A Contextual-adaptive Location Disclosure Agent for General Devices in the Internet of Things

Mahmoud Elkhodr, Seyed Shahrestani and Hon Cheung
School of Computing, Engineering and Mathematics
University of Western Sydney
Sydney, Australia

***Abstract*—The Internet of Things (IoT) has the potential to transform our daily lives and societies. This is, at least in part, due to its massively distributed and ubiquitous nature. To realize the benefits of the IoT, security and privacy issues associated with the use of the IoT need to be identified and addressed properly. In this paper, our focus is on protecting the privacy of the users of location-based services in the IoT. To achieve this protection, we propose a context-aware adaptive approach for general devices in the IoT, where the general devices are used by users in accessing the location-based services. The proposed approach is based on developing and utilizing an agent to manage location privacy in the context of requested network-based services. The results of an experiment conducted to show the effectiveness and efficiency of this approach are also reported.**

***Index Terms*—Communication Agent, Internet of Things, Location-based Services, Privacy, Smart Objects.**

I. Introduction

The Internet of things (IoT) is a technology that connects physical objects and not only computer devices to the Internet, making it possible to access data/services remotely and to control a physical object from a remote location. The ITU describes the IoT as an infrastructure which interconnects physical and virtual things together using existing and evolving interoperable information and communication technologies [2]. Things can be physical or virtual objects which are capable of being identified and integrated into communication networks. Examples of physical objects are industrial robots, wireless sensors and smart phones; while examples of virtual objects are multimedia contents and application software. There are three main categories of objects in the IoT:

- General objects: objects in this category have embedded processing and communication capabilities. Examples are industrial and electrical machines, smart cars, robots and smart phones.
- Sensing and actuating objects: sensing objects collect information about their surroundings or environment using sensors. Actuators are objects that can manipulate their environment using mechanical movements, remotely via the Internet.
- Data-capturing and data-carrying objects: these objects communicate using technologies such as RFID and NFC. An EFTPOS machine is an example of a data-capturing object, while a credit card is an example of a data-carrying object.

Interactions between all categories of objects are via a communication network. The communication network can be of various types as illustrated in Fig. 1. This distributed nature of technologies found in the IoT gives rise to numerous security and privacy concerns. In a previous work on the IoT [5], it has been discussed how embedded objects in public areas could create weak links that malicious entities can exploit and can perform illegitimate surveillance, tracing, tracking, and profiling of the users' movements and activities. A number of automated attack vectors on privacy in the IoT are also introduced. In another earlier work on the mobile location privacy in the IoT [4], the existing and inherited privacy issues, such as the incidents which lead to threats to location privacy of users, have been reported. It has been further discussed how it is possible to collect the personal information of users from IoT's objects without the individual's knowledge or consent. These privacy challenges, which confront the IoT, are of a new nature when compared to those experienced in today's online communication which mostly involves the user directly. Ordinarily, in a traditional online environment, a user is in control of his or her privacy choices as he or she is directly involved in the access to the Internet. This can no longer be the case with automated IoT's objects where access controls and privacy policies on information need to be pre-determined and executed without the real-time involvement of the user. The principle known as notice and choice is being challenged in the IoT. A question is also raised on whether users have the control over their location disclosure in the IoT and the principles that should govern the deployment of IoT location enabled technologies.

This paper proposes a method for providing a context-aware adaptive technique for the protection of location privacy for general objects in the IoT. This method is based on the use of an agent, referred to as the Dynamic Location Disclosure Agent (DLDA). The paper makes the following contributions:

- The development of a context-aware adaptive method for the IoT. By adaptive we mean the method implies automated privacy choices tailored to specific contexts and/or privacy requirements.
- It includes: A method based on the obfuscation technique for protecting location privacy in the IoT. It defies data mining by degrading an object's precise location in a given situation or more precisely, based on the context.

978-1-4799-0540-9/13/$31.00 ©2013 IEEE

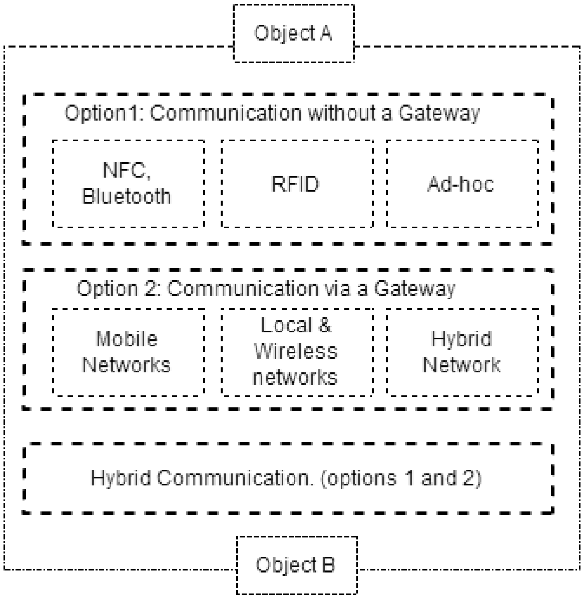

Object A
Option1: Communication without a Gateway
NFC, Bluetooth
RFID
Ad-hoc
Option 2: Communication via a Gateway
Mobile Networks
Local & Wireless networks
Hybrid Network
Hybrid Communication. (options 1 and 2)
Object B

Fig. 1. IoT's communication

- Some experimental works that validate the proposed solution.

The reminder of this paper is organized as follows: Section II introduces context with particular reference to context awareness related to this topic. A brief introduction to location privacy protection techniques is provided in Section III. Section IV introduces the DLDA agent. Section V sets out the Dynamic Disclosure-Control Method (DDCM). Applying the contextual model to the DDCM method is provided in Section VI. This Section also reports on the experimental works. Conclusion remarks, and possible future works are provided in Section VII

II. CONTEXT AWARENESS

With the introduction of pervasive computing and ubiquitous computing, and the proliferation of portable and wireless enabled devices, the term context awareness has become more prominent [12]. Originally, context awareness describes devices that can sense their physical environment, and change their behavior accordingly. Location awareness, on the other hand, has also emerged as a growing trend in hardware and software applications and has become more prevalent on portable devices. Recent developments in location technologies have allowed portable devices to become aware of their location. This leads to the introduction of location based services and location based applications.

The innovations in wireless and communication networks, which power location-aware portable devices, have made the protection of privacy a cumbersome task, and specifically the location privacy. Users are rapidly losing the control over the disclosure decisions of their personal information. For instance, location based services applications are mostly considered vulnerable to a large-scale of privacy loss. These applications inherently depend on the users' location in which users entrust to applications running on un-trusted third-party servers. In [4], it has been shown how contextual aware systems, such as mobile applications, are actively collecting the user personal information, specifically location information, without the user's consent or knowledge. While in a normal situation, the user might choose willingly to disclose his or her location information, he or she might not wish to disclose their location to everyone at all time. In fact, location information becomes highly sensitive when it is combined with other contextual data such as the user's identity. Knowledge of identity, location and other contextual data such as the users' shopping habits has improved services presented to users by enhancing the quality of service provided to users. On the other hand, when this information falls in the wrong hands, or when this information is collected by unauthorized parties without the users' consent, privacy issues arise.

A. Context Awareness in the IoT

The IoT extends the interactions between humans and applications to a new dimension of communication via objects. Rather than always interacting with the human users, objects will be interacting with each other autonomously, performing actions on behalf of the users and updating their daily schedules. Therefore, context awareness is seen as an enabling technology for the IoT. Context aware objects in the IoT are concerned with the acquisition of context, for instance through the use of sensors to sense the environment and therefore perceiving a situation based on that; or performing a mechanical movement with the use of actuators. Object are also concerned in analyzing a context, e.g., matching services to a context, and in the recognition of a context, e.g. performing some actions or triggering event based on a recognized context. Thus, sophisticated and complex contextual interaction models are perceived in the IoT for the support and delivery of context-aware services.

Mainly, contextual data in the IoT is used to provide tailored services, increase the quality/precision of information, discovery of nearby services and making implicit users' interactions. However, this comes at a price. In [5], the attacks on privacy envisioned in the IoT and how inference attacks can be generated by collecting contextual data belonging to a user has been explored. It has been also discussed how an automated invasion attack can be formed. In brief, an automated invasion attack is an incremental process of inference attacks in which the attacker gradually gathers more knowledge on the user's life or activities through the combination and linking of the information collected from various source of objects owned, operated or in contact with the user. Contextual information can relate directly to a user, or it can be associated with the users' tasks or activities, and their social interactions. Location, date, time, identity, knowledge of shopping habits, type of communication, task performance, and physical parameters (noise, light, and temperature) are all forms of contextual data.

Consequently, as with the current context-aware systems, context-aware objects in the IoT will also challenge the users' privacy. However, the impact on the users' privacy is seen to be higher than those found in the current context-aware systems. In the IoT, the user is no longer the implicit source of information; and therefore privacy choices cannot explicitly rely on the users' decisions- a burden we wish to avoid, if at all possible, since the communication in the IoT is, in a great part, autonomous between objects, which does not necessary involve the human user directly. Therefore, the challenges remain on providing privacy solutions which autonomously adapt to variations in contexts.

B. Motivational Example

In order to motivate this paper, consider the followings scenario in the IoT:

Bob is a traveling finance consultant. He drives to work in his smart interactive car Monday to Friday on a weekly basis. Bob's smart car interacts autonomously with a number of Location-Based Services (LBSs) on the way. They provide him with information on nearby traffic jams and discount petrol prices, and also update him on the daily currency exchange rates and the share information in the stock market. Bob's car sends his location, during business hours, back to the office system which manages the clients' appointments based on Bob's current location. This helps reducing the time spent traveling from a client's location to another. To stay connected with his family, Bob smart phone alerts him when his kids reach school and when any of his kids is mobile. During the day, Bob also receive a few notifications on any object's activities occurring in his smart home.

In the above scenario, Bob wishes to provide only an approximate location to some information service providers, e.g., nearby restaurants, a precise location to others, e.g., his work, and a completely fake location to other providers, e.g., when he is checking the currency exchange rate. In addition, Bob prefers to reduce the precision of his location during personal activities. He also would like his car and any portable/wearable objects he carries to stop sending his precise location once these objects are connecting to the Internet using any public wireless network. Bob is one of the authorized people who have access to the location of his kids as well. Above all of these, Bob is not an expert in technology and he would like these privacy requirements to be arranged automatically by the IoT.

This simplified example shows that the variations in location precision requirements are based on several contextual factors such as the requester, service provider, time and date, current location, type of networks, the user's preferences and other parameters. In this work, these contextual factors define the context.

III. PROTECTION TECHNIQUES FOR LOCATION PRIVACY

In order to perform location privacy protection, most of the computational techniques used for privacy protection alter the location information in a way of reducing the information granularity. The key techniques used for privacy protection are briefly discussed in this Section.

A. The randomization technique

Randomization is a core principle in statistical theory. It is the process of making a data stream random. The study in [1] uses a decision-tree classifier to randomize data. This results in a new data stream which looks different from the original data stream. A reconstructed distributions procedure is proposed to accurately estimate the distribution of the original data. The issue with this method is that it does not offer the flexibility needed in the protection of location information.

B. Regulatory based techniques

This method relies on the government rules and the regulations in protecting the personal information of users. The work in [8] reports the status of privacy legislations and fair information practices in a number of countries. The problem with this method is that regulations vary from a country to another. In addition, they usually lag behind newly developed technologies.

C. Privacy policies

This is a trust-based agreement policy arranged between the user and service provider. However, similar to the regulatory method, privacy policies cannot offer a complete solution since they are vulnerable to malicious disclosure of private information [7].

D. Anonymity

This method uses pseudonyms, normally to hide the identity of a user, in order to anonymize the user personal information, e.g. the work in [10]. This challenges personalized services by eliminating authentications and personalization techniques [9].

E. Obfuscation

The term obfuscation is introduced in [3]. It is described as the practice of deliberately degrading the quality of location information in some way, in order to protect the privacy of an individual to whom that location information refers. Location obfuscation is a technique used to protect a user's location by generalizing the location information, or using substitution or alteration. The obfuscation concept can also be linked to the principle of need-to know. The obfuscation technique offers a good approach for preserving the location privacy of users. However, obfuscating the location information is ineffective when owners of location information might not wish to obfuscate their location information at all time or in all situations. The challenge is then in providing a solution that would vary the degree of location privacy by using different levels of obfuscation. Determining the level of obfuscation is based on the context of the communication and the privacy policies defined by the user. To achieve this, the method presented in this work complements and make use of the existing privacy protection techniques described previously. This method is referred to as the Dynamic location Disclosure Agent (DLDA).

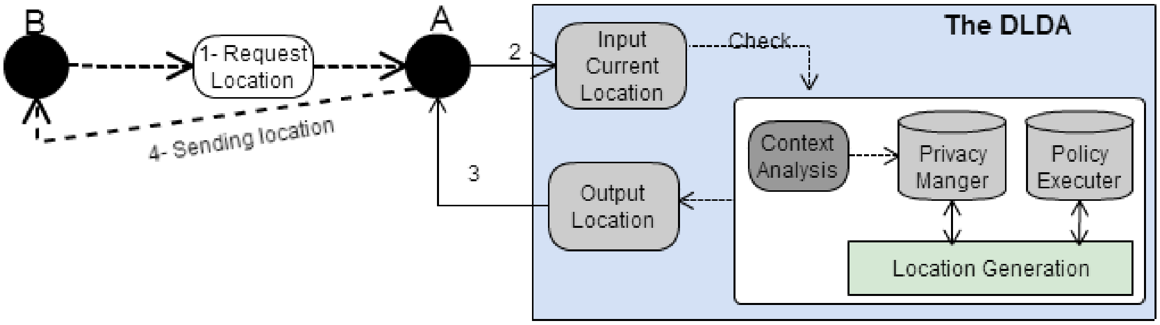


Fig. 2. The DLDA agent. Object B requests the location of object A (1). Object A forwards the request to the DLDA agent, attached to it, by inputting the current location (2). The DLDA agent determines the location output using the context analysis, policy executer and the location generation components. The location output is forwarded back to object A (3). Object A sends object B the location output as his current location (4).

IV. DYNAMIC LOCATION DISCLOSURE AGENT

This work assumes the followings: the objects fall under the general device category, as categorized by the ITU. Therefore, objects are assumed to have embedded processing and communication capabilities. An object requests the location information of another object over wireless or mobile networks. An object, say object B, has no information about another objects' location, say object A, other than the information in which the object A chooses to reveal.

The Dynamic Location Disclosure Agent (DLDA) is represented in Fig. 2. The agent takes the current location of object A as an input and outputs an obfuscated location that varies in degree or precision based on the context of the communication and based on the values of the contextual parameters. For example, at a certain location, for two different requesters of location information, the agent may provide different location information, each according to their parameters that are based on the context.

The DLDA contains four major components: a context analysis component which allows the agent to be contextually aware of the current location of object, mobility status, type of the Internet connection and the requester among other computed parameters. The second component is the privacy manager which stores the users' defined privacy preferences. The third component is the policy executer that retrieves the relevant policies and executes disclosure-control methods, according to the current context. The agent then determines whether location can be revealed to the requester and the level of obfuscation to be used. This is done using the Location Generation component which applies some spatial constraints to each location output. Spatial constraints can be in the form of constraints based on time, date and the expiry time and date, also known as Time-to-Live (TTL) of each location output.

Specifically, the DLDA's components interact as follows: An object B requests the location of object A. This could be a direct request as part of a communication request, or object A may request some LBS information from object B which in turn asks object A for its location. Object A refers to the DLDA agent and place a location disclosure request by providing its current true location. The agent requests from object A some other information necessary for the context analysis component. This information describes the current context of object A, for instance, its mobility status and its current network settings. Similarly, the agent also requests some information on object B that might be known by object A. This can be in the form of any identification information for object B and its current networks settings. Next, the agent requests a permission from the privacy manager, in order to retrieve any defined users' privacy policies. The policy executer component then computes, using the Dynamic Disclosure-Control Method (DDCM), the level of obfuscation needed. Obfuscation levels are discussed in the next section in details. It then refers to the location generation component which in turn generates the new output location and sends it back to object A. Object A then uses the location output in the communication with object B. The process repeats if another object, for instance object C, requested the location of object A even if object A is still at the same current location.

V. THE DYNAMIC DISCLOSURE-CONTROL METHOD

A. Architecture

The Dynamic Disclosure-Control Method (DDCM) implements five levels of obfuscation. Each level provides different location outputs for the same location input. The obfuscation levels range from level 0 (disclosing true location) to level 4 (generating a dummy location) with a variation of location precision in between level 0 and 4. That is, the location precision degrades subsequently from a level to another. The DDCM is also contextual dynamic and adapts from a context to another. The obfuscation level is computed and determined based on analysing the current context using the context analysis component. A context is analysed using the contextual parameters of four layers: the Network, Location, Period and the Requester layers as shown in Fig 3. Thus, for a given scenario n, a context denoted by C is defined using the following statement:

$$Cn = F(N) + F(L) + F(P) + F(R) \qquad (1)$$

In the above equation, F(N) represents the contextual parameters related to the network settings. E.g. mobile network

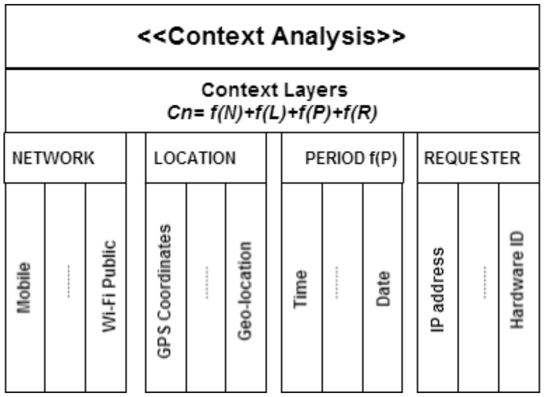


Fig. 3. The context analysis component

or Wi-Fi Home. F(L) represents the current location of the object. F(P) includes the time and date of the interaction. F(R) represents the contextual parameters which identify an object from another such as the object's identifier or IP address.

On the other hand, the privacy manager component has also a major role in determining the level of obfuscation used for each context. The privacy manager is discussed in a subsequent section.

B. The obfuscation levels

As discussed in Section V-A, there are 5 levels of location outputs L0,L1,L2,L3 and L4. Level 0 (L0) discloses the true location of the object and Level 4 (L4) generates a dummy location. The remaining three obfuscated location levels (L1, L2, L3) are computed for each location input and are discussed in this Section.

The position of any location on earth, on a 2D scale, can be determined using the conjugate graticule, which is where the latitude and longitude intercept. Determining the precise latitude and longitude coordinates of a location is available using many technologies such as a global positioning satellite receiver, which can communicate with satellites over the Earth to triangulate to a certain position. Therefore, an object's location in geographic space can be represented as a point on a map and denoted by L, where L is a 2-tuple (latitude, longitude). Define L to be a member of a set LS such that L$\in$ LS. LS is a collection of locations. For every element L$\in$ LS, define a base point LS (Xsi ,Ysj) to represents each L$\in$ LS. Let the set LS be a subset of another set $\wp$(LS). In turn, let $\wp$(LS) be is a subset of a master set $\wp$($\wp$(LS)). Each of these three sets has a base point that can represent L each in its correspondent subset. Therefore, by selecting a set, a different base point location can be used and hence different levels of obfuscation are provided using different base points. Figure 4 depicts this logic. The DDCM method, given in Table 1, describes how these sets are formulated and how the base points are derived.

TABLE I
THE DDCM METHOD

Data: The geographic location of a device L is determined by the longitude X and the latitude Y and represented by LX,Y

Input: Li (Xi, Yj) where Li is the true location with current longitude Xi and latitude Yj
Output: Lo (Xi, Yj) where Lo is the obfuscated location where
Li (Xi, Yj)$\in$ Lo (Xi,Yj) and Li $\subseteq$ Lo.

Procedures:
1- Let Li (Xi,Yj) be the true location with longitude Xi and the latitude Yj
2- Let LS be a set of $\{$(Xa,Yb), (Xc,Yd) (Xi,Yj) (Xn,Ym)$\}$; Where n and m are unique representation of the longitude and latitude of a true location. That's for a given set of locations denoted by LS1,
$\forall$ (Xn,Ym) [(Xn,Ym) $\setminus \in$ LS1]
where "$\setminus \in$" means "strictly an element of"

Define the base point 1: LS (Xsi,Ysj) to represent any (Xn,Ym) included in a particular LS set in a way that:
If (Xn, Xm) $\in$ LS then $\forall$ (Xn,Ym) there exist
(Xsi,Ysj) $\in$ LS such that [(Xsi,Ysj)$\angle$ (Xn,Ym)]
where "represent any" is denoted by "$\angle$"

3- A collection of sets of LS is denoted by $\wp$(LS)= $\{$LS1, LS2,...,LSp$\}$ where p is an integer representing the number of subsets in $\wp$(LS) such that$\wp$(LS)=$\{$ K $\mid$ K $\subseteq$ LS $\}$
Let ψ= $\wp$(LS)

Define the base point 2: ψ (Xti,Ytj) to represent any (Xn,Ym) included in any subset of ψ in a way that:
If (Xn, Xm) $\in$ LS and LS $\in \psi$ then $\forall$ (Xn,Ym) there exist
(Xti,Ytj) $\in \psi$ such that [(Xti,Ytj) $\angle$ (Xn,Ym)]
4- Define$\wp(\psi)$ to be the master set of ψ
where $\wp(\psi)$=ψ1$\sqcup$ ψ2$\sqcup\psi$f;
where f is an integer representing the number of ψ subsets available.
Letξ =$\wp(\psi)$.

Define the base point 3: ξ(XCi,YCj) to represent any (Xn,Ym) included in any subset of ξ in a way that:
If (Xn, Xm) $\in$ LS and LS $\subseteq\psi$ and $\psi\subseteq \xi$ then $\forall$ (Xn,Ym) there exist
(XCi,YCj) $\in \xi$ such that [(XCi,YCj)$\angle$ (Xn,Ym)]

5- Therefore if Li (Xi, Yj) $\in$ LS and LS $\subseteq\psi$ and $\psi \subseteq \xi$
There exist:

$\forall$(Xi, Yj) [(Xsi ,Ysj) $\in$ LS), ((Xti,Ytj) $\in\psi$), ((XCi,YCj) $\in\xi$)$\angle(Xi, Yj)$]

C. Privacy Manager and Policy Executer

Privacy policies can be defined by the user in the privacy manager component. For instance, a policy can be defined to not disclose the accurate location of the object on a certain time or date. Another policy can attach time and date restrictions to a location output for specific requesters on specific networks. The user might as well define which obfuscation level to be used in certain contexts. For example, from the scenario provided in Section 3, Bob can define a privacy policy enforced on his smart car which states: On Mon to Fri between 12:00 and 13:00 pm, do not disclose my exact location; instead only disclose my location as in Sydney, for example. A default privacy profile can also be defined in the privacy manager component. This default profile will be enforced in the absence of any defined privacy policies. For example, a default location privacy policy could define to not disclose the objects precise location information to unknown objects on specific time of the day and on specific networks.

By analyzing the contextual information and comparing the

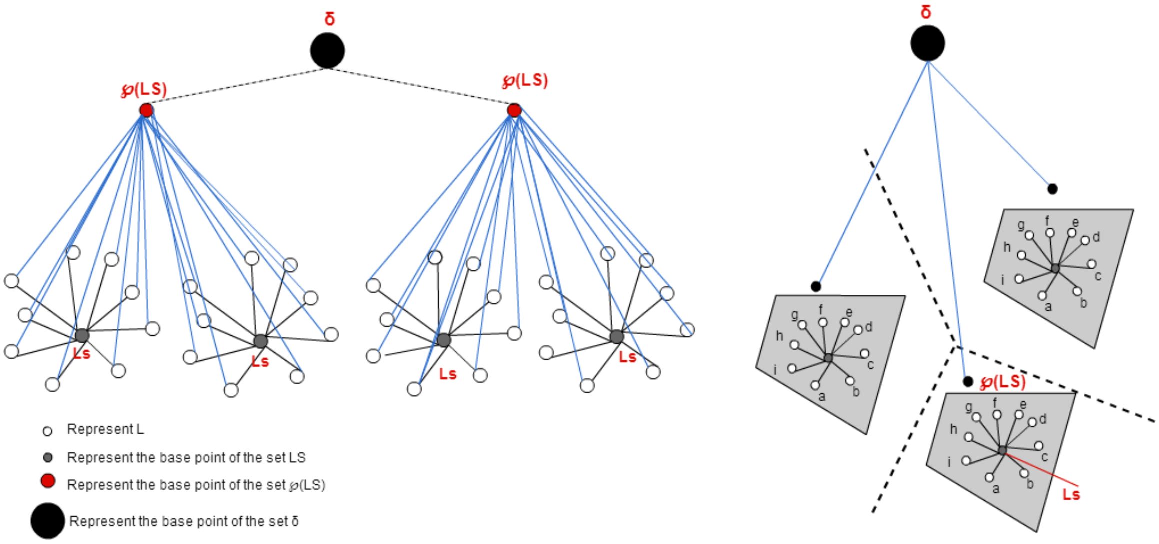

δ
℘(LS)
℘(LS)
Ls
Ls
Ls
Ls
Represent L
Represent the base point of the set LS
Represent the base point of the set ℘(LS)
Represent the base point of the set δ
δ
g f e d h c i a b
g f e d h c i a b
g f e d h c i a b
℘(LS)
Ls

Fig. 4. Cartesian representation of the three sets.

results against any defined privacy policies, the policy executer component is able to determine the most suitable obfuscation level that guarantees the use of the best appropriate privacy protection measures. Next, a time-to-live (TTL) constraint is also set by the policy Executer component such that it sets an expiry date and time for a location output in order to prevent continuous tracking of locations. The use of TTL ensures that location of an object is not continuously tracked. Therefore, the DLDA agent allows users or object's operators to define the accuracy and extent to which their location information is revealed. Thus, offering a contextual-adaptive solution that varies in the degree of granularity and restrictiveness.

VI. EXPERIMENTAL WORKS

In order to evaluate the DLDA agent an experiment is setup. The experiment has the followings aims:

- To test and validate the DDCM method by testing each level of the obfuscation method in different contexts.
- To test and validate the privacy policies and spatial constraints applied to locations outputs.
- To verify the context-adaptive feature of the DLDA.

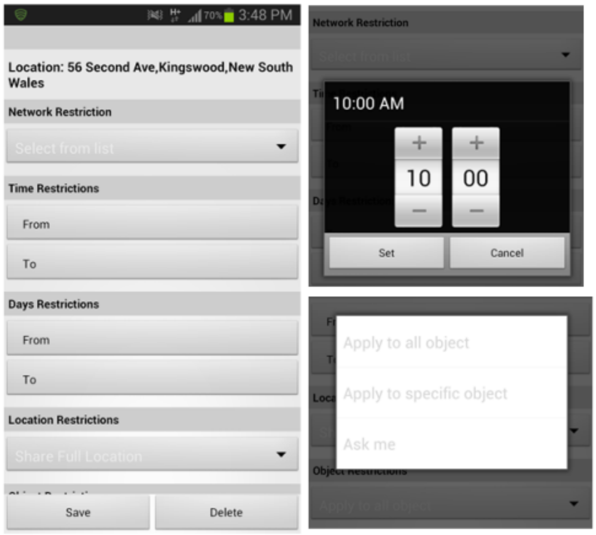
To this end, the experiment is developed to model the interaction described between object A and B. A scenario is setup where a computer application, representing object B, requests the location of a mobile device which represents object A. The interactions are as follows: Using the UI of the computer application (object B), a request for the location of object A (the mobile device) is placed through the Internet. The mobile device receives this request and refers to the agent attached to it by acquiring its current location. The mobile device also provides the agent with the contextual information needed for the context analysis. These are the time, date, and its current network name e.g. WiFi home, and any known information on the requester e.g. its IP address. In addition, the agent has a control panel which allows the user to define the privacy policies beforehand. These privacy policies are compared against the current contextual information received. This results in the selection of a suitable obfuscation level. The agent generates the base points location coordinates for this selected obfuscation level, add a TTL and any other restrictions defined in the privacy manager to it, and send this location output to the mobile device. The mobile device replies back to the computer application with the location output as its current location. The agent has a control panel UI which allows the definition of the followings privacy policies:

- Network restrictions: it's an enforced policy restriction on a particular network e.g. mobile network.
- Time restrictions: it's where a restriction on time can also be enforced e.g. between 9:00 AM and 17:00 PM.
- Date restrictions: it's where a restriction on the week days can be enforced as well. E.g. Monday to Friday.
- Location restrictions: This is a policy which enforces a specific obfuscation level.
- Objects restrictions: This allow the enforcement of the above mentioned policies to specific objects.
- Default settings: Allow the enforcement of a specific set of configurations (restrictions) as a default profile. This default profile is used in the absence of specific policies that govern specific objects.

The screenshot of the control panel is given in Fig. 5. It shows how operator of an object (the mobile device in this example) are able to attach restrictions to a location output by setting restrictions on the network, time, days, location and objects. Figure 5 also shows the default location restrictions settings where a default profile can be configured.

A. Obtaining location input

The mobile device requests the current position (longitude and latitude) via GPS. If GPS is not available then it requests it via the Wi-Fi network. If Wi-Fi is not available as well,



3:48 PM

Location: 56 Second Ave,Kingswood,New South Wales

Network Restriction: Select from list

Time Restrictions: From / To

Days Restrictions: From / To

Location Restrictions: Share Full Location

Save | Delete

Network Restriction: Select from list

10:00 AM

10 : 00

Set | Cancel

Apply to all object

Apply to specific object

Ask me

Object Restrictions: Apply to all object

Fig. 5. The Agent's Control Panel

then it extracts the location coordinates from the mobile's network. The device then forwards its position to the agent. The agent sets the current true location as L0 and generates a dummy location to be used for Level 5 (L5). The agent then proceeds into finding the coordinates of the three base points' corresponding for the three levels (L1, L2, and L3). These are the base points of the sets previously described. The process is as follows:

1) The agent sends the current location coordinates to Google Geocoder API in the format of (X,Y) where X and Y are the integers representing the latitude and longitude. It should be noted that Google Geocoder API is only used as a service for converting the coordinates into a possible readable address. The Google Geocoder API is not made aware of the object's true location.
2) Google API converts these coordinates into readable addresses in the format of (Street Name, Suburb, State and Country).
3) The agent defines the suburb as the first set LS, the state as the second set ψ and the country as the third set ξ. It then communicates back with the Google API to extract the coordinates of each of the base point found in each set using reverse lookup. Therefore, the agent identifies 4 coordinates that can be representing the current location each in a different set (given L0 is the true exact location). The fake location L5 is generated by randomizing the latitude and longitudes values. Figure 6 shows how the sets are derived from a given address.

After identifying the base points in each set, the agent proceeds into analyzing the context and the privacy policy, in order to determine which base point is going to be used as a location output. In the absence of any relevant privacy policies, the agent uses the default settings profile.

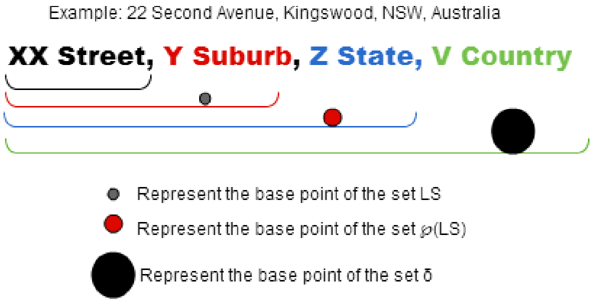

Example: 22 Second Avenue, Kingswood, NSW, Australia
XX Street, Y Suburb, Z State, V Country
Represent the base point of the set LS
Represent the base point of the set ℘(LS)
Represent the base point of the set δ

Fig. 6. The conversion of the base points into real addresses

B. Experimental Results

A test plan has been developed to evaluate the agent and the results were noted. The test plan included twenty test cases which test the generation of each of the obfuscation levels. Location inputs were automatically collected by the mobile device across Sydney and the surrounding suburbs. Location inputs were also collected using various networks (Wi-Fi and mobile' network); and with and without the use of a GPS.

By comparing the current location (location input) against the expected location (location output) and the location received (this is the location received by the computer application representing object B), we were able to verify if the location outputs were successfully generated by the agent for a given context. For each of the twenty test cases performed in this test plan, the agent generated location output with nil errors. An example of a test case is given in Table 3. Figure 7 shows the location output (response) received by object B (the computer).

C. Evaluation

While the DLDA agent generated location outputs with no errors, it failed to obfuscate the location in one particular case. If the current true location of object A is located very close to the coordinates of the level 1 base point location (suburb), and if the agent is set to disclose the suburb location (obfuscation L1); then it is noted that the location outputs of L0 and L1 are geographically close to each other if not the same. In order to address this shortcoming, instead of using a fixed base point for each set, the base point will be randomly generated in the future works.

In addition, this experiment suffers from two major limitations. Firstly, the DLDA agent is implemented in a centralized fashion by attaching it to the object. Given the decentralized nature of the IoT, a centralized solution is not considered the optimum approach to adopt in such environments. For instance, the solution is considered unreliable for managing multiple objects as it will require the installation of the agent on each object. Secondly, because the agent is attached to the object, the agent relies on the object for all the needed computations. To this end, future works will expand the current work to provide a holistic solution for objects in the IoT, by providing a decentralized location protection technique suitable to use, not only with general objects, but also with lite

TABLE II
A SAMPLE TEST CASE

Disclosure Settings	Location input	Obfuscations level	Expected output	Actual output
Only Suburb	56 Second Av, Kingswood NSW	Level 1 (Suburb)	Kingswood	Kingswood
Only Suburb	1 Ocnelst, Kingswood Av NSW	Level 1 (Suburb)	Kingswood	Kingswood
Only State	56 Second Av, Kingswood NSW	Level 2 (State)	NSW	NSW
Do not Disclose	56 Second Av, Kingswood NSW	Level 5 (fake)	A location different form the location input	A location different form the location input

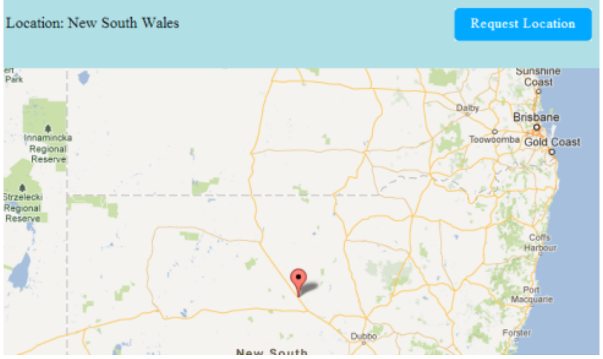

Location: New South Wales

Fig. 7. The server's responce

objects which does not necessarily have heavy computation or processing powers. Consequently, we will look into the possibility of incorporating cloud computing characteristics [6], VPN [11], and other techniques that could improve the DLDA agent by making it suitable for adoption in a decentralized environment such as the IoT. Computation capabilities need to be provided to the agent independently from the object as well. In addition, research into suitable experimental setups for a decentralized IoT environment will be explored.

VII. CONCLUSION

This paper has argued that context is a keystone of an overall approach to location privacy in the Internet of Things. A context-aware adaptive technique is presented in the paper, offering protection for location privacy throughout an agent. The agent provides a location privacy method adaptive to variations in contexts using an efficient context analysis process. In addition, the method takes into consideration the users' or objects operators' privacy preferences. In the development of the agent architecture, we have attempted to set out our assumptions in a clear and methodological way. The experimental works confirm that by applying the DLDA agent, the location privacy of an object has significantly improved. Planned future works have the objective of relaxing the assumptions made in this work by incorporating techniques that would promote the operation of the agent in decentralized environments such as the IoT. Future works will also look into ways that could confront the limitations challenging the experimental works reported in this paper.

ACKNOWLEDGMENT

The authors would like to thank the five anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

REFERENCES

[1] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, May 2000. [Online]. Available: http://doi.acm.org/10.1145/335191.335438

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

[3] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Pervasive computing*. Springer, 2005, pp. 152–170.

[4] M. Elkhodr, S. Shahrestani, and H. Cheung, "A review of mobile location privacy in the internet of things," in *2012 10th International Conference on ICT and Knowledge Engineering, (ICT Knowledge Engineering)*, 2012, pp. 266–272.

[5] M. Elkhodr, S. Shahrestani, and H. Cheung, "The internet of things vision and challenges," in *Proceedings of the IEEE Tencon spring conference*. IEEE, 2013.

[6] N. Giweli, S. Shahrestani, and H. Cheung, "Enhancing data privacy and access anonymity in cloud computing," *Communications of the IBIMA*, vol. 1, no. 462966, 2013.

[7] M. Gruteser and D. Grunwald, "A methodological assessment of location privacy risks in wireless hotspot networks," in *Security in pervasive computing*. Springer, 2004, pp. 10–24.

[8] N. Huijboom and T. Van den Broek, "Open data: an international comparison of strategies," *European journal of ePractice*, vol. 12, no. 1, pp. 1–13, 2011.

[9] L. Liu, "Privacy and location anonymization in location-based services," *SIGSPATIAL Special*, vol. 1, no. 2, pp. 15–22, 2009.

[10] M. Mano and Y. Ishikawa, "Anonymizing user location and profile information for privacy-aware mobile services," in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, ser. LBSN '10. New York, NY, USA: ACM, 2010, pp. 68–75. [Online]. Available: http://doi.acm.org/10.1145/1867699.1867712

[11] K. S. Munasinghe and S. A. Shahrestani, "Virtual private networks over a wireless infrastructure: Evaluation and performance analysis," in *Proceedings of the 9th WSEAS International Conference on Computers*. World Scientific and Engineering Academy and Society (WSEAS), 2005, p. 29.

[12] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," *Network, IEEE*, vol. 8, no. 5, pp. 22–32, 1994.