack [132], [134], [133], [135]

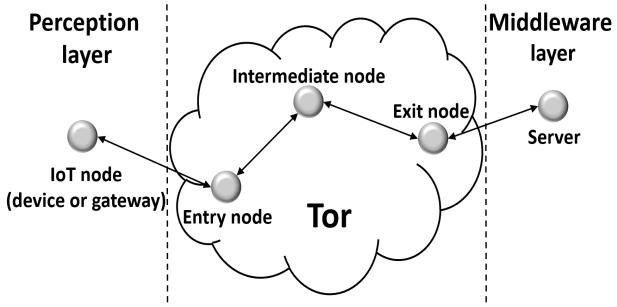


Fig. 4: Tor over IoT [12], [148]

an X.509 certificate. In addition to DTLS, the CoRE working group also proposed a draft for using CoAP with IPsec [142]. The adoption of IPsec can make use of the built-in link-layer encryption hardware and perform transparently towards the application layer. However, due to its well-known issues with using the firewalls and Network Address Translation (NAT), the IPsec is not always available. In addition, the configuration and management of IPsec in IoT is very difficult due to the huge number of heterogeneous devices [143].

Based on the IETF protocol stack, there are some other IoT protocol stacks proposed by other standardization bodies and industry alliances. We briefly review some representatives among them. The Thread stack [144] adopts 6LoWPAN to support IPv6 and leverages DTLS to secure UDP. The Thread stack has been widely adopted for connecting home devices and applications. The IPSO Alliance [145] argued that using standardized protocols (e.g., IETF stack) may fail to ensure interoperability at the application layer. They proposed the IPSO Smart Objects, an object model that provides high-level interoperability between applications and devices. The core idea is to leverage the open Mobile Alliance Lightweight Specification (OMA LWM2M) on top of CoAP to enable device management operations such as bootstrapping and firmware updates. Again, DTLS is in charge of security. The Industrial Internet of Things (IIoT) was proposed by the Industrial Internet Consortium (IIC), with the aim to connect industrial objects to enterprise systems and business processes [146]. Its reference architecture adopts DDSI-RTPS [147]/CoAP for UDP and MQTT/HTTP for TCP, respectively. Therefore, its security requires both TLS and DTLS.

B. Anonymous communication

The end-to-end security provided by either IPsec or TLS/DTLS can only hide the content of the messages, but not the meta-data, such as the identity (e.g., IP) of the two sides or the time, frequency and amount of the communications. Therefore, PETs enabling anonymous communication are required to handle the privacy problem due to the disclosure of meta-data, especially the identity of the initiator of the communication. For example, when health data or smart home data has to be sent to the middleware layer to get some service, it is better to make the data subject anonymous so that the personal health condition or living habits cannot be easily linked to the data subject. Such an objective can be achieved through the implementation of the anonymization and perturbation mechanisms in the perception layer, but the anonymous communication makes it also possible to handle in the networking layer.

The communication can be anonymized through the Proxy, the Virtual Private Network (VPN) and the onion router (Tor) [12], [148]. Among them, Tor is considered an important anonymous communication PET because of its strong attack resilience [149]. We show a potential Tor-based anonymous communication framework in Fig. 4. An IoT node, either a device or a gateway, wants to communicate with the middleware to get service without revealing its identity (e.g., IP address). For this purpose, instead of directly communicating with the middleware, the IoT node can first connect with the Tor network to anonymize itself. The Tor network is a distributed network with thousands of volunteers all around the world performing as the onion routers [150]. Its scale, as monitored by the torstatus website, is around 7000-8000 nodes in 2018 [151]. To process the request of the IoT node, Tor will build a path (circuit) formed by one entry node, one or multiple intermediate nodes and one exit node. The raw package sent by the IoT node is then encrypted by the public keys of the nodes on the path one by one, from the entry node to the exit node, forming a layered structure, just like an onion. Each node on the path, on receiving a package from its predecessor, should decrypt one layer of the package with its private key, learn the IP of its successor and transmit the decrypted package to the successor. Each node on the path only knows the IP of its predecessor and successor and hence, the IP address of the IoT node is only revealed to the entry node and the middleware only knows the IP address of the exit node.

The implementation of Tor over smart home was evaluated in [149] in which, Tail, a subproject of Tor, was set up to be the central smart home gateway passed by all the outgoing data packages generated by the appliances. The results showed that Tor works well for multimedia transmission (smart TV) but not the voice-over-Internet protocol application such as Skype, due to the short time-to-live duration of UDP packets. This work demonstrated the practicability of Tor in IoT. However, several key challenges still need to be addressed. First, the access point to the Tor network should be designed to make it available to the capacity-constrained IoT devices. Second, as Tor does not support UDP, for the devices unable to encapsulate the UDP into TCP packets, mechanisms are required to enable UDP transmission over Tor. Third, the affordability of the Tor

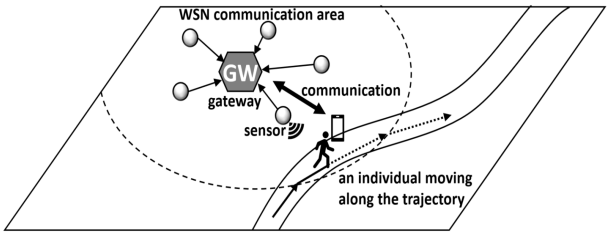


Fig. 5: Notification in the WSN

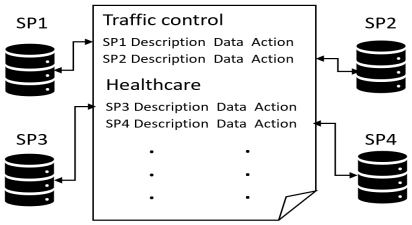


Fig. 6: Central control platform

network in terms of the massive data generated by the billions of IoT nodes should be evaluated.

VI. Privacy at Middleware Layer

In this section, we present the interaction-enhancing PETs fulfilling *Inform* and *Control* strategies and discuss the compliance-enhancing PETs enabling *Enforce* and *Demonstrate* strategies. We evaluate existing middlewares on their support for these four process-oriented strategies.

A. Interaction-enhancing techniques

The main objective of interaction-enhancing techniques is to break the isolation between data subjects and their data so that data subjects can track the status of their data (*Inform* strategy) and also remotely control their data (*Control* strategy). The GDPR [28] requires data subjects to get notification both before and after the data collection. Before the data collection, in addition to the data collection notification itself, data subjects should also be notified more information such as identity and contact details of data collector and purpose of the processing (Article 13). After the data collection, *Inform* strategy can be combined with *Control* strategy to assist data subjects to safeguard their rights, such as the right of access (Article 14), right to rectification or erasure of personal data and restriction of processing (Article 15) and right to know the personal data breach (Article 30).

In the traditional Internet, *Inform* strategy is easy to be implemented because it is the data subjects who actively determine whether to click the link to enter a website. The PETs such as the P3P [152] aim to assist the end users with little privacy knowledge or with no patience to quickly understand the privacy condition of the visiting websites in an automatic and usable manner [34]. Specifically, the privacy policies provided by most websites are both long and obscure with dense legalese, which makes the visitors hard to understand how their private data such as browsing history is handled. The P3P solved this problem by providing both a computer-readable format for websites to standardize the privacy policies and a protocol for the web browsers to understand the privacy policies and automatically process them based on the pre-determined privacy preference. Unfortunately, things become harder in IoT. Unlike the traditional Internet where the end users can easily interact with the websites through static web browsers, it is essential to figure out how to effectively build the communication between data subjects and data controllers in dynamic IoT scenarios to enable *Inform* and *Control* strategies. To build such a communication for active collection is not hard. An example is the privacy coach [153], a phone application to help end users decide whether to buy products with RFID tags by actively reading RFID tags to learn corresponding privacy policies. However, to do the same thing for passive collection is more challenging. Consider the example in Fig. 5 where an individual quickly passes a WSN area, the gateway has to quickly and actively get connected with the personal phone to notify the data collection, get the consent and leave information for future notifications. All these should be completed within a short period of time before the communication is disconnected.

For *Control* strategy, the main challenge is not how to technically implement the actions such as revision and deletion but how to design a centralized platform to simplify the control of data subjects when there are multiple data controllers. In active collection, each data subject can actively upload private data for different data controllers to a common personal space in cloud to simplify the tracking and control of their data [154], [45]. In passive collection, as personal data of a data subject may be passively uploaded by data controllers to different storage places, a centralized user control platform is required, such as the one in Fig. 6. A data subject, after login, should be able to check the list of his/her personal data collected by different data controllers. Each data controller, after collecting the data, should report the collection to this central platform, link its database to the platform and provide APIs to allow the authorized data subjects to control their data. The format of a report should contain identity of the data collector, description of collection purpose, collected data and a list of possible actions that can be made by data subjects. Then, data subjects can remotely revise or delete their data.

B. Compliance-enhancing techniques

The goal of compliance-enhancing techniques is to enforce and demonstrate compliance with the privacy policy. The *Enforce* and *Demonstrate* strategies are highly related. First, the *Enforce* strategy requires a privacy policy compatible with laws to be in place and a set of proper PETs to technically enforce it in engineering so that a data controller has the ability to comply with privacy laws. We require the *Demonstrate* strategy here to enforce it so that the data controllers can technically prove their compliance.

As the first step, a privacy policy should be in place to guide the processing of private data. By considering personalization, this privacy policy can be replaced by a privacy preference in many cases to also reflect personal privacy demands. Such

a privacy preference should be in place during the entire lifecycle of the personal data [155]. That is, even if the personal data is disseminated from the initial data controller to the others, the privacy preference of the original data subject should be simultaneously transmitted along with the data. In other words, the privacy preference should be stuck to the corresponding data in the complicated middleware layer, which can be supported by the PET. Such a scheme was named sticky policy [156]. The privacy model of sticky policy requires data to be first encrypted by data subjects. Then, the encrypted data and the sticky policy are sent to the data controller while the decryption key is sent to a Trusted Third Party (TTP). Any party who wants to decrypt the data, including the initial data controller and later ones, should submit a request to the TTP with the sticky policy and credentials. The TTP will then check the integrity and trustworthiness of them to decide whether the decrypted key can be given. During the whole process, data subjects can join or check the decision making through the TTP. To sum up, the privacy preference must also flow along with the data and its existence should be enforced and monitored by the TTP. A similar approach was proposed in [45], where the data is encrypted by the data subjects at their gateways and attached with semantic data handling annotations as the privacy preference.

After the privacy preference is in place, the PETs that can fulfill the privacy preference are required. To make it automatic, the sticky policy is recommended to be used as machine-readable semantic annotations that can be parsed by the middleware to configure the corresponding PETs. The implementation of the policy can be supported by access control mechanisms [5]. In terms of purpose limitation, the mechanism proposed in [157] require the data requesters to declare their purpose of usage and the range of required data so that the current data controller is able to compare the declaration with the sticky annotations to make decisions. Another choice is the Hippocratic database [158]. As a database designed to fulfill the Fair Information Practices [17] and especially the purpose limitation, the Hippocratic database requires the queries to be tagged with a purpose and only access the columns and tuples matching the purpose.

Finally, the most common solution to verify the compliance is the audit mechanism. That is, any interaction with private data should either be pre-checked or logged for later inspection. An example of pre-checking is the sticky policy [156], where data requesters must first submit the sticky policy and credentials to the TTP and accept the inspection of TTP about their environment. An audit approach using the log was proposed in [45], where personal data is encrypted in personal sphere by a gateway and then stored in a cloud platform. The cloud platform offers a database abstraction layer that can log every access of a data controller to the data with detailed information such as the access time and purpose. Next, the data subject should verify that the usage of the data complies with the privacy preference. However, even with the log information and available source code of the service, data subjects may not have the expertise to audit it. Therefore, a trusted auditor is deployed to verify the data usage in the service implementation by checking the source code.

TABLE III: Evaluation of middlewares ($\surd$ supported with PETs, $\bigcirc$ mentioned without details, $\times$ not mentioned)

Middleware	year	inform	control	enforce	demonstrate
COUGAR [159]	2001	$\times$	$\times$	$\times$	$\times$
Impala [160]	2003	$\times$	$\times$	$\times$	$\times$
IrisNet [161]	2003	$\times$	$\times$	$\bigcirc$	$\times$
Adaptive [162]	2004	$\times$	$\times$	$\times$	$\times$
TinyLIME [163]	2005	$\times$	$\times$	$\times$	$\times$
Melete [164]	2006	$\times$	$\times$	$\times$	$\times$
SENSEI [165]	2010	$\times$	$\times$	$\times$	$\times$
UbiROAD [166]	2010	$\times$	$\times$	$\times$	$\times$
GSN [55]	2006	$\times$	$\times$	$\times$	$\times$
Xively [167]	2007	$\times$	$\surd$	$\bigcirc$	$\times$
Paraimpu [168]	2012	$\times$	$\surd$	$\bigcirc$	$\times$
Webinos [154]	2012	$\surd$	$\surd$	$\surd$	$\times$
OpenIoT [169]	2013	$\times$	$\times$	$\times$	$\times$
Google Fit [170]	2014	$\surd$	$\times$	$\times$	$\times$
Calvin [171]	2015	$\times$	$\times$	$\times$	$\times$
Node-RED [58]	2015	$\times$	$\times$	$\times$	$\times$
OpenHAB [172]	2010	$\surd$	$\surd$	$\surd$	$\surd$
AllJoyn [173]	2013	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$
NOS [174]	2016	$\surd$	$\surd$	$\surd$	$\times$

C. Evaluation of existing middlewares

Currently, only a few middlewares support privacy protection. Among 61 middlewares reviewed by a recent survey [175], only eight of them were labeled to support privacy. We evaluate their performance over the *inform*, *control*, *enforce* and *demonstrate* strategies. As can be seen in the first part of Table III, among the eight middlewares, only the IrisNet mentioned the importance of the enforcement of privacy policies. In the second part of Table III, we present middlewares reviewed by another recent survey [59]. In Xively, permission is not required for data collection and sharing, but users are allowed to review, update or change their data in the account, which satisfies the *control* strategy. Similar to Xively, the Paraimpu middleware tries to support user privacy according to the privacy laws. Both Xively and Paraimpu have the privacy policy, but the details on the enforcement are not clearly presented. The Webinos middleware can meet the three strategies in terms of protecting user privacy. In Webinos, applications require permission to access the private data. The private data is processed and stored in a local Personal Zone Proxy (PZP) and a remote Personal Zone Hub (PZH) so the users can fully control their data. Besides, through the eXtensible Access Control Markup Language (XACML) and the Webinos policy enforcement framework, users can define fine-grained access control policies that will be enforced by the PZP and PZH to mediate every access to a Webinos API.

Additionally, we have reviewed some other IoT middlewares and software frameworks regarding their adoption of the *inform*, *control*, *enforce* and *demonstrate* strategies. The results are shown as the third part of Table III. The OpenHAB [172] is

a software framework designed for managing home automation systems. It makes all the devices and data stay in the local network and provides a single channel to enter the local network. It allows users to decide automation rules and has the ability to enforce the rules. It provides logging information for user-defined rules. Therefore, it satisfies all the four strategies. The AllJoyn [173] is a software framework aimed to create dynamic proximal networks by enhancing interoperability among devices and applications across manufacturers. Such proximal networks can make private data stay inside the local network and therefore has the potential to satisfy all the four strategies. The middleware based on NetwOrked Smart objects (NOS) [174] extracts privacy information from incoming data as part of security metadata at the Analysis layer, which is then used to annotate the data at the Data Annotation layer. It requires users to actively register and input private information to annotate their data. Further, the privacy protection can be enforced by the Integration layer and thus, the NOS-based middleware satisfies the three strategies.

In summary, we found that not all middlewares emphasize privacy protection. Although the recent middlewares have better protection than the previous ones, there are still privacy requirements that may be implemented at the middleware layer through PbD privacy strategies.

VII. Privacy at Application Layer

The unprecedented proximity between physical and digital worlds facilitated by IoT creates a huge number of applications [1], [4]. Different IoT applications may face different kinds of privacy risks as data collected in IoT applications may contain sensitive information related to the users. For instance, in smart home applications, religious beliefs of users may be inferred from smart refrigerators and similarly, daily schedules of users may be inferred from smart lamps. In automobile driving applications, dozens of internal sensors monitor data related to vehicle speed and seatbelt usage that can be used by insurance companies to determine insurance premium for the users. In healthcare and fitness applications, wearable devices may collect data that may reflect users' health information [176]. Similarly in smart meters, by applying energy disaggregation over the power usage data, it may be possible to learn when and how a home appliance was used by the residents [177]. In general, many of the application-level privacy risks can be handled at lower layers of the IoT architecture stack using PETs presented in Section IV to Section VI. For example, software frameworks such as OpenHAB [172] can make smart home a personal sphere so that data can be securely stored locally and any interaction with the data can be examined and logged. As another example, differential privacy mechanisms [65], [87] can be applied to perturb the smart meter data [178], [179], where the injected noises can be added by an in-home device. However, it is important to ensure that the PETs employed to achieve the privacy goals does not adversely affect the utility of the target IoT application. For example, perturbation PETs such as differential privacy when applied to healthcare data that require high accuracy to be retained, the resulting perturbed data may not retain the desirable clinical efficacy and as a result, it may lead to lower application utility [180]. In such cases, a cross-layer understanding of the impact of the employed PETs on the application-level utility is critical in determining the privacy-utility tradeoffs while designing the applications.

VIII. Related work

Research on privacy in IoT has become an important topic in the recent years. A number of surveys have summarized various challenges and potential solutions for privacy in IoT. Roman *et al.* [181] analyzed the features and challenges of security and privacy in distributed Internet of Things. The authors mentioned that data management and privacy can get immediate benefit from distributed IoTs as every entity in distributed IoTs has more control over the data it generates and processes. In [182], the authors discussed several types of PETs and focused on building a heterogeneous and differentiated legal framework that can handle the features of IoT including globality, verticality, ubiquity and technicity. Fink *et al.* [183] reviewed the challenges of privacy in IoT from both technical and legal standpoints. Ziegeldorf *et al.* [10] discussed the threats and challenges of privacy in IoT by first introducing the privacy definitions, reference models and legislation and reviewed the evolution of techniques and features for IoT. In both [184] and [185], security risks, challenges and promising techniques were presented in a layered IoT architecture but the discussion on privacy protection is limited to the techniques related to security problems.

Although most of the existing surveys review privacy in IoT from either a technical standpoint or a legal standpoint, to the best of our knowledge, none of the existing surveys analyzed the IoT privacy problem through a systematic fine-grained analysis of the privacy principles and techniques implemented at different layers of the IoT architecture stack. In this paper, we study the privacy protection problem in IoT through a comprehensive review of the state-of-the-art by jointly considering three key dimensions, namely the state-of-the-art principles of privacy laws, architecture of the IoT system and representative privacy enhancing technologies (PETs). Our work differentiates itself by its unique analysis of how legal principles can be supported through a careful implementation of various privacy enhancing technologies (PETs) at various layers of a layered IoT architecture model to meet the privacy requirements of the individuals interacting with the IoT systems.

IX. Conclusion

The fast proliferation of low-cost smart sensing devices and the widespread deployment of high-speed wireless networks have resulted in the rapid emergence of the Internet-of-things. In this paper, we study the privacy protection problem in IoT through a comprehensive review of the state-of-the-art by jointly considering three key dimensions, namely the architecture of the IoT system, state-of-the-art principles of privacy laws and representative privacy enhancing technologies (PETs). We analyze, evaluate and compare various PETs that can be deployed at different layers of a layered IoT architecture to meet the privacy requirements of the individuals

interacting with the IoT systems. Our analysis has shown that while many existing PETs (e.g., differential privacy, Tor) demonstrate a great potential for use in the IoT, the adoption of these techniques requires a careful consideration of the unique features associated with the IoT, including the use of heterogeneous power-limited devices and the massive need for streaming data flow. We expect this study to provide a broader understanding of the state-of-the-art principles in privacy legislation associated with the design of relevant privacy enhancing technologies (PETs) and how privacy legislation maps to privacy principles which in turn drives the design of necessary privacy enhancing technologies to be employed in the IoT architecture stack.

REFERENCE

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] L. Malina, J. Hajny, R. Fujdiak, and J. Hosek, "On perspective of security and privacy-preserving solutions in the internet of things," Computer Networks, vol. 102, pp. 83–95, 2016.

[3] "Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016," http://www.gartner.com/newsroom/id/3598917, 2017.

[4] B. D. Weinberg, G. R. Milne, Y. G. Andonova, and F. M. Hajjat, "Internet of things: Convenience vs. privacy and secrecy," Business Horizons, vol. 58, no. 6, pp. 615–624, 2015.

[5] G. D'Acquisto, J. Domingo-Ferrer, P. Kikiras, V. Torra, Y.-A. de Montjoye, and A. Bourka, "Privacy by design in big data: An overview of privacy enhancing technologies in the era of big data analytics," arXiv preprint arXiv:1512.06000, 2015.

[6] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by internet of things," Transactions on Emerging Telecommunications Technologies, vol. 25, no. 1, pp. 81–93, 2014.

[7] "Blackmirror," https://www.netflix.com/title/70264888.

[8] D. J. Solove, "A taxonomy of privacy," U. Pa. L. Rev., vol. 154, p. 477, 2005.

[9] "Ieee internet of things survey provides clarity around definition, future uses and challenges," http://www.prnewswire.com/news-releases/ieee-internet-of-things-survey-provides-clarity-around-definition-future-uses-and-challenges-193865271.html, 2013.

[10] J. H. Ziegeldorf, O. G. Morchon, and K. Wehrle, "Privacy in the internet of things: threats and challenges," Security and Communication Networks, vol. 7, no. 12, pp. 2728–2742, 2014.

[11] L. F. Cranor, "Necessary but not sufficient: Standardized mechanisms for privacy notice and choice," J. on Telecomm. & High Tech. L., vol. 10, p. 273, 2012.

[12] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, D. L. Metayer, R. Tirtea, and S. Schiffner, "Privacy and data protection by design-from policy to engineering," arXiv preprint arXiv:1501.03726, 2015.

[13] J.-H. Hoepman, "Privacy design strategies," in IFIP International Information Security Conference. Springer, 2014, pp. 446–459.

[14] M. Langheinrich, "Privacy by designprinciples of privacy-aware ubiquitous systems," in Ubicomp 2001: Ubiquitous Computing. Springer, 2001, pp. 273–291.

[15] S. D. Warren and L. D. Brandeis, "The right to privacy," Harvard law review, pp. 193–220, 1890.

[16] A. F. Westin, "Privacy and freedom," Washington and Lee Law Review, vol. 25, no. 1, p. 166, 1968.

[17] R. Gellman, "Fair information practices: A basic history," Browser Download This Paper, 2016.

[18] E. U.S. Department of Health and Welfare, "Report of the secretarys advisory committee on automated personal data systems," Records, Computer, and the Rights of Citizens, 1973.

[19] A. Levin and M. J. Nicholson, "Privacy law in the united states, the eu and canada: the allure of the middle ground," U. OTTAWA L. & TECH. J., vol. 2, p. 357, 2005.

[20] stanford, "Principles of privacy in the university," http://web.stanford.edu/group/privacyproject/legalPrinciplesOfPrivacy.html.

[21] D. of Homeland Security, "Privacy policy guidance memorandum," https://www.dhs.gov/xlibrary/assets/privacy/privacy_policyguide_2008-01.pdf.

[22] A. Act, "Health insurance portability and accountability act of 1996," Public law, vol. 104, p. 191, 1996.

[23] P. PROTECTION, "Children's online privacy protection act," 2002.

[24] OECD, "Guidelines on the protection of personal privacy and transborder flows of personal data," http://www.oecd.org/sti/ieconomy/oecdguidelinesontheprotectionofprivacyandtransborderflowsofpersonaldata.htm#part2.

[25] E. Directive, "95/46/ec-the data protection directive," Official Journal of the European Communities, 1995.

[26] U. D. of Commerce, "Safe harbor," https://www.export.gov/safeharbor/.

[27] "Privacy shield," https://www.privacyshield.gov/.

[28] E. U. Regulation, "General data protection regulation," Official Journal of the European Union, vol. 59, pp. 1–88, 2016.

[29] EUGDPR, "Gdpr key changes," http://www.eugdpr.org/key-changes.html.

[30] "onem2m requirements technical specification," http://www.ttc.or.jp/jp/document_list/pdf/j/TS/TS-M2M-0002v0.6.2.pdf, 2013.

[31] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012, pp. 13–16.

[32] C. Project, "Final report, rfid and the inclusive model for the internet of things," https://docbox.etsi.org/zArchive/TISPAN/Open/IoT/low%20resolution/www.rfidglobal.eu%20CASAGRAS%20IoT%20Final%20Report%20low%20resolution.pdf.

[33] E. TS102689, "Machine-to-machine communications (m2m): M2m service requirements."

[34] S. Spiekermann and L. F. Cranor, "Engineering privacy," IEEE Transactions on software engineering, vol. 35, no. 1, pp. 67–82, 2009.

[35] S. Lahlou, M. Langheinrich, and C. Röcker, "Privacy and trust issues with invisible computers," Communications of the ACM, vol. 48, no. 3, pp. 59–60, 2005.

[36] B.-J. Koops and R. Leenes, "Privacy regulation cannot be hardcoded. a critical comment on the privacy by designprovision in data-protection law," International Review of Law, Computers & Technology, vol. 28, no. 2, pp. 159–171, 2014.

[37] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the internet of things (iot)," IEEE Internet Initiative, no. 1, 2015.

[38] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "Role of middleware for internet of things: A study," International Journal of Computer Science and Engineering Survey, vol. 2, no. 3, pp. 94–105, 2011.

[39] I. D. Addo, S. I. Ahamed, S. S. Yau, and A. Buduru, "A reference architecture for improving security and privacy in internet of things applications," in Mobile Services (MS), 2014 IEEE International Conference on. IEEE, 2014, pp. 108–115.

[40] S. Funke, J. Daubert, A. Wiesmaier, P. Kikiras, and M. Muehlhaeuser, "End-2-end privacy architecture for iot," in Communications and Network Security (CNS), 2015 IEEE Conference on. IEEE, 2015, pp. 705–706.

[41] G. Sun, S. Huang, W. Bao, Y. Yang, and Z. Wang, "A privacy protection policy combined with privacy homomorphism in the internet of things," in Computer Communication and Networks (ICCCN), 2014 23rd International Conference on. IEEE, 2014, pp. 1–6.

[42] C. Perera, C. McCormick, A. K. Bandara, B. A. Price, and B. Nuseibeh, "Privacy-by-design framework for assessing internet of things applications and platforms," in Proceedings of the 6th International Conference on the Internet of Things. ACM, 2016, pp. 83–92.

[43] M. Dabbagh and A. Rayes, "Internet of things security and privacy," in Internet of Things From Hype to Reality. Springer, 2017, pp. 195–223.

[44] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," Future Generation Computer Systems, vol. 56, pp. 684–700, 2016.

[45] M. Henze, L. Hermerschmidt, D. Kerpen, R. Häußling, B. Rumpe, and K. Wehrle, "A comprehensive approach to privacy in the cloud-based internet of things," Future Generation Computer Systems, vol. 56, pp. 701–718, 2016.

[46] X. Jia, Q. Feng, T. Fan, and Q. Lei, "Rfid technology and its applications in internet of things (iot)," in Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on. IEEE, 2012, pp. 1282–1285.

[47] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[48] L. Tan and N. Wang, "Future internet: The internet of things," in Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, vol. 5. IEEE, 2010, pp. V5–376.

[49] Z. Specification, "Zigbee alliance," ZigBee Document 053474r06, Version, vol. 1, 2006.

[50] B. SIG, "Bluetooth core specification v5.0," https://www.bluetooth.com/specifications/adopted-specifications, 2016.

[51] W.-F. Alliance, "Wi-fi halow," http://www.wi-fi.org/discover-wi-fi/wi-fi-halow.

[52] I. WP5D, "Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond," 2015.

[53] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "A survey of middleware for internet of things," in Recent trends in wireless and mobile networks. Springer, 2011, pp. 288–296.

[54] M. A. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the internet of things," in Collaboration Technologies and Systems (CTS), 2012 International Conference on. IEEE, 2012, pp. 21–26.

[55] K. Aberer, M. Hauswirth, and A. Salehi, "A middleware for fast and flexible sensor network deployment," in Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, 2006, pp. 1199–1202.

[56] M. Eisenhauer, P. Rosengren, and P. Antolin, "Hydra: A development platform for integrating wireless devices and sensors into ambient intelligence systems," The Internet of Things, pp. 367–373, 2010.

[57] A. Gómez-Goiri and D. López-de Ipiña, "A triple space-based semantic distributed middleware for internet of things," Current Trends in Web Engineering, pp. 447–458, 2010.

[58] Node-Red, "A visual tool for wiring the internet-of-things," http://nodered.org, 2015.

[59] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "Iot middleware: A survey on issues and enabling technologies," IEEE Internet of Things Journal, vol. 4, no. 1, pp. 1–20, 2017.

[60] P. Fremantle and P. Scott, "A security survey of middleware for the internet of things," PeerJ PrePrints, Tech. Rep., 2015.

[61] C. Gentry et al., "Fully homomorphic encryption using ideal lattices." in STOC, vol. 9, no. 2009, 2009, pp. 169–178.

[62] L. Sweeney, "k-anonymity: A model for protecting privacy," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 05, pp. 557–570, 2002.

[63] J. L. H. Ramos, J. B. Bernabé, and A. F. Skarmeta, "Towards privacy-preserving data sharing in smart environments," in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2014 Eighth International Conference on. IEEE, 2014, pp. 334–339.

[64] L. Sweeney, "Simple demographics often identify people uniquely," Health (San Francisco), vol. 671, pp. 1–34, 2000.

[65] C. Dwork, "Differential privacy: A survey of results," in International Conference on Theory and Applications of Models of Computation. Springer, 2008, pp. 1–19.

[66] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in TCC, vol. 3876. Springer, 2006, pp. 265–284.

[67] S. Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data," IEEE access, vol. 4, pp. 2751–2763, 2016.

[68] M. R. Rieback, B. Crispo, and A. S. Tanenbaum, "The evolution of rfid security," IEEE Pervasive Computing, vol. 5, no. 1, pp. 62–69, 2006.

[69] C.-k. Chung, Y.-k. Hsieh, Y.-h. Wang et al., "Aware and smart member card: Rfid and license plate recognition systems integrated applications at parking guidance in shopping mall," in Advanced Computational Intelligence (ICACI), 2016 Eighth International Conference on. IEEE, 2016, pp. 253–256.

[70] B. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," ACM Computing Surveys (CSUR), vol. 42, no. 4, p. 14, 2010.

[71] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on. IEEE, 2006, pp. 24–24.

[72] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on. IEEE, 2007, pp. 106–115.

[73] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on. IEEE, 2006, pp. 25–25.

[74] T. Iwuchukwu and J. F. Naughton, "K-anonymization as spatial indexing: Toward scalable and incremental anonymization," in Proceedings of the 33rd international conference on Very large data bases. VLDB Endowment, 2007, pp. 746–757.

[75] C. C. Aggarwal, "On k-anonymity and the curse of dimensionality," in Proceedings of the 31st international conference on Very large data bases. VLDB Endowment, 2005, pp. 901–909.

[76] G. Ghinita, Y. Tao, and P. Kalnis, "On the anonymization of sparse high-dimensional data," in Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on. Ieee, 2008, pp. 715–724.

[77] J. Cao, B. Carminati, E. Ferrari, and K.-L. Tan, "Castle: Continuously anonymizing data streams," IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 3, pp. 337–352, 2011.

[78] J. Li, B. C. Ooi, and W. Wang, "Anonymizing streaming data for privacy protection," in Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on. IEEE, 2008, pp. 1367–1369.

[79] A. Meyerson and R. Williams, "On the complexity of optimal k-anonymity," in Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2004, pp. 223–228.

[80] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu, "Anonymizing tables," in International Conference on Database Theory. Springer, 2005, pp. 246–258.

[81] H. Park and K. Shim, "Approximate algorithms for k-anonymity," in Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM, 2007, pp. 67–78.

[82] K. Wang, B. C. Fung, and G. Dong, "Integrating private databases for data analysis," in International Conference on Intelligence and Security Informatics. Springer, 2005, pp. 171–182.

[83] W. Jiang and C. Clifton, "A secure distributed framework for achieving k-anonymity," The VLDB JournalThe International Journal on Very Large Data Bases, vol. 15, no. 4, pp. 316–333, 2006.

[84] X. Xiao and Y. Tao, "Personalized privacy preservation," in Proceedings of the 2006 ACM SIGMOD international conference on Management of data. ACM, 2006, pp. 229–240.

[85] Y. Xu, X. Qin, Z. Yang, Y. Yang, and K. Li, "A personalized k-anonymity privacy preserving method," JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE, vol. 10, no. 1, pp. 139–155, 2013.

[86] C. Dwork, A. Roth et al., "The algorithmic foundations of differential privacy," Foundations and Trends® in Theoretical Computer Science, vol. 9, no. 3–4, pp. 211–407, 2014.

[87] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on. IEEE, 2007, pp. 94–103.

[88] C. Dwork, "Differential privacy in new settings," in Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms. SIAM, 2010, pp. 174–183.

[89] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in Proceedings of the forty-second ACM symposium on Theory of computing. ACM, 2010, pp. 715–724.

[90] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," ACM Transactions on Information and System Security (TISSEC), vol. 14, no. 3, p. 26, 2011.

[91] L. Fan and L. Xiong, "Real-time aggregate monitoring with differential privacy," in Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012, pp. 2169–2173.

[92] G. Ács and C. Castelluccia, "I have a dream!(differentially private smart metering)." in Information hiding, vol. 6958. Springer, 2011, pp. 118–132.

[93] S. Kotz, T. Kozubowski, and K. Podgorski, The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance. Springer Science & Business Media, 2012.

[94] S. Goryczka, L. Xiong, and V. Sunderam, "Secure multiparty aggregation with differential privacy: A comparative study," in Proceedings of the Joint EDBT/ICDT 2013 Workshops. ACM, 2013, pp. 155–163.

[95] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010, pp. 735–746.

[96] Z. Jorgensen, T. Yu, and G. Cormode, "Conservative or liberal? personalized differential privacy," in Data Engineering (ICDE), 2015 IEEE 31st International Conference on. IEEE, 2015, pp. 1023–1034.

[97] S. Heron, "Advanced encryption standard (aes)," Network Security, vol. 2009, no. 12, pp. 8–12, 2009.

[98] R. L. Rivest, A. Shamir, and L. M. Adleman, "Cryptographic communications system and method," Sep. 20 1983, uS Patent 4,405,829.

[99] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," Journal of Computer Security, vol. 19, no. 5, pp. 895–934, 2011.

[100] A. C. Yao, "Protocols for secure computations," in Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on. IEEE, 1982, pp. 160–164.

[101] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography," NIST DRAFT NISTIR, vol. 8114, 2016.

[102] NXP, "Rs08ka," http://www.nxp.com/products/microcontrollers-and-processors/more-processors/8-16-bit-mcus/8-bit-rs08/8-bitgeneral-purpose-ultra-low-end-market-ka-mcus:RS08KA.

[103] M.-J. O. Saarinen and D. W. Engels, "A do-it-all-cipher for rfid: Design requirements," IACR Cryptology EPrint Archive, vol. 2012, p. 317, 2012.

[104] C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Ultra-lightweight implementations for smart devices–security for 1000 gate equivalents," in CARDIS, vol. 5189. Springer, 2008, pp. 89–103.

[105] K. Ideguchi, T. Owada, and H. Yoshida, "A study on ram requirements of various sha-3 candidates on low-cost 8-bit cpus." IACR Cryptology ePrint Archive, vol. 2009, p. 260, 2009.

[106] P. Gauravaram, L. R. Knudsen, K. Matusiewicz, F. Mendel, C. Rechberger, M. Schläffer, and S. S. Thomsen, "Grøstl-a sha-3 candidate," in Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

[107] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2005, pp. 457–473.

[108] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and communications security. Acm, 2006, pp. 89–98.

[109] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Security and Privacy, 2007. SP'07. IEEE Symposium on. IEEE, 2007, pp. 321–334.

[110] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in Proceedings of the 17th ACM conference on Computer and communications security. ACM, 2010, pp. 735–737.

[111] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in Annual international conference on the theory and applications of cryptographic techniques. Springer, 2011, pp. 568–588.

[112] Q. Huang, L. Wang, and Y. Yang, "Decent: secure and fine-grained data access control with policy updating for constrained iot devices," World Wide Web, vol. 21, no. 1, pp. 151–167, 2018.

[113] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of attribute-based encryption: Toward data privacy in the iot," in Communications (ICC), 2014 IEEE International Conference on. IEEE, 2014, pp. 725–730.

[114] M. Ambrosin, M. Conti, and T. Dargahi, "On the feasibility of attribute-based encryption on smartphone devices," in Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems. ACM, 2015, pp. 49–54.

[115] M. Ambrosin, A. Anzanpour, M. Conti, T. Dargahi, S. R. Moosavi, A. M. Rahmani, and P. Liljeberg, "On the feasibility of attribute-based encryption on internet of things devices," IEEE Micro, vol. 36, no. 6, pp. 25–35, 2016.

[116] N. Oualha and K. T. Nguyen, "Lightweight attribute-based encryption for the internet of things," in Computer Communication and Networks (ICCCN), 2016 25th International Conference on. IEEE, 2016, pp. 1–6.

[117] Y. Yang, X. Zheng, and C. Tang, "Lightweight distributed secure data management system for health internet of things," Journal of Network and Computer Applications, vol. 89, pp. 26–37, 2017.

[118] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," Future Generation Computer Systems, vol. 49, pp. 104–112, 2015.

[119] A. Biryukov and L. Perrin, "Lightweight cryptography lounge," http://cryptolux.org/index.php/Lightweight_Cryptography, 2015.

[120] Z. Gong, S. Nikova, and Y. W. Law, "Klein: A new family of lightweight block ciphers." RFIDSec, vol. 7055, pp. 1–18, 2011.

[121] M. R. Albrecht, B. Driessen, E. B. Kavun, G. Leander, C. Paar, and T. Yalçın, "Block ciphers–focus on the linear layer (feat. pride)," in International Cryptology Conference. Springer, 2014, pp. 57–76.

[122] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An ultra-lightweight block cipher," in CHES, vol. 4727. Springer, 2007, pp. 450–466.

[123] R. Rivest, "The rc5 encryption algorithm," in Fast software encryption. Springer, 1995, pp. 86–96.

[124] A. Poschmann, G. Leander, K. Schramm, and C. Paar, "New lightweight des variants suited for rfid applications," in FSE, vol. 4593, 2007, pp. 196–210.

[125] J. Guo, T. Peyrin, and A. Poschmann, "The photon family of lightweight hash functions," Advances in Cryptology–CRYPTO 2011, pp. 222–239, 2011.

[126] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varıcı, and I. Verbauwhede, "Spongent: A lightweight hash function," Cryptographic Hardware and Embedded Systems–CHES 2011, pp. 312–325, 2011.

[127] M. Hell, T. Johansson, A. Maximov, and W. Meier, "The grain family of stream ciphers," Lecture Notes in Computer Science, vol. 4986, pp. 179–190, 2008.

[128] S. Babbage and M. Dodd, "The mickey stream ciphers," in New Stream Cipher Designs. Springer, 2008, pp. 191–209.

[129] C. De Canniere and B. Preneel, "Trivium," New Stream Cipher Designs, pp. 244–266, 2008.

[130] P. Paillier et al., "Public-key cryptosystems based on composite degree residuosity classes," in Eurocrypt, vol. 99. Springer, 1999, pp. 223–238.

[131] V. Oleshchuk, "Internet of things and privacy preserving technologies," in Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on. IEEE, 2009, pp. 336–340.

[132] K. Seo and S. Kent, "Security architecture for the internet protocol," 2005.

[133] T. Dierks and C. Allen, "Rfc 2246: The tls protocol," IETF, January, 1999.

[134] N. Kushalnagar, G. Montenegro, and C. Schumacher, "Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals," Tech. Rep., 2007.

[135] I. Ishaq, D. Carels, G. K. Teklemariam, J. Hoebeke, F. V. d. Abeele, E. D. Poorter, I. Moerman, and P. Demeester, "Ietf standardization in the field of the internet of things (iot): a survey," Journal of Sensor and Actuator Networks, vol. 2, no. 2, pp. 235–287, 2013.

[136] A. Banks and R. Gupta, "Mqtt version 3.1. 1," OASIS standard, vol. 29, 2014.

[137] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," 2014.

[138] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of ipv6 packets over ieee 802.15. 4 networks," Tech. Rep., 2007.

[139] S. Raza, T. Chung, S. Duquennoy, T. Voigt, U. Roedig et al., "Securing internet of things with lightweight ipsec," 2010.

[140] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, "Secure communication for the internet of thingsa comparison of link-layer security and ipsec for 6lowpan," Security and Communication Networks, vol. 7, no. 12, pp. 2654–2668, 2014.

[141] E. Rescorla and N. Modadugu, "Datagram transport layer security version 1.2," 2012.

[142] C. Bormann, "Using coap with ipsec," 2012.

[143] T. A. Alghamdi, A. Lasebae, and M. Aiash, "Security analysis of the constrained application protocol in the internet of things," in Future Generation Communication Technology (FGCT), 2013 second international conference on. IEEE, 2013, pp. 163–168.

[144] "Thread," https://www.threadgroup.org/, 2018.

[145] "Ipso alliance," https://www.ipso-alliance.org/, 2018.

[146] "Industrial internet of things volume g4: Security framework," http://www.iiconsortium.org/pdf/IIC_PUB_G4_V1.00_PB-3.pdf, 2016.

[147] "Ddsi-rtps," https://www.omg.org/spec/DDSI-RTPS/About-DDSI-RTPS/, 2014.

[148] N. P. Hoang and D. Pishva, "Anonymous communication and its importance in social networking," in Advanced Communication Technology (ICACT), 2014 16th International Conference on. IEEE, 2014, pp. 34–39.

[149] ——, "A tor-based anonymous communication approach to secure smart home appliances," in Advanced Communication Technology (ICACT), 2015 17th International Conference on. IEEE, 2015, pp. 517–525.

[150] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Naval Research Lab Washington DC, Tech. Rep., 2004.

[151] "Tor network status," http://torstatus.blutmagie.de/, 2017.

[152] L. Cranor, M. Langheinrich, and M. Marchiori, "A p3p preference exchange language 1.0 (appel1. 0)," W3C working draft, vol. 15, 2002.

[153] G. Broenink, J.-H. Hoepman, C. v. Hof, R. Van Kranenburg, D. Smits, and T. Wisman, "The privacy coach: Supporting customer privacy in the internet of things," arXiv preprint arXiv:1001.4459, 2010.

[154] H. Desruelle, J. Lyle, S. Isenberg, and F. Gielen, "On the challenges of building a web-based ubiquitous application platform," in Proceedings of the 2012 ACM conference on ubiquitous computing. ACM, 2012, pp. 733–736.

[155] C. V. Berghe and M. Schunter, "Privacy injector-automated privacy enforcement through aspects," in Privacy Enhancing Technologies, vol. 4258. Springer, 2006, pp. 99–117.

[156] M. C. Mont, S. Pearson, and P. Bramhall, "Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services," in Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on. IEEE, 2003, pp. 377–382.

[157] M. C. Mont and R. Thyne, "A systemic approach to automate privacy policy enforcement in enterprises," Lecture Notes in Computer Science, vol. 4258, pp. 118–134, 2006.

[158] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Hippocratic databases," in Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, 2002, pp. 143–154.

[159] P. Bonnet, J. Gehrke, and P. Seshadri, "Towards sensor database systems," in Mobile Data Management. Springer, 2001, pp. 3–14.

[160] T. Liu and M. Martonosi, "Impala: A middleware system for managing autonomic, parallel sensor systems," in ACM Sigplan Notices, vol. 38, no. 10. ACM, 2003, pp. 107–118.

[161] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan, "Irisnet: An architecture for a worldwide sensor web," IEEE pervasive computing, vol. 2, no. 4, pp. 22–33, 2003.

[162] M. C. Huebscher and J. A. McCann, "Adaptive middleware for context-aware applications in smart-homes," in Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing. ACM, 2004, pp. 111–116.

[163] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. L. Murphy, and G. P. Picco, "Mobile data collection in sensor networks: The tinylime middleware," Pervasive and Mobile Computing, vol. 1, no. 4, pp. 446–469, 2005.

[164] Y. Yu, L. J. Rittle, V. Bhandari, and J. B. LeBrun, "Supporting concurrent applications in wireless sensor networks," in Proceedings of the 4th international conference on Embedded networked sensor systems. ACM, 2006, pp. 139–152.

[165] V. Tsiatsis, A. Gluhak, T. Bauge, F. Montagut, J. Bernat, M. Bauer, C. Villalonga, P. Barnaghi, and S. Krco, "The sensei real world internet architecture," 2010.

[166] V. Terziyan, O. Kaykova, and D. Zhovtobryukh, "Ubiroad: Semantic middleware for context-aware smart road environments," in Internet and web applications and services (iciw), 2010 fifth international conference on. IEEE, 2010, pp. 295–302.

[167] "Xively," http://xively.com.

[168] A. Pintus, D. Carboni, and A. Piras, "Paraimpu: a platform for a social web of things," in Proceedings of the 21st International Conference on World Wide Web. ACM, 2012, pp. 401–404.

[169] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko et al., "Openiot: Open source internet-of-things in the cloud," in Interoperability and open-source solutions for the internet of things. Springer, 2015, pp. 13–25.

[170] "Google fit," https://developers.google.com/fit/overview.

[171] P. Persson and O. Angelsmark, "Calvin–merging cloud and iot," Procedia Computer Science, vol. 52, pp. 210–217, 2015.

[172] "Openhab," https://www.openhab.org/, 2018.

[173] "Alljoyn," https://openconnectivity.org/developer/reference-implementation/alljoyn, 2018.

[174] S. Sicari, C. Cappiello, F. De Pellegrini, D. Miorandi, and A. Coen-Porisini, "A security-and quality-aware system architecture for internet of things," Information Systems Frontiers, vol. 18, no. 4, pp. 665–677, 2016.

[175] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for internet of things: a survey," IEEE Internet of Things Journal, vol. 3, no. 1, pp. 70–95, 2016.

[176] M. Meingast, T. Roosta, and S. Sastry, "Security and privacy issues with health care information technology," in Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE. IEEE, 2006, pp. 5453–5458.

[177] A. Ukil, S. Bandyopadhyay, and A. Pal, "Iot-privacy: To be private or not to be private," in Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on. IEEE, 2014, pp. 123–124.

[178] L. Sankar, S. R. Rajagopalan, and S. Mohajer, "Smart meter privacy: A theoretical framework," IEEE Transactions on Smart Grid, vol. 4, no. 2, pp. 837–846, 2013.

[179] J. Zhao, T. Jung, Y. Wang, and X. Li, "Achieving differential privacy of data disclosure in the smart grid," in INFOCOM, 2014 Proceedings IEEE. IEEE, 2014, pp. 504–512.

[180] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing." in USENIX Security Symposium, 2014, pp. 17–32.

[181] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," Computer Networks, vol. 57, no. 10, pp. 2266–2279, 2013.

[182] R. H. Weber, "Internet of things–new security and privacy challenges," Computer law & security review, vol. 26, no. 1, pp. 23–30, 2010.

[183] G. A. Fink, D. V. Zarzhitsky, T. E. Carroll, and E. D. Farquhar, "Security and privacy grand challenges for the internet of things," in Collaboration Technologies and Systems (CTS), 2015 International Conference on. IEEE, 2015, pp. 27–34.

[184] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1250–1258, 2017.

[185] D. M. Mendez, I. Papapanagiotou, and B. Yang, "Internet of things: Survey on security and privacy," arXiv preprint arXiv:1707.01879, 2017.

Chao Li is currently a 4th year Ph.D. student in the School of computing and information, University of Pittsburgh. He is also a member of the Laboratory of Education and Research on Security Assured Information Systems (LERSAIS). Before that, he got his MSc degree in Communication and Signal Processing from Imperial College London and BEng degree in Electronics and Electrical Engineering from both University of Edinburgh and Dalian University of Technology. His current research interests include location and data privacy and blockchain-based protocol design. He is a student member of the IEEE.

Balaji Palanisamy is an Assistant Professor in the School of computing and information in University of Pittsburgh. He received his M.S and Ph.D. degrees in Computer Science from the college of Computing at Georgia Tech in 2009 and 2013, respectively. His primary research interests lie in scalable and privacy-conscious resource management for large-scale Distributed and Mobile Systems. At University of Pittsburgh, he codirects research in the Laboratory of Research and Education on Security Assured Information Systems (LERSAIS). He is a member of the IEEE. Dr. Palanisamy is currently serving as an Associate Editor for the IEEE Transactions on Services Computing journal.