A Lightweight Anonymity & Location Privacy Service

Konstantinos Fysarakis, Charalampos Manifavas, Ioannis Papaefstathiou
Dept. of Electronic & Computer Engineering
Technical University of Crete
Chania, Crete, Greece
kfysarakis@isc.tuc.gr, cmanifavas@isc.tuc.gr, ypg@mhl.tuc.gr

Anastasios Adamopoulos
Dept. of Computer Science
University of Crete
Heraklion, Crete, Greece
anadam@csd.uoc.gr

***Abstract*—As embedded systems, in their various forms, become ubiquitous, new types of services are offered, aiming to enhance every aspect of the everyday life and allowing users to enjoy pervasive and personalized access to information. Along with the benefits come considerable threats to users' privacy, as these enhanced services operate on private sensitive information, such as the true identity and location of users. This paper proposes a lightweight, distributed anonymity and location privacy service which can be deployed on mobile embedded systems (e.g. smart clothing, smart phones), even heterogeneous in nature, allowing users to form communities which help them hide their identities and exact location. Acting as a proxy the proposed service can sanitize users' requests prior to forwarding them to a location-based service (LBS). The deployment of this type of information sanitization proxy would facilitate the wider adoption of the new generation of enhanced reality applications and services, alleviating the justified concerns regarding users' privacy.**

Keywords—location privacy; k-anonymity; embedded systems;

I. INTRODUCTION

Location-based services (LBSs) are a relatively new and rapidly expanding market, owing to the widespread use and advances both in mobile devices and positioning systems. Enhanced Reality applications, and pervasive computing in general, emerge and are expected to spread in the coming years, moving towards the Internet of Things (IoTs). Applications pertaining to the abovementioned concept include - but are not limited to - smart clothing, smart home and working environments, location-aware emergency response services, entertainment facilities and targeted advertisement mechanisms. Furthermore, even in privacy-sensitive applications like the ones mentioned above, it is still important to be able to monitor the network's and nodes' health, security incidents and generally aggregate various related data and alerts.

The location of individual users is necessary in order to enable the abovementioned services but, even though its disclosure may not pose a security risk for the embedded device itself, said information constitutes sensitive personal data of the user or users associated with each device and should be handled accordingly. Disclosure of user location to unauthorized parties raises significant concerns about the user's privacy as it can reveal information about her daily schedule, personal relationships, political affiliations, medical condition etc. Disclosure of such information may even pose a threat to user's safety as it could enable a malicious individual to harass, blackmail or even harm the user (e.g. entering her residence when she is away or asleep). In other instances, the LBS provider itself may be considered as an adversary since it can have economic benefit from exploiting user's sensitive data, like in the case of targeted advertising. Additionally, in smart office deployments certain legal and ethical implications stem from explicitly monitoring employee locations [1].

The rest of this paper is organized as follows. Section 2 includes some basic anonymity and location privacy concepts and introduces mechanisms utilized in this work, while Section 3 lists related work in the area. Section 4 details the scheme presented in this work, including the general architecture as well as node and server implementations. Further refinements and improvements to the proposed scheme are suggested in Section 5 and the paper concludes in Section 6.

II. ANONYMITY & LOCATION PRIVACY CONCEPTS

Location privacy can be protected through anonymity and cloaking services, i.e. hiding a node's identity among a set of peers and substituting its exact location with a larger region containing said peers. It should be mentioned that in many cases simply hiding the node's identity does not safeguard its location privacy. If a user anonymously submits service requests with unsanitized location information, the privacy of the user can be easily compromised by an untrusted or dishonest LBS provider. As an example, considering a smart-office environment, it is safe to assume that the LBS provider can have access to common user locations (e.g. the location of an employee's office). An observation of requests coming from an area around said office, correlated with employee's work hours, days off etc. if necessary, is more than enough for the LBS to infer that these requests come from the specific employee.

In order to guarantee anonymity and location privacy, an anonymizer component can typically deployed, being responsible for decentralizing data, encrypting it, changing the traffic pattern, flooding the network or implementing a k-

This work has been supported by the Greek General Secretariat for Research and Technology (GSRT), under the ARTEMIS JU research program nSHIELD (new embedded Systems arcHItecturE for multi-Layer Dependable solutions) project. Call: ARTEMIS-2010-1, Grand Agreement No.:269317.

anonymity scheme for location cloaking. The latter is the privacy concept that will be implemented in this work. The basic principle is that an entity's location information must always be sent in a form that makes said entity indistinguishable from k-1 other neighboring entities [2]. For example, assuming a k value of 10, any information transmitted to an LBS must be sanitized in such a way that the service provider will not be able to identify a service user from 9 of her peers. It is important to note that the k-level should be set appropriately for each scenario. A 5-anonymity scheme might be acceptable for a small office environment but, obviously, not enough to anonymize a family of five living in a detached house. Another parameter that needs to be considered is that a lower k value provides less privacy protection but also better quality of location monitoring and service accuracy, while the opposite stands for higher k values. In general, there are two important and mostly unavoidable tradeoffs: a tradeoff between privacy and quality of service and a tradeoff between privacy and personalization [3][4].

The anonymization service implemented in this work is based on the k-anonymity privacy concept and, more specifically, the TinyCasper scheme. Said scheme was designed as a privacy-preserving location monitoring system and is introduced and formally validated in [5]. In more detail, the system consists of two modules: the in-network location anonymization (at node level) and the aggregate query processing over the anonymized locations (which takes place at the server). The core of the system implemented in this work is the in-network location anonymization module, which involves direct communication between nodes, allowing deployment in applications where the server cannot be fully trusted. During this phase a node collaborates with peers to calculate a cloaked area containing at least k users and it transmits said area, along with the number of contained users, to the server. The authors include two anonymization algorithms, a resource-aware variant which aims to minimize communication and computational cost and a quality-aware one which aims to maximize the accuracy of the aggregate locations by minimizing their monitored areas.

By exploiting the abovementioned anonymization scheme, we develop a service which, under the assumption that the nodes form a trusted community, reverses the information flow of the original scheme and enables these users to take advantage of location-based services without compromising their privacy. The abovementioned enhancement over the original system introduces further complications and requires certain provisions (e.g. catering for node mobility) which will be addressed in the sections to follow. On the other hand, as no modifications were made to TinyCasper's core mechanisms (i.e. the cloaked area calculations and area validation), its security and privacy properties remain unaffected.

III. Related Work

Literature dealing with location privacy issues includes a few recurring methods and their variations. The pseudonym-based methods involve disposable pseudonyms for each node in the location service [6]. These pseudonyms change over time, being used as temporal identifiers in a way that makes it hard for an attacker to track the users. This concept is further expanded with the introduction of silence periods, as detailed in [7], a mechanism which could be explored in the implementation presented in this work as well. Alternative schemes rely on path perturbation, i.e. crossing paths in areas where at least two users meet [8]. Still, anonymization methods aiming to remove identifying information using generalizations and suppressions, are the most popular in the literature, with k-anonymity (detailed above) being the basic mechanism used, as in [9]. Authors in [10] propose the use of personalized k values, for systems with context-sensitive privacy requirements. Said work, though, is centralized nature, with a trusted anonymity server doing all the data sensitization via a message perturbation engine. With L-diversity [11], another dimension can be added to k-anonymity, where L is a set of distinct locations. Furthermore, the use of dummy locations [12] and semantic information [13] are proposed to address issues that are not solved by plain k-anonymity mechanisms.

IV. Service Analysis

To facilitate the deployment on resource constrained devices and heterogeneous environments, it was decided to avoid assumptions about the capabilities of devices that will have to run the anonymity service. Thus, the resource-aware version of the original scheme was implemented. Choosing a lightweight scheme produces a universal solution which can be run in heterogeneous systems, from highly resource constrained devices like wearable units, wireless sensors and various embedded systems to high-power devices like modern smart phones. There are no prerequisites regarding network topology since communication is based on a distributed tree; only a path from each node to the server is needed.

As already mentioned, the scheme presented here is not limited to location monitoring of objects (as in the original TinyCasper). Nodes can also act as proxies to serve requests of the objects (i.e. users) to services requiring their location. The introduction of this reverse information flow, i.e. having users forward their location-related requests to a node in range, allows the utilization of relevant services by the users without disclosing the true identity and exact location. In fact, the nodes could be actual users of the service themselves; i.e. a subset of the user base who opt to allow their devices to act as proxies for other users of the anonymity service. This concept is similar to how the TOR network [14] operates, where the users can, upon installation, choose if they want to join the anonymization network as simple clients or as clients and proxies, helping anonymize the traffic of other TOR users on the Internet. An overview of this functionality can be seen in Fig. 1. It is worth noting that, as all entities are mobile, the exit node role is shared between proxy nodes, depending on their proximity to the server at the time a request is served.

This significant modification on the original system necessitated the introduction of certain refinements in parts of the service. The latter was also extended to cater for nodes with different capabilities as various users may have heterogeneous devices. More importantly, provisions have been made for mobile nodes (e.g. wearable nodes), moving away from the static infrastructure deployment of the original

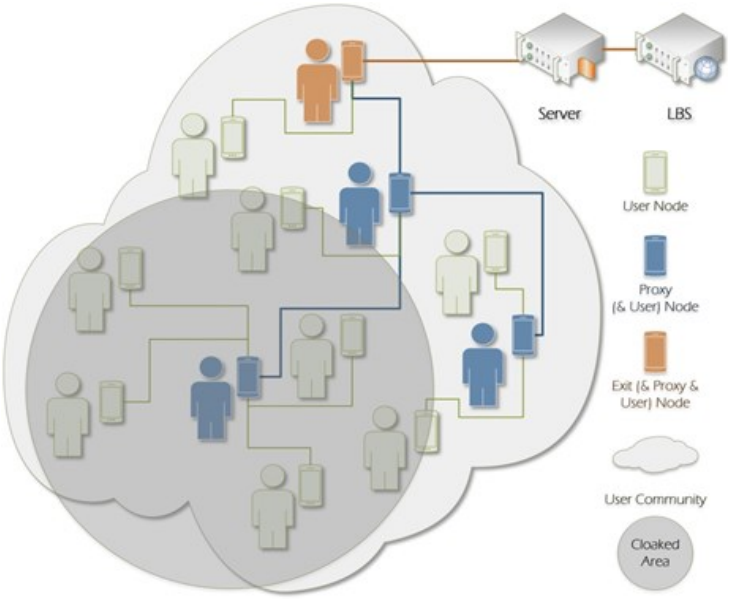

Fig. 1. Overview proposed scheme, including main entities and roles

service. To enhance this functionality, special features present on the test platforms were exploited, namely using an accelerometer to detect when the node is mobile and trigger the relevant events. Finally, a corresponding application was developed on the server side, featuring a graphical user interface (GUI) to facilitate system monitoring.

A. Test Equipment

In terms of the hardware utilized, the server functionality was implemented on a mains-powered Beagleboard-xM platform [15], featuring a 1GHz ARM Cortex-A8 processor, 512MB of RAM and a minimal Linux-based operating system with the lightweight LXDE window manager. The nodes were deployed on Beaglebone platforms [16], equipped with a 720MHz ARM Cortex-A8 processor, 256MB of RAM, a minimal Linux-based operating system. A battery cape (expansion board) was attached on the Beaglebones to enable the use of four 2000mAh NiMH batteries as power source. All platforms were equipped with RTL8192-based Wi-Fi adapters operating in ad-hoc mode, while node platforms were also equipped with Analog Devices ADXL345, 3-axis digital accelerometer modules (to detect node movement and trigger position update routines).

B. Node Platforms

As already mentioned, the system implemented relies on the k-anonymity privacy concept that aims to make a user indistinguishable from k-1 of her neighbors. In more detail, every sensor node is connected to its neighbors. The location of every node and its coverage area is modeled as coordinates of a grid, which can trivially be mapped to GPS coordinates in actual deployments. Nodes may have different ranges and sensing areas due to hardware discrepancies and limitations. It is expected that the server will not have direct communication with most system nodes, but, since a distributed tree scheme is used, this is not an issue as long as there is a path between the two communication endpoints.

The coverage area (or different sizes of coverage areas) supported by a node must be set during its initialization. For example, referring to Fig. 2, red sensors RS={J} have a 12x12 coverage area, purple sensors PS={H,I,B,F,E,D} cover a 5x5 area and green sensors GS={A,C,GI} have an area of 7x7 on the grid. An additional feature was introduced to allow for each node to switch dynamically between supported coverage areas, enabling the system to better adapt to various situations, like a node entering a low power state and, thus, covering a smaller area. For accuracy and reliability purposes, it is important that, at any given time, these assigned areas are smaller than the actual range of the nodes.

Every node communicates with its neighbors, broadcasting its pseudonym, sensing area and number of served users. The purpose of these lists is to allow nodes to find k users in an area as close as possible. When peers receive these messages, they rebroadcast them until all their neighboring nodes have enough number of users, i.e. at least k users. If a node does find the required number of users, it notifies its neighbors. Otherwise the node informs its peers that it is still trying to find k users. In response, each neighbor sends its peer list to the node that needs help reaching its k-level. Whenever a node does reach the desired k level based on information received, it computes a score for every peer in its peer list, as in:

$$Score = (users\ in\ area\ of\ peer) / (distance\ to\ peer) \quad (1)$$

After the abovementioned score is calculated, the node selects the peers with the highest score and computes the cloaked area. The cloaked area is, therefore, a minimum bounding rectangle of the area (of the node and its chosen peers) which contains at least K users. Finally, the node sends

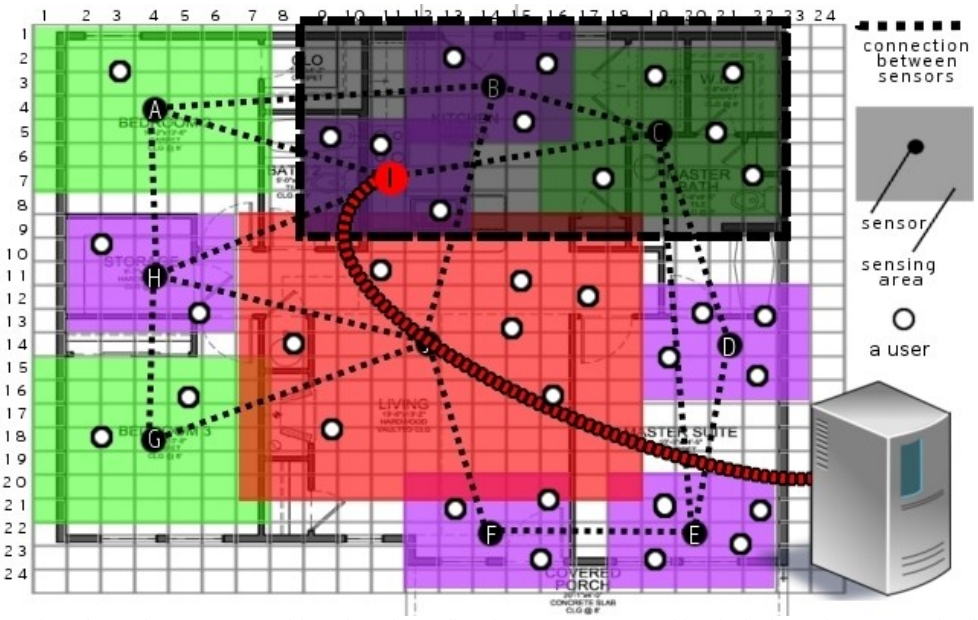

Fig. 2. Sample node topology on server grid (projected over floor layout). Node "I" and its cloaked area for 8-anonymity (i.e. k=8)

the cloaked area it just calculated to its neighboring nodes, validates that this area is unique and sends it to the server along with the total number of users contained in the area. An example of this process can be seen in Fig**. 2**, where the anonymity level is assumed to be 8 (i.e. k=8). In this case, node "I" picks a cloaked area which includes its own coverage areas and that of its peers "B" and "C", encompassing 11 users in total (thus exceeding the minimum required, i.e. 8).

The validation step mentioned above is essential and aims to avoid reporting cloaked areas with a containment relationship. If such cloaked areas were to be reported, an adversary could infer the number of users in the non-overlapping area, possibly breaking the k-level condition.

Once the above process is complete and the cloaked area is validated, the node can be used as a proxy, serving requests of users and allowing said users to utilize LBSs without disclosing their true identity and exact location. These LBS-related messages are forwarded on behalf of the user to the server. In this phase, inter-node communication is accomplished via the Ad Hoc On-Demand Distance Vector (AODV, [17]) routing protocol. At the last hop and before a message leaves the community of nodes (i.e. to be sent to the anonymity server), it is stripped of any identifying information regarding the source of the request and is then transmitted to the server. The choice of the exit node only depends on its proximity to the server and, given that nodes and users in general are mobile, any node may be assigned that role at a point in time.

When a node moves to another location, it broadcasts a new message to discover its peers. In this implementation the function is triggered by an accelerometer which is installed on the chosen test platforms, but other triggers can be used as well (e.g. new position reported by the GPS unit). As long as the node keeps moving (i.e. the accelerometer continuously detects movement), the node repeats the above process at pre-set intervals. The actual timing of these intervals should be set depending on the grid dimensions, as requirements vary between, e.g., an indoor space, with grid units covering 1 square meter and an outdoor space with 20 square meters of actual area covered per grid position. Other factors that should be considered when deciding optimal update intervals include the average expected speed of objects, sensor's resources (long intervals for lower performance overhead and, thus, lower power consumption) and, last but not least, the location accuracy required by the LBSs users typically utilize.

In terms of the software implementation, every node structure contains all the necessary information (e.g. pseudonym, user count, address) as well as dynamic linked lists with information about its neighbors (i.e. list of neighbors, peer list of the neighbors that reached the k-anonymity level, neighbors' cloaked areas). All data structures are dynamic to cater for the lack of memory on some resource-constrained nodes.

A valid message interchange between nodes must have one of the structures defined in Fig. 3, which presents a typical node message exchange. The supported *callsign* types are *SNINF* (a node info package), *NOTIF* (notification package that sender reached K anonymity level), *CAREA* (cloaked

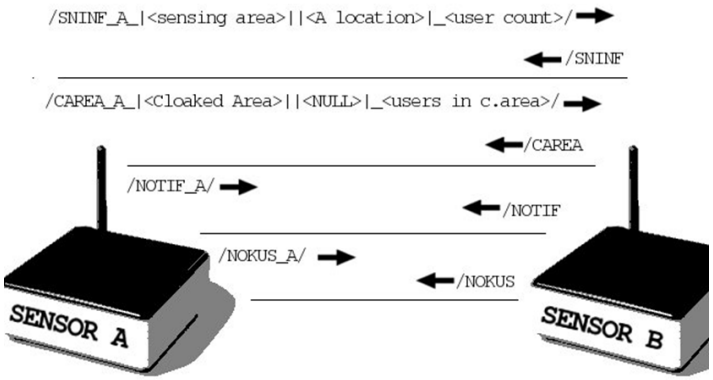

Fig. 3. Typical message exchange between nodes.

Area of the sender) and *NOKUS* (sender did not reach K anonymity level). If the receiver accepts the message, it replies with the callsign. If the reply is valid, the sender considers that the message has been delivered successfully.

1) Performance Analysis

In order to assess the performance of the mechanism developed, a scenario involving a smart hospital environment was adopted. In such a case, a potential LBS request could involve locating the nearest free examination room. Thus, several proxy nodes where deployed (spawning at random locations for each iteration), one of which had to serve such a request. Moreover, two test cases where investigated, involving 5 and 10 proxy nodes. The latter was chosen as a worst case scenario in terms of overhead, as ten proxy nodes in the relatively limited 24x24 area of our deployment scenario can be considered a congested community in terms of proxy participation. The performance figures reported are the average of ten test runs for each test case.

An important parameter that was analyzed is the time elapsed before a proxy node is fully initialized, i.e. before the node is ready to process, anonymize and forward LBS requests. This averaged 0.54 and 0.95 seconds in the 5 and 10 node scenarios respectively. The memory footprint of the service, when deployed on the Beaglebone test platforms, was measured at 0.7MB when deploying 5 nodes and 1.23MB in the case of 10 nodes. Evidently, both of the abovementioned aspects of the performance deteriorate when moving from the more realistic 5-node deployment to the crowded 10-node one, but the extra overhead remains insignificant in terms of the memory footprint and not enough to be deemed problematic or prohibitive in terms of delay time.

Further profiling of the anonymity service application and its execution on the proxy nodes reveals that the bulk of the performance overhead can be attributed to proxies communicating with each other (to populate the peer lists, ask for more lists, notify that k-level has been reached etc.). Additionally, significant part of the procedure is devoted to calculating and validating the cloaked area itself. A detailed breakdown of the initialization phase on the 5-node scenario can be seen in Fig. 4. The equivalent profiling figure pertaining to the 10-node test case has been excluded, as the only notable difference between the two test cases is an unsurprising increase in the percentage devoted to proxies communicating with their peers (up from 38% to 46%) with a

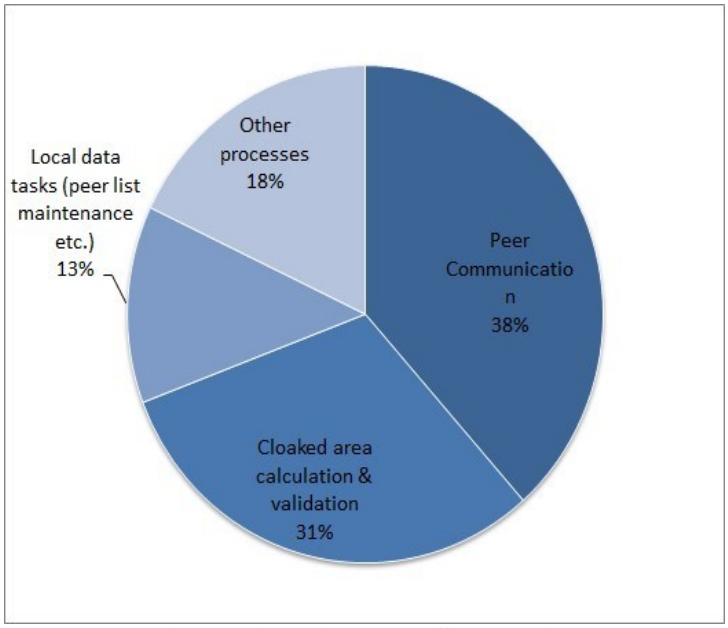

Fig. 4. Service performance profiling – 5 nodes

corresponding decrease in the percentage allocated to cloaked area-related and other processes.

C. *Server Platform*

The main task of the server is to act as a proxy between the actual LBS service and the community of users and nodes. Hence, the server will be responsible for formulating service requests received by the nodes in a way that these will be compatible with the supported LBSs. E.g. if an LBS requires an explicit location to provide its services, the server can select a random position inside the cloaked area and report that position to the LBS server. Additionally, since enhancements introduced in this work do not interfere with the functionality of the original scheme, if desired, the user histogram can be calculated on the server (as described in [5]). The only requirement is that the nodes accurately report their total user count along with their cloaked area. This allows for privacy-preserving location monitoring of the users and facilitates the deployment of services that require such monitoring (emergency response, smart power management etc.).

As the server is not considered trusted, it works on a need-to-know basis. It is not aware of the type of each sensor, its actual sensing area nor its real identity but just the pseudonym included in the request. Moreover, as already mentioned, any message arriving to the server from the community of nodes is stripped of all but the necessary information. Therefore the server is only aware of the exit point of the community and not the actual source of the request. Contrary to the original TinyCasper scheme, the server is not able to set the desired k-level either. This is a parameter set by the nodes themselves and can, optionally, be different between nodes. This facilitates the adaptation of the anonymity service to node capabilities and requirements. More importantly, it eliminates the potential threat of a malicious server setting a very low (or even zero) k-level, effectively negating the scheme.

The server implementation was designed to be deployable in most platforms, without imposing a significant performance overhead. The Java programming language was used in order to have a platform-unaware application. Moreover it was designed to be as lightweight as possible without compromising functionality and was intentionally tested on an embedded ARM platform with relatively limited resources (i.e. Beagleboard xM). The application features a graphical user interface (GUI) which displays the system grid, latest cloaked areas received and the pseudonym associated with those areas. The application is deployed as a bundle for Knopflerfish [18], an open source service platform following the Open Services Gateway initiative (OSGi, [19]) specification. The anonymity service is expected to be deployed alongside other services on the server platform. In view of that, the modular and dynamic service deployment as well as the service orchestration features provided by the OSGi framework will be advantageous in actual deployments. A screenshot of the server interface can be seen in Fig. 5, where the proposed scheme is deployed in a "smart hospital" scenario. The floor layout is visible, divided into grid squares, with the latest node cloaked area and request overlaid upon it. In this scenario the request received involved a query for a free exam room, nearest to the cloaked are reported by the node.

In terms of the resources required, the server application only occupies a low amount of memory when idle, i.e. 36MB,

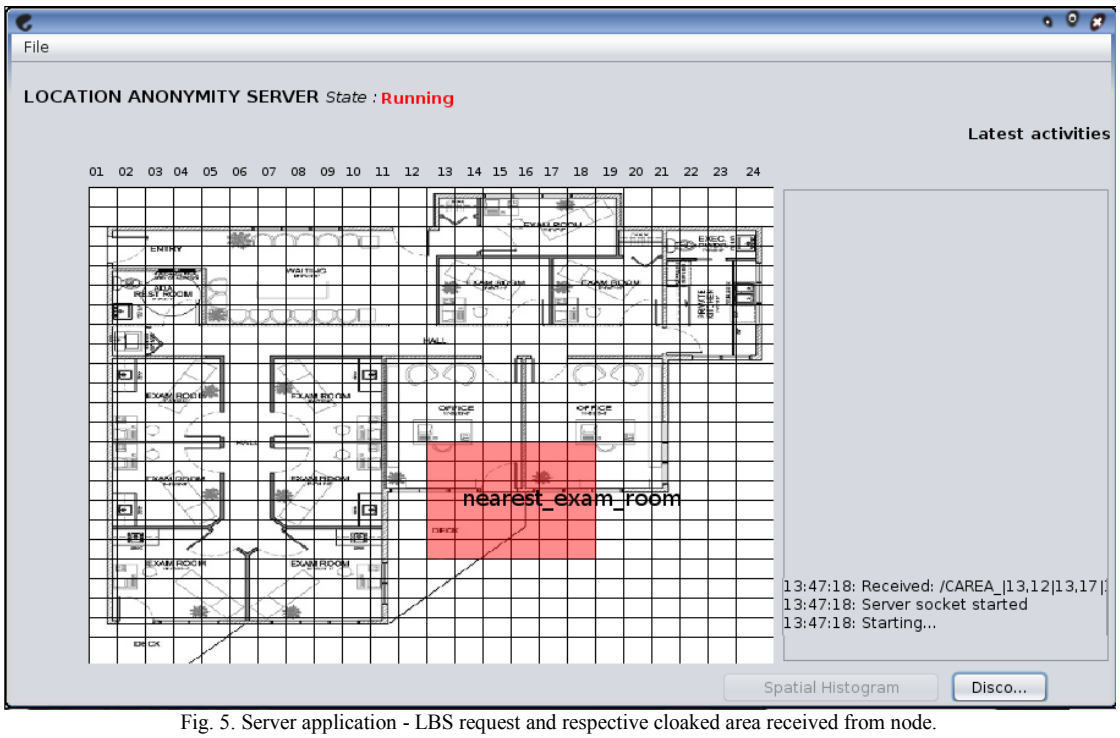

Fig. 5. Server application - LBS request and respective cloaked area received from node.

which rises and stabilizes to about 62MB under load. The reported memory footprint can be considered negligible if the server is deployed on a personal or power computing system, but is also suitable for deployment on somewhat resource-constrained embedded devices like the Beaglebone-xM platform used in this work.

V. FUTURE WORK

Further work on the presented system could focus on the development of an alternative mode of operation, fully distributed in nature, for deployments where the server is not required at all and nodes communicate directly with the LBS, would be desirable. In such cases, the nodes' pseudonyms should change over time, introducing silence periods to make it harder to correlate old and new pseudonyms, further obfuscating the real identities of the nodes, as proposed in [7]. The potential disclosure of location semantic information should also be examined, adding, if necessary, mechanisms to ensure that cloaking is done with semantically heterogeneous locations, as proposed in [13]. This enhancement would, naturally, impose an extra overhead in cloaked area calculations, so its applicability will have to be assessed in the context of the proposed scheme, especially in deployments featuring time-critical services.

Another potential enhancement would be to safeguard the proposed scheme against malicious and/or selfish nodes in the community. There are various mechanisms in the literature which could be used to accomplish that, like combining the AODV routing protocol already used for inter-node communications with a reputation system (e.g. as presented in [20]), or even substituting AODV with an anonymous routing protocol, like ANODR [21].

VI. CONCLUSION

With the wider adoption of smart devices, be it handheld, embedded in smart infrastructures or even in smart-clothing, the Internet of Things is coming closer to realization. One of the key benefits of ubiquitous computing is the provision of personalized, enhanced reality services. Significant privacy concerns arise, however, by the adoption of such services, as they base their operation on users' private sensitive data (their true identity, location, habits etc.). It is therefore essential to provide safeguards and guarantee users' privacy, if these types of services are to be widely adopted.

This work proposed a scheme which aims to protect users' privacy while allowing them to enjoy the benefits of said enhanced services. The proposed anonymity and location service is based on the k-anonymity concept and allows for privacy-aware monitoring of users as well as allowing users to forward their own requests to potential LBSs.

It is essential to provide safeguards and guarantee users' privacy, if personalized, location-based and enhanced-reality services are to be widely adopted. The anonymity and location privacy preserving service presented in this work helps alleviate most of the aforementioned concerns and could facilitate the move to the pervasive computing future and its applications.

REFERENCES

[1] Kaupins G., Minch P.: Legal and Ethical Implications of Employee Location Monitoring. In Proc. of the 38th Hawaii International Conference on System Sciences, Los Alamitos, IEEE Computer Society. 2005.

[2] Sweeney L.: K-anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), pp. 557-570. 2002.

[3] Beresford R. A.: Location Privacy in Ubiquitous Computing. University of Cambridge, Computer laboratory, Technical report no. 612, UCAM-CL-TR-612. 2005.

[4] Liu L.: From Data Privacy to Location Privacy: Models and Algorithms. In Proc. of the 33rd international conference on Very large data bases (VLDB '07), VLDB Endowment pp. 1429-1430.2007.

[5] Chow C., Mokbel M. F., He T.: A Privacy-preserving Location Monitoring System for Wireless Sensor Networks. IEEE Transactions on Mobile Computing 10, 1, pp. 94-107. Jan. 2011.

[6] Gruteser M., Grunwald D.: Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A quantitative analysis. WMASH'03, San Diego, California USA. 2003.

[7] Huang L., Matsuura K., Yamane H., Sezaki K.: Enhancing Wireless Location Privacy Using Silent Period. In IEEE Wireless Communications and Networking Conference (WCNC 2005), New Orleans, LA, USA. March 2005.

[8] Hoh B., Gruteser M.: Protecting Location Privacy Through Path Confusion. In Proc. of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, IEEE Computer Society. Athens, Greece, September 2005.

[9] Zhong G., Hengartner U.: Toward a Distributed K-anonymity Protocol for Location Privacy. In Proc. of the 7th ACM workshop on Privacy in the Electronic Society (WPES '08). ACM, New York, NY, USA, pp. 33-38. 2008.

[10] Gedik, B.; Ling L.: Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. IEEE Transactions on Mobile Computing, Volume 7, Issue 1, pp. 1-18. 2008.

[11] Machanavajjhala A., Kifer D., Gehrke J., Venkitasubramaniam M.: L-diversity: Privacy Beyond k-Anonymity". ACM Trans. on Knowl. Discov. Data 1, 1, Article 3. 2007.

[12] Gupta K., Yadav A.S., Yadav S.: Location Privacy Using User Anonymity and Dummy Locations. International Journal of Innovative Technology & Creative Engineering, Volume 1, No 10, pp. 5-8. 2011.

[13] Byoungyoung L., Jinoh O., Hwanjo Y., Jong K.: Protecting Location Privacy Using Location Semantics. ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). San Diego, California, USA, August 2011.

[14] TOR Project: Anonymity Online [Online]. Available: https://www.torproject.org/ [Accessed: June 21, 2013]

[15] Beagleboard xM platform specifications [Online]. Available: http://beagleboard.org/hardware-xM [Accessed: June 21, 2013]

[16] Beaglebone platform specifications [Online]. Available: http://beagleboard.org/bone [Accessed: June 21, 2013]

[17] Perkins, C., Belding-Royer E., Das S.: Ad hoc On-Demand Distance Vector (AODV) Routing. IETF. RFC 3561. July 2003. [Online] Available: https://tools.ietf.org/html/rfc3561 [Accessed: June 21, 2013]

[18] Knopflerfish open source service platform [Online]. Available: http://www.knopflerfish.org [Accessed: June 21, 2013]

[19] Open Services Gateway Initiative (OSGi) [Online]. Available: http://www.osgi.org [Accessed: June 21, 2013]

[20] Anker, T., Dolev, D., Hod, B.: Cooperative and Reliable Packet Forwarding on top of AODV. In Proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad-hoc, and Wireless Networks, Boston, MA, April 2006.

[21] Jiejun K., Xiaoyan H.: ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In Proc. of the 4th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc). ACM, New York, NY, USA, 291-302. 2003.