IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 9, 2018, accepted March 24, 2018, date of publication March 29, 2018, date of current version May 2, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2820716

A User-Friendly Privacy Framework for Users to Achieve Consents With Nearby BLE Devices

SHI-CHO CHA[1], (Senior Member, IEEE), MING-SHIUNG CHUANG[1,2], KUO-HUI YEH[3], (Senior Member, IEEE), ZI-JIA HUANG[1], AND CHUNHUA SU[4]

[1]Department of Information Management, National Taiwan University of Science and Technology, Taipei 10607, Taiwan
[2]Criminal Investigation Bureau, Taipei 11072, Taiwan
[3]Department of Information Management, National Dong Hwa University, Hualien 97401, Taiwan
[4]Division of Computer Science, University of Aizu, Aizuwakamatsu 965-8580, Japan

Corresponding author: Chunhua Su (chsu@u-aizu.ac.jp)

This work was supported in part by the Academia Sinica, in part by the Taiwan Information Security Center, and in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 105-2221-E-011-079-MY3, Grant MOST 105-2221-E-259-014-MY3, Grant MOST 105-2221-E-011-070-MY3, and Grant MOST 106-3114-E-011-003. The work of C. Su was supported by JSPS KAKENHI under Grant 15K16005.

ABSTRACT The deployment of IoT devices with significant data collection capabilities around the world raises concerns about user privacy. People are worried about ubiquitous IoT devices collecting and sharing their data with unknown parties without their awareness or consent. Currently, several governmental agencies have stated that IoT service providers should obtain user consent before collecting and using their personal data. However, to the best of our knowledge, there is no standard means for users to reach agreements on privacy practices for IoT applications. Among different types of IoT applications, this paper focuses on the scenario in which people use their personal smartphones to access nearby IoT devices via Bluetooth Low Energy (BLE). To address the privacy issue in the scenario, this paper proposes a privacy preferences expression framework for BLE-based applications named PrivacyBat. The framework defines specifications for users to achieve agreements on privacy practices with nearby BLE devices. In addition, this framework provides guidelines for a device to process user requests according to the agreement. To demonstrate how the framework operates, this paper further provides a proof of concept implementation. As the proposed framework can improve the privacy policy agreement process in IoT applications, this paper can hopefully contribute to increasing user trust in IoT applications.

INDEX TERMS Bluetooth low energy (BLE), BLE privacy, informed consent, Internet of Things (IoT), IoT privacy.

I. INTRODUCTION

With the advances of IoT technologies, an increasing number of organizations can now deploy sensors, actuators and other IoT devices around the world to provide IoT application services. However, when people interact with the IoT devices, the devices may collect personal data. For example, operators of auto lighting control systems, such as Philips Hue, can collect lightbulb control events and use the information to derive the behavior patterns of users. According to recent surveys [1], [2], there is a considerable proportion of people worried about the privacy risks of ubiquitous IoT devices with significant data collection capabilities. Such privacy concerns may become a critical obstacle to the growth and adoption of IoT applications. In light of this, several governmental agencies, such as the US FTC and EU Article 29 Working Party, have stated that IoT service providers should obtain user consent before collecting and using personal data [3], [4]. However, most IoT applications may not properly inform and acquire user consent. For example, IoT application providers may just post notices on walls to notify users of the existence of IoT devices. This results in ‘‘low-quality’’ consents described by the report of the EU Article 29 Working Party and leads to users lacking trust in IoT devices.

As major smartphone platforms, such as iOS and Android, support the BLE (Bluetooth Low Energy) specification, BLE has become a de facto standard in scenarios where a user employs his/her smartphone to access nearby IoT or wearable devices. This study focuses on this scenario and proposes a Privacy Preferences Expression Framework for BLE-based applications named PrivacyBat. The framework provides a standard format and method for administrators of IoT devices to present the privacy policies of their devices to the user. The framework enables device administrators to register their devices and privacy polices. Therefore, when users discover nearby BLE devices, they can find associated privacy policies via the interface defined in the proposed framework. The proposed framework also defines a standard means for users to notify BLE devices of their privacy preferences. The devices can then follow user preferences to process personal data. To the best of our knowledge, there is no standard or research on how users can reach an agreement with BLE-based IoT application providers on privacy practices. This study can hopefully contribute to the improvement of the quality of consent for BLE-based IoT applications.

The rest of this paper is organized as follows: Section 2 introduces the preliminary knowledge and related work. Section 3 provides an overview of our proposed framework. This study then describes the major components of the proposed framework and the rationales behind the components from Section 4 to Section 6. We further demonstrates how the proposed framework works with a proof of concept implementation in Section 7. Finally, conclusions are drawn in Section 8 along with recommendations for future research.

II. BACKGROUND KNOWLEDGE AND RELATED WORK

In this section, we present the preliminary knowledge related to BLE security and privacy, potential threats from privacy invasions and possible countermeasures.

A. BLUETOOTH LOW ENERGY AND ITS SECURITY AND PRIVACY MECHANISMS

BLE was merged into the Bluetooth standard with Bluetooth 4.0 Core Specification [5]. Compared to the traditional Bluetooth specification, BLE provides a means for devices to communicate with one another with lower power consumption [6]. Generally, a BLE-enabled device (or simply a device) can use the following means to communicate with other devices:

- A device can broadcast (or advertise) messages containing information to nearby devices. Therefore, a nearby device can receive the advertised messages.
- A device (or a central device) can connect to another device (or a peripheral device) to use services provided by the peripheral device.

The bonding process is critical to BLE security and privacy mechanisms. After a central device connects to a peripheral device, either one of the devices can request to initiate the bonding process [5] and [7]. Before the two devices establish a bonding relationship, the two devices need to pair with each other. In the pairing process, the two devices will exchange security features, such as I/O capability, to decide a pairing scheme. There are four different pairing schemes:

- Numeric comparison. The two devices generate a six digit number mutually and display the number on their screens. Therefore, the owner of a device can authenticate that his/her device is pairing with the right device by checking whether the two devices display the same number.
- Passkey entry. One device displays a randomly generated six digit number and requests the owner of the other device to input the displayed number. The other device will then transfer the inputed number back to the first device for checking. Therefore, the former can ensure the owner of the latter device has seen the displayed number and input the number for authentication.
- Out of band. A device can generate a key and transfer the key to the other device using channels other than BLE, such as NFC and USB sticks. Then, the device can use the key to verify that it is pairing with the device that the key is delivered to.
- Just work. Two devices exchange information to generate a temporary key for the bonding process.

To sum up, BLE allows a device to verify the device it is pairing with if a numeric comparison, passkey entry, or out of band pairing scheme is used. No matter which scheme is adopted, the two devices will exchange information to generate a temporary key for further use. Then, a device can use the temporary key to exchange the following keys with the other device in the bonding process:

- A Long Term Key (LTK) for the device to encrypt data transferred to the other device.
- A Connection Signature Resolving Key (CSRK) for the device to generate signatures of transferred data to enable the other device to verify the integrity of the transferred data.
- An Identity Resolving Key (IRK) used by the device to generate random addresses to prevent others from tracking its address. To protect user privacy, the Bluetooth specification defines the random device address feature [5]. Simply speaking, the random address feature enables BLE devices to change their Bluetooth MAC addresses so that others cannot track the devices based on their addresses. The random addresses can further be classified into non-resolvable addresses and resolvable addresses. A non-resolvable address is generated randomly and there is no way to identify the device using the address. If the device uses a resolvable address, then the other devices will be able to identify the device if they have the IRK used by the device to generate the resolvable address.

Note that the original BLE pairing protocol is vulnerable to brute force attacks [8]. Malicious individuals may eavesdrop on the messages exchanged in the pairing process and derive the temporary keys and ultimately be able to eavesdrop on all future communication. To address this vulnerability, the Bluetooth specification has enabled two devices to use elliptic curve cryptography to establish secure connections for key exchanging from version 4.2 onward. TABLE 1 summarizes the security and privacy mechanisms of BLE.

TABLE 1. Implementation environment.

Objective	Description
Authentication	A device can authenticate other devices during the pairing process.
Confidentiality	Two devices can encrypt transferred data to protect data confidentiality.
Integrity	A device can check the integrity of data transferred from a bonded device based on the associated signature.
Privacy	A device can use random addresses to prevent itself from being tracked.

B. THREATS OF PRIVACY INVASION AND COUNTERMEASURES

This subsection discusses the threats of privacy invasion when people use their smartphones to access nearby BLE devices. As depicted in Figure 1, when a user communicates with a BLE-based sensor using his/her smartphone, a malicious BLE device may try to invade user privacy passively or actively.

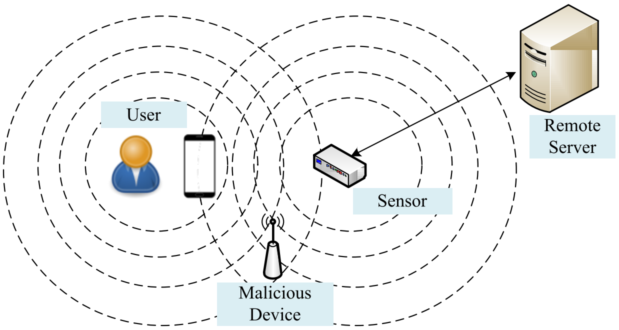


FIGURE 1. The scenario for threat modeling.

First, a malicious device may eavesdrop on messages transferred between the user's smartphone and the sensor passively. Then, the malicious device may obtain the smartphone's BLE MAC address or other advertised identifiers to track the user. If the sensor is a wearable device, the malicious device may be able to track the user via the MAC address of the sensor. To solve this issue, the Bluetooth specification has defined the random address scheme to prevent a device from being identified by unauthorized parties, as described in Section 2.1. Moreover, to address the problem that legacy BLE devices could not support the random address scheme, Fawaz et al. proposed the BLE-Guardian to "hide" a BLE device by invoking jamming to prevent adversaries from obtaining advertised messages of the device [9].

The malicious device may also collect personal data from messages transferred between the user's smartphone and the sensor. To mitigate the threat, the user's smartphone and the sensor can encrypt personal data before transferring the data. For example, the user's smartphone and the sensor can adopt the means defined in the Bluetooth specification to encrypt transferred messages.

A malicious device may invade user privacy actively. Even if a BLE device adopts the random address scheme, if the device responds to scanning requests issued by an another device, a malicious person may record the scanning requests and replay the requests to identify the device. Therefore, Ping Wang proposed to use counters to enhance the existing Bluetooth specification to overcome this vulnerability [10]. If a sensor provides GATT services that enable others to access stored personal data, unauthorized people may access the personal data if the sensor does not adopt an appropriate authentication and access control mechanism. Besides relying on the BLE authentication mechanism, BLE application providers may also implement their own authentication and access control mechanisms [11]. Furthermore, unauthorized people may steal personal data via physical attack. This study does not address the threats of physical attacks.

Even if there is no malicious device trying to collect personal data, when a person uses his/her smartphone to access a BLE device, the person may not be able to obtain the privacy policies of the device. This may violate current personal data protection rules [3], [4]. To address this issue, [12] and [13] have proposed negotiation mechanisms for application providers to negotiate with users. Using these mechanisms, application providers can reach agreements with users on privacy practices. However, the proposed negotiation mechanisms do not apply for device to device BLE communication.

As there have been limited efforts focusing on enabling a BLE device to obtain user consent to collect and use personal data, this study provides a means for a user to obtain the privacy practices of a BLE device and send his/her privacy preferences to the device over BLE.

III. OVERVIEW OF THE PROPOSED FRAMEWORK

Figure 2 gives an overview of the proposed PrivacyBat framework. The kernel of the framework is a *Device Information Service*. As illustrated in Figure 2, the *Device Information Service* provides two major interfaces:

- The *Device Registration and Management* interface enables an authorized device administrator to register a device using its device identity. Each device that supports the PrivacyBat framework has a unique 128-bit UUID (Universally unique identifier). The administrator can upload and manage device information and associated privacy policies by means of device identity via the interface. In this case, the framework defines *Ontologies for Device Information and Privacy Policies* based on UPnP and P3P, respectively. This study will describe the details in Section 4.
- Each device that supports the PrivacyBat framework should advertise its UUID periodically, similar to iBeacons [14]. Therefore, a person can collect UUIDs of nearby BLE devices and query the Device Information

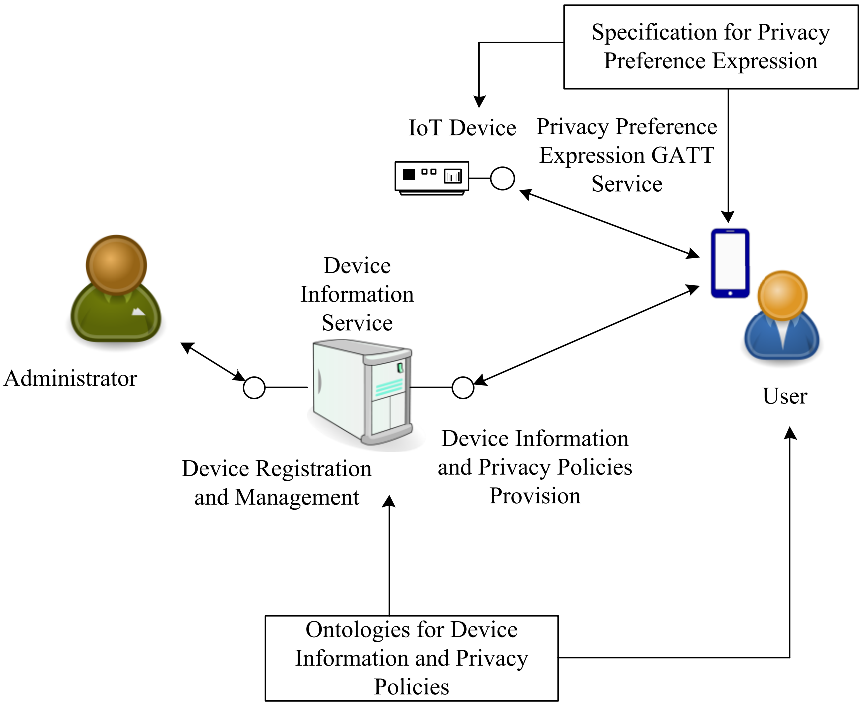

Specification for Privacy Preference Expression
IoT Device
Privacy Preference Expression GATT Service
Device Information Service
Administrator
User
Device Registration and Management
Device Information and Privacy Policies Provision
Ontologies for Device Information and Privacy Policies

FIGURE 2. Overview of the proposed framework.

Service through the *Device Information and Privacy Policy Provision* interface to obtain device information and associated privacy policies.

The PrivacyBat framework provides what we call the *Specification for Privacy Preference Expression*, which defines the Privacy Preference Expression GATT service. A PrivacyBat compatible device should implement this service. After receiving the privacy policies of a device, a user can notify the device whether or not he/she accepts the policies via its Privacy Preference Expression GATT service.

IV. ONTOLOGIES

To help a person to determine whether or not to allow a device to collect and use their personal data, this study borrows from the UPnP device schema to provide device information. Simply speaking, the UPnP technology defines protocols for a device to discover nearby devices and obtain names, capabilities, and other configuration information specific to the devices [15]. This study selects attributes in the UPnP device schema to describe a device. As shown in Figure 3, a device has a *Unique Device Name (UDN)* attribute to represent its UUID and a *friendlyName* attribute to provide short description for users. The *manufacturer* attribute of a device provides information on the device's manufacturer. Moreover, the *deviceType*, *modelName*, *modelNumber*, and *serialNumber* attributes describes the features of a device. Users can obtain more detailed information germane to a device from the URL links in the *manufacturerURL* and the *modelURL* attributes. Users can also obtain surrounding images of a device stored in the *iconList* attribute. Therefore, users can locate the device based on the images and its *location* description.

The device information format proposed in this study extends the UPnP device schema in the following respects:

- In the proposed framework, a device may have one or more privacy policies. This study uses the *policyList* attribute to store identities of the privacy policies. Users can query the Device Information Service and retrieve the policies using their identities from the *policyList* attribute.
- Although the UPnP device schema has defined the *serviceList* attribute to describe the services of a UPnP device, the BLE GATT service is different from the UPnP service. Therefore, this study defines types of *Services* and *Characteristics* to describe GATT services provided by a device.

This study defines the privacy policies based on P3P. Simply speaking, P3P provides a vocabulary and specification for a Website to express its privacy policies in XML-based machine readable format [16]. Although P3P has become less popular recently, to the best of our knowledge P3P is still the most well-known specification used to express privacy policies. Therefore, researchers use and adapt the specification in areas such as database accessing [17], e-Commerce [18], RFID applications [19], cloud computing [20], smartphone applications [21] and so on.

For each privacy policy, a device administrator should specify who collects and uses personal data, the means for users to access collected personal data, and how disputes will be solved. The core of a privacy policy is a set of statements.

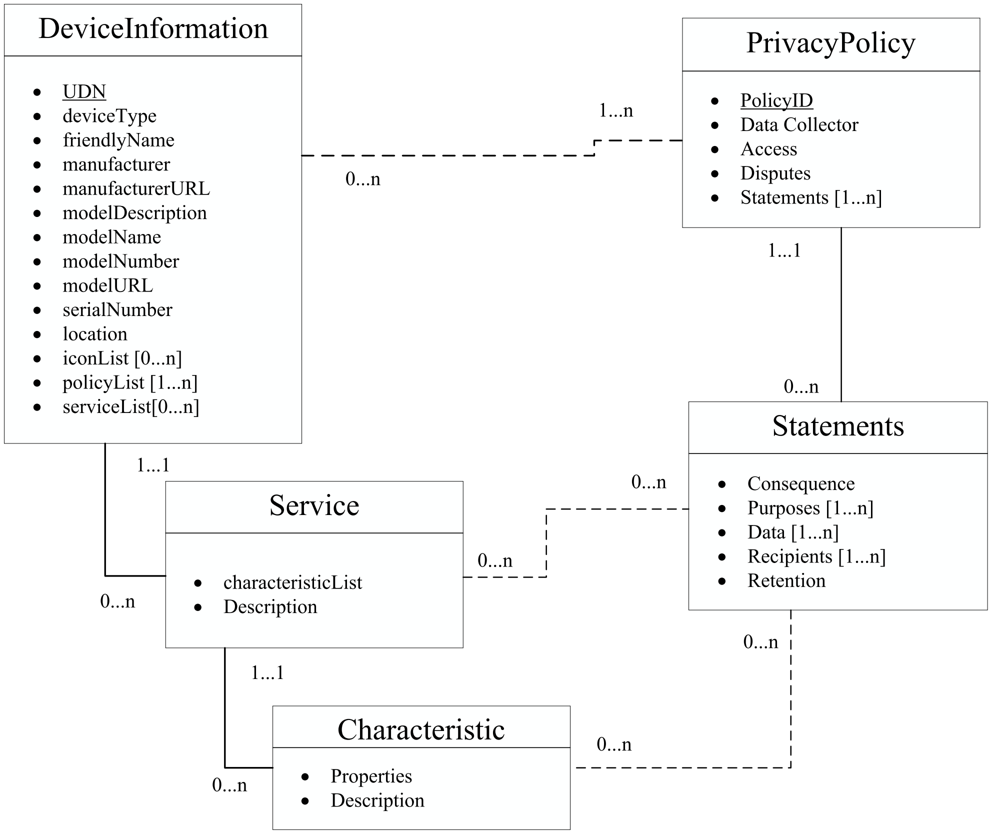


FIGURE 3. Data schema of device information and privacy policie.

A statement describes what personal data are collected and used for what purposes, with whom the collected data will be shared, and the retention period of the collected data. This study modifies the consequence attribute in P3P by linking the consequence attribute in a statement to associated GATT services and characteristics. Therefore, the administrator can limit user access to specified GATT services and characteristics if the user is not willing to accept related statements.

V. THE PRIVACY PREFERENCE EXPRESSION GATT SERVICE

In the PrivacyBat framework, every BLE device that collects and uses personal data should implement the Privacy Preference Expression Service. This study uses Figure 4 to illustrate the concept of the service. The Privacy Preference Expression Service contains three characteristics:

- *Policy ID*. The characteristic is a string and can be written by the user to specify to which policy the user wishes to express his/her preference.
- *Action*. An integer for a user to specify which action he/she wishes to adopt for the policy. Currently, the proposed framework provides three types of actions: First, a user can query the preference for the policy that is currently written in the Policy ID characteristic. A user can also express whether to *accept* or *decline* a privacy policy using this characteristic.

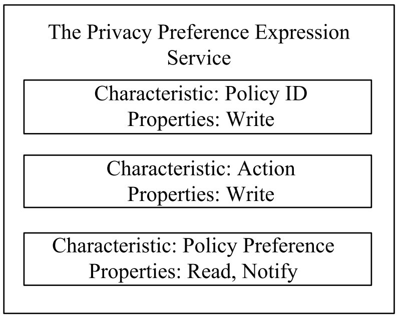

The Privacy Preference Expression Service
Characteristic: Policy ID
Properties: Write
Characteristic: Action
Properties: Write
Characteristic: Policy Preference
Properties: Read, Notify

FIGURE 4. Concept of the privacy preference expression GATT service.

- *Policy Preference*. This characteristic exposes the preference (accepted or declined) of the policy specified in the Policy ID characteristic. The characteristic is readable and can send notifications in the event that changes occur during a connection.

Figure 5 illustrates the flowchart for a PrivacyBat compatible peripheral device to process a request from a user device to its Privacy Preference Expression Service. The peripheral device starts to handle a user request when it receives a write request to its action characteristic. The peripheral device first checks whether it can identify the user device that issued the request. The PrivacyBat framework uses a device's real

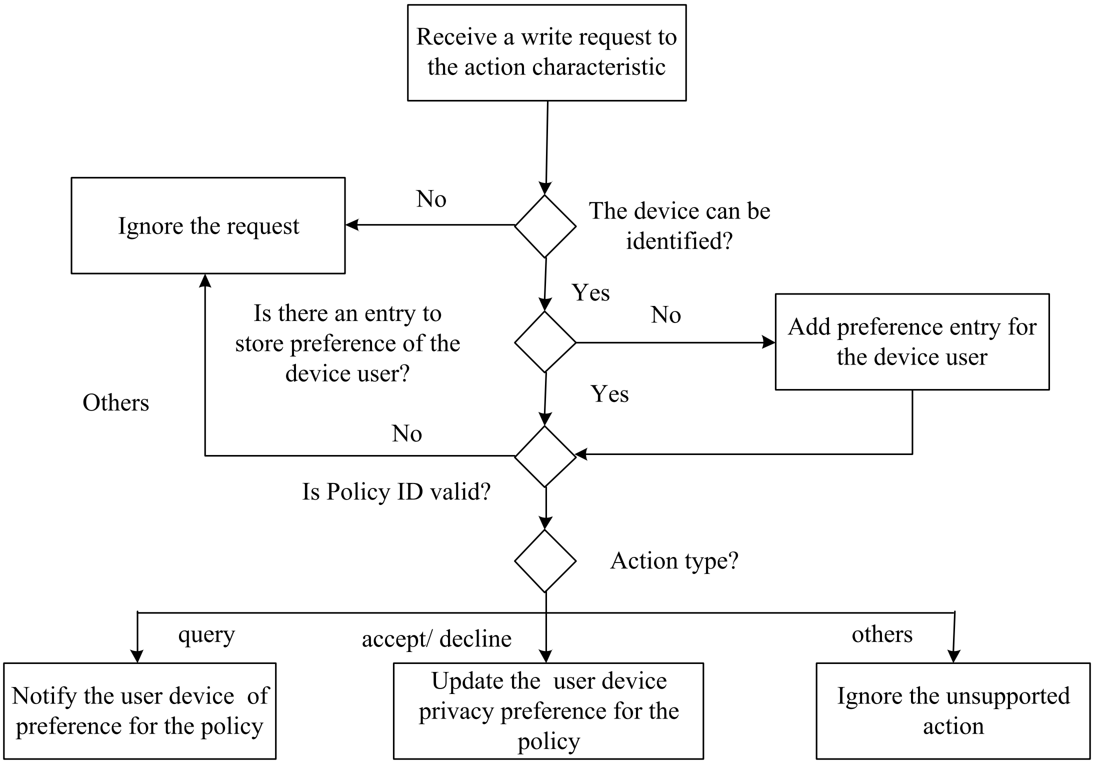

Receive a write request to the action characteristic
The device can be identified?
No
Ignore the request
Yes
Is there an entry to store preference of the device user?
No
Add preference entry for the device user
Yes
Others
No
Is Policy ID valid?
Action type?
query
accept/ decline
others
Notify the user device of preference for the policy
Update the user device privacy preference for the policy
Ignore the unsupported action

FIGURE 5. The flowchart for dealing with requests to the privacy preference expression service.

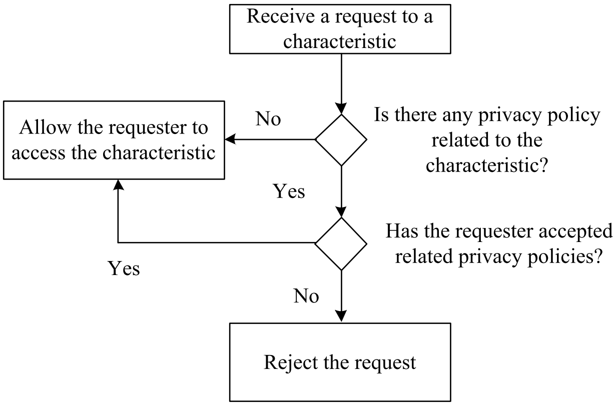

Receive a request to a characteristic
Is there any privacy policy related to the characteristic?
No
Allow the requester to access the characteristic
Yes
Has the requester accepted related privacy policies?
Yes
No
Reject the request

FIGURE 6. The flowchart for handling requests to normal characteristics.

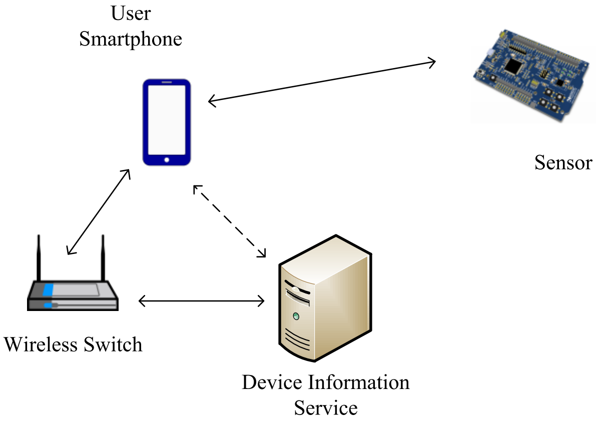
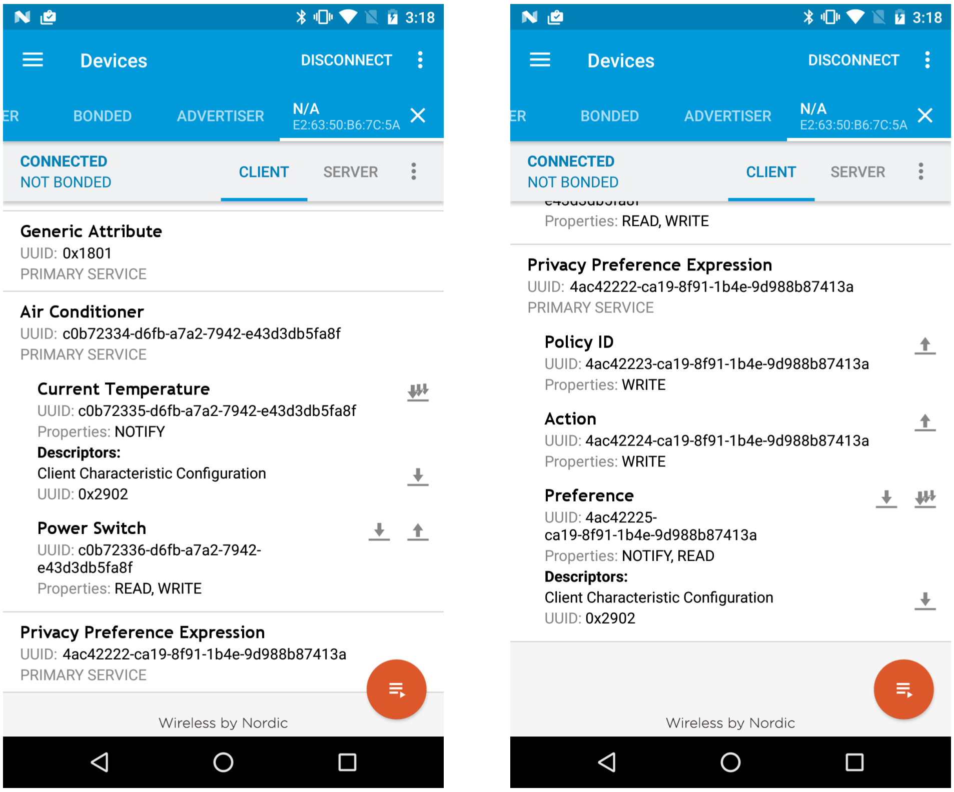


FIGURE 7. The experimental environment.

bluetooth MAC address to identify a device. If the user device adopts the random address scheme and has not bonded with the peripheral device, the peripheral device will ignore the request.

This study assumes that each peripheral device has a table of tuples (DID_i, $Pref_{i\_1}, \ldots Pref_{i\_n}$) and that a device is related to privacy policies. For user device DID_i, the peripheral device will store the user preference for each policy. After obtaining the identity of the user device, the peripheral device checks whether the table contains the tuple to represent the preference settings of the user device. If the peripheral device cannot find the tuple of the user device, the peripheral device generates a new one for the user device and stores the tuple in its storage. Note that a peripheral device can only store privacy policy preferences for a limited number of user devices because of resource constraints. If the storage is full, the peripheral device may replace the least recently used tuple with the new one.

The peripheral device will then check the policy ID characteristic to determine the targeted policy of the request. The request will be ignored by the peripheral device if the policy ID in the characteristic is not valid. Finally, the peripheral device can handle the request based on request type: If the request is to query the privacy preference for a specified policy, the peripheral device will notify the user device with the value "accepted" or "declined" to reflect whether the



3:18

Devices DISCONNECT

ER BONDED ADVERTISER N/A E2:63:50:B6:7C:5A

CONNECTED
NOT BONDED
CLIENT SERVER

Generic Attribute
UUID: 0x1801
PRIMARY SERVICE

Air Conditioner
UUID: c0b72334-d6fb-a7a2-7942-e43d3db5fa8f
PRIMARY SERVICE

Current Temperature
UUID: c0b72335-d6fb-a7a2-7942-e43d3db5fa8f
Properties: NOTIFY
Descriptors:
Client Characteristic Configuration
UUID: 0x2902

Power Switch
UUID: c0b72336-d6fb-a7a2-7942-e43d3db5fa8f
Properties: READ, WRITE

Privacy Preference Expression
UUID: 4ac42222-ca19-8f91-1b4e-9d988b87413a
PRIMARY SERVICE

Wireless by Nordic

3:18

Devices DISCONNECT

ER BONDED ADVERTISER N/A E2:63:50:B6:7C:5A

CONNECTED
NOT BONDED
CLIENT SERVER

e43d3db5fa8f
Properties: READ, WRITE

Privacy Preference Expression
UUID: 4ac42222-ca19-8f91-1b4e-9d988b87413a
PRIMARY SERVICE

Policy ID
UUID: 4ac42223-ca19-8f91-1b4e-9d988b87413a
Properties: WRITE

Action
UUID: 4ac42224-ca19-8f91-1b4e-9d988b87413a
Properties: WRITE

Preference
UUID: 4ac42225-ca19-8f91-1b4e-9d988b87413a
Properties: NOTIFY, READ
Descriptors:
Client Characteristic Configuration
UUID: 0x2902

Wireless by Nordic

FIGURE 8. The services provided by the sensor device.

user device has accepted the policy before. A user device can choose to decline a privacy policy that the user accepted previously and vice versa.

VI. REQUEST PROCESSING

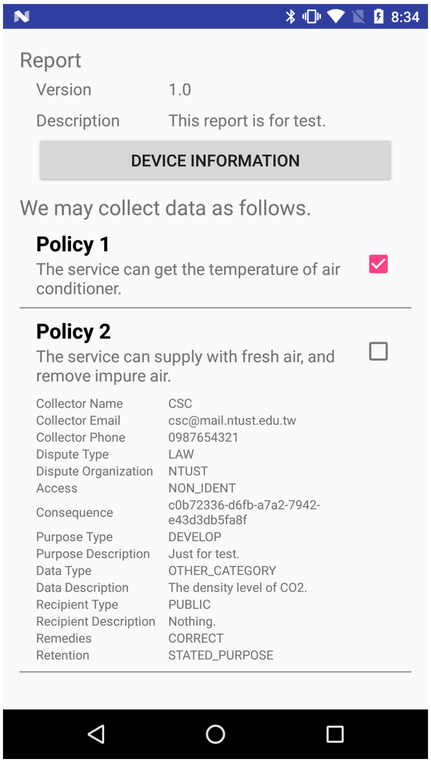
This section describes how a PrivacyBat compatible peripheral device processes requests to normal characteristics (or characteristics that do not belong to the Privacy Preference Expression GATT Service). As described in Section 4, the PrivacyBat framework allows device administrators to prohibit a user from accessing specified services or characteristics of a device if the user does not accept the associated privacy policies. Figure 6 illustrates the process for a PrivacyBat compatible peripheral device to handle requests to access normal characteristics.

After receiving a request from a user device to access a normal characteristic, a PrivacyBat compatible peripheral device will first look up which service the characteristic belongs to. Then, the peripheral device checks if there are privacy policies that have statements linked to the characteristic or service in the consequence field. As described in the previous section, the peripheral device records which policies the user device has accepted. Therefore, the peripheral device can determine whether or not to allow the request.

VII. PROOF OF CONCEPT IMPLEMENTATION

This study has implemented a prototype system to verify that the proposed framework has practical potential. As depicted in Figure 7, the Device Information Service is hosted on a desktop with Intel Core i7-4790 3.6GHz CPU and 16G RAM running Windows 10 professional. The service is implemented with Java Servlet, JDK 8(u131), Jetty application server version 9.3.6 and MySQL Community Database Server version 5.7.18. Note that although this study implements the service on a centralized server, it is possible to implement the service using a distributed architecture. The desktop is connected to a D-Link DIR 850L switch with Gigabit Ethernet. Therefore, user smartphones can connect to the switch through WiFi to communicate with the desktop. An experimental application is implemented and deployed on a Nexus 5X with Qualcomm Snapdragon 808 1.8GHz processor and 2G RAM running Android 6.0.1. Finally, this study implements the Privacy Preference GATT Service on a Nordic nRF52 DK board with nRF52832 SOC to simulate a sensor or a IoT device that will be accessed by the user's smartphone.

In our experiment, the simulated IoT device provides an *Air Conditioner* GATT service (Figure 8). The Air Conditioner GATT service contains two characteristics: the *Current Temperature* characteristic showing the current temperature and the *Power Switch* characteristic enabling a user to turn on/off the air conditioner. The simulated IoT device advertises its 128-bit device identity periodically. The implemented Android application can be used to discover nearby IoT devices, obtain privacy policies of the devices if any, and express user preferences on the policies. After receiving a



Report

Version	1.0
Description	This report is for test.

We may collect data as follows.

Policy 1
The service can get the temperature of air conditioner. ☑

Policy 2
The service can supply with fresh air, and remove impure air. ☐

Collector Name	CSC
Collector Email	csc@mail.ntust.edu.tw
Collector Phone	0987654321
Dispute Type	LAW
Dispute Organization	NTUST
Access	NON_IDENT
Consequence	c0b72336-d6fb-a7a2-7942-e43d3db5fa8f
Purpose Type	DEVELOP
Purpose Description	Just for test.
Data Type	OTHER_CATEGORY
Data Description	The density level of CO2.
Recipient Type	PUBLIC
Recipient Description	Nothing.
Remedies	CORRECT
Retention	STATED_PURPOSE

FIGURE 9. The screenshot of the experimental application.

device identity, the application retrieves the device information using the received identity. The user can further use the smartphone to query the contents of privacy policies of interest. This study uses JSON to represent privacy policies rather than XML because JSON is more lightweight compared to XML. In our experiment, if a privacy policy has one statement, it has a size of about 2K bytes. A user smartphone can retrieve the privacy policy in less than 1 second (about 0.285 sec). Upon obtaining a privacy policy, the experimental application displays the privacy policy and lets the user choose between accepting or declining the policy by checking or unchecking the associated check boxes (Figure 9). Hereafter, the application can connect to the simulated IoT device to express the user's privacy preference as described in Section 5.

As illustrated in Figure 8, the simulated IoT device also provides the Privacy Preference Expression service. This study measures the time taken for the experimental application to query the privacy preference of a privacy policy from the IoT device by averaging the result from 100 experiments. The average time of such experiments is 0.316 seconds.

VIII. CONCLUSION AND FUTURE WORK

To provide a user-friendly means for users to achieve agreement on privacy practices with nearby BLE-based IoT devices, this study has proposed a Framework of Privacy Preferences Expression for BLE-based applications called PrivacyBat. The PrivacyBat framework provides a standard means for users to discover nearby devices and obtain device information along with associated privacy policies. In addition, the framework defines machine-processable ontologies of device information and privacy policies. Therefore, application developers can develop applications to display device information and privacy policies of nearby devices as user notifications. Moreover, the proposed framework offers the Privacy Preference Expression GATT service. A PrivacyBat compatible device should implement this service. Consequently, a user can connect to a PrivacyBat compatible device and express their preferences for received privacy policies through such a service. To demonstrate how the framework works, this study offers a proof of concept implementation and performs experiments to evaluate the performance of major operations. Experimental results indicate that the proposed framework can be implemented with commercially available products. As the proposed framework improves the process for IoT application providers to obtain user consent, this study can hopefully contribute to increasing user trust in IoT applications.

This study has certain limitations that point the way toward future research. First, legacy BLE devices may not be able to support the proposed framework. To address the issue, a gateway can be developed to connect to the legacy devices and provide the privacy preference expression services their behalf. Users can store their privacy preferences in the gateway. Therefore, when users wish to access the devices through the gateway, the gateway can play the role of a gatekeeper to control whether users can access the devices based on their preferences. In this case, designing and implementing such a gateway would be a challenging task. Second, this study only developed an experimental smartphone application to validate the proposed framework. However, this study does not consider the user's attitude toward the application. Usability tests need to be perfomed on the application to help improve the user experience. Last but not least, people may be curious about whether a device follows the accepted privacy policies. To address the issue, we can develop several kinds of tools to detect suspicious devices. For example, we can develop tools for a user smartphone to detect whether or not it transfers personal data to an IoT device. In addition, we can use tools like IoTScanner [22] to monitor the outgoing traffic of an IoT device to detect whether it transfers personal data to remote hosts. However, users still cannot know how a device will deal with their personal data. In this case, it would be interesting future work to develop vetting systems on IoT devices.

REFERENCES

[1] J. Groopman and S. Etlinger. (Jun. 2015). Consumer perceptions of privacy in the Internet of Things: What brands can learn from a concerned citizenry. Altimeter Group. Accessed: Feb. 9, 2018. [Online]. Available: http://www.altimetergroup.com/pdf/reports/Consumer-Perceptions-Privacy-IoT-Altimeter-Group.pdf

[2] V. Pureswaran and P. Brody, "Device democracy: Saving the future of the Internet of Things," IBM Inst. Bus. Value, Tech. Rep. GBE03620USEN, 2015, accessed: Feb. 9, 2018. [Online]. Available: https://www-935.ibm.com/services/multimedia/GBE03620USEN.pdf

[3] *EU Article 29 Data Protection Working Party, Opinion 8/2014 on the on Recent Developments on the Internet of Things*, European Commission, Brussels, Belgium, 2014.

[4] *US Federal Trade Commission, The Internet of Things: Privacy and Security in a Connected World, Federal Trade Commission Staff Reports*, DIANE Publishing Company, Collingdale, PA, USA, 2015.

[5] *Bluetooth Core Specification*, Bluetooth SIG, Inc., Kirkland, WA, USA, 2016.

[6] R. Davidson, K. Townsend, C. Wang, and C. Cufí, *Getting Started With Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*. Sebastopol, CA, USA: O'Reilly, 2014.

[7] N. Gupta, *Inside Bluetooth Low Energy*. Norwood, MA, USA: Artech House, 2013, accessed: Feb. 9, 2018. [Online]. Available: https://books.google.com.tw/books?id=-LMq0NhoEQgC

[8] M. Ryan, "Bluetooth: With low energy comes low security," in *Proc. 7th USENIX Workshop Offensive Technol. (WOOT)*, Washington, DC, USA, Aug. 2013, accessed: Feb. 9, 2018. [Online]. Available: https://www.usenix.org/system/files/conference/woot13/woot13-ryan.pdf

[9] K. Fawaz, K.-H. Kim, and K. G. Shin, "Protecting privacy of BLE device users," in *Proc. 25th USENIX Secur. Symp. (USENIX Security)*, Austin, TX, USA, Aug. 2016, pp. 1205–1221.

[10] P. Wang, "Bluetooth low energy—Privacy enhancement for advertisement," M.S. thesis, Dept. Telematics, Norwegian Univ. Sci. Technol., Trondheim, Norway, Jun. 2014, accessed: Feb. 9, 2018. [Online]. Available: https://brage.bibsys.no/xmlui/handle/11250/263047

[11] S.-C. Cha, C.-Y. Dai, and J.-F. Chen, "Is there a tradeoff between privacy and security in BLE-based IoT applications: Using a smart vehicle of a major Taiwanese brand as example," in *Proc. IEEE 5th Global Conf. Consum. Electron. (GCCE)*, Kyoto, Japan, Oct. 2016, pp. 1–4.

[12] N. Davies, N. Taft, M. Satyanarayanan, S. Clinch, and B. Amos, "Privacy mediators: Helping IoT cross the chasm," in *Proc. 17th Int. Workshop Mobile Comput. Syst. Appl. (HotMobile)*, St. Augustine, FL, USA, Feb. 2016, pp. 39–44.

[13] A. Ukil, S. Bandyopadhyay, J. Joseph, V. Banahatti, and S. Lodha, "Negotiation-based privacy preservation scheme in Internet of Things platform," in *Proc. 1st Int. Conf. Secur. Internet Things (SecurIT)*, Kollam, India, Aug. 2012, pp. 75–84.

[14] M. S. Gast, *Building Applications With iBeacon: Proximity and Location Services With Bluetooth Low Energy*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2014.

[15] (Oct. 15, 2008). *UPnP Forum, UPnP Device Architecture 1.1*. Accessed: Feb. 9, 2018. [Online]. Available: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf

[16] L. Cranor *et al.* (Nov. 13, 2006). *The Platform for Privacy Preferences 1.1 (P3P1.1) Specification, W3C Specification*. Accessed: Feb. 9, 2018. [Online]. Available: https://www.w3.org/TR/P3P11/

[17] R. Agrawal, P. Bird, T. Grandison, J. Kiernan, S. Logan, and W. Rjaibi, "Extending relational database systems to automatically enforce privacy policies," in *Proc. 21st Int. Conf. Data Eng. (ICDE)*, Tokyo, Japan, Apr. 2005, pp. 1013–1022.

[18] A. A. Said, A. R. C. Hussin, H. M. Dahlan, and M. M. H. Pour, "Privacy policy preference (P3P) in e-commerce: Key for improvement," in *Proc. Int. Conf. Inf. Retr. Knowl. Manage. (CAMP)*, Kuala Lumpur, Malaysia, Mar. 2012, pp. 177–181.

[19] S.-C. Cha, K.-J. Huang, and H.-M. Chang, "An efficient and flexible way to protect privacy in RFID environment with licenses," in *Proc. IEEE Int. Conf. RFID*, Las Vegas, NV, USA, Apr. 2008, pp. 35–42.

[20] M. Olurin, C. Adams, and L. Logrippo, "Platform for privacy preferences (P3P): Current status and future directions," in *Proc. 10th Annu. Int. Conf. Privacy, Secur. Trust (PST)*, Paris, France, Jul. 2012, pp. 217–220.

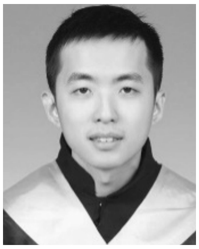
[21] S.-C. Cha, C.-M. Shiung, T.-C. Liu, S.-C. Syu, L.-D. Chien, and T.-Y. Tsai, "A framework for major stakeholders in Android application industry to manage privacy policies of Android applications," in *Proc. 4th Annu. Privacy Forum (APF)*, Frankfurt, Germany, Sep. 2016, pp. 153–170.

[22] S. Siby, R. R. Maiti, and N. Tippenhauer. (Jan. 2017). "IoTScanner: Detecting and classifying privacy threats in IoT neighborhoods." Accessed: Feb. 9, 2018. [Online]. Available: https://arxiv.org/abs/1701.05007

SHI-CHO CHA (SM'17) received the B.S. and Ph.D. degrees in information management from National Taiwan University, in 1996 and 2003, respectively. He is currently an Associate Professor with the Department of Information Management, National Taiwan University of Science and Technology, where he has been a Faculty Member since 2006. He is a certified PMP, CISSP, CSSLP, CCFP, and CISM. From 2003 to 2006, he was a Senior Manager with PricewaterhouseCoopers, Taiwan. His current research interests include security and privacy of blockchain applications, IoT security and privacy, and information security risk management.

MING-SHIUNG CHUANG received the master's degree in criminal justice from the Nation Center Police University, Taiwan. He is currently pursuing the Ph.D. degree with the Department of Information Management, National Taiwan University of Science and Technology. He is currently with the Hi-Tech Crime Center of Criminal Investigation Bureau, Taiwan. His research interests include cyberspace security, cyber-crime, digital forensics, and information security risk management.

KUO-HUI YEH (SM'16) received the M.S. and Ph.D. degrees in information management from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2005 and 2010, respectively. He is currently an Associate Professor with the Department of Information Management, National Dong Hwa University, Hualien, Taiwan. He has authored over 90 articles in international journals and conference proceedings. His research interests include IoT security, Android security and privacy, NFC/RFID security, digital signature, and network security.



ZI-JIA HUANG is currently pursuing the master's degree with the Department of Information Management, National Taiwan University of Science and Technology. His current research interests are in the area of security and privacy of blockchain applications, information security management, and IoT security.

CHUNHUA SU was a Research Scientist with the Cryptography and Security Department, Institute for Infocomm Research, Singapore, from 2011 to 2013. From 2013 to 2016, he was an Assistant Professor with the Japan Advanced Institute of Science and Technology. He was with Osaka University as an Assistant Professor from 2016 to 2017. He is currently an Associate Professor with the University of Aizu, Japan. He His research interests include cryptographic protocols, privacy-preserving technologies, and IoT security and privacy.