An Approach for Protecting Privacy in the IoT

George O. M. Yee

Computer Research Lab
Aptusinnova Inc.
Ottawa, Canada
email: george@aptusinnova.com

Dept. of Systems and Computer Engineering
Carleton University
Ottawa, Canada
email: 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 proposes 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.**

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

I. INTRODUCTION

The objective of this paper is to present an approach 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.

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, Personal Digital Assistants (PDAs), smartphones, workstations, smart sensors, smart appliances) 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.

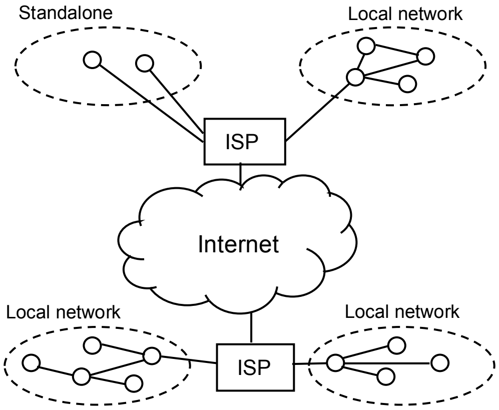


Figure 1. IoT network environment (ISP = Internet Service Provider, circles are smart devices)

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 approach. Section IV gives an example of applying the approach. Section V evaluates the approach by discussing some strengths and weaknesses. Section VI examines related work. Section VII concludes the paper and lists some ideas for future research.

II. PRIVACY POLICIES

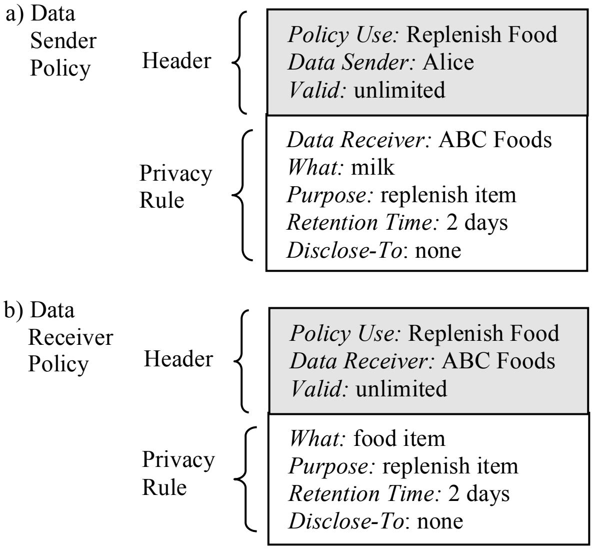
A. *Privacy*

As defined by Goldberg et al. in 1997 [1], 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 her receiver privacy policy (which is compatible with the data sender's privacy policy). Foolproof mechanisms must be in place to ensure compliance. The mechanics of privacy policy matching [2] and negotiation [3] are outside the scope of this work.

Fig. 2 shows example sender and receiver privacy policies for a smart refrigerator.



a) Data Sender Policy

Header

Policy Use: Replenish Food
Data Sender: Alice
Valid: unlimited

Privacy Rule

Data Receiver: ABC Foods
What: milk
Purpose: replenish item
Retention Time: 2 days
Disclose-To: none

b) Data Receiver Policy

Header

Policy Use: Replenish Food
Data Receiver: ABC Foods
Valid: unlimited

Privacy Rule

What: food item
Purpose: replenish item
Retention Time: 2 days
Disclose-To: none

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

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. 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. Thus, a data receiver who complies with such privacy policies also complies with a data sender's legislated privacy rights.

III. APPROACH

For each smart device, the approach 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, or mobile device such as a smart phone or tablet. The components and functionality of the PC are given in Table 1.

TABLE 1. 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
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. 3 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.

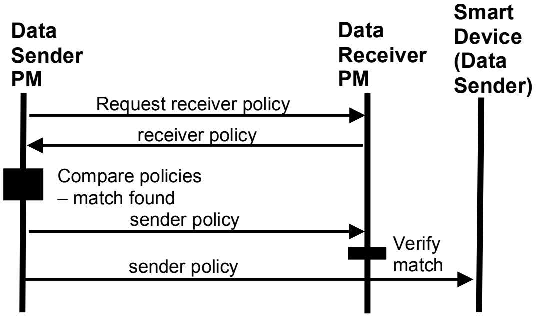


Figure 3. Message sequence chart showing the interactions for a first time successful policy match.

Fig. 4 shows the same scenario as Fig. 3 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.

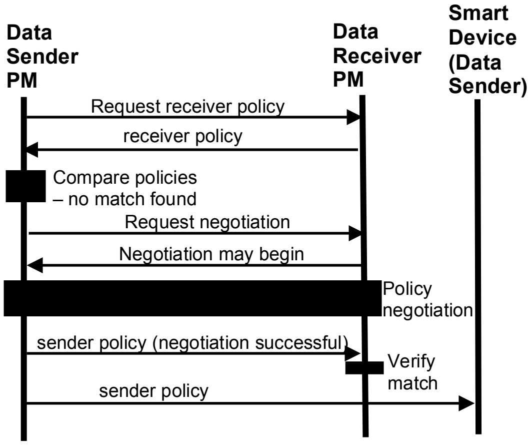


Figure 4. Message sequence chart showing the interactions for a first time unsuccessful policy match and the ensuing negotiation (assumed successful).

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 "attached" to the device. The components and functionality of the CC are given in Table 2. In Table 2, 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 2. COMPLIANCE CONTROLLER (CC)

CC Component	Functionality
Compliance Module (CM) – Data Sender	Requests the LM 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

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. 5 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.

The non-privacy preserving IoT network of Fig. 1 is converted to a privacy-preserving IoT network by adding a CC to each smart device or node (Fig. 6). In Fig. 6, the double arrows in the CC blow-up represent expected communication directions based on the functionalities described in Table 2. However, the actual communications will depend on how the CC is implemented.

Prior to using a smart device to send or receive data, the user accesses the device 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 (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 C below).

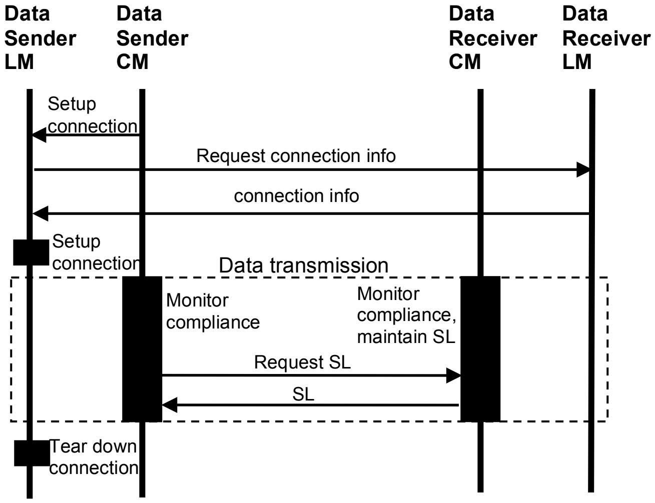


Figure 5. Message sequence chart showing the interactions for a connection setup, data transmission, policy compliance monitoring, and connection teardown.

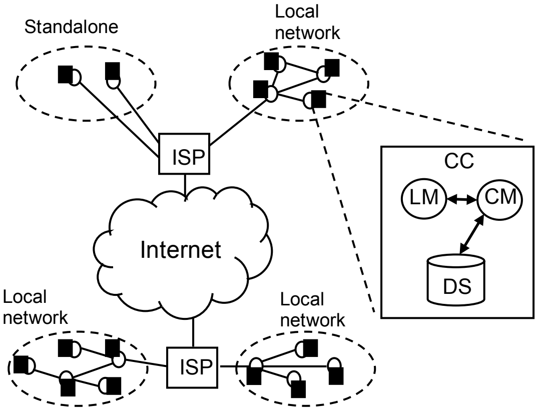


Figure 6. Proposed privacy preserving IoT network; each smart device (small circle) has a CC (black rectangle); a blow up of a CC is also shown (all acronyms defined above).

C. Implementation Notes

Some implementation aspects of the approach 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). 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.

The heterogeneous nature of today's smart devices may present some implementation problems for the proposed approach. Some devices may not have sufficient computing power to host the CC in addition to the required software for the device. However, this may depend on how the CC is implemented. If the CC is implemented as a stand-alone module running on a tablet or smartphone "attached" to the device (as mentioned above), and only needs to communicate with the smart device to operate, then the device can be less powerful.

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. [4] 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.

What does matching of policies mean between data sender and data receiver? There needs to be a way of comparing two policies using some measure of compatibility such as levels of privacy [2].

Privacy policies need to be amenable to machine processing. Policy languages such as APPEL [5] that are XML-based are good choices.

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 3 suggests some security mechanisms that may be employed.

TABLE 3. 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)
smart device software, CC software	anti-malware tools (e.g., Kaspersky)

ISBN: 978-1-61208-512-8

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 [6]). 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 EXAMPLE

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. 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 along with their reputations. (Note: The details of grocer 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 by choosing and including three online grocers (e-services), based on their reputations.

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.

V. EVALUATION

Some strengths of the proposed approach 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 CCs, 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 approach 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 approach 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 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 for the aged, or any other type of data transmission session that requires privacy protection. Another strength is the fact that the proposed approach is highly scalable. The privacy preserving IoT can be easily expanded by adding CCs to devices that do not yet possess them. Each additional device so equipped would also require a privacy policy exchange session. However, the increase cost per additional device is linear. The addition of 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 of this work have pointed out additional weaknesses, as follows: 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 approach 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 [11] (see next Section) 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 [10] address this issue (see next Section) by proposing two semi-automated methods of privacy policy derivation. The approach can be applied to less powerful devices by implementing the CC as a software module running on a smartphone or tablet which is connected to the device, as mentioned above. In this scenario, the smart device merely has to forward its data to the smartphone or tablet 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 smartphone or a tablet. 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 2) there can be no violation of the vendor's privacy. It should also be noted here that the SL would 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 should 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 [7] in which we applied "privacy controllers" to protect privacy in web services. In this work, we have updated and re-designed the components in [7] 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 [8] proposed a hybrid centralized / P2P architecture for ubiquitous computing that also protects privacy using privacy policies. Yee and Korba [9] examine privacy policy compliance for web services, and Yee and Korba [10] discuss privacy policy derivation. Another related work in this area, as suggested by a reviewer, is Mont and Thyne [11], 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.

In the privacy literature for IoT, Kanuparthi et al. [12] describe privacy protection through the use of security measures such as encryption. Alquassem [13] presents a privacy and security requirements framework for developing IoT, taking account of these requirements from the beginning of development. Zhang et al. [14] describe the security challenges in the IoT and examine conventional security mechanisms (e.g., authentication) to look for countermeasures. Davies et al. [15] state that unease over data privacy will retard consumer acceptance of IoT deployments. They consider this to be primarily due to lack of user control over raw data that is streamed directly from sensors to the cloud and propose the use of privacy mediators on every data stream. Savola et al. [16] 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. A reviewer has suggested Appavoo et al. [17] as another related work. These authors address privacy-preserving access to sensor data for IoT based services such as health monitoring services. Appavoo et al. 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 effective approach to protect privacy in the IoT, making use of compliance controllers together with sender and receiver privacy policies. In this approach, privacy is protected through compliance with privacy preferences, expressed as sender privacy policies.

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 approach presented here is only theoretical. The effectiveness of the approach remains to be proven through prototyping and experimentation.

Future work includes the construction of a prototype to fine-tune the proposed approach, determine its effectiveness, and investigate some of the ideas discussed in the implementation notes, such as the use of reputation 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, as well as applying the approach to the transmission of private data from e-health smart devices in the wearable world (e.g., Apple watch).

REFERENCES

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

[2] 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/

[3] 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/

[4] 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, pp. 144-152, 2003.

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

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] K. L. Zhang, M. C. Y. 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.

[15] 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.

[16] R. M. 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.

[17] 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.