Privacy-Triggered Communications in Pervasive Social Networks

Murtuza Jadliwala[†], Julien Freudiger[†], Imad Aad[‡], Jean-Pierre Hubaux[†], and Valtteri Niemi[‡]

[†] School of Computer and Communication Sciences, EPFL, Switzerland
firstname.lastname@epfl.ch
[‡] Nokia Research Center, Lausanne, Switzerland
firstname.lastname@nokia.com

ABSTRACT

Pervasive communications extend social networking by enabling users to share local contextual data in a peer-to-peer fashion by using their mobile devices. Such pervasive social networks bring along new privacy challenges in terms of location and data privacy. In order to address some of these challenges, there has been research on privacy preserving mechanisms and on privacy measurement techniques. However, little has been done so far on conditioning users' communications to their privacy. For example, a user may communicate personal information when he feels private enough and refrain from communicating when he feels "exposed". In this paper, we introduce the concept of *privacy-triggered communications* that enables users to make communication decisions based on their privacy level. To do so, we provide tools and techniques to visualize privacy and control communications. We evaluate existing privacy metrics and analyze with simulations the network performance when the underlying communications are privacy-triggered. We also provide an implementation of the concept on a testbed of mobile devices.

1. INTRODUCTION

Pervasive communications enable mobile devices to wirelessly share data with each other in a peer-to-peer fashion without user intervention. As more mobile devices become equipped with peer-to-peer technologies, such as WiFi and Bluetooth, friends or strangers can use their mobile phones to dynamically exchange information when they are in proximity. These peer-to-peer communications enable *context-awareness* on mobile devices and thus connect Internet applications to the real world. For example, users can share information in real-time for local-area social networking [8, 16], dating [1, 2, 37], personal safety [43], or microblogging [24]. These applications enable users to enter information in their mobile phones, it is then shared with other phones nearby unbeknownst to their users, effectively obtaining pervasive social networks.

Some users of online web services have an aversion to sharing their contextual information with infrastructure-based services because of privacy concerns [29]. Sharing personal information locally in a peer-to-peer fashion prevents this problem, but leaks personal information to wireless eavesdroppers, thus making possible the continuous tracking of users location and habits. In particular, personal information exchanged between mobile devices can identify their owners and threaten their privacy. The promises of peer-to-peer wireless sharing of information may turn into a pervasive nightmare if user privacy is not protected.

The upcoming generation of mobile computing thus requires mechanisms to share information anonymously in a peer-to-peer fashion. Privacy enhancing technologies for wireless communications have become a central topic of research and various approaches to user privacy have been suggested. Several solutions propose to remove identifiers [27] from exchanged messages or to change them over time [13, 53] to make it more difficult for an adversary to link data to users. Other solutions rely on anonymous authentication, such as group signatures or anonymous credentials, to authenticate and/or encrypt communications.

Anonymous authentication and data encryption enable users to share data with friends (with whom they share security credentials) in a privacy-preserving fashion. Nevertheless, such mechanisms do not work for sharing data with strangers because of the unavailability of shared security credentials. One solution for preserving privacy in this scenario is to share data only if several nodes are in proximity (as otherwise there is only one possible sender for a message). This allows users to share information in a privacy-preserving fashion with strangers.

In this work, we introduce the concept of *privacy-triggered communications* that enables users (or their devices) to make communication decisions based on their privacy level. In contrast with existing approaches, it allows users to regulate the sharing of their information by making dynamic privacy-based decisions. Hence, the

advantage of our approach is that users can identify appropriate moments to share data with nearby nodes: for example, share intimate data only if their privacy level is sufficiently high.

To do so, we compute user privacy based on simple network characteristics and on well-known metrics. We then implement efficient tools to visualize privacy and user controls for privacy-triggered communications on a testbed of mobile devices. Privacy-triggered communications affect the response time between devices (often delaying communications). Hence, we also evaluate the introduced delay on large networks using ns-2 simulations and derive the relation between privacy and quality-of-service of context-aware applications. To the best of our knowledge, this paper is the first to propose, evaluate and implement such concept for mobile devices. With our approach, we tackle one of the important issues that has hindered the use of peer-to-peer wireless communications between mobile devices.

2. STATE OF THE ART

Context and location awareness turn phones into ubiquitous tools for quantifying personal patterns and habits. Several efforts from the industry and academia have proposed context-aware platforms for mobile phones. Dey, Salber and Abowd [20] introduce a conceptual framework for context-aware systems architecture that Korpipaa and Mantyjarvi [39] extend to mobile phones. Raento *et al.* [44] implement a prototype for Nokia phones that enable context-aware applications. Recently, Nokia introduced an *awareness networking solution* [8] for the transport of small amounts of locally relevant data in a power-efficient manner between mobile phones. Their work is a first step towards making pervasive communications energy-efficient. We consider the proposal of Nokia and focus on privacy-problems introduced by context-awareness for social networking applications.

There are several solutions to provide privacy in a pervasive environment. Confab, proposed by Hong et al., is the first toolkit for facilitating the development of privacy-sensitive ubiquitous computing applications [34]. SmokeScreen is another proposal by Cox et al. that introduces flexible privacy controls for presence-sharing [18]. Our work complements both Confab and SmokeScreen by introducing privacy regulation controls that deal with data sharing. Another field of privacy research focuses on privacy-aware routing in ubiquitous networks [9, 52]. These schemes complement our approach to provide privacy when multi-hop routing is required. Other mechanisms protect user *location privacy* by using multiple pseudonyms [13, 53], path-cloaking [26, 31], or k-anonymity [35]. In our work, we assume the use of such mechanisms to protect user location privacy. Similarly, previous work studies the protection of privacy in people sensing applications [17]. Our work focuses instead on the peer-to-peer interactions between mobile devices. Another related work is privacy-aware data sharing in online social networks [41]. The authors propose to measure the sensitivity and visibility of information in social networks and provide a privacy score as a feedback for users to decide with whom to share information. Similarly, our work suggest mechanisms to share data but focuses on peer-to-peer wireless communications.

As investigated by Altman in a sociological study in the seventies [10], users manage their privacy as a dialectic and dynamic regulation process. In other words, users decide to share data depending on context. Later work studied how the management of privacy changes with the evolution of technology [42] and suggested that privacy management should be a dynamic response to circumstances rather than a static enforcement of rules. Yet educating users and giving precise privacy controls to dynamically tune privacy can make things worse [49] because it becomes increasingly difficult for users to understand the mechanisms underpinning the technology. Instead, in this work, we propose simple privacy metrics to help users regulate their privacy dynamically depending on the context.

Note that it is possible to identify devices relying on their distinctive characteristics (i.e., fingerprints) at the physical, link, OS, and application layers. At the physical layer, the wireless transceiver has a wireless fingerprint that can be used to identify it in the long term using modulation-based techniques [15], transient-based techniques [19], amplitude-based techniques [50] or a combination of features [28, 45]. However, these techniques are only evaluated with specific scenarios (static, small scale experiments with expensive hardware) and countermeasures could be developed. Hence, in mobile networks, it remains unclear how much identifying information can be extracted from the physical layer. At the link layer, it is possible to distinguish between a number of devices and drivers [23]. At the application layer, devices can also be identified based on clock skews [38]. However, such techniques require an active adversary and can be countered by ignoring the requests sent by the adversary. Similarly, a reduction of the differences between drivers would limit the effectiveness of such attacks. Note that independently from the presence of fingerprinting attacks, higher layer privacy mechanisms remain useful. Some applications may, for example, require storing location traces for a while (e.g., for congestion analysis).

3. SYSTEM MODEL

In this section, we introduce the assumptions made throughout the paper.

3.1 Network and Communication Model

We study a pervasive network composed of autonomous mobile devices equipped with one or more wireless interfaces (e.g., WiFi or Bluetooth). Devices can communicate with each other in an ad hoc fashion upon coming in radio range. Ad hoc (or peer-to-peer) wireless communications complement communications with an infrastructure such as cellular or WLAN.

Mobile nodes have an *always-on* wireless interface for ad hoc communications, which allows for continuous neighborhood awareness. Without loss of generality, we assume that mobile users advertise their presence by periodically broadcasting proximity beacons containing the nodes identifying information. This beaconing can be done anonymously using a privacy-preserving mechanism such as [12, 18]. Moreover, beacons do not contain intimate information and thus do not pose a data privacy threat. After discovering their neighborhood, mobile devices can automatically exchange information unbeknownst to their users. Typically, the shared information relates to users' personal interests and context. The shared data can either be generated on-the-fly based on the device sensors or be preloaded on the device by users.

As suggested by Athiainen *et al.* in [8], we consider that communications between nodes are small in size (100 bytes per packets) to guarantee a low computation overhead on the mobile nodes. Thus, small packets make peer-to-peer communications energy-efficient and enable always-on networking.

As commonly assumed in pervasive networks, the sharing of information relies on a push/pull system: Mobile nodes can push information to the network by making it public, or pull information from nearby nodes by asking questions. For example, a node can advertise (push) the allergen concentration measured in another region of the network, or ask (pull) its neighbor their relationship status. Devices then collect and archive the published information for later use by the owner of the mobile device. The decision to push/pull information is made by the devices unbeknownst to their owners.[1] Without loss of generality, we consider that messages are broadcasted in the following format:

|| User pseudonym || Message ||

We consider that communications are multi-hop to a certain maximum hop-count defined by the system. Hence, traditional routing algorithms can be used [36].

3.2 Threat Model

We consider an adversary that attempts to obtain information about mobile nodes based on observed messages. In practice, the adversary can be a rogue individual that deploys its own infrastructure (e.g., by placing eavesdropping devices in the network) or a set of malicious legitimate nodes. In the worst case, the adversary is *global*: it has a complete coverage and eavesdrops communications throughout the entire network. The adversary has the capability to use directional antennas to detect the source of transmissions.

The goal of the adversary is to track users' locations and break his data privacy. By collecting messages together with their location, the adversary learns a device's whereabouts and can implicitly obtain the true identity of the owner of the device from the analysis of pseudonymous location traces [13, 33]. Similarly, by observing the exchanged messages, the adversary can learn intimate information about users and their social networks.

Note that even if messages are encrypted, the adversary (being a legitimate member of the network) authenticates himself to nearby nodes to establish communications. Hence, in this work, we consider without loss of generality that all messages are unencrypted.

4. PRIVACY-TRIGGERED COMMUNICATIONS

In this section, we introduce the concept of *privacy-triggered communications*. It consists of a privacy-wrapper that dynamically determines when to share personal information (Fig. 1). Intuitively, the privacy-wrapper monitors the surroundings of mobile users and determines whether the context provides enough privacy for users to share their personal information. Hence, the context acts as a trigger of the networking activity of the nodes.

In practice, mobile phones check their privacy level and then decide to push/pull data from neighboring nodes. An example where privacy-triggered communications can be useful is an application used by commuters in the subways of Tehran, Iran. Mobile-phone owners use Bluetooth to circumvent prohibition laws and exchange political jokes or sexually-related content in the metro. The large number of people in the metro provides an appropriate level of anonymity [37]. In this case, privacy-triggered communication is manually enforced by communicating in high density areas such as metros.

4.1 Privacy-Preserving Mechanisms

In wireless networks, user privacy mostly depends on the protection of *data privacy* and *location privacy*. An adversary should not be able to link collected data (data privacy) and locations (location privacy) to a user's identity. These two problems are interdependent and need to be considered together.

In order to prevent trivial linking of users' data and locations by the adversary, we assume that the mobile

[1]Note that to facilitate the exchange of information, nodes may be grouped into communities based on their interest and social affiliation. Like publish/subscribe systems, communities enable users to publish information in a coordinated fashion, and to subscribe to certain topics.

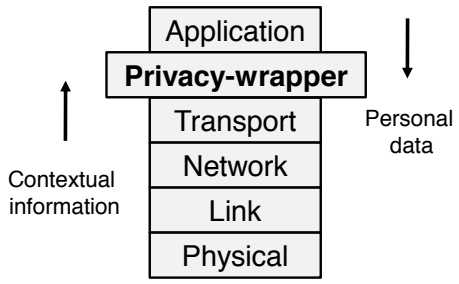


Figure 1: Illustration of the privacy-wrapper for privacy-triggered networking. The privacy-wrapper uses contextual information to control the sharing of user information.

devices use pseudonymous communications. Any suitable pseudonym mechanism can be used. For example, mobile devices can use multiple pseudonyms over time [13, 53], remove pseudonyms from their messages, or use anonymous credentials. By means of these pseudonym mechanisms, mobile users can achieve a basic level of anonymity. In addition to these mechanisms, the overall privacy of users is also context-dependent. For example, a single user is not private even if he regularly changes pseudonyms, but can achieve some anonymity by mixing in a group of people (and ofcourse use and change pseudonyms). Such context-derived privacy can be measured using various metrics, as discussed next.

4.2 Privacy Metrics

In order to use privacy-triggered communications, mobile devices must monitor the level of privacy provided by their context. To do so, we rely on well-known privacy metrics that capture the level of privacy of mobile nodes.

For any message sent, the *k-anonymity* metric [48] guarantees that k nodes might be the originators of the message. In wireless networks, nodes are k-anonymous if they have $k-1$ neighbors. Yet, in wireless networks, an adversary using directional antennas can still use the wireless signal power to find the originator of a message [11]. In this work, we model the strength of the adversary with a *confusion distance* that is the maximum distance in meters between two devices, where an attacker is not able to distinguish who among the two is the originator of the transmission. Hence, nodes must estimate how distant they are from their neighbors. In its simplest form, distance can be estimated from the received signal strength. Note that there are more elaborate schemes that allow to bound the distance between mobile users [51]. Thus, the level of privacy brought by k-anonymity not only depends on the number of nodes k, but also on the ability of the adversary to distinguish nodes from their neighbors.

The adversary may also have statistical information about the originator of a message. To take this into account, several researchers [21, 46] proposed to compute an *effective anonymity set* that measures the level of uncertainty of the adversary in guessing the originator of a message (i.e., the *entropy* of the anonymity set).

User location privacy also depends on the traceability of nodes. Even if interactions between users are anonymous, the traceability of nodes' locations over time enables an adversary to obtain mobile users real identities [12, 32, 40]. There are various metrics, as classified by Shokri *et al.* in [47], for measuring the success of the adversary in tracking nodes locations. *Clustering-based* metrics measure the error of the adversary in assigning events from the set of all possible events to users [22, 30]. *Traceability-based* metrics measure the extent to which a user can be tracked with high certainty by the adversary [31, 33]. *Distortion-based* metrics measure the distortion between the actual trajectory of users and the trajectory guessed by the adversary [47].

In practice, metrics that measure the traceability usually require a priori knowledge of mobility patterns of the mobile users. This information is hard to estimate. In addition, several metrics do not provide a real-time measure of privacy and rely instead on data collected over a period of time. For these reasons, it is difficult to use these metrics in our work.

We consider simple metrics that can help estimate the level of privacy of the nodes. More specifically, we consider the k-anonymity metric that measures the neighborhood density, i.e., the number of nodes in proximity. Then, we model the strength of the adversary by considering various confusion distances. In the future, we intend to test our system with more elaborate metrics.

5. SYSTEM ARCHITECTURE AND DESIGN

In this section, we introduce a privacy-triggered communication application for pervasive social networks. Figure 2 shows the architecture of our application. The application provides users with a Messaging Service for sharing data with other users. As mentioned earlier, the Messaging Service is just one way of generating and sharing local contextual data in pervasive social networks. Other types of information exchange may share data from the various device sensors (e.g., camera, GPS, etc.). The application assumes the existence of a pervasive social networking platform such as the Nokia awareness platform [8]. The awareness platform provides the basic user management and communication primitives such as joining, leaving and revoking users, sending messages, receiving messages and forwarding and routing messages to the destination. In addition to user interfaces for generating data, the Messaging Service also provides tools for visualizing privacy and data

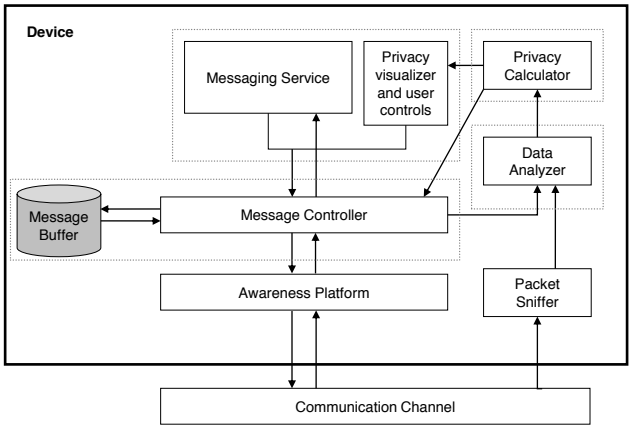


Figure 2: Overall system architecture.

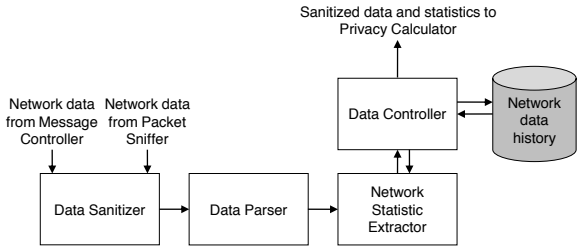


Figure 3: Data analyzer.

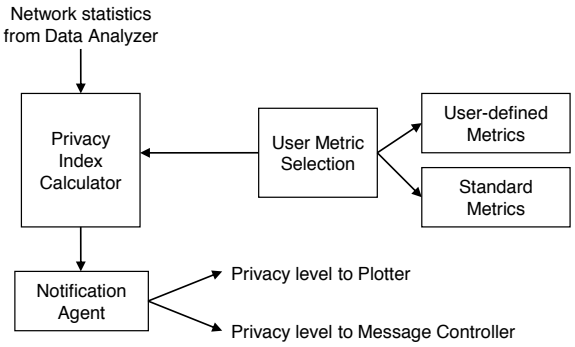


Figure 4: Privacy calculator.

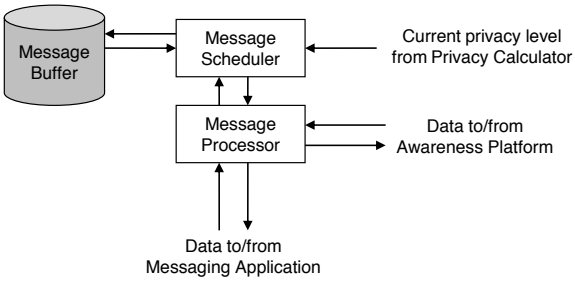


Figure 5: Message controller.

regulation based on privacy.

The Messaging Service passes the data to the Message Controller. The Message Controller regulates the transmission of data to the network depending on the current privacy level and the privacy level required for the data. The current user privacy level is calculated by the Privacy Calculator. The Privacy Calculator computes the current level of user privacy based on the network statistics provided by the Data Analyzer and the privacy metric chosen by the user. The Data Analyzer computes important network statistics for use in the privacy level computation from the network data provided by the Awareness platform or optionally by a third-party packet sniffing tool. We now describe in more detail each of the four main components.

The Data Analyzer component shown in Figure 3 provides a Data Sanitizer that takes the raw network input from the awareness platform and/or the Packet Sniffer and eliminates unwanted and duplicate information. The Data Sanitizer passes the sanitized data to the Data Parser that captures information useful from the point-of-view of privacy computation. The Network Statistic Extractor uses this data together with previously archived network data to generate and update useful network statistics. This process of data aggregation and statistic generation by the Network Statistic Extractor is also useful in reducing the amount of network data required to be stored locally on the devices. The Data Controller service controls the flow of these network statistics to the Privacy Calculator.

The Privacy Calculator shown in Figure 4 provides a metric selection service that allows users to select the specific privacy metric to be used during privacy computation. The Privacy Index Calculator computes the privacy value of the user, at regular time intervals, from the network statistics and based on the user-selected metric. This privacy level is passed to a Notification Agent that issues notifications to the Plotter service for visualization purposes and to the Message Scheduler of the Message Controller component.

The Message Controller component, as shown in Figure 5, is responsible for controlling the messages in and out of the application. It contains a Message Scheduler that takes messages together with the required privacy level chosen by the user and decides if the message is ready to be sent to the awareness platform. If the current privacy level is greater than or equal to the desired privacy level for the message, then the message is instantly scheduled for delivery to the awareness platform. If the current privacy level is smaller than the desired privacy level then the message is stored in the local message buffer for delivery at a later point in time. The Message Scheduler receives regular interrupts

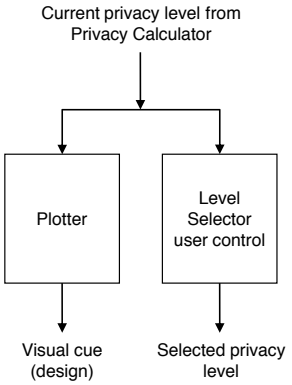


Figure 6: User privacy controls.

Figure 7: Nokia N810 running the privacy-triggered communication application.

from the Notification Agent of the Privacy Calculator with the current privacy level. Everytime an interrupt is received, the Message Scheduler checks the Message Buffer for messages that need to be transmitted at (or below) the current privacy level and schedules them for delivery to the awareness platform. The Message Scheduler uses a Message Processor to control the messages to and from the Messaging tool and the awareness platform.

The privacy visualizer controls, as shown in Figure 6, are a part of the user interface along with the Messaging Service. It provides a Plotter service that takes the privacy value provided by the Privacy Calculator and plots a visual cue of the privacy for the user. This visual cue is indicative of the current level of user's privacy. The visualizer control also provides a Level Selector control that allows users to select the desired level of privacy, relative to the level indicated by the Plotter, for particular messages. This selection indicates the privacy level at which that message is expected to be sent out on the awareness network. The message is then regulated accordingly by the Message Controller.

Finally, the privacy-triggered communication application can optionally employ a third-party Packet Sniffer such as Wireshark [3] or tcpdump. The main use of the Packet Sniffer is to collect network data that cannot be obtained directly from the awareness platform. This can help in better estimating users privacy. For hardware specific design constraints (such as simultaneous operation of the awareness platform and the sniffer on the same wireless interface), we do not employ a Packet Sniffer for the data collection operations. Thus in the current implementation, we only use the data provided by platform. Alternatively, in order to avoid using third-party sniffing tools, privacy computation specific network data collection procedures can also be integrated within the awareness platform itself. In summary, this is just a system specific constraint, which does not affect the overall application architecture.

6. SYSTEM IMPLEMENTATION

The privacy-triggered communication application is developed and tested on a testbed of Nokia N810 devices as shown in Figure 7. In the following sections, we give an overview of the implementation, including the details about the hardware and software platforms and the various application components[2].

6.1 Hardware and Software Platforms

We develop the proposed privacy-triggered communication application on top of the Nokia awareness platform [8] using the python programming language[3] (version 2.6.4) in the Linux (Ubuntu 9.04) environment. A research prototype of the awareness platform, called AwareNet[4], was developed for the Nokia N810 mobile devices by Nokia Research Center, Helsinki. AwareNet is a power-efficient, always-on, and fully distributed messaging system for mobile devices such as Nokia N810s. AwareNet uses peer-to-peer wireless connections between devices (in a connectionless fashion) to carry very small-sized local contextual information between clients without human intervention. AwareNet implements an efficient query-reply mechanism, with a smart flooding algorithm to disseminate data queries and a smart routing algorithm to route back replies. AwareNet also provides other network, link and physical layer functionalities to make the entire process very power efficient. AwareNet is written in C on the Maemo platform [6]. Maemo is an open-source Linux-based operating system used in the Nokia N810 devices. In summary, AwareNet provides a

[2]A video illustrating our implementation is available at: http://icapeople.epfl.ch/freudiger/videos/PTNVideo.mp4. username: submission, password: mobisys10

[3]PyCairo [4] and PyGtk [5] libraries were used for the GUI.

[4]AwareNet is a research platform and not currently available commercially.

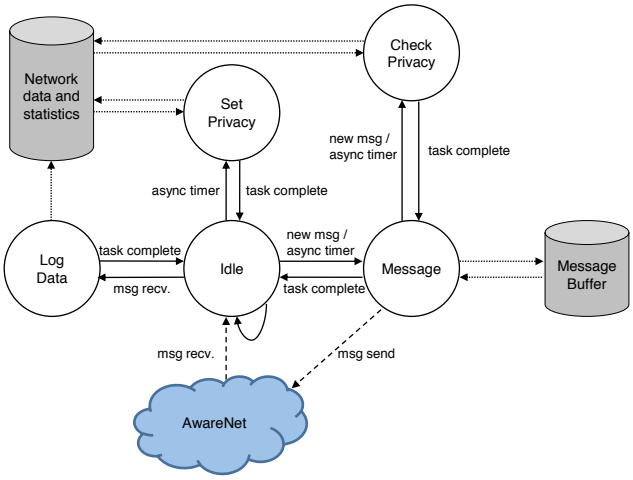


Figure 8: State diagram of the privacy-triggered communication application.

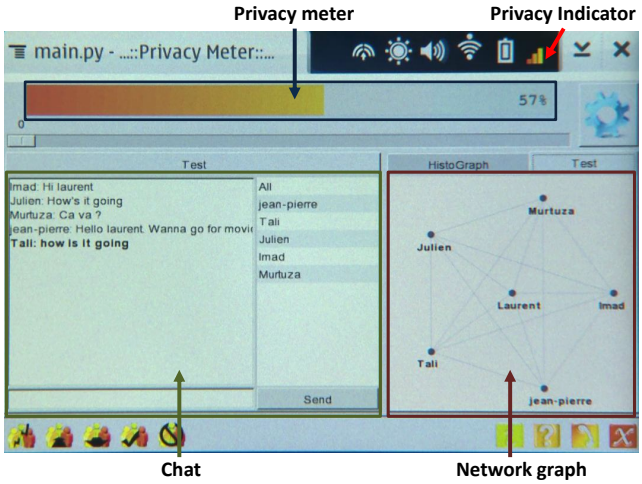


Figure 9: Privacy-triggered communication application.

platform for implementing the notion of pervasive social networks introduced earlier.

6.2 Application

Let us first describe the basic operation of our privacy-triggered application by means of state diagram as shown in Figure 8. By default, the application is in the *Idle* state. When a new message is received from the AwareNet platform, the application transitions to the *Log Data* state and logs the received message to the local storage. This storage maintains a short history of the network data and is used to compute network statistics and the privacy level of the user. Once completed, the application returns to the Idle state and waits for a new message either received from the AwareNet platform or generated by the user. When the user generates a new message, the application transitions to the *Message* state. In the Message state, the current privacy level is checked. If the privacy level required in the message is less than or equal to the current privacy level, as determined in the *Check Privacy* state, the message is sent to the AwareNet platform. Otherwise the message is stored locally in the *Message Buffer*. While in the Idle state, the application also transitions regularly to the Message state to check if there are any messages in the Message Buffer that now satisfy the privacy requirement. If there are such messages, then they are sent to the AwareNet platform. Similarly, while in the Idle state, the application regularly transitions to the *Set Privacy* state to update the network statistics and the current user privacy level.

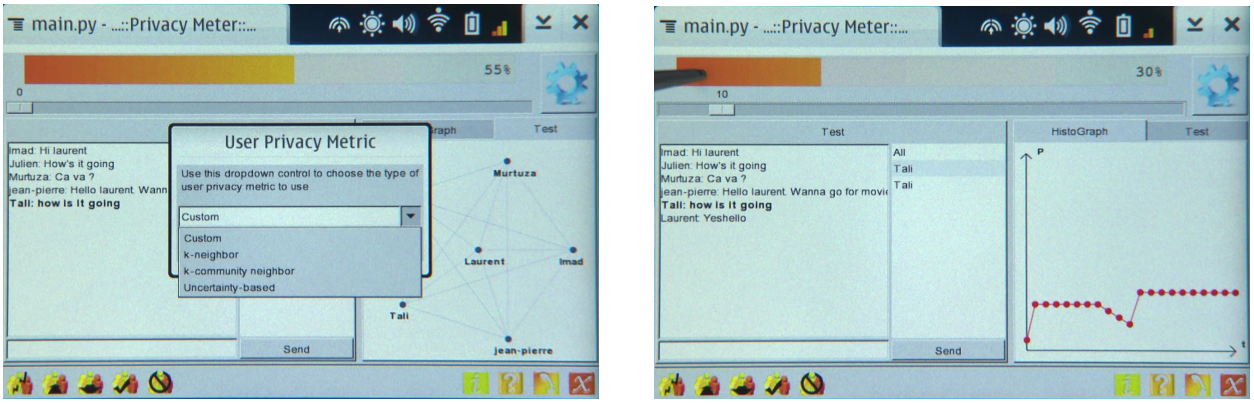
The actual prototype application and its user interface is illustrated in Figures 9 and 10. We can see from Figure 9 that the application provides a messaging or chat tool, similar to other popular IM tools, to enable users to send and receive messages from nearby users over the AwareNet platform. Note that pseudonyms are not changed while users are chatting. Let us discuss the privacy specific components in the user interface of the application.

6.2.1 *Privacy Visualization*

In Figure 9, we can see that the privacy meter, located above the chat control, displays the current level of user privacy based on the privacy metric chosen by the user. A privacy indicator (similar to the wireless signal strength indicator) is also provided in the task bar on the top right of the device display. This indicator can be used by the user to view the privacy level in the case the application is minimized or inactive. The network graph is located on the right-hand side of the chat control. Users can use the network graph to view other members in the vicinity of their device. The relative distances to these neighbors is visible from the network graph. The application also provides a privacy history graph in another tab on the right-hand side of the chat control. The history graph, as shown in Figure 10 (b), displays the evolution of user's privacy over the last 60 seconds.

6.2.2 *Privacy Metric Selection*

Users can choose the metric for measuring privacy by using the privacy metric selection control, as shown in Figure 10 (a). As discussed before, currently we have only implemented the well-known k-anonymity metric that measures privacy based on the neighborhood of users and the confusion distance to these neighbors. Implementation of other metrics is an ongoing work. Privacy metrics implemented in this application are restricted by the amount of information provided by the



(a) (b)

Figure 10: Privacy-triggering controls in the application (a) Privacy metric selection control (b) Privacy visualization, set privacy threshold and privacy history visualization controls.

platform.

6.2.3 *Communication Controller*

From Figure 10 (b), we can see that below the privacy meter is located a slider control that can be used for triggering communications based on user privacy. The user can select the position of the slider, with respect to the current privacy level shown in the privacy meter above the slider, to indicate the privacy level at which he wants to exchange messages. After choosing and fixing the position of the slider, all messages to be sent thereafter are associated with the privacy level indicated by the slider. These messages will be sent on the network by the application only when the desired privacy level is reached. The user can use this control to regulate his communications based on his current privacy level, for example, select a low privacy level for sending general data and a high privacy level for sending much more intimate information.

It is obvious that privacy-triggered communications will affect the overall quality-of-service (QoS) of users in the network, i.e., the communication delay. Due to the limitations (in terms of size) of the current testbed, we perform network simulations to observe the effect of privacy-triggered communications on the QoS in large mobile networks. The results of these simulations are discussed in the following section.

7. NETWORK PERFORMANCE

In this section, we perform simulation experiments using the ns-2 network simulator [7] in order to analyze the impact of privacy-triggering on user communications in large-sized networks. Results from these simulations provide insights on how simple privacy constraints on user communications affect the QoS in pervasive social networks, and how this QoS evolves with changes in user mobility and the underlying mobility area. These simulations are useful in evaluating the critical network parameters required to maintain a reasonable QoS of user communications while providing the desired level of user privacy. In the following section, we first describe our precise simulation setup. After that, we discuss the outcome of the conducted simulation experiments and analyze the impact of privacy-triggered communications.

7.1 Simulation Setup

The simulation setup consists of wireless mobile devices moving over a two-dimensional (rectangular), unobstructed terrain. All devices in the simulation are similar and the initial positions of these devices are uniformly chosen over the two dimensional area. The devices communicate in a peer-to-peer fashion using a wireless radio similar to a 914MHz Lucent WaveLAN DSSS. The physical, link and network layer properties of these devices are summarized in Table 1. As

Table 1: Properties of Wireless Devices

Parameters	Value
Antenna	Omnidirectional
Antenna Gain	1.0
Antenna Height	1.5m
Radio frequency	914MHz
Radio propagation	TwoRayGround
Radio range (default)	250m
MAC	802.11
Routing protocol	AODV (RFC 3561)

discussed earlier in Section 3, wireless devices communicate with each other by sending small-sized packets. The packet size that we consider in this simulation has a constant length of 100 bytes.

Each device sends two types of packets, namely beacon packets and data packets, at pre-defined regular

intervals. Beacon packets are used for neighbor discovery, hence beacon packet intervals are kept small, for example, 1 second. The actual data that devices want to exchange are sent as data packets. Data packets are sent less frequently as compared to the beacon packets. For the sake of simplicity, we assume that each device has a data packet to be sent every 15 seconds. The QoS measurements in these simulations are done only with respect to the data packets. All packets are sent as broadcasts; beacon packets are only broadcast up to 1 hop, whereas data packets are broadcast up to m hops. For the current simulations, we assume that m is a constant for all data packets and its value is 3. Each device maintains an application layer buffer to store data packets that cannot be immediately sent on the network because of the required privacy level. In these simulations, we assume a large local buffer of 50 KB, which is enough to hold a maximum of 500 data packets. This is a reasonable assumption considering the excellent data storage capabilities of current wireless mobile phones.

The simulations are carried out for two different types of random mobility models in which the entire mobility is divided into a number of short trips. The first mobility model that we use in the current simulations is the classical *Random Waypoint model with pauses (RW)* in which the nodes move (in each trip) on randomly chosen straight line segments on the two-dimensional area at uniform speeds. At the end of each trip along the line segment, the node pauses for some time. After the pause, it chooses uniformly at random the next endpoint from the two dimensional area and a speed below some maximum speed and moves along that line segment. In our simulations, we use the above random waypoint model with parameters as shown in Table 2.

Table 2: Parameters for RW mobility model

Parameters	Value
Simulation area	500m x 500m
Average node speed	5 m/sec
Average pause time	10 secs

We also perform simulations using a more realistic model, called the *Random Waypoint on a City selection (RWC)*. In this model, the rectangular area consists of a fixed set of line segments (signifying roads) and each is assigned a maximum speed. Nodes can only move along these line segments. At the end of each trip, a node selects uniformly at random a destination point on the intersection of one of these segments and moves along the shortest path from its original point to the destination point. A node can optionally pause between trips. Le Boudec and Vojnović [14] extensively study the stability and time-invariability properties of such mobility models. They provide algorithms and tools to initialize node mobility state so that the distribution of the node state is time-stationary throughout a simulation. We use their tool to generate the required random mobility patterns. For the RWC mobility model, we use node mobility patterns on a random U.S. city map, as shown in Figure 11. Other simulation parameters for this mobility model are outlined in Table 3.

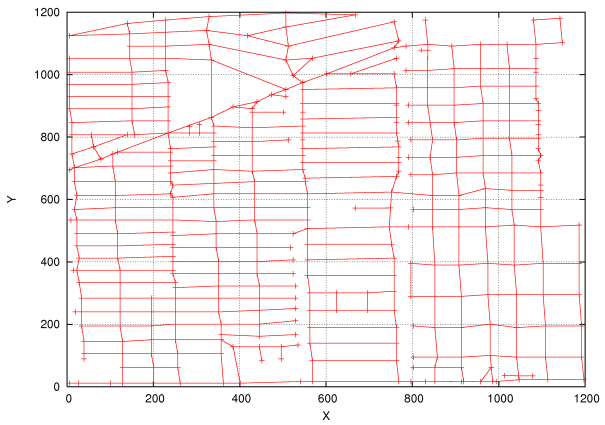


Figure 11: West University Place, Texas, USA city map.

Table 3: Parameters for RWC mobility model

Parameters	Value
Simulation area	1200m x 1200m
Number of roads	1188
Number of intersections	383
Average speed on all roads	5 m/sec
Average pause time	10 secs

As discussed earlier in Section 4, each data packet is associated with a required privacy level that symbolizes the privacy level that needs to be attained by the device in order for the packet to be sent on the network. Here, we use a simple k-anonymity metric to determine the privacy level of nodes. In this metric, the privacy level of a node in a given time interval, for a given confusing distance d, is the *total number of neighbors*[5] within distance d of the node. In order to observe the system response to adversaries with varying capabilities, we repeat the simulations for various values of the confusion distance; low confusion distances for strong adversaries and high confusion distances for very weak adversaries. In the current simulations, we choose $k = 6$ for all the nodes. In other words, for each data packet and for a given confusing distance d, each device needs at least 5 neighbors within a distance of d meters (in a fixed time interval) for the data packet to be sent out on the network.

[5]Device B is a neighbor of device A if A can directly hear packets sent by B (in 1 hop)

7.2 Simulation Results

In this section, we outline the two different simulation scenarios for privacy-triggered communications and discuss the results of these simulations. In the first scenario, all the wireless devices or nodes follow the classical random waypoint model (with pauses) as outlined in the previous section. In this scenario, we run the simulations for an increasing number of nodes from 20 up to 100 nodes in steps of 20 nodes and for increasing values of confusion distance from $50m$ up to $250m$ in steps of $50m$. Each simulation is run for a total duration of 5000 secs. All other parameters are as discussed in the earlier section. Next, we repeat the simulations using the random waypoint on a city selection mobility model instead of the classical random waypoint model. All other simulation parameters remain unchanged. The results of these simulations are discussed in the following two sections.

7.2.1 Simulation Results for RW Mobility Model

The results of the simulation runs using the classical random waypoint model (with pauses) are outlined in Figure 12. For each simulation run, we record the following three statistics that measure the QoS provided by the system. The first statistic is the average delay before data packets are transmitted on the network. This delay is the difference (in seconds) between the time the data packet is originally scheduled and the time it is actually broadcast, i.e., the time spent in the local buffer. The average is over all (buffered) data packets and over all the nodes in the network. The delay before transmission is plotted in Figure 12(a). We also observe the average displacement (Figure 12(b)) before packet transmission on the network, which is the Euclidean distance between the location that the data packet is originally scheduled at and the location it is actually broadcast on the network. Both average delay and average displacement are important QoS properties because of the real-time and local (context-dependent) nature of the data that is shared in pervasive social networks. The utility of the data in pervasive social networks is directly related to the delay and displacement associated with it.

From Figure 12(a) and (b), we can observe that, for any value of the confusion distance, the average delay and displacement increases as the total number of nodes in the network decreases. This is intuitive because as the current privacy requirement for the data packets is k-anonymity (with $k = 6$), fewer nodes in the network would lead to sparser neighborhoods (and the k-anonymity requirement not being satisfied) and eventually more waiting time or delay for the data packets. As nodes are mobile, more wait time translates directly to longer displacements. Another trend that we can observe in these figures is that average delay and displacement increases with a decrease in the confusion distance. This is also easy to explain as a decrease in the confusion distance makes it difficult to satisfy the k-anonymity requirement for the data packets resulting in longer delays and displacements. One interesting observation is that this increase is exponential when the nodes follow a classical random waypoint mobility model. For example, the delay is 0.1 seconds for a confusion distance of $250m$, which increases to 10 seconds for a confusion distance of $50m$ when there are 100 nodes in the network. As in this model, there are at least 250000 end-points and nodes choose random end-points (meeting points), when the confusion distance decreases it becomes extremely difficult to find k-neighbors for privacy-triggering.

Another QoS related statistic that we observe is the average device data buffer occupancy or the average number of data packets in the buffer at any time. The plot for buffer occupancy for the simulations using the classical random waypoint model is shown in Figure 12(c). From the figure, we can see that the buffer occupancy increases with the decrease in the number of the nodes in the network or in the confusion distance. This is also intuitive because a decrease in the number of nodes or the confusion distance makes it harder for nodes to satisfy the k-anonymity requirement, as discussed earlier. Similarly to the previous two plots, the buffer occupancy increases exponentially with the decrease in the confusion distance. The plot of buffer occupancy is also consistent with the previous two results. Greater buffer occupancy in nodes should lead to larger data packet delays (and displacements), which is also evident from the above plots.

7.2.2 Simulation Results for RWC Mobility Model

The results of the simulation runs using the random waypoint model on a city map are outlined in Figures 12 (d),(e) and (f). Similar to the previous results, for each simulation run the average delay before transmission (Figure 12(d)), the average displacement before transmission (Figure 12(e)) and the average buffer occupancy (Figure 12(f)) are observed and plotted.

We can see from Figures 12 (d),(e) and (f) that the plots of the simulation results for the random waypoint model on a city map follow a very similar trend to those for the classical random waypoint model as discussed in Figures 12 (a),(b) and (c). Similar to the earlier results, all the QoS properties, namely, average delay, average displacement and average buffer occupancy, increase as the number of nodes in the network decreases. Similarly, the average delay, average displacement and average buffer occupancy also increase with the decrease in the confusion distance. One main difference with the earlier results is that the average delay, average displacement and average buffer occupancy in this case are

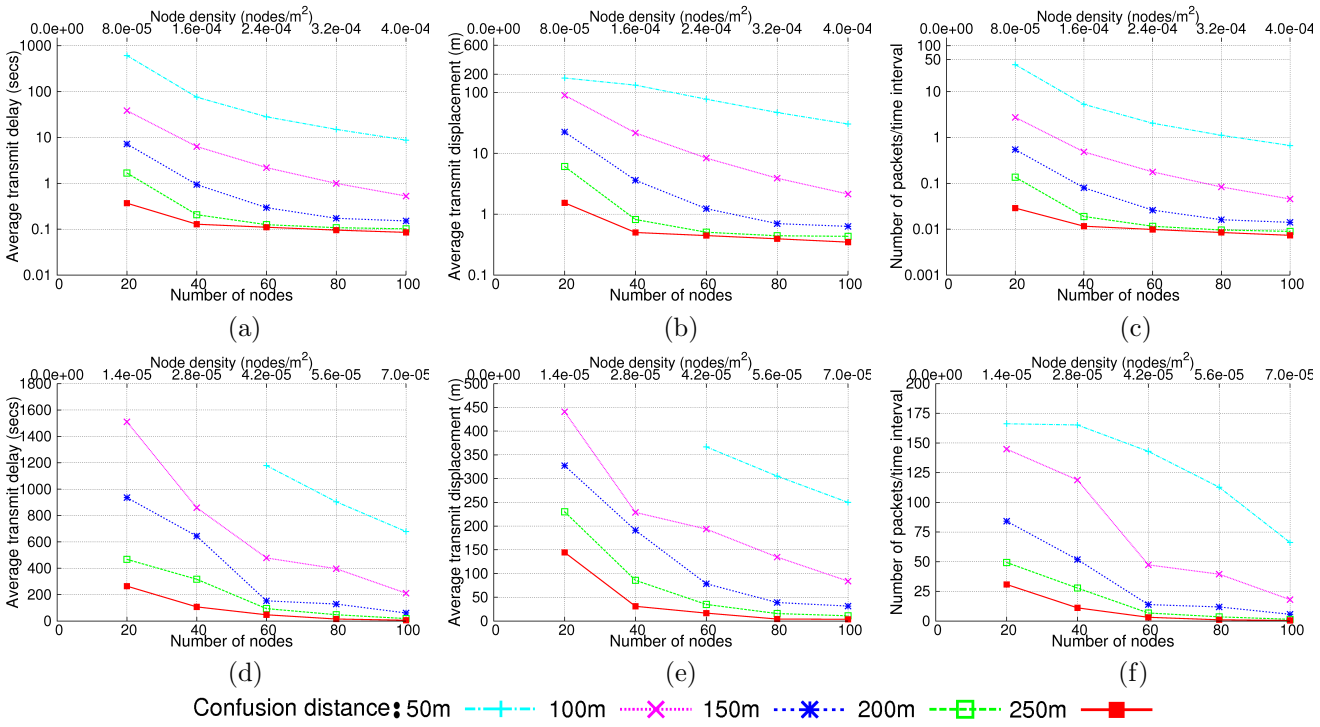


Figure 12: Simulation results: (a), (b) and (c) Classical random waypoint model with pauses; (d), (e) and (f) Random waypoint on a city selection.

always higher as compared to the corresponding plots for the classical random waypoint model. Moreover the increase in the values of these properties with the confusion distance is much slower as compared to the earlier case. Another important observation that we make from these results is that for the low values of confusion distance (for example, $50m$) and the total number of nodes (for example, 20 and 40), there is no service. In other words, none of the nodes are able to send out a single data packet on the network.

Up to this point, we have seen that the system behaves in a predictable fashion in terms of the QoS received by the users when their communications are privacy triggered. This clearly reflects the trade-off that exists between privacy and QoS. Now, let us further analyze the above simulation results to study the effect that node mobility has on the privacy-triggered communication mechanism and how moving along restricted paths (and a limited set of encounter points) performs against moving randomly, from the point of view of privacy-triggered communications.

7.3 Discussion

Let us compare the simulation results for the two different mobility models. From Figure 12, we observe that on average the delay for the RWC model is higher than the RW model, which is not surprising, given the restricted motion of the nodes and the fewer meeting points. For the classical RW model, assuming only integer coordinates, there are at most 250,000 possible meeting points as compared to 383 for the RWC model. Even on scaling down the area used for the RWC model, we get at most 2300 meeting points for the RWC case. For the RW model with a node density of 8e-05 (20 nodes), the delays are 0.36, 1.67, 7, 38 and 610, all in seconds, when the confusion distance decreases from $250m$ to $50m$. For the RWC model with approximately the same density of $7e{-}05$ (100 nodes), the delays are 6, 18, 61, 210 and 678, all in seconds, when the confusion distance decreases from $250m$ to $50m$. For lower values of the confusion distance, the difference in the delay between the two models is small. These results indicate that privacy-triggered communications perform better for the RW model as compared to the RWC model.

Let us see how these results translate to real-life situations. Large public gatherings such as markets, fairs, sporting events, etc., are good situations for privacy-triggered communication whereas walking or driving on less crowded streets may not be so favorable. Large public gatherings generally correspond to Points of Interest (POI), e.g., school, workplace, bus stops, etc., and people move towards these POIs. Intuitively, privacy-triggered communication will work better as people approach a POI or are within a POI. For example, the application does not communicate at all as a person is walking towards a bus stop, but starts communicating

as he reaches the bus stop or boards a bus.

One observation is that as users approach a POI (or within a POI), the communications become bursty. This leads to high congestion near and in the POIs. This also leads to sharing lots of personal data at once. This can be observed from the buffer occupancy results shown in figures 12 (c) and (f). The average number of packets in the buffer per time interval for the RW model is far less compared to the RWC model. There are more encounters and thus regular communications in a RW model than the RWC model, which leads to a much more uniform traffic pattern. In order to overcome this problem, one solution is to gradually start and resume communications inside a POI where privacy is good.

Finally, our results show that privacy does not come for free and the price to pay is to reduce the QoS. By providing the necessary tools and services for privacy regulation in such networks, we provide a better understanding of this tradeoff. In addition, our solution enables users to make their own choice in this regard. Those who prefer high privacy can have it, but at the cost of a lower QoS.

8. CONCLUSION

In this paper, we introduced the concept of triggering communications based on privacy in peer-to-peer wireless networks. First, we derived the requirements to build privacy-triggered applications and showed that it is not tied to any privacy metric. Then, we designed and implemented the first operational prototype of this concept on the Nokia N810 devices. By means of simulations, we showed how privacy-triggered communications affect the overall QoS provided by the system. In particular, we showed that for the k-anonymity metric the usability of privacy-triggered communication and the resulting QoS depend not only on users privacy requirements, but also on their mobility patterns. This work is a first step towards providing privacy tools that consider the wireless context to control user privacy. A particularly interesting result of such approach is the ability to regulate communications based on the user environment.

For future work, we would like to test our prototype with other privacy metrics as different classes of people may have different privacy requirements. Moreover, we would like to extend our current prototype to integrate some of the regulation tasks to the underlying communication platform, and thus avoid user intervention. Finally, we would also like to integrate an "always-on" monitoring module with our current prototype to continuously measure privacy, even when the main application is turned-off.

9. ACKNOWLEDGMENTS

We would like to thank Laurent Bindschaedler and Tali Gutman for implementing the prototype of the privacy-triggered communication application. We also thank Marcin Poturalski and Reza Shokri for providing useful feedbacks. Finally, we are grateful to Nokia Research Center, Lausanne for funding this project.

10. REFERENCES

[1] http://en.wikipedia.org/wiki/Lovegetty.
[2] http://en.wikipedia.org/wiki/Bluedating.
[3] http://www.wireshark.org/.
[4] http://www.cairographics.org/pycairo/.
[5] http://www.pygtk.org/.
[6] http://maemo.org/intro/platform/.
[7] http://www.isi.edu/nsnam/ns/.
[8] A. Ahtiainen, K. Kalliojarvi, M. Kasslin, K. Leppanen, A. Richter, P. Ruuska, and C. Wijting. Awareness networking in wireless environments: Means of exchanging information. *IEEE Vehicular Technology Magazine*, September 2009.
[9] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi. Routing through the mist: Privacy preserving communication in ubiquitous computing environments. In *ICDCS*, page 74, 2002.
[10] I. Altman. *The Environment and Social Behavior: Privacy, Personal Space, Territory and Crowding.* Brooks/Col Pub. Co., 1975.
[11] K. Bauer, D. McCoy, B. Greenstein, D. Grunwald, and D. Sicker. Physical layer attacks on unlinkability in wireless lans. In *PETS*, August 2009.
[12] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *Pervasive Computing, IEEE*, 2(1):46–55, 2003.
[13] A. R. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *PerSec*, March 2004.
[14] J.-Y. Le Boudec and M. Vojnovic. The random trip model: stability, stationary regime, and perfect simulation. *IEEE/ACM Trans. Netw.*, 14(6):1153–1166, 2006.
[15] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *MobiCom*, pages 116–127, 2008.
[16] S. Buchegger, D. Schiöberg, L-H Vu, and A. Datta. Peerson: P2P social networking: early experiences and insights. In *EuroSys Workshop on Social Network Systems (SNS)*, pages 46–52, 2009.
[17] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, and M. Shin. Anonysense: Privacy-aware people-centric sensing. In *MobiSys*, 2008.
[18] L. P. Cox, A. Dalton, and V. Marupadi. Smokescreen: flexible privacy controls for presence-sharing. In *MobiSys*, pages 233–245,

2007.

[19] B. Danev and S. Capkun. Transient-based identification of wireless sensor nodes. In *IPSN*, 2009.

[20] A. Dey, D. Salber, and G. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human Computer Interaction*, 16:97–166, 2001.

[21] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *PET*, 2002.

[22] L. Fischer, S. Katzenbeisser, and C. Eckert. Measuring unlinkability revisited. In *WPES*, 2008.

[23] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX*, 2006.

[24] S. Gaonkar, J. Li, R. R. Choudhury, L. P. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In *MobiSys*, pages 174–186, 2008.

[25] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, 2003.

[26] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis. *Mob. Netw. Appl.*, 10(3):315–325, 2005.

[27] M. Gruteser and B. Hoh. On the anonymity of periodic location samples. In *Security in Pervasive Computing*, pages 179–192, 2005.

[28] J. Hall, M. Barbeau, and E. Kranakis. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. In *CIIT*, 2004.

[29] B. Hayes. Cloud computing. *Commun. ACM*, 51(7):9–11, 2008.

[30] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *SECURECOMM*, 2005.

[31] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *MobiSys*, 2008.

[32] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5(4):38–46, 2006.

[33] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *CCS*, 2007.

[34] J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *MobiSys*, pages 177–189, 2004.

[35] T. Jiang, H. J. Wang, and Y.-C. Hu. Preserving location privacy in wireless LANs. In *MobiSys*, 2007.

[36] I. Kang and R. Poovendran. A comparison of power-efficient broadcast routing algorithms. In *GLOBECOM*, pages 387–392, 2003.

[37] M. Khiabani. Metro-sexual. http://bit.ly/theranMetroSexual, 2009.

[38] T. Kohno, A. Broido, and K.C. Claffy. Remote physical device fingerprinting. *TDSC*, 2, 2005.

[39] P. Korpipaa and J. Mantyjarvi. An ontology for mobile device sensor-based context awareness. In *Context*, pages 451–458, 2003.

[40] J. Krumm. Inference attacks on location tracks. In *Pervasive*, 2007.

[41] K. Liu and E. Terzi. A framework for computing the privacy scores of users in online social networks. In *ICDM*, December 2009.

[42] L. Palen and P. Dourish. Unpacking privacy for a networked world. In *CHI*, pages 129–136, 2003.

[43] E. Paulos and E. Goodman. The familiar stranger: anxiety, comfort, and play in public places. In *CHI*, pages 223–230, 2004.

[44] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. Contextphone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 5:51–59, 2005.

[45] B. Rasmussen and S. Capkun. Implications of radio fingerprinting on the security of sensor networks. In *SECURECOMM*, 2007.

[46] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *PET*, 2002.

[47] R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux. A Distortion-based Metric for Location Privacy. In *WPES*, 2009.

[48] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[49] S. Trudeau, S. Sinclair, and S. W. Smith. The effects of introspection on creating privacy policy. In *WPES*, 2009.

[50] O. Ureten and N. Serinken. Wireless security through RF fingerprinting. *Canadian J. Elect. Comput. Eng.*, 32, 2007.

[51] S. Vasudevan, J. Kurose, and D. Towsley. On neighbor discovery in wireless networks with directional antennas. In *Infocom*, 2005.

[52] H. Wang, B. Sheng, and Q. Li. Privacy-aware routing in sensor networks. *Comput. Netw.*, 53(9):1512–1529, 2009.

[53] F.-L. Wong and F. Stajano. Location privacy in Bluetooth. In *ESAS*, pages 176–188, 2005.