An Elaborated Framework for Protecting Privacy in the IoT

George O. M. Yee
Computer Research Lab, Aptusinnova Inc., Ottawa, Canada
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada
email: george@aptusinnova.com, gmyee@sce.carleton.ca

***Abstract*—The Internet of Things (IoT) is attracting great interest within the research community. Yet, there is little research on how data generated by the "things" can be shared while respecting the privacy wishes of the data's owners. Consider a smart refrigerator as one of the "things". It keeps track of which food items are consumed, in order that the consumer can know when and what foods need to be replenished. Suppose the smart refrigerator sends this consumption information to online grocers that can automatically schedule deliveries to replenish the food. The consumption information may contain personal information (e.g., foods identifying a particular medical condition) leading to privacy concerns. This paper extends the CYBER 2016 paper "An Approach for Protecting Privacy in the IoT". The original version proposed an approach that utilizes personal privacy policies and policy compliance checking to protect privacy in the IoT, using the smart refrigerator as an example to illustrate the approach. This paper adds additional explanations and diagrams, a health monitoring example, and more discussion on related work.**

Keywords-privacy protection; IoT; privacy policy; compliance; controller.

I. INTRODUCTION

The objective of this paper is to present an elaborated framework that makes use of privacy policies and policy compliance checking to protect privacy in the IoT. Privacy protection is in the context of smart devices (defined below) that supply data to e-services (defined below). The smart devices themselves may also be providing e-services. The objective of this paper is achieved by focusing on a smart device as sending data that needs privacy protection.

This work extends Yee [1] by expanding all sections with additional details. In particular, an additional example using health monitoring has been added, and the section on related works has been enlarged.

A "smart" device is any physical device endowed with computing and communication capabilities. Some smart devices may have more computing and communication capabilities (e.g., smartphones) than others (e.g., sensors). An e-service is a grouping of computation that optionally takes input and produces output (the service). For example, the connected smart refrigerator would access the food replenish e-service from the online grocer and transmit its food consumption information (the input) to the food replenish e-service. In response, the food replenish e-service would schedule food deliveries (the output). As another example, a sensor would provide an e-service of transmitting data to another e-service that requested the data. In this case, the sensor e-service would not require any input (except for the request to transmit data).

This work addresses an Internet of things environment (see Fig. 1) with the following characteristics:

- Smart devices (e.g., laptops, smartphones, workstations, smart sensors, smart appliances, smart home switches and cameras, smart speakers) are optionally locally networked (e.g., Ethernet, Wi-Fi, IrDA, Bluetooth) or standalone (i.e., not locally networked). The locally networked or standalone smart devices are connected to the Internet via an Internet Service Provider (ISP).
- The locally networked or standalone smart devices are owned by a human or an organization.
- Human users employ these devices to make use of e-services, to offer e-services, or both. A user who makes use of an e-service sends information to that e-service and is called a *data sender*. One who offers an e-service receives information needed by that e-service and is called a *data receiver*. A user who both makes use of e-services and offers e-services is both a data sender and a data receiver.

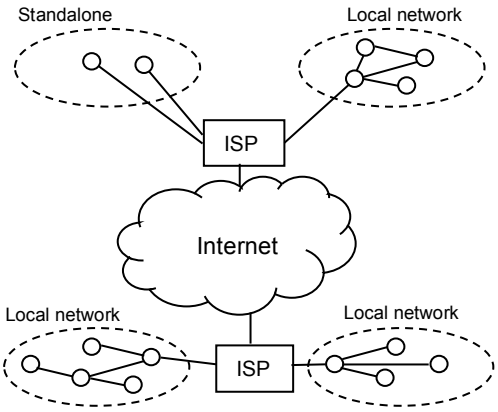

Figure 1. IoT network environment (ISP = Internet Service Provider, circles are smart devices)

The remainder of this paper is organized as follows. Section II looks at privacy and the use of privacy policies. Section III presents the proposed framework. Section IV

gives two examples of applying the framework. Section V discusses some strengths and weaknesses of the framework. Section VI examines related work. Section VII concludes the paper and lists some ideas for future research.

II. PRIVACY POLICIES

A. Privacy

As defined by Goldberg et al. in 1997 [2], privacy refers to the ability of individuals to *control* the collection, retention, and distribution of information about themselves. This is the definition of privacy used for this work. Protecting an individual's privacy then involves endowing the individual with the ability to control the collection, retention, and distribution of her personal information.

B. Use of Privacy Policies

In this work, a data sender is given control over her private information as follows. The data sender specifies in her sender privacy policy how she wants her personal information handled by the data receiver; the data receiver, on the other hand, specifies in her receiver privacy policy what personal information her service requires from the data sender and how she plans to handle the data sender's information. The data sender's policy has to be compatible or match the data receiver's policy before information sending can begin. If the policies do not match, the data sender can either negotiate with the data receiver to try to resolve the disagreement or choose a different data receiver. Once the information is sent, the data receiver has to comply with the sender's privacy policy (which is compatible with her own receiver privacy policy). Foolproof mechanisms must be in place to ensure compliance. The detailed mechanics of privacy policy matching [3] and negotiation [4] are outside the scope of this work, although we do explain below the meaning of matching.

Fig. 2 shows example sender and receiver privacy policies for a smart refrigerator. We have not expressed these policies in any specific policy language, preferring to keep our meaning clear and unencumbered with language details (see Section III D and Section VI). Referring to Fig. 2, a privacy policy for sending personal information consists of a header section (shaded) followed by one or more privacy rules, where there is one rule for each item of personal information. The fields within the header have the following meaning: *Policy Use* identifies the e-service (e.g., replenish food), *Data Sender / Data Receiver* gives the name of the party that owns the policy, and *Valid* indicates the period of time during which the policy is valid. The fields in each privacy rule have the following meaning: *Data Receiver* identifies the party that receives the information, *What* describes the nature of the information, *Purpose* identifies the purpose for which the information is being sent or received, *Retention Time* specifies the amount of time the data receiver can keep the information, and *Disclose-To* identifies any parties who will receive the information from the data receiver. Fig. 3 shows example sender and receiver privacy policies for a smart watch, which is able to monitor the wearer's heart rate, skin temperature, sleep pattern, and exercise pattern. In this case, the e-service identified in *Policy Use* is Health Monitor, which is an online service that continuously monitors a person's health by gathering and processing health indicators such as heart rate and skin temperature. The other fields in the privacy policies have the same meaning as described above for Fig. 2.

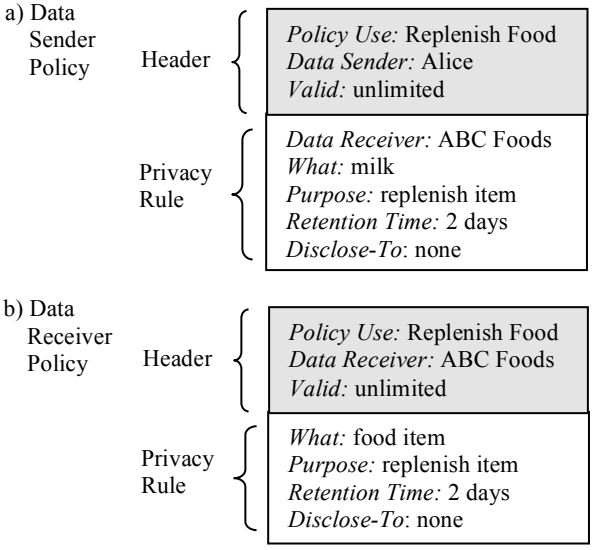

Figure 2. Example data sender / data receiver privacy policies for a smart refrigerator. Each policy can have as many privacy rules as are needed.

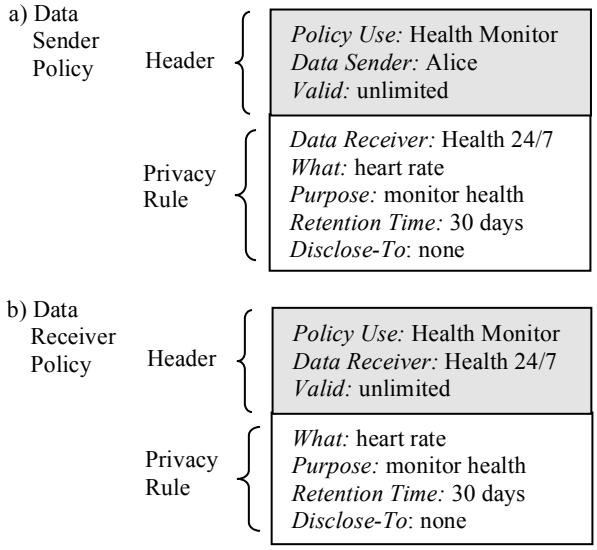

Figure 3. Example data sender / data receiver privacy policies for a smart watch. Each policy can have as many privacy rules as are needed.

It was mentioned above that the sender and receiver privacy policies have to "match". Matching means that the

values of the fields *What*, *Purpose*, and *Disclose-To* are the same in both sender and receiver policies for the same data item. It further means that the *Retention Time* of the receiver policy is not longer than the *Retention Time* of the sender policy for any data item. Finally, both sender and receiver policies must be valid during the utilization period of the e-service. For example, the policies in Fig. 3 would not match if the receiver policy were to have the following values: *Retention Time:* 40 days, *Disclose-To:* John, where John is Alice's husband (Alice does not want John to be concerned if she has an abnormal heart rate).

The above privacy rules and fields conform to Canadian privacy legislation, which is representative of privacy legislation in many parts of the world, including the European Union and the United States. The fields *What*, *Purpose*, *Retention Time*, and *Disclose-T*o correspond to fair information principles 4, 2, 5, and 5, respectively, as shown in Table I. Policy matching corresponds to principle 3 (consent). The fair information principles form the foundation of the Canadian Personal Information Protection and Electronic Documents Act (PIPEDA) [5]. Thus, a data receiver who complies with a data sender's privacy policy also complies with the sender's legislated privacy rights. Furthermore, the framework proposed here would apply in the European Union, the United States, and elsewhere in the world where privacy legislation similar to PIPEDA exist, with only minor changes to the content of the privacy policies.

TABLE I. PIPEDA FAIR INFORMATION PRINCIPLES

Principle	***Description***
1. Accountability	An organization is responsible for personal information under its control. It must appoint someone to be accountable for its compliance with these fair information principles.
2. Identifying Purposes	The purposes for which personal information is collected must be identified by the organization before or at the time of collection.
3. Consent	The knowledge and consent of the individual are required for the collection, use, or disclosure of personal information, except when inappropriate.
4. Limiting Collection	The collection of personal information must be limited to that which is needed for the purposes identified by the organization. Information must be collected by fair and lawful means.
5. Limiting Use, Disclosure, and Retention	Unless the individual consents otherwise or it is required by law, personal information can only be used or disclosed for the purposes for which it was collected. Personal information must only be kept as long as required to serve those purposes.
6. Accuracy	Personal information must be as accurate, complete, and up-to-date as possible in order to properly satisfy the purposes for which it is to be used.
7. Safeguards	Personal information must be protected by appropriate security relative to the sensitivity of the information.
8. Openness	An organization must make detailed information about its policies and practices relating to the management of personal information publicly and readily available.
9. Individual Access	Upon request, an individual must be informed of the existence, use and disclosure of their personal information and be given access to that information. An individual shall be able to challenge the accuracy and completeness of the information and have it amended as appropriate.
10. Challenging Compliance	An individual shall be able to challenge an organization's compliance with the above principles. Their challenge should be addressed to the person accountable for the organization's compliance with PIPEDA, usually their Chief Privacy Officer.

III. FRAMEWORK

For each smart device, the framework consists of two phases: a privacy policy agreement (PPA) phase and a privacy policy compliance (PPC) phase. These phases apply to both data senders and data receivers.

A. PPA Phase and Design of Policy Controller

The PPA phase consists of the composition and exchange of privacy policies between data sender and data receiver, using a Policy Controller (PC), which runs on a desktop, laptop, a mobile device such as a smart phone or tablet, or on the IoT node itself if it has sufficient computing power. The components and functionality of the PC are given in Table II.

TABLE II. POLICY CONTROLLER (PC)

PC Component	**Functionality**
Policy Module (PM) - Data Sender	Partially composes the data sender policy; searches for e-services (data receivers) and obtains their receiver policies; determines if data receiver policies match the sender policy; selects a data receiver with a matching policy and completes the data sender policy by filling in the name of the data receiver; sends the sender privacy policy to the selected data receiver; sends the sender policy to the smart device; optionally sets up a privacy policy negotiation between the data sender and a data receiver for a particular policy pair that does not match, in order to try to arrive at a match (where possible)
PM - Data Receiver	Composes the data receiver privacy policy; sends the data receiver privacy policy to the PM of the data sender when requested; receives the data sender privacy policy and verifies that the sender policy matches its own policy; optionally cooperates to set up a privacy policy negotiation with the owner of a data sender
Policy Store (PS) – Data Sender	Holds the data sender privacy policy; holds the privacy policies received from data receivers
PS – Data Receiver	Holds the data receiver privacy policy; holds the privacy policies received from data senders

Fig. 4 presents a message sequence chart showing the interactions between the PMs of a data sender and a data receiver (only one receiver shown and policy composition excluded for simplicity). A first time successful privacy policies match is assumed.

Fig. 5 shows the same scenario as Fig. 4 except that the first time policy match is unsuccessful, resulting in the need for policy negotiation, assumed to be successful. If the

negotiation was unsuccessful, the sender would not be able to proceed any further with the receiver and would have to select a new receiver or find some way to satisfy the receiver's policy.

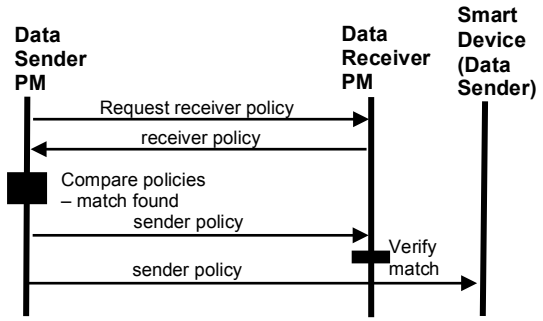

Figure 4. Message sequence chart showing the interactions for a first time successful policy match.

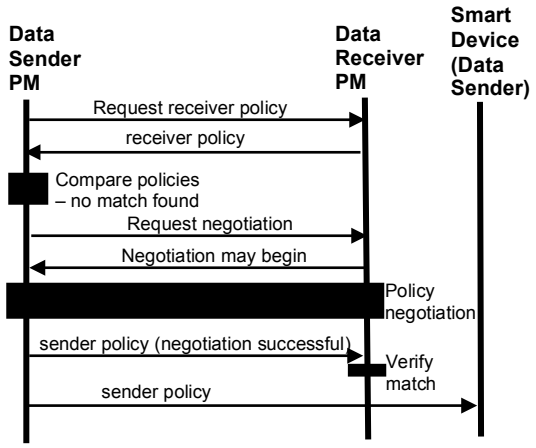

Figure 5. Message sequence chart showing the interactions for a first time unsuccessful policy match and the ensuing negotiation (assumed successful).

In the case where the data sender's PM finds a data receiver that has received data from the data sender in the past, it is very likely that the associated privacy policies already match. To account for this case, the PM always verifies if the data receiver found is one that has received data from the data sender in the past, prior to determining if the policies match. If this verification is positive, the PM further verifies if the sender and receiver policies are the same as in the previous interaction, and if yes, bypasses determining if the policies match. These checks may improve the performance of the data sender's PM. These checks are incorporated in the task "determines if data receiver policies match the sender policy" for the data sender's PM in Table II. They are also part of the "Compare policies" module in Fig. 4 and Fig. 5. Similarly, provided that no negotiation occurred, the data receiver's PM verifies if the receiver has received data from the data sender in the past, prior to verifying if the policies match. If so, the receiver's PM will further verify if the policies are the same as during the last interaction, and if they are the same, does not verify if the policies match. This may improve the performance of the data receiver's PM if no negotiation occurred. These verifications are incorporated in the task "verifies that the sender policy matches its own policy" for the data receiver's PM in Table II. They are also part of the "Verify match" module (Data Receiver PM) in Fig. 4 and Fig. 5.

B. PPC Phase and Design of Compliance Controller

In the PPC phase, the data sender sends its data to the data receiver, while ensuring that both sender and receiver privacy policies are respected. This phase is carried out using software called a Compliance Controller (CC), which runs on the smart device or on a computing platform (e.g., tablet) that is "linked" to the device. The components and functionality of the CC are given in Table III. The data sender's CC makes a single connection to the data receiver's CC per data sending session. In Table III, for a particular smart device, Compliance Module (CM) functionality depends on whether the device sends data, receives data, or both sends and receives data. In the latter case, each component would have the functionalities prescribed for a data sender and data receiver combined.

TABLE III. COMPLIANCE CONTROLLER (CC)

CC Component	Functionality
Compliance Module (CM) – Data Sender	Requests the Link Module (described below) to set up a connection with the data receiver; periodically requests the secure log (SL) from the data receiver to verify policy compliance; automatically verifies compliance and warns the user if the verification fails
CM – Data Receiver	Ensures that a data receiver complies with the privacy policy of a data sender; maintains a SL of all transactions involving the sender's private data; sends the SL to the sender when requested
Link Module (LM) – Data Sender	Sets up a connection for sending data to the selected data receiver with a matching privacy policy; tears down the connection once the associated data sending session is finished
LM – Data Receiver	Cooperates with the LM of the data sender to set up the connection for data reception, e.g., provides the port number to use in case there is a need to bypass a firewall
Data Store (DS) – Data Sender	Holds the sender's private information that is to be sent to the data receiver; holds the sender privacy policy received from the sender's PC
DS – Data Receiver	Holds the private information received from the data sender; holds the data receiver privacy policy

The CM uses the secure log to verify that the data receiver complies with the data sender's privacy policy, in terms of the policy fields *Purpose*, *Retention Time*, and *Disclose-To* (see Section II B). Note that it is not necessary to verify *Data Receiver* and *What*, since the data sender sends only the data items mentioned in the matched sender and receiver privacy policies to the data receiver in her policy. The data receiver is responsible for generating the

secure log, with the following requirements: a) there is a header containing the data sharing session identifier, the name of the data sender, and a time-stamp of when the log was started, b) has entries, where each entry is of the form *[time-stamp, operation, (personal data item)]* where *personal data item* may or may not be present, c) an entry is made every time an operation occurs, and d) the log is secured in that an entry once written cannot be altered. Fig. 6 shows an example of a secure log for heart rate (pulse) data, corresponding to the sender privacy policy in Fig. 3a. The secure log entries in Fig. 6 show two types of operations: a "Verify Pulse" operation and an "Erase-29.04.2018" operation. Each entry starts with a time-stamp in the format *dd.mm.yy:hour.minute.second*. Alice's heart rate (pulse) arrives at the data receiver's computer system approximately every 2 minutes. With each arrival, the Verify Pulse operation checks to see if the received pulse (at the end of the entry) is within expected bounds. The Erase-29.04.2018 operation erases all the heart rate data collected on 29.04.2018 (April 29, 2018), which is 31 days from the current date of 30.05.2018 (May 30, 2018). Using this secure log to check the data receiver's compliance with Alice's sender privacy policy (fig. 3a), the CM is able to verify compliance as follows: *Purpose* – the Verify Pulse operation in the secure log is compatible with the purpose of monitoring health and there are no operations that suggest a different purpose, *Retention Time* – the Erase-29.04.2018 operation has deleted pulse data that is 31 days old (the CM keeps track of past Erase operations and knows that the data receiver has always erased data older than 30 days), and *Disclose-To* – there are no operations that suggest that the data has been disclosed to any other party.

Session ID: 21673 Data Sender: Alice Log Started: 30.05.2018:09.29.11
30.05.2018:09.31.10-Verify Pulse-70
30.05.2018:09.33.09-Verify Pulse-72
30.05.2018:09.35.10-Verify Pulse-75
30.05.2018:09.37.11-Verify Pulse-71
30.05.2018:09.39.10-Verify Pulse-69
30.05.2018:09.41.09-Verify Pulse-68
30.05.2018:09.43.10-Verify Pulse-70
30.05.2018:09.45.10-Verify Pulse-70
30.05.2018:09.47.11-Erase-29.04.2018
30.05.2018:09.49.10-Verify Pulse-73
30.05.2018:09.51.09-Verify Pulse-72
30.05.2018:09.53.10-Verify Pulse-71
30.05.2018:09.55.10-Verify Pulse-70
30.05.2018:09.57.11-Verify Pulse-69

Figure 6. Example secure log for heart rate data corresponding to the data sender privacy policy of Fig. 3a.

In addition to the CC itself, the following are also required: a) local and global networking as shown in Fig. 1, and b) interfaces to connect the CC to the smart device. Local and global networking are assumed to be what is most commonly available, i.e., Ethernet, Wi-Fi, IrDA, or Bluetooth for local, and the Internet for global. Smart devices need to have appropriate interfaces that inter-work with the Compliance Controller to carry out policy compliance management (e.g., checking a secure log to verify compliance), connection setup for sending data, and the storage and retrieval of private data.

Fig. 7 presents a message sequence chart showing the interactions between the LMs and CMs of a data sender and a data receiver (only one receiver is shown for simplicity) for a data sending session. As shown, the CC makes a single connection

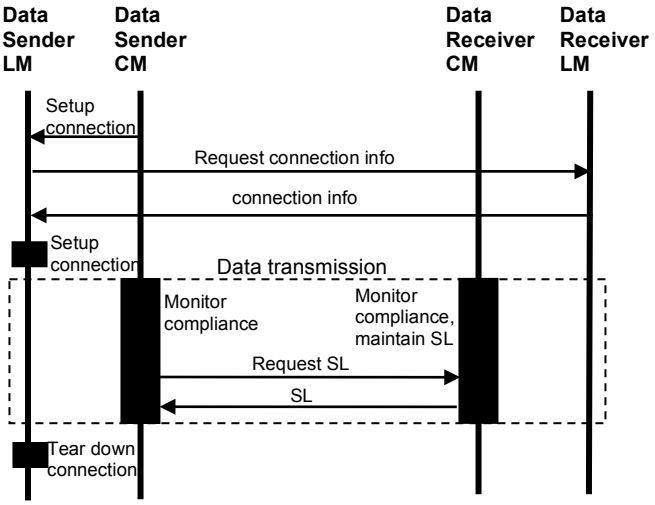

Figure 7. Message sequence chart showing the interactions for a connection setup, data transmission, policy compliance monitoring, and connection teardown.

C. System Configuration

This section considers the system configuration or where the different modules run. If the IoT node that is to send and receive data has sufficient computing capability, the PC and CC may both reside in and run in the node. Examples of such nodes are laptops and smartphones. If this node is less capable computationally but is more capable than the least capable node, one may experiment with having the PC reside in and run in a desktop, laptop or smartphone, while the CC resides in and runs in the node. An example of such a node is a smart refrigerator. Finally, if the IoT node that is to send and receive data is very limited in terms of computational power, both the PC and CC may reside in and run on a desktop, laptop or smartphone, using basic control signals to trigger the node to send data, receive data, or both. In this case, the data would be sent or received through the desktop, laptop, or smartphone. An example of such a node is a simple temperature sensor. Fig. 8 illustrates these three configurations for low, medium, and high computing power. Further, for the medium and low power cases in Fig. 8, each desktop, laptop, or smartphone may run

multiple instances of PC or multiple instances of a PC-CC pair, as required, corresponding to multiple IoT nodes.

The non-privacy preserving IoT network of Fig. 1 is converted to a privacy-preserving IoT network by adding a PC and CC to each smart device or node (Fig. 9) using one of the configurations shown in Fig. 8. In Fig. 9, the double arrows in the PC and CC blow-ups represent expected communication directions based on the functionalities described in Tables II and III. However, the actual communications will depend on how the PC and CC are implemented. Note that as mentioned above, more than one PC or more than one PC-CC pair may run in a desktop, laptop, or smart phone, so the relationship between desktop, laptop, or smart phone and IoT node may be one-to-many. However, Fig. 8 shows this relationship as one-to-one to keep the complexity manageable.

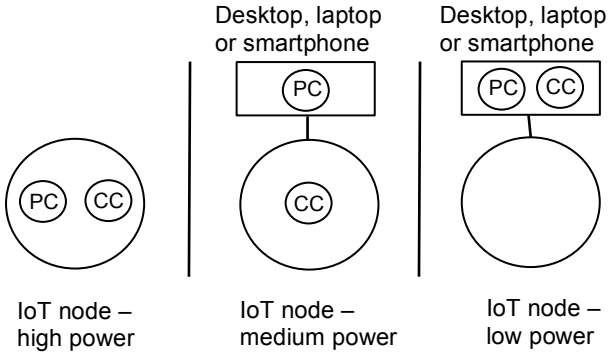

Figure 8. System configurations for the deployment of PC and CC depending on the computational power of the IoT node.

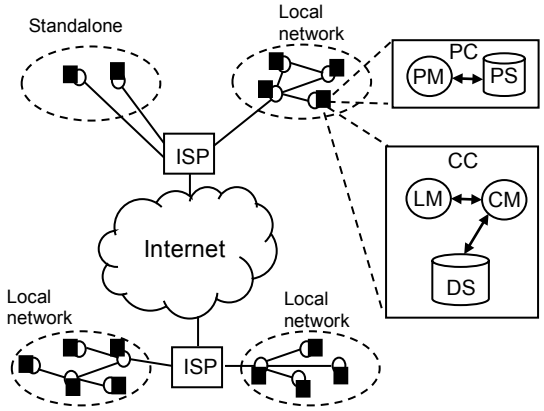

Figure 9. Proposed privacy preserving IoT network; each smart device (small circle) has a PC and CC (solid black rectangle) using one of the configurations in Fig. 7; blow-ups of a PC and CC are also shown (all acronyms defined above).

Although the functionalities of the PC and CC are different between data sender and data receiver, as stated in Tables II and III, the configurations in Fig. 8 apply to both data sender and data receiver. However, in the case where the data receiver is a commercial enterprise, its IoT nodes may consist of desktops or laptops.

Prior to using a smart device to send or receive data, the user accesses the device (possibly through a laptop or smartphone) using some secure form of authentication, such as 2-factor authentication requiring a password and a fingerprint scan. This is needed to protect the user's personal data that is stored in the device and can be satisfied by authentication software within the user's device or within the laptop or smartphone (e.g., part of operating system). As well, any additional security needed to secure the data sender's personal information and privacy policies from attack must be in place. This is satisfied by additional security measures such as certificates and encryption (discussed in Section III D below).

D. Implementation Notes

Some implementation aspects of the framework are considered here.

How does the owner of a data sender come up with her sender privacy policy? It is proposed that data receivers (e-services) routinely advertise their data requirements on the Internet. Note that this is in a way being done today by service websites (e.g., when the user is asked to fill out an online form) and would appear to be a natural way for the receiver to share requirements. Data sender owners can then use the PM to compose the sender policy based on these data requirements. The owners of data receivers also use the PM to compose receiver privacy policies based on how they would like to handle the private information that they receive. Further, data senders and data receivers may only need to create new privacy policies infrequently as they can re-use previous policies from past interactions with the same data receiver or data sender.

The heterogeneous nature of today's smart devices may present some implementation problems for the proposed framework. Some devices may not have sufficient computing power even to be considered as a low power device per Fig. 8. In this case, we concede that such a device would need to be excluded from participation in the proposed framework. A low power device must at least have sufficient computing power to receive signals telling it to send or receive data via the desktop, laptop, or smartphone, as shown in Fig. 8. However, such functionality would require very little computing power, and we expect that the majority of smart devices would possess the power needed.

Further to the need for interfaces to connect the CC to the smart device, mentioned in Section III B, the interfaces and communication links between the desktop, laptop, or smartphone and the smart device would need to be

implemented for medium and low power devices, as shown in Fig. 8. These interfaces and links may be overlaid on top of the ones used for the devices to communicate with one another and the Internet (networked devices, see Section I) and to communicate with the Internet (stand alone devices).

The search for data receivers in the PM may return a reputation value for each receiver. This would help the owner of a data sender to choose which receiver to include in her sender privacy policy. The reputation value may be calculated based on the receiver's history of past transactions, as is done on eBay.com for buyers and sellers. Gupta et al. [6] investigate the design of a reputation system for P2P networks like Gnutella. These authors believe that having reliable reputation information about peers can guide decision making such as whom to trust for a file, similar to this work. However, the choice of a data receiver such as an online grocery store, may depend on other factors such as availability of product, and even personal relationships, e.g. a friend of the data sender works at the grocery store. Nevertheless, a reputation value would be a good place to start.

What does matching of policies mean between data sender and data receiver? This has already been discussed in Section II B. However, an alternative way of comparing two privacy policies is to use a measure of compatibility such as levels of privacy [3]. For this work, matching policies has the meaning explained in Section II B.

Privacy policies need to be amenable to machine processing. Policy languages such as APPEL [7] that are XML-based are good choices. Section VI gives some references for choosing a suitable policy language for implementation.

Any additional security needed to secure the data sender owner's private information and her privacy policies from attack must be installed. Suitable authentication mechanisms, such as the use of certificates, will be needed for data sender / data receiver authentication. Other security mechanisms such as the use of encryption to encrypt the private information will need to be applied or developed and applied. Table IV suggests some security mechanisms that may be employed.

TABLE IV. ADDITIONAL SECURITY MECHANISMS

System Component Requiring Protection	Security Protection Mechanism
data sender / data receiver authentication	SSL with 2-way authentication
Internet communication channels	SSL with 2-way authentication
privacy policies stored in PS and DS	encryption (e.g., 3DES)
personal information stored in DS	encryption (e.g., 3DES)
Software for smart device, PC, and CC	anti-malware tools (e.g., Kaspersky)

In addition, the CC and in particular, the CM, need to be protected from malicious tampering. Since the CM plays the important role of checking for compliance, critical elements of the CM may be implemented in hardware to resist tampering (e.g., by using the Trusted Platform Module [8]). In fact, to further resist tampering, the entire CC may be implemented as a stand-alone hardware module that plugs into the smart device to operate (e.g., via a USB port). It can then be standardized and certified by a trusted authority such as a privacy commissioner to increase user trust.

IV. APPLICATION EXAMPLES

This section provides 2 application examples of the framework described in Section III, a smart refrigerator and a smart watch.

A. Smart Refrigerator

Suppose Alice has a smart refrigerator, which is running low on a number of food items. Alice's refrigerator is connected to the Internet through WI-FI as a node in the privacy-preserving IoT network proposed in this work (see Fig. 10). Before ordering these food items replenished, Alice's refrigerator compares their prices at three online grocers and orders the items from the grocers with the lowest price for each item. The following steps are performed:

1) Alice accesses her laptop (after entering her password), gets on the Internet, and launches her PC. Using network software that was packaged with her PC, she requests to see all grocers located within 10 kilometers of her home who are online. Alice receives a listing of online grocers located within 10 kilometers of her home. (Note: The details of grocer lookup and online messaging are assumed to go on in the background).

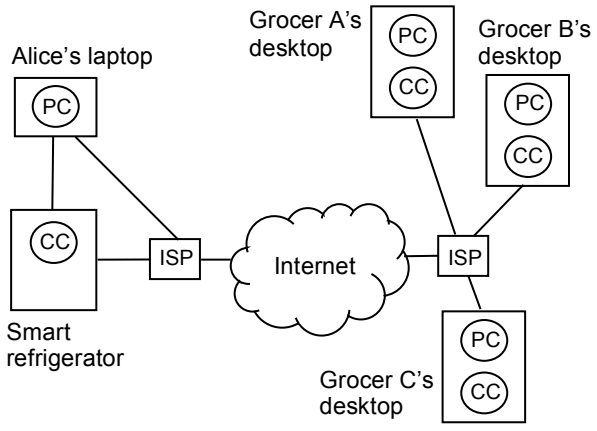

Figure 10. System configuration for smart refrigerator example; the connections from Alice's laptop and from the refrigerator may be Wi-Fi.

2) Alice uses her PC to retrieve her pre-specified privacy policy from her laptop's local storage (PS) and completes it by choosing and including three online grocers (e-services), based on her comfort level with their brand names (e.g., Loblaws, Metro, but shown generically in Fig. 10). Alice actually completes a policy for each grocer, each policy differing only in the *Data Receiver* field. We refer to these policies as Alice's privacy policy.

3) Alice's PC requests the privacy policy of each online grocer that Alice specified in her privacy policy after mutual authentication with each grocer. With the arrival of each grocer's policy, Alice's PC compares Alice's policy with the grocer's policy to see if the policies match up. All grocers' policies match except for one. Alice is asked if she wants to negotiate with the non-matching grocer to try to resolve the non-match. Alice agrees to negotiate and is able to negotiate to a successful conclusion. Now all policies match. Alice's PC sends her sender policy to the PC of each grocer whose policy matches Alice's policy. For added safety, the PC of each grocer receiving Alice's policy does a quick verification of the policy match. If a non-match is found here (unlikely since already checked by Alice's PC) the grocer's PC could terminate the interaction with Alice. Alice's PC sends the sender policy to the CC of the smart refrigerator.

4) The CC in Alice's refrigerator sets up connections between Alice's refrigerator and the three online grocers with the cooperation of the grocers' CCs. Alice's refrigerator then starts sending data to the grocers.

Alice's refrigerator sends personal consumption information to the grocers, such as Alice's favorite brand of food item, her consumption rate for each food item, and the prices that she expects to pay. In return, the online grocers provide Alice's refrigerator with the food items' prices. Alice's refrigerator completes the data transmission, ordering food items from the grocers with the lowest prices. In addition, during and after the transmission, the CM modules of the grocers' respective CCs, continuously checks the grocers' handling of Alice's personal information to ensure compliance with Alice's sender privacy policy. These CM modules log all private data activities to secure logs and sends them to Alice's CC when requested. Alice's CC verifies these secure logs for policy compliance and notifies Alice upon detection of any discrepancy, so that Alice can challenge the grocers' handling of her data when warranted.

B. Smart Watch

Suppose Alice has a smart watch which keeps track of her heart rate, skin temperature and how many steps she's walked during the day. Alice's smart watch is connected to the Internet through cellular LTE as a node in the privacy-preserving IoT network proposed in this work. Alice subscribes to an online health monitoring service called Top Health that encourages her to take remedial action whenever she has not taken a sufficient number of steps in a day, or is about to come down with an illness. The smart watch sends the data mentioned above to the service for use in its diagnoses of Alice's health (see Fig. 11).

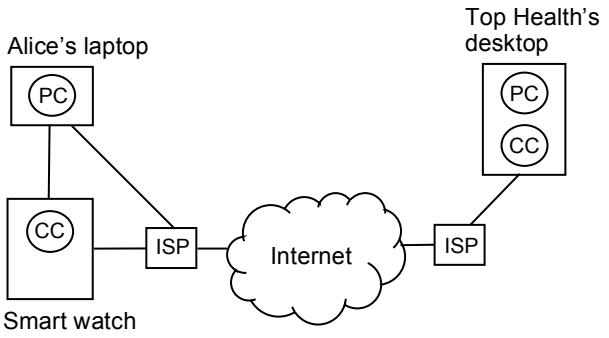

Figure 11. System configuration for smart watch example; the connections are as follows: Bluetooth between the laptop and the smart watch, Wi-Fi between the laptop and the ISP, and cellular LTE between the smart watch and the ISP (provider of both Internet and cellular services).

Alice performs the following steps when she first subscribes to Top Health:

1) Alice accesses her laptop (after entering her password), gets on the Internet, and launches her PC. Using network software that was packaged with her PC, she requests to see all online health monitoring services located in her province. Alice receives a listing of online health monitoring services located in her province. (Note: The details of service lookup and online messaging are assumed to go on in the background).

2) Alice uses her PC to retrieve her pre-specified privacy policy from her laptop's local storage (PS) and completes it with the name of the Top Health health monitoring service, based on her comfort level with Top Health after reading the reviews and recommendations from satisfied customers.

3) Alice's PC requests the privacy policy of Top Health after mutual authentication with it. With the arrival of Top Health's policy, Alice's PC compares Alice's policy with Top Health's policy to see if the policies match up. The policies match. Alice's PC sends her sender policy to the PC of Top Health, which does a quick verification of the policy match for added safety. If a non-match is found here (unlikely since already

checked by Alice's PC), Top Health's PC could terminate the interaction with Alice. Alice's PC sends the sender policy to the CC of the smart watch.

4) The CC in Alice's smart watch sets up a connection between Alice's smart watch and Top Health with the cooperation of Top Health's CC. Alice's smart watch then starts sending data to Top Health.

Alice's smart watch continuously sends the personal health information mentioned above to Top Health, which provides Alice with encouragements to take remedial action when needed. In addition, during the data transmission, the CM module of Top Health's CC, continuously checks Top Health's handling of Alice's personal information to ensure compliance with Alice's sender privacy policy. The CM module logs all private data activities to secure logs and sends them to Alice's CC when requested. Alice's CC verifies these secure logs for policy compliance and notifies Alice upon detection of any discrepancy, so that Alice can challenge Top Health's handling of her data when warranted. Of course, Alice would need to start a new data sending session with Top Health whenever the connection between Alice's smart watch and Top Health is unexpectedly broken, e.g., the smart watch runs out of power. Setting up a new session would simply require a repeat of step 4 above, assuming that the respective privacy policies have not changed since the previous data sharing session, and it is most likely that they have not, since the loss of connection was unintended. If either Alice or Top Health decides to change her/its privacy policy so that the policies no longer match up, then either party would notify the other party and Alice's use of Top Health's service would be ended. Alice may then use her PC to negotiate with Top Health in order to arrive once again at a policy match up. If this negotiation is unsuccessful, Alice may start again at Step 1 in order to choose a different health monitoring service.

V. Strengths and Weaknesses

Some strengths of the proposed framework are: a) upholds personally specified privacy preferences, b) can theoretically be used for all smart devices and all types of receivers or e-services, c) highly scalable due to the use of PCs and CCs (explained below), and d) easy to retrofit an existing non-privacy preserving IoT into a privacy preserving one. One weakness may be that the CM is not trusted to enforce privacy policy compliance. These points are elaborated below.

In terms of the strengths, the proposed framework allows each user to specify her privacy preferences in a privacy policy and for this policy to be upheld. Further, disagreements in privacy policies may be negotiated. Next, the framework allows a privacy preserving "session" to be set-up in which a data sender sends data to a data receiver. It leaves open what computing can be done in the session. Therefore, the session can be an e-commerce session where the data sender is a buyer and the data receiver is a seller, as in the above smart refrigerator example, or a health monitoring session where the data sender is a smart body worn sensor and the data receiver is a medical monitoring service as in the above smart watch example, or any other type of data transmission session that requires privacy protection. Another strength is the fact that the proposed framework is highly scalable. The privacy preserving IoT can be easily expanded by adding or linking PCs and CCs to devices that do not yet possess them (per Fig. 8), where if needed, multiple PC sessions may run in a desktop, laptop, or smartphone. Each additional IoT node so equipped may require a separate privacy policy exchange session. However, the increased cost per additional device is linear. The addition of PCs and CCs does increase network traffic, e.g., requests for the receiver's SL. However, the increased traffic can be accommodated by increasing network capacity, which is consistent with network growth and is not a limiting factor on scalability.

In terms of the weakness of trusting the CM, it must be made clear that malicious attacks on the CC and CM are always possible and could result in violation of privacy. One defense is to make it as hard as possible for those attacks to succeed, by protecting the CM. Ways to protect the CM and build trust for it have already been suggested above.

Reviewers have pointed out the following additional weaknesses: a) enforcement using SLs is not foolproof, i.e., the receiver can still leak personal information using channels not captured by SLs, b) people would need help in defining privacy policies, c) the framework may not apply to less powerful IoT devices, d) the CC may have performance issues in all that it is asked to do, and e) continuous checking of the vendor's handling of private information (Section IV above) could violate the vendor's privacy. These weaknesses are acknowledged, attenuated, or removed as follows. While enforcement using SLs is not foolproof, there is probably no method that is foolproof. As well, there would be tradeoffs to consider between using a more complex enforcement scheme, which is potentially more effective, and the complexity involved in the enforcement. For example, Mont and Thyne [13] (see Section VI) propose a potentially more effective enforcement scheme but which is more complex and thereby more error prone. Nevertheless, replacing SLs with a potentially more effective enforcement method is part of future work. People do need help defining privacy policies, usually through automation. Yee and Korba [12] address this issue (see Section VI) by proposing two semi-automated methods of privacy policy derivation. The framework can be applied to less powerful devices by implementing the PC and CC as software modules running on a desktop, laptop, or smartphone which is connected to the device, as mentioned above. In this scenario, the smart device merely has to be signaled to forward/receive its data

to/from the desktop, laptop, or smartphone running the CC, a change that should be implementable on even the least compute capable smart device. In terms of the CC potentially having performance issues, this is a possibility, especially if the smart device is not very powerful. This potential problem would be mitigated to some extent if the CC were to run on a desktop, laptop, or smartphone. In any case, this potential issue will be addressed through prototyping the CC, a part of future work. Finally, with regard to the possible violation of the vendor's privacy by the continuous checking of the vendor's handling of private information, note that this continuous checking is performed by the vendor's CC running on the vendor's platform for the benefit of the vendor so that the vendor can be assured that it is complying with the sender's privacy policy. Since there is no data associated with this checking that is forwarded back to the sender (only the SL is forwarded back to the sender – see Table III) there can be no violation of the vendor's privacy. It should also be noted here that the SL does not violate the vendor's privacy either, as it only refers to the sender's private information and how the receiver processed it in terms of the sender's privacy policy. In other words, the SL does not contain any vendor private information.

VI. Related Work

This work shares the notion of using controllers to monitor privacy policy compliance with an earlier work [9] in which we applied "privacy controllers" to protect privacy in web services. In this work, we have updated and re-designed the components in [9] to apply to the IoT.

Works that are related in terms of the application of personal privacy policies to implement privacy preferences are as follows. Yee [10] proposed a hybrid centralized / P2P architecture for ubiquitous computing that also protects privacy using privacy policies. Yee and Korba [11] examine privacy policy compliance for web services, and Yee and Korba [12] discuss privacy policy derivation. Another related work in this area is Mont and Thyne [13], which gives an approach for automatic privacy policy enforcement within an enterprise, by making data access control privacy-aware. Their approach incorporates a "Privacy Policy Decision Point" which makes decisions for allowing access based on privacy policies, and a "Data Enforcer" which intercepts attempts to access personal data and enforces the decisions made at the Privacy Policy Decision Point. Thus, their work is an example of enforcement other than checking a secure log as done in this work. However, their approach does not cover other privacy requirements such as purpose, retention time, and data disclosure. In terms of implementation languages for privacy policies, Kasem-Madani and Meier [14] overview 27 security and privacy policy languages and categorize them using a categorization framework. They also identify areas not covered by the languages. Kumaraguru et al. [15] summarize the literature available (at the time of their paper's publication) on privacy policy languages. They describe the features, characteristics, and requirements of the languages. They also provide a comprehensive framework for analysis and expect their work to aid the implementer in choosing a suitable language.

In the privacy literature for IoT, the following authors identify or analyze the security and privacy issues of the IoT. Loi et al. [16] develop a systematic method for identifying the security and privacy shortcomings of various IoT devices in order to alert consumers, manufacturers, and regulators to the associated risks. They apply their method to evaluate twenty market-ready consumer IoT devices and present their findings. Liu et al. [17] examine solutions for establishing smart devices in a smart home and demonstrate that such solutions may be error prone in terms of security and privacy. They argue that if the security and privacy issues are not considered, devices using a solution are inevitably vulnerable, seriously threatening the security and privacy of the smart home. Siby et al. [18] propose IoTScanner, a system that allows for passive, real-time identification and monitoring of an existing wireless infrastructure used for connecting IoT devices. Using this system, they identify privacy threats and investigate metrics for classifying devices. Kumar et al. [19] discuss the privacy and security concerns in IoT focusing on common IoT vulnerabilities such as distributed denial of service and data modification attacks. Their goal is to present the security and privacy concerns of IoT environments and the existing protection mechanisms.

In the privacy literature for IoT, the following authors propose partial or entire privacy protection solutions for the IoT. Kanuparthi et al. [20] describe privacy protection through the use of security measures such as encryption, which is the traditional approach to protection rather than checking for compliance to privacy policy as in our work. Alqassem [21] presents a privacy and security requirements framework for developing IoT, taking account of these requirements from the beginning of development, which is a software engineering approach distinct from our approach. Zhang et al. [22] consider privacy preservation in the application layer of the IoT and construct application scenarios to identify privacy preservation challenges. They look at privacy preservation in terms of maintaining confidentiality and examine various authentication schemes as means for providing confidentiality whereas we look at privacy preservation in terms of legislated privacy rights. Davies et al. [23] look at privacy protection for raw data that is streamed directly from IoT sensors to the cloud. They propose the use of a privacy mediator on every raw sensor stream. Each mediator is part of the same administrative domain as the sensors whose data is being streamed, and

dynamically enforces the privacy policies of the sensor owners. The use of privacy mediators in [23] is similar to the use of privacy controllers in our work. However, the application area is restricted to raw data streamed by sensors to the cloud and there is no mention of privacy policy matching or negotiation. Savola et al. [24] consider e-health applications in the IoT, such as biomedical sensor networks, as holding great promise but security and privacy are major concerns. They propose a high-level adaptive security management mechanism based on security metrics to cope with these concerns. Thus their approach is quite different than ours in their use of metrics to drive security management. Joy et al. [25] present a scheme to ensure granular location privacy for GPS enabled IoT devices. They accomplish this by designing and implementing a privacy module within the GPSD daemon, a low level GPS interface that runs on GPS enabled devices, thus giving data owners granular control over the release of their GPS location. Their proposal differs from ours in that they are only concerned with location privacy for GPS enabled devices, and of course, their method for protecting this privacy is different from ours. Pacheco et al. [26] study privacy preserving architectures for integrating the IoT with cloud computing. Their main concern for these architectures is to investigate the feasibility of implementing security and privacy mechanisms in IoT devices that are severely constrained in terms of computing resources. Their intention is to show that if such mechanisms can work in these constrained devices, they will work in almost all other devices. Thus, their approach is to safeguard privacy using traditional security and privacy mechanisms installed within the IoT devices (e.g., encryption), which is different than our approach of using privacy policies and verifying compliance to the policies. Appavoo et al. [27] address privacy-preserving access to sensor data for IoT based services such as health monitoring services. They observed that a large class of applications can function based on simple threshold detection, e.g., blood pressure above a pre-determined threshold. They propose a privacy-preserving approach based on this observation, their goal being to minimize privacy loss in the presence of untrusted service providers. The main algorithm in their proposed approach is an anonymization scheme that uses a combination of sensor aliases to hide the identity of the sensor data source, together with initialization vectors (or filters) to reveal information only to relevant service providers. Appavoo et al.'s work differs from this work in at least two ways. First, their work addresses a particular segment of services (monitoring services) whereas this work is applicable to all types of services. Second, they protect privacy through anonymizing the source of private information and restricting the private information to service providers that need to know. This work protects privacy through privacy policies and ensuring that the service provider complies with the policies.

VII. Conclusions and Future Work

This work has proposed a straightforward elaborated framework to protect privacy in the IoT, making use of compliance controllers together with sender and receiver privacy policies. In this framework, privacy is protected through compliance with sender privacy preferences, expressed as sender privacy policies. This work has greatly expanded the original CYBER 2016 paper by providing additional details for all sections.

Once privacy is protected, the smart devices in the IoT can engage in many applications, such as e-commerce (smart refrigerator using replenish food services) and e-health (smart body worn sensors using a health monitoring service).

The framework presented here is only theoretical. The effectiveness of the framework remains to be proven through prototyping and experimentation. However, much like a blueprint for a building, some security, performance and scalability aspects can already be predicted.

Future work includes the construction of a prototype to fine-tune the proposed framework, determine its effectiveness, and investigate some of the ideas discussed in the implementation notes, such as the use of reputation and other factors to help data senders decide which data receivers to select. We also plan to investigate other means of enforcing compliance with privacy policy that do not involve verifying SLs. Lastly, we plan to take a closer look at applying the framework to the transmission of private data from e-health smart devices in the wearable world (e.g., fitness trackers, smart watches, elders' call-for-help pendants).

References

[1] G. Yee, "An Approach for Protecting Privacy in the IoT," Proc. The First International Conference on Advances in Cyber-Technologies and Cyber-Systems (CYBER 2016), pp. 60-66, Oct. 9-13, 2016.

[2] I. Goldberg, D. Wagner, and E. Brewer, "Privacy-Enhancing Technologies for the Internet," Proc. IEEE COMPCON'97, pp. 103-109, Feb. 23-26, 1997.

[3] G. Yee and L. Korba, "Comparing and Matching Privacy Policies Using Community Consensus," Proc. 16th IRMA International Conference, in Managing Modern Organizations with Information Technology, edited by Mehdi Khosrow-Pour, pp. 208-211, 2005. Available Aug. 23, 2016: http://www.irma-international.org/proceeding-paper/comparing-matching-privacy-policies-using/32576/

[4] G. Yee and L. Korba, "The Negotiation of Privacy Policies in Distance Education," Proc. 14th IRMA International Conference, pp. 702-705, May 18-21, 2003. Available Aug. 23, 2016: http://www.irma-international.org/proceeding-paper/negotiation-privacy-policies-distance-education/32116/

[5] Office of the Privacy Commissioner of Canada, "PIPEDA fair information principles," available as of January 15, 2018 at:

https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/p_principle/

[6] M. Gupta, P. Judge, and M. Ammar, "A Reputation System for Peer-to-Peer Networks," Proc. 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '03), pp. 144-152, June 2003.

[7] W3C, "A P3P Preference Exchange Language 1.0 (APPEL1.0)," available as of May 14, 2016 at: http://www.w3.org/TR/P3P-preferences/

[8] Trusted Computing Group, "Trusted Platform Module (TPM)," available as of May 14, 2016 at: http://www.trustedcomputinggroup.org/work-groups/trusted-platform-module/

[9] G. Yee, "A Privacy Controller Approach for Privacy Protection in Web Services," Proc. ACM Workshop on Secure Web Services (SWS '07), pp. 44-51, Oct. 29 – Nov. 2, 2007.

[10] G. Yee, "A Privacy-Preserving UBICOMP Architecture," Proc. Privacy, Security, Trust 2006 (PST 2006), pp. 224-232, 2006.

[11] G. Yee and L. Korba, "Privacy Policy Compliance for Web Services," Proc. 2004 IEEE International Conference on Web Services (ICWS 2004), pp. 158-165, July 6-9, 2004.

[12] G. Yee and L. Korba, "Semi-Automatic Derivation and Use of Personal Privacy Policies in E-Business," International Journal of E-Business Research, Vol. 1, Issue 1, pp. 54-69, 2005.

[13] M. C. Mont and R. Thyne, "A Systemic Approach to Automate Privacy Policy Enforcement in Enterprises," Proc. 6th Annual International Workshop on Privacy Enhancing Technologies (PET 2006), pp. 118-134, June 28-30, 2006.

[14] S. Kasem-Madani and M. Meier, "Security and Privacy Policy Languages: A Survey, Categorization and Gap Identification," in Cornell University Library, Dec. 2015. Available May 30, 2018: https://arxiv.org/abs/1512.00201

[15] P. Kumaraguru, L. F. Cranor, J. Lobo, and S. B. Calo, "A Survey of Privacy Policy Languages," Workshop on Usable IT Security Management (USM 07), in Proc. 3rd ACM Symposium on Usable Privacy and Security, pp. 1-4, July 2007. Available May 30, 2018: https://cups.cs.cmu.edu/soups/2007/workshop/Privacy_Policy_Languages.pdf

[16] F. Loi, A. Sivanathan, H. Gharakheili, A. Radford, and V. Sivaraman, "Systematically Evaluating Security and Privacy for Consumer IoT Devices," Proc. 2017 Workshop on Internet of Things Security and Privacy (IoTS&P '17), pp. 1-6, 2017.

[17] H. Liu, C. Li, X. Jin, J. Li, Y. Zhang, and D. Gu, "Smart Solution, Poor Protection: An Empirical Study of Security and Privacy Issues in Developing and Deploying Smart Home Devices," Proc. 2017 Workshop on Internet of Things Security and Privacy (IoTS&P '17), pp. 13-18, 2017.

[18] S. Siby, R. Maiti, and N. Tippenhauer, "IoTScanner: Detecting Privacy Threats in IoT Neighborhoods," Proc. 3rd ACM International Workshop on IoT Privacy, Trust, and Security (IoTPTS '17), pp. 23-30, 2017.

[19] N. Kumar, J. Madhuri, and M. ChanneGowda, "Review on Security and Privacy Concerns in Internet of Things," Proc. 2017 International Conference on IoT and Application (ICIOT), pp. 1-5, 2017.

[20] A. Kanuparthi, R. Karri, and S. Addepalli, "Hardware and Embedded Security in the Context of Internet of Things," Proc. 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles (CyCAR'13), pp. 61-65, Nov. 4, 2013.

[21] I. Alqassem, "Privacy and Security Requirements Framework for the Internet of Things (IoT)," Proc. ICSE Companion'14, pp. 739-741, May 31-June 7, 2014.

[22] Z.-K. Zhang, M. Cho, and S. Shieh, "Emerging Security Threats and Countermeasures in IoT," Proc. 10th ACM Symposium on Information, Computer and Communications Security (Asia CCS '15), pp. 1-6, 2015.

[23] N. Davies, N. Taft, M. Satyanarayanan, S. Clinch, and B. Amos, "Privacy Mediators: Helping IoT Cross the Chasm," Proc. 17th International Workshop on Mobile Computing Systems and Applications (HotMobile '16), pp. 39-44, 2016.

[24] R. Savola, H. Abie, and M. Sihvonen, "Towards Metrics-Driven Adaptive Security Management in e-health IoT Applications," Proc. 7th International Conference on Body Area Networks (BodyNets '12), pp. 276-281, 2012.

[25] J. Joy, M. Le, and M. Gerla, "LocationSafe: Granular Location Privacy for IoT Devices," Proc. Eighth Wireless of the Students, by the Students, and for the Students Workshop (S3), pp. 39-41, 2016.

[26] L. Pacheco, E. Alchieri, and P. Barreto, "Enhancing and Evaluating an Architecture for Privacy in the Integration of Internet of Things and Cloud Computing," Proc. 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), pp. 1-8, 2017.

[27] P. Appavoo, M. C. Chan, A. Bhojan, and E.-C. Chang, "Efficient and Privacy-Preserving Access to Sensor Data for Internet of Things (IoT) Based Services," Proc. 8th International Conference on Communication Systems and Networks (COMSNETS), pp. 1-8, 2016.