Computer Networks 138 (2018) 136–148

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

MSP: Providing location privacy in WLAN networks with a MAC Swapping Protocol

O. Arana*, F. Garcia, J. Gomez, V. Rangel

Department of Telecommunications Engineering, National Autonomous University of Mexico, Mexico City 04510, Mexico

ARTICLE INFO

Article history:
Received 30 June 2017
Revised 15 December 2017
Accepted 27 March 2018
Available online 28 March 2018

Keywords:
Location privacy
MAC exchange
Anonymity
Location estimation

ABSTRACT

Location privacy has been widely studied in the context of location-based services (LBS). However, a far more serious location privacy threat arises when malicious eavesdroppers listen to wireless transmissions from an unsuspecting mobile user in order to pinpoint his location and figure out his identity. This new scenario is known as *location estimation* (LE). While there are several strategies to mitigate the threats posed by LBS scenarios, only a few researchers deal with countermeasures for LE scenarios. This paper proposes MSP, a MAC Swapping Protocol that allows two mobile users to discreetly exchange their MAC addresses without malicious eavesdroppers being able to detect it. In this way, although potential eavesdroppers can still pinpoint the location of a transmitting node, they will get its identity wrong. Over time, MSP eliminates the eavesdroppers' ability to link the position and identity of a transmitting source. In contrast to related research, the identity exchange in MSP takes into account information from the mobile users' physical and MAC layers simultaneously, so an attack in one layer does not expose the identity exchange in the other layer. In order to provide location privacy, MSP uses two algorithms. The first algorithm works at the physical layer, allowing two mobile nodes to decide when and where to exchange their MAC addresses. The second algorithm uses virtual interfaces to guarantee that the identity exchange does not exhibit any abnormal behavior at the MAC layer. Test-bed and simulation experiments demonstrate that MSP is able to guarantee location privacy even with attackers eavesdropping at the physical and MAC layers simultaneously.

1. Introduction

Location-based services (LBS) are becoming more and more popular because of the exponential use of mobile devices. Nowadays, most of these devices are equipped with multiple embedded sensors (e.g., accelerometer, GPS, and so on), facilitating the development of a variety of applications based on location information. Interacting with LBS usually requires mobile users to provide their current location (typically acquired by GPS) to obtain information such as specific directions to reach a destination, or request services from a taxi company, for example. However, this type of transaction creates location privacy concerns for mobile users since their location could be used by LBS for other purposes not authorized by the user. In [1], the authors defined the location privacy problem as *"the ability to prevent other parties from learning one's current or past location."* To mitigate the location privacy problem, several techniques have been proposed in recent years in the context of LBS. Most of these techniques can be grouped into obfuscation and anonymity categories. Obfuscation techniques require mobile users to provide a non-accurate location to the LBS while still being able to receive meaningful information. Anonymity techniques, on the other hand, consider *"the dissociation of information about an individual, such as location, from that individual's actual identity"* [2]. Although both techniques have been widely used to mitigate the location privacy problem in LBS scenarios, they do not take into consideration the scenario in which a set of fixed nodes eavesdrops on a mobile node's wireless transmissions in order to pinpoint its location (e.g., using multilateration or trilateration techniques [3]) and figure out its identity. This scenario is known as location estimation (LE) [4], and represents a serious location privacy risk for mobile users since they are unaware that third parties are silently tracking them. This is in contrast to LBS scenarios in which at least mobile users are aware of the location privacy risk involved. Furthermore, an attacker in LE scenarios could gather the mobile user's location over time not only to predict future positions, but also to figure out the mobile user's movement patterns. Unfortunately, obfuscation and anonymity techniques de-

* Corresponding author.
E-mail addresses: arana.h.oscar@comunidad.unam.mx (O. Arana), fgarcia@fi-b.unam.mx (F. Garcia), javierg@fi-b.unam.mx (J. Gomez), victor@fi-b.unam.mx (V. Rangel).

https://doi.org/10.1016/j.comnet.2018.03.030
1389-1286/

veloped for LBS scenarios are not suitable for the new challenges posed by LE scenarios, since the user's location is acquired without any user intervention. Consequently, countermeasures to mitigate the risk of being tracked by third parties would therefore be completely different for the LBS and LE scenarios.

A well-known technique to mitigate location privacy risks in LE scenarios consists in letting the user vary the wireless interface's transmission power, also known as transmission power control (TPC). This technique confuses the attackers since they usually employ location algorithms that relate received signal strength indicator (RSSI) to distance, assuming the mobile user is transmitting with the default nominal power. As a result of using TPC, the attackers might estimate a distance different from the mobile node's actual distance. However, it has been shown that the effectiveness of this technique diminishes as the number of attackers eavesdropping on the mobile user's transmissions increases. Consequently, TPC cannot guarantee location privacy for mobile users, especially in densely deployed wireless networks.

Another technique used to mitigate location privacy risks in LE scenarios consists of frequently changing identity parameters, such as the MAC and/or IP addresses [5,6]. In this technique, a mobile user changes his own MAC/IP address or that of other users, making it harder for attackers to associate the location of a transmitting node to its true identity. However, information from other layers, in particular from the physical layer, might expose a MAC or IP address modification. This becomes possible because an attacker can relate the RSSI to the mobile users' actual location. Thus, the attacker may detect a change of MAC/IP address by comparing similarities between RSSI values before and after the change took place. Furthermore, when a MAC address changes, the mobile node's operating system must reboot its wireless interface, requiring the interface to re-associate with the access point (AP), which in turns resets the IEEE 802.11s' sequence number field back to zero. The presence of re-association control packets and the resetting of the number field provide the attackers with clues that a node changed its MAC address.

In contrast to previous proposals that did not consider an attacker having access to physical (e.g., RSSI parameter) and MAC layer information, this work clearly demonstrates the need to simultaneously consider the information from both layers in order to provide location privacy for mobile users in LE scenarios. In order to achieve this goal, this paper introduces MSP, an algorithm that allows mobile users to swap their MAC addresses in order to avoid detection in LE scenarios. In MSP, protecting location privacy for mobile users is divided into two steps. In the first step, we propose an algorithm named *Safe-zone* to solve the problem related to physical layer detection when two mobile users swap their MAC addresses. Safe-zone continuously monitors RSSI behavior to select not only the best candidate, but also the best location in which to perform a MAC address exchange between two users. As a second step, we propose an algorithm named virtual interfaces MAC address exchange (*VIME*) in order to solve the aforementioned problems related to MAC layer detection. VIME uses virtual interfaces to control the IEEE802.11's header fields before delivering packets to the wireless interface. VIME allows a mobile user to change his MAC address discreetly, without having to reboot the wireless interface or re-associate with an AP. Combining Safe-zone and VIME algorithms provides an integral solution to mobile users that prevents potential eavesdroppers from detecting a MAC address exchange. This combination dissociates a mobile user's identity from his actual location, causing attackers to unwittingly track the wrong user.

In MSP, this MAC exchange takes place between two users only, in contrast to other proposals in which a user changes his MAC address either alone or within a group. We argue that swapping MAC addresses between two mobile users is more likely to pass unnoticed than when a single mobile user performs the change by his own means. For instance, consider the case in which a single mobile user changes his MAC address from MAC *A* to MAC *B*. From the attacker's view, MAC *A* suddenly disappeared from the network while, at the same time, MAC *B* appeared. This behavior clearly informs an attacker that a user has changed his MAC address, thus nullifying the intended protection. On the other hand, hiding the user's identity within a group (mix-zone) presents several disadvantages. For example, discovering a mix-zone of $k-1$ mobile users willing to cooperate is challenging, specially if k increases. Moreover, users experience connectivity loss while they remain inside the mix-zone.

To sum up, previous proposals did not consider an attacker having access to information from both the physical and MAC layers. This issue raises serious location privacy concerns for mobile users since attackers can nullify the intended protection implemented in one layer by using information from the other layer. MSP, on the other hand, prevents attackers from discovering the MAC address exchange by considering information from both layers simultaneously. To achieve this goal, MSP uses two complementary algorithms. At the physical layer, the Safe-zone algorithm looks for the best place and time to perform a MAC address exchange between two mobile users. The second algorithm uses virtual interfaces to guarantee that the identity exchange does not exhibit any abnormal behavior at the MAC layer. The combination of both algorithms guarantees location privacy for mobile users even if attackers eavesdrop information from both layers simultaneously. Simulation and test-bed experiments showed that when users implemented Safe-zone, attackers could detect at most five percent of MAC address exchanges using the available physical layer information. On the other hand, VIME was able to deceive attackers eavesdropping information from the MAC layer without triggering any alarm. Overall, the results showed that MSP was able to protect mobile users' location privacy even when the attackers had access to information from both layers. While VIME adds about 100 μs to process each packet before transmission, the experiments showed no throughput degradation.

The rest of this paper is organized as follows: Section 2 presents an overview of the work related to location privacy in LBS and LE scenarios. Section 3 describes Safe-zone and VIME algorithms. Section 4 presents test-bed experiments under diverse conditions. Finally, Section 5 presents the conclusions.

2. Related work

This section reviews the most relevant work related to location privacy techniques for LBS and LE scenarios, as well as the most relevant MAC spoofing attacks.

The most studied scenario related to location privacy is by far the one in which a mobile user requests LBS for location-dependent information. The key concern with this transaction is that mobile users must provide their current location to LBS. This type of request raises serious location privacy concerns since the mobile users' location becomes readily available to potentially malicious LBS.

In recent years, various techniques have been proposed to provide location privacy while users interact with LBS. These techniques can be classified into obfuscation and anonymity techniques. Obfuscation techniques base their operation on blurring the mobile user's exact location before submitting a request to LBS [7–9]. Anonymity techniques, on the other hand, base their operation on dissociating mobile users' current location from their true identities [1,10]. However, both obfuscation and anonymity techniques proposed for LBS scenarios cannot fully provide location privacy since mobile users must provide their location with a minimum level of accuracy in order to receive meaningful information.

In contrast to LBS scenarios, where mobile users are aware of the risk involved, mobile users in LE scenarios are completely unaware that third parties may be tracking them. We argue that this scenario represents a far more serious location privacy risk for mobile users compared to LBS scenarios. Location attacks in LE scenarios basically involve a group of attackers that discreetly eavesdrop on the mobile users' wireless transmissions in order to pinpoint their location and establish their identity. Once a mobile user sends radio waves into the air, it is fairly simple for a set of listeners to pinpoint the mobile user's location using lateration or multilateration methods [11]. These methods usually measure one or more parameters from the mobile user's transmission such as *time of arrival* (ToA), *time difference of arrival* (TDoA), *angle of arrival* (AoA) and RSSI [12]. By far, RSSI is the most often used parameter in practice since it does not require specialized hardware and it is readily available in most commercial radios.

Similar to LBS scenarios, in LE scenarios obfuscation and anonymity techniques can mitigate the location privacy risks involved. However, their operation is fundamentally different. For instance, in [13] the authors proposed transmission power control (TPC), an obfuscation strategy in which a mobile user reduces his transmission power in a way that reduces the number of attackers listening to his transmission, thus reducing location accuracy. However, the effectiveness of this strategy strongly depends on the number and spatial distribution of the attackers listening to the mobile user's transmission. Another example of obfuscation is the use of beam-forming antennas [14] to concentrate the mobile user's signal in a specific direction only, thus decreasing the number of attackers detecting the presence of the mobile user. However, this technique requires a specialized and expensive hardware, not available on most commercial devices [15].

Anonymity techniques proposed for LE scenarios, on the other hand, are based on the idea that even if the attackers can establish the location of a wireless source, they cannot correlate such location to the source's true identity. To achieve this goal, users should periodically change identity parameters such as MAC and/or IP addresses. For instance, in [5] the authors proposed that mobile users use a different MAC address every time they associate with a new AP. In [6], the authors proposed a MAC address coordinator in order to hide the identities of various users. However, a disadvantage of changing ID parameters, either alone or within a group, is the fact that this process involves a temporary loss of connectivity with the AP [16,17]. Similarly, this sudden glitch in the connection and the following re-association provides attackers with a clue that something unusual has occurred, such as a MAC address exchange. There is in the literature a mature body of work known as MAC spoofing, intended to discover when a non-authorized user takes the MAC address of a valid user to obtain service [18]. Even if they were not intended to provide location privacy, MAC spoofing techniques are an important research area to consider, since they can be used by attackers to detect a mobile user trying to hide his identity by changing his MAC address with another user.

MAC spoofing methods are designed to identify abnormal behavior in physical or MAC layers. MAC spoofing techniques usually assume that the malicious node (i.e., the node stealing a valid MAC address) and the victim node (the valid owner of the MAC address) transmit their packets from different places at the same time. From the MAC spoofing detector's viewpoint (i.e., the attacker), the stealing of a MAC address creates sudden fluctuations of RSSI levels from one received packet to the next, thus allowing a detector to identify MAC spoofing [19]. In [20], for instance, a detector creates a node's profile by computing a histogram of RSSI measurements, and then the detector compares the node's histogram to discover any discrepancy. Authors in [21] proposed a MAC spoofing detector based on k-means algorithm. This technique divides n RSSI measurements associated to a MAC address into k sets (clusters), so that each cluster gathers similar RSSI measurements. As a result, during a spoofing event clusters without correlation are separated from each other, evidencing an attack. The authors in [18] use Fourier's analysis to convert time-varying RSSI values into a frequency domain. Since in a spoofing attack RSSI levels fluctuate drastically, it creates high frequency components in the frequency domain that clearly indicate a spoofing attack.

MAC spoofing detectors used at the MAC layer base their operation on sequence number analysis. The IEEE 802.11 standard requires the sequence number to increase its value monolithically for each newly created packet. This operation guarantees the correct reassemble of frames at the receiver side. According to the IEEE 802.11 standard [22], the sequence number field is 12 bits long, representing 4096 possible sequence numbers. Any anomaly in the behavior of the sequence number indicates a possible spoofing attack. For instance, the sequence number gap detector (SNG) [23] computes the difference between the sequence numbers of the ith frame with the ith $- 1$ frame. If the gap between two consecutive sequence numbers is greater than an established threshold, a MAC spoofing alert is raised. Sequence number rate analysis (SNRA) [18] bases its operation on the maximum number of frames that the wireless interface can send per second. For instance, the maximum number of frames per second using the IEEE 802.11b standard is 98,214; assuming a maximum transmission rate of 11 Mbps. In this way, if the number of frames transmitted exceeds the maximum number of frames the wireless interface can handle per second, an alarm will be raised.

Summarizing related proposals, although there are several methods addressing the location privacy problem in LBS and LE scenarios, none of these methods consider an integral solution comprising information obtained from both physical and MAC layers. In contrast, MSP considers a simultaneous attack on both layers so information from one layer does not expose an exchange in the other layer. MSP's operation does not require rebooting the mobile users' wireless interface as well as it keeps the integrity of the sequence number field intact before and after the MAC exchange takes place. Moreover, our proposal uses the safe-zone algorithm to hide the MAC exchange against MAC spoofing detectors that consider the physical layer information. Overall, MSP provides location privacy to mobile users, as it raises no alarms from the attackers' viewpoint. In this way, even if the attackers can locate a transmitting node, they will get its identity wrong, thus preserving the anonymity of mobile users.

3. MSP

This section introduces MSP, a strategy that allows two mobile users to swap their MAC addresses in a way that passes unnoticed to a group of attackers eavesdropping on the two mobile users' transmissions at the physical and MAC layers. To achieve this goal, MSP uses two algorithms named Safe-zone and VIME that operate at the physical and MAC layers, respectively.

It is important to note that in this paper we only consider attackers having access to physical and MAC layer information in order to establish the location and identity of a wireless source. While attackers might also exploit information from higher layers, such as information about visited destinations (i.e., IP addresses), and socket status, among others, we assert that considering attacks at all layers and their respective countermeasures cannot be covered in detail in a single piece of research. For this reason, in this paper both mobile users and attackers have been restricted to consider physical and MAC layers-related information only. Threats arising from exploiting information from higher layers will be dealt with in a future paper.

3.1. Physical layer

At the physical layer, it is assumed that a group of attackers continually monitor wireless transmissions from mobile nodes in order to establish their location and identity, and also to detect any abnormal behavior that might suggest mobile users are attempting to evade the attackers. An alarm indicating that a mobile node possibly changed its MAC address is triggered every time a sudden RSSI variation occurs between two consecutive packets having the same MAC address. Based on this possible attack scenario, in this paper we propose the Safe-zone algorithm as a countermeasure to such physical layer attacks. Safe-zone permits two mobile users to swap their MAC addresses without setting off any alarms, in case a group of attackers is listening to their transmissions. At the core of the Safe-zone operation is a tight coordination among mobile users before and after a MAC address exchange takes place. In Safe-zone, mobile users communicate with other users located nearby to explore which of them is the best candidate to perform a MAC exchange. Before explaining the Safe-zone algorithm, we will formalize the operation of the attacker at the physical layer.

3.1.1. Attacker model

A group of sparsely deployed APs will be considered to be the attackers. These APs are located at fixed locations and are able to retrieve RSSI information from mobile nodes located within range. The attackers use this information to estimate the location of the transmitting source and also to detect any abnormal variation of RSSI measurements that suggest a mobile user changed his MAC address. In [24–26], the authors show that the accuracy of localization algorithms strongly depends on the number of attackers (APs) detecting the mobile user's transmissions; the more attackers eavesdropping on the signal from a mobile user, the more accurate the position estimated by the attackers. As a result, the best scenario to perform a MAC address exchange for a mobile node implementing Safe-zone appears when only one AP is within range. This scenario guarantees that Safe-zone has a higher probability of finding a candidate without triggering any alarm.

Even if it becomes possible for two mobile nodes to perform a MAC address exchange in the absence of any APs (i.e., a hole in the network coverage), as soon as the two nodes reenter the network coverage, the MAC exchange may be exposed by the gap in the sequence numbers and ensuing presence of re-association packets. Similarly, if we constrain MSP to allow MAC swapping events to occur only in places with no network coverage, it limits the ability of nodes to hide their location as many networks have little or no holes in their coverage. We consider that having a strategy such as MSP solves these limitations as it operates in scenarios with none, one, two or more attackers listening to the node's signals. Regardless of the number of attackers listening to the mobile users' transmissions, MSP is able to protect users' location privacy, since the attackers will unwittingly track a wrong user. In the rest of this paper, the discussion will focus on the scenario in which only one AP is within range. However, the same description applies to other scenarios with more APs listening.

The previous section reviewed works related to physical layer attacks, in particular, the body of work related to MAC spoofing. None of the proposed methods detect when two mobile users swap their MAC addresses. However, the *k*-means algorithm proposed in [21] can be modified in such a way that it is able to detect the MAC address swapping, and for that reason we will use it to attack the physical layer. Before explaining how the *k*-means algorithm can be modified to become an attacker in LE scenarios, we will explain in general terms how a MAC swapping detector operates at the physical layer.

It is well documented that RSSI decreases as the distance between transmitter and receiver increases [27]. In basic terms, the ratio between transmitted and received signal strength is determined by the path loss exponent, which is typically modeled using Eq. (1) [28]

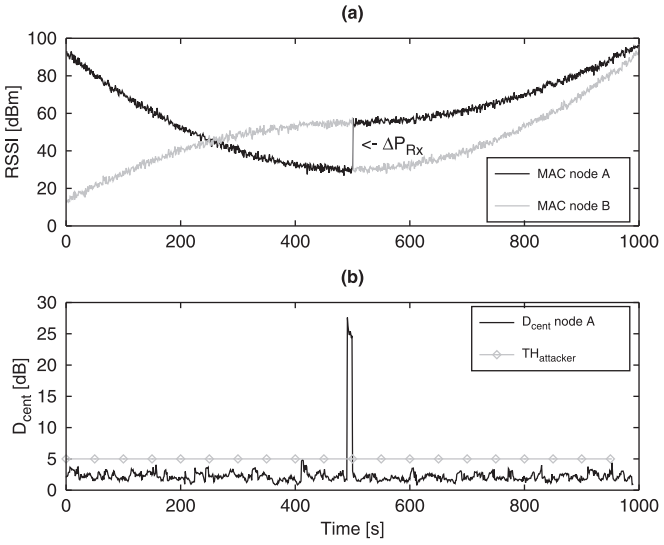


Fig. 1. KSD operation.

$$P_{Rx}(d) = P_{Tx} - PL(d_0) - 10\gamma log\left(\frac{d}{d_0}\right) + X_\sigma \ [dBm] \tag{1}$$

where P_{Tx} is the transmission power, P_{Rx} is the received power, $PL(d_0)$ is the path loss at a reference distance (typically d_0 is set to $1\,m$), d is the distance between transmitter and receiver nodes, γ is the path loss exponent, and finally X is a zero-mean Gaussian random variable with σ standard deviation, representing the shadow-fading factor. From the AP's viewpoint, the received power from a mobile node varies according to Eq. (1). The difference between two consecutive RSSI measurements can be computed using Eq. (2), where P_{Rx_1} is the received power from $node_1$ at distance d_1, and P_{Rx_2} is the received power from $node_1$ at distance d_2

$$\Delta P_{Rx} = P_{Rx1} - P_{Rx2}. \tag{2}$$

Assuming the path loss exponent is similar for all mobile nodes within the AP's coverage area, and considering a constant $K = P_{Tx} - PL(d_0)$, we can substitute Eq. (1) into Eq. (2), resulting in:

$$\Delta P_{Rx} = 10\gamma log\left(\frac{d_2}{d_1}\right) + X_\delta \tag{3}$$

where δ is the standard deviation equal to $\sqrt{2}\sigma$, and ΔP_{Rx} is the received power difference measured at two different locations. Since γ is constant, this equation depends only on the ratio $(\frac{d_2}{d_1})$. A MAC detector operating at the physical layer expects ΔP_{Rx} not to change abruptly between two consecutive RSSI measurements from the same MAC address, as it is assumed that they originate from the same node and there is little or no mobility. For instance, consider the example depicted in Fig. 1(a), where two mobile nodes roam within the AP's coverage. Mobile nodes *A* and *B* exchange their MAC addresses at $time = 500$ s, consequently, node *A* takes the identity of node *B* and vice-versa. From the AP's viewpoint, a MAC address exchange is noticeable only if ΔP_{Rx} exceeds a predefined threshold between two consecutive RSSI values belonging to the same MAC address [29].

The *k*-means spoofing detector (KSD) proposed in [21] could be modified to detect abrupt RSSI variations using a sliding window along the RSSI trace of consecutive received packets from the same MAC address. By considering a window of fixed size *m*, KSD takes

into account a set of RSSI values $(x_1, x_2, \ldots, x_m)$. Then, KSD separates the m values into k sets, so that RSSI values of similar characteristics are clustered together. To detect a MAC exchange, the KSD algorithm should first compute the distance between the centroids (D_{cent}) of the formed clusters. In this way, if such distance exceeds a predefined threshold ($\mathcal{TH}_{attacker}$), an alarm will be raised indicating a possible MAC exchange. In Section 4, $\mathcal{TH}_{attacker}$ was estimated experimentally. Fig. 1(b) shows this operation in which the KSD detector became aware of the exchange at $time = 500$ s since, in this example, D_{cent} exceeded the established $\mathcal{TH}_{attacker} = 5$.

3.1.2. Safe-zone

To overcome MAC swapping detection at the physical layer, an algorithm that allows two mobile users swap their MAC addresses without raising any alarm was proposed (*Safe-zone*). In Safe-zone, a mobile node intending to swap its MAC address broadcasts a MAC address request (MACRQ). Mobile nodes receiving this message reply with a MAC address reply (MACR). Safe-zone assumes that the wireless channel is symmetric (i.e., the RSSI measurements from a mobile node to the AP is similar to RSSI measurements from the AP to the mobile node). Then, the initiator node computes $P_r\{\Delta P_{Rx} > \mathcal{TH}_{Safe-zone}\}$ for each replying candidate and selects those candidates whose $P_r \leq Pr_{max}$, where P_r is the probability that an attacker can detect the MAC address exchange, $\mathcal{TH}_{Safe-zone}$ is the maximum variation of ΔP_{Rx} that passes unnoticed to the attackers, and Pr_{max} is a fixed bound defined by the initiator node to choose the best candidates and discard those whose probability could expose the exchange. To illustrate how Safe-zone works, Fig. 2 shows a set of possible candidates (whose detection probability is below Pr_{max}) colored in black. Fig. 2(a)–(c) show the cases in which one, two and three APs are within range, respectively (the initiator is represented by a star). The initiator must choose the best candidate by negotiating with the node whose P_r is the lowest. If this node does not authorize the exchange, then the initiator node asks the next candidate in ascending order, and so on. If no node authorizes the MAC exchange, the initiator node will start all over again but in a different location. To compute $P_r\{\Delta P_{Rx} > \mathcal{TH}_{Safe-zone}\}$, the initiator node estimates the value of ΔP_{Rx} (see Eq. (3)) that depends on distances d_1 and d_2, in which d_1 is the distance between the initiator and the AP, and d_2 is the distance between the candidate and the AP. To compute d_1, the initiator collects a set of samples of RSSI measurements from the AP. Similarly, candidates send their collected RSSI samples within the MACR packet. Both estimated distances are in the form of a range, since they are computed using Eq. (1) that depends on the Gaussian random variable. Therefore, each distance has a lower and upper bound. As a consequence, the Safe-zone algorithm computes the worst case that maximizes $|10\gamma\log(\frac{d_2}{d_1})|$ in order to guarantee that any combination between d_1 and d_2 does not exceed $\mathcal{TH}_{Safe-zone}$. To achieve this, Safe-zone selects the largest value between $|10\gamma\log(\frac{d_{2_{upper}}}{d_{1_{lower}}})|$ and $|10\gamma\log(\frac{d_{2_{lower}}}{d_{1_{upper}}})|$. Then, Safe-zone computes $P_r\{\Delta P_{Rx} > \mathcal{TH}_{Safe-zone}\}$ as follows:

$$Pr\{\Delta P_{Rx} > \mathcal{TH}_{Safe-zone}\} = \int_{\mathcal{TH}_{Safe-zone}}^{\infty} \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{(x-\mu)^2}{2\delta^2}} dx \quad (4)$$

where μ is the largest value in $|10\gamma\log(\frac{d_2}{d_1})|$, δ is the standard deviation equal to $\sqrt{2}\sigma$ and σ is taken from Eq. (1). Algorithm 1 shows the operation of the Safe-zone algorithm in pseudo-code.

Finally, it is important to note that Safe-zone requires a private channel to transmit all their signaling packets (e.g., MACR or MACRQ) without the attackers being able to listen to the ongoing negotiation between the initiator and candidate nodes. To achieve this, several methods can be used. For instance, in [30] users keep data away from the attackers' sight by using header fields to transport secret information. A different method is to shift to a higher modulation than the maximum one supported by the AP at that distance. However, in this paper we use TPC [13] to transmit MSP's signaling packets to guarantee that these packets can only be decoded by mobile nodes located within a certain area. Nevertheless, the Safe-zone algorithm might work with any mechanism that prevents attackers from receiving signaling packets.

Algorithm 1 Safe-zone.

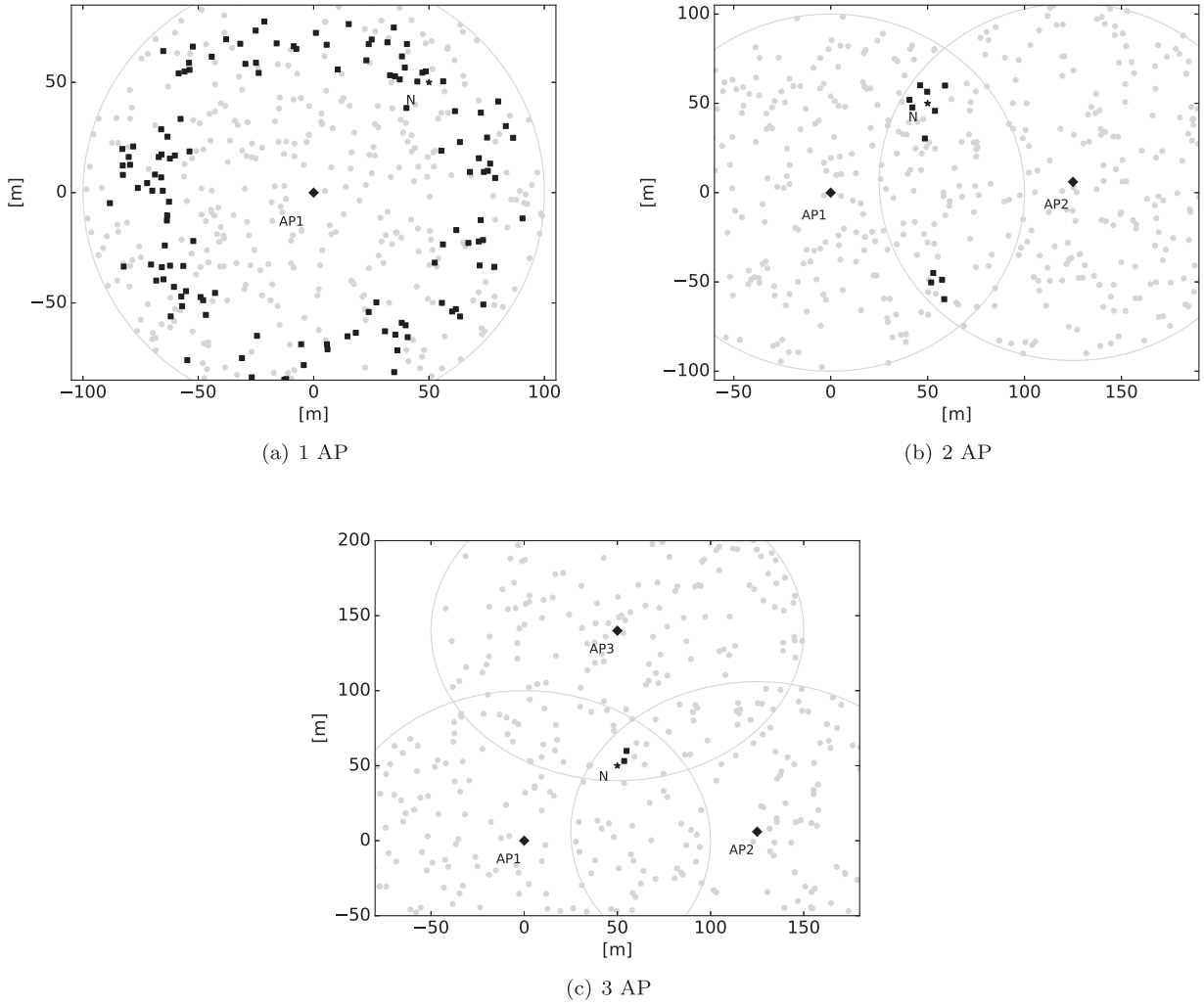
```
1:  procedure Main Loop
2:      while TRUE do
3:          Send MACRQ
4:          if candidate responds with MACR then
5:              Compute d1
6:              for each candidate do
7:                  Compute d2
8:                  Compute μ = max|ΔPRx|
9:                  Compute Pr{ΔPRx > THSafe−zone}
10:                 if Pr < Prmax then
11:                     Append candidate to possible list
12:
13:         Sort possible candidates list by ascending order
14:         for each candidate in the list do
15:             Send MAC exchange start-negotiation
16:             if Candidate Accept then
17:                 Send MAC exchange confirm-negotiation
18:                 Update MAC address in VIME config
19:                 Send MAC exchange end-negotiation
20:                 Break for loop
21:             else
22:                 Send MAC exchange end-negotiation
23:         Wait time interval t0
```

3.2. MAC layer

Even if two mobile users are able to deceive a group of attackers at the physical layer, operating systems require mobile nodes to restart their wireless interfaces every time a node changes its MAC address. This involves resetting the IEEE 802.11s' sequence number field back to zero as well as conducting a re-association process with the AP. Attackers monitoring MAC-related information may trigger an alarm as a result of observing this abnormal behavior from packets belonging to the same MAC address. To overcome this potential threat, we designed VIME, an algorithm that operates at the MAC layer, and is based on virtual network interfaces. VIME allows mobile users to change their MAC addresses without resetting sequence numbers, and without having to re-associate with the AP. Before explaining VIME's operation, we will explain how a MAC layer attacker operates, in general.

3.2.1. Attacker model

Most MAC spoofing detectors reviewed in Section 2 are rendered useless when two users swap their MAC addresses. In [18], for example, the owner and thief nodes are assumed to simultaneously transmit packets using the valid MAC address. This assumption does not hold in MSP, where mobile users will never use the same MAC address at the same time. The MAC spoofing detectors proposed in [23] and [31], however, have a higher probability of detecting a MAC swapping event since their operation is based on observing sequence number discrepancies. In particular, the SNG technique proposed in [23] is appealing since it requires a single computation only (e.g., the difference between two consecutive sequence numbers) in order to detect a MAC exchange, instead of a set of computations as in [31].



(a) 1 AP (b) 2 AP (c) 3 AP

Fig. 2. Potential candidates for different number of APs.

An attacker operating at the MAC layer can be seen as a function $y = f(SN_{gap})$, where SN_{gap} is the difference between two consecutive sequence numbers, while y expresses the probability that a MAC exchange occurs. According to our experiments, the sequence number difference between two consecutive packets can be considered normal if it falls between 1 and 25, since 99.07% of the sequence number difference between consecutive packets falls below 25. Sequence number gaps are mainly due to packet losses and collisions. An alarm using the proposed attacker model is triggered every time the y value exceeds 0.5, since SN_{gap} exceeded the fixed threshold ($SN_{threshold}$) set at 25. In contrast to [23], we also considered the case when the sequence number reached 4096 before returning to zero, which is normal behavior [22] (this is represented in the last term of Eq. (5)). This addition improves the attacker's ability to avoid false positives while monitoring sequence number gaps. This model is shown in Eq. (5)

$$y = f(SN_{gap}) = 1 - e^{-\frac{ln(2)*SN_{gap}^2}{SN_{threshold}^2}} - e^{-\frac{ln(2)*(SN_{gap}-4096)^2}{SN_{threshold}^2}}. \quad (5)$$

Now, let's look at the re-association aspect of the MAC exchange detection. According to the IEEE 802.11 standard, every time a wireless interface re-associates with the AP, a set of authentication packets is mandatory. A re-association event can tip off the attackers of mobile users' countermeasures to avoid detection. We enhanced the attacker model by monitoring re-association events. The IEEE 802.11 header comprises the fields shown in Table 1. We modeled the attacker detector as a function that considers the type and subtype values shown in Table 1 in order to compare whether one of these values appears in the IEEE 802.11 header of any transmitted packet; if such an event should occur, an alarm is triggered (i.e., alarm ← one), otherwise it shows no re-association information (i.e., alarm ← zero).

Table 1
IEEE 802.11 header type and subtype combinations.

Type b3 b2	Type description	Subtype b7 b6 b5 b4	Subtype description
00	Management	0000	Association request
00	Management	0001	Association response
00	Management	0010	Reassociation request
00	Management	0011	Reassociation response
00	Management	0100	Probe request
00	Management	0101	Probe response
00	Management	1010	Disassociation
00	Management	1011	Authentication
00	Management	1100	Deauthentication

3.2.2. VIME

Because attackers can detect a MAC spoofing event that is implementing the SNG technique, we designed *VIME* as a countermeasure that allows mobile users to modify any field in the IEEE 802.11 header before delivering packets to the wireless interface

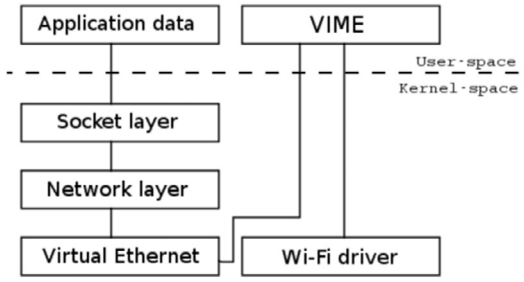


Fig. 3. Relationship between VIME and the Unix networking system.

by using virtual interfaces. Consequently, VIME does not need to restart the wireless interface after a node changes its MAC address, thus avoiding any abnormal behavior in the communication pattern with the AP. VIME guarantees that an attacker analyzing MAC layer information cannot detect any abnormal behavior even if two users swap their MAC addresses simultaneously.

On Unix systems, the TUN/TAP driver [32] allows a user to redirect application packets to a file-descriptor (e.g., typically placed in /dev/net/tun) instead of sending them to a physical interface. This driver can be configured in two different ways. In TUN configuration, the driver creates a logical interface (virtual interface), which is used to deliver traffic between two endpoints. In TAP mode, the driver creates a virtual Ethernet interface that takes packets from the upper layer. These packets are encapsulated using the IEEE 802.3 format and are written into a file descriptor. VIME exploits the TAP mode operation in order to develop a user application that can easily modify the packet's header before delivering it to the wireless NIC. In this way, VIME becomes an intermediary between the virtual Ethernet and the wireless NIC.

Fig. 3 depicts a simplified diagram showing the relationship between VIME and the main blocks involved in the Unix networking system. Whenever an application running in user-space sends data packets to a remote device, the Unix's kernel creates a socket that is placed inside the operating system's kernel, and serves as an interface between applications and network protocols. The Network layer block (see Fig. 3) consists of the TCP/IP communication model. This block performs TCP and IP encapsulation functions, as well as routing functions within the operating system. Finally, the kernel sends IP packets to the virtual Ethernet block. As aforementioned, this block encapsulate IP packet into IEEE 802.3 header packet format. Then VIME can access to each 802.3 packet by reading the file descriptor placed in "/dev/net/tun" in order to remove this header and replace it with a new 802.11 header. This is an essential step of VIME, since at this point VIME has the ability to modify specific fields in order to avoid detection by attackers analyzing MAC layer information. Specifically, VIME updates the packet's MAC address and sequence number fields with the new values matching those of the chosen candidate selected by the Safe-zone algorithm. Once re-encapsulation takes place, VIME injects the modified packets directly to the wireless NIC for transmission. The process of updating the MAC address and the sequence number field to their new values take place instantly, so there is no packet loss in the flow of outgoing packets.

VIME operates in two different ways: when an application layer packet is forwarded to the Wi-Fi driver (see Fig. 3), as well as when the packet goes from the Wi-Fi driver to the application layer. In the latter case, VIME prepares IEEE 802.11 packets before sending them to the virtual Ethernet by replacing the IEEE 802.11 header with the IEEE 802.3 header. An advantage of using virtual interfaces is that VIME runs as a normal user application and uses conventional protocol stack on UNIX systems. VIME Algorithm is shown as pseudo-code in Algorithm 2.

Algorithm 2 VIME.

```
1:  procedure MAIN LOOP
2:      Point to the file descriptor /dev/net/tun
3:      Create raw socket on wireless interface
4:      while TRUE do
5:          if packet in /dev/net/tun then
6:              Read packet
7:              Remove Ethernet header
8:              Append Wi-Fi header
9:              Send packet to wireless interface
10:         if packet in wireless interface then
11:             Read packet
12:             if packet is for this terminal then
13:                 Remove Wi-Fi header
14:                 Append Ethernet header
15:                 Send packet to /dev/net/tun
```

3.3. Exchanging identities in MSP

While a single MAC exchange confuses the attackers with the wrong user's identity, the attackers can potentially gather enough information over time to establish the user's real identity by other means. Therefore, it is recommended that users periodically exchange their MAC addresses (i.e., identifier). How long an identifier should last is an open research problem [33,34]. Various techniques have been proposed to solve this issue, for example, in [17], the authors modeled pseudonym age (i.e., the period of time an identifier is used) as the probability that at least one candidate be found, as well as the exchange cost. In [34], the authors suggested that the identifier exchange frequency in vehicular ad hoc networks (where connectivity is intermittent) be inversely proportional to the time interval in which the mobile user is being tracked. However, for networks in which connectivity remains present, such as pedestrian networks, the number of attackers eavesdropping the mobile node can be used as a metric to establish pseudonym age. In general, the more the attackers eavesdrop the user's signal, the more accurate the resulting location is, thus increasing the user's need to perform an exchange. In [4], the authors proposed a mechanism that allows a user to estimate how accurately the attackers are computing his current position. MSP can use this approach in order to modify the age of identifiers according to the number of attackers eavesdropping a mobile node. However, MSP can work with any mechanism that triggers the Safe-zone algorithm to look for a candidate.

It is important to mention that initiators will only send a MACRQ packet when the age of their present identifiers has expired. Similarly, candidates will only respond with a MACR packet when their identifiers' age has also expired. Once a MAC exchange takes place between two nodes, their identifiers' age is reset. This operation guarantees that no node performs frequent MAC exchanges even if it receives multiple MACRQ packets from various initiator nodes.

4. Experiments and results

This section first describes the methodology used to implement both the attackers and countermeasures proposed in this paper, and subsequently presents the experiments conducted at the physical and MAC layers in order to evaluate the effectiveness of MSP to provide location privacy to mobile users in LE scenarios. Finally, a security and performance analysis as well as a comparison of MSP against attackers and similar countermeasures is also presented.

Table 2
Terms used in safe-zone and VIME.

Param.	Description
D_{cent}	Distance between centroids in KSD algorithm
$\mathcal{TH}_{attacker}$	Physical layer attacker threshold
$\mathcal{TH}_{safe-zone}$	Maximum allowed variation of ΔP_{Rx} considered in safe-zone
P_r	Probability of being detected by eavesdroppers when performing a MAC address exchange
Pr_{max}	Maximum allowed probability to pass unnoticed to the attackers
SN_{gap}	Sequence number gap between two consecutive packets
$SN_{threshold}$	MAC layer attacker threshold considered by SNG

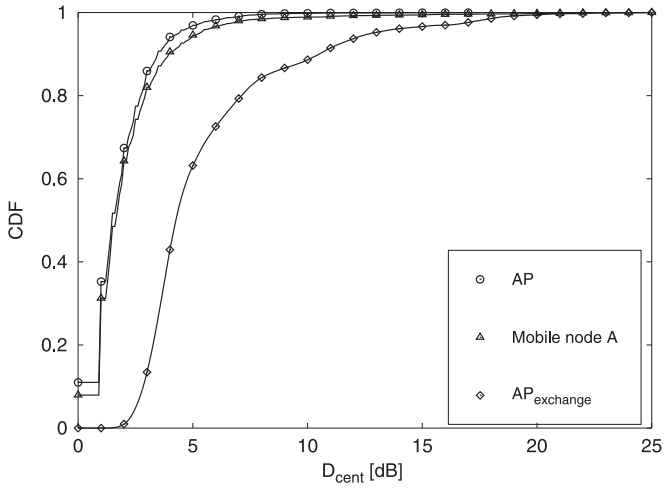


Fig. 4. D_{cent}'s CDF.

Table 2 gives a brief description of the terms used in Safe-zone and VIME algorithms. A detailed description of them is found in Section 3.

As described earlier, the Safe-zone algorithm requires computing probability $P_r\{\Delta P_{Rx} > \mathcal{TH}_{Safe-zone}\}$ in order to select not only the best location, but also the best candidate for a MAC address exchange. However, finding $P_r\{\Delta P_{Rx} > \mathcal{TH}_{Safe-zone}\}$ requires computing parameters such as γ, K and σ in order to estimate ΔP_{Rx}. To achieve this, we deployed an outdoor AP and collected RSSI measurements at several distances with a mobile node equipped with an Atheros chip-set at 2.4 GHz, implementing IEEE 802.11n standard running *tcpdump*. Such distances were in the range from 5 to 100 m in 5 m steps. By minimizing the mean square error between the collected RSSI values and the estimated power, using Eq. (1), γ became 1.34 dBm, K became −44.12 dB and finally σ equaled 2.23 dBm [35].

To evaluate the effectiveness of the attacker algorithm, the false positive rate was defined as the percentage of the experiments where the attacker erroneously believed a MAC address exchange had taken place. The attacker will use this metric to select the best $\mathcal{TH}_{attacker}$ threshold that minimizes the false positive rate. To compute $\mathcal{TH}_{attacker}$, we conducted two experiments. In experiment 1, we characterized the variation of D_{cent} in the absence of MAC address exchanges by using the KSD algorithm. For this experiment, we deployed a fixed AP and a mobile node A roaming within the AP's coverage area (maximum communication range was about 100 m). The AP collected 1×10^5 RSSI samples from packets transmitted by the mobile node. We fixed the window size at 10 and set k equal to 2 for the KSD algorithm. Fig. 4 shows the D_{cent}'s CDF for measurements taken by the AP (line with circles). Similarly, mobile node A also collected the same number of RSSI samples from the AP to characterize ΔP_{Rx} in the absence of MAC address exchange by means of the KSD algorithm. Fig. 4 also shows the D_{cent}'s CDF for measurements taken by mobile node A (line with triangles). In this figure, the CDF obtained by the AP and the CDF acquired by the mobile node A are similar, which supports our claim that the wireless channel can be considered symmetrical.

Table 3
$\mathcal{TH}_{attacker}$ threshold vs false positive rates.

$\mathcal{TH}_{attacker}$	AP's false positive	Node A's false positive
1	0.64	0.68
2	0.32	0.35
3	0.14	0.18
4	0.05	0.09
5	0.03	0.05
6	0.01	0.03
7	0.009	0.02
8	0.004	0.014
9	0.0023	0.012
10	0.0021	0.010
15	0.0004	0.005

In experiment 2, a second mobile node (B) roaming within the AP's coverage area was considered. Again, we collected 1×10^5 RSSI samples as node A exchanged its MAC address with node B while the AP computed D_{cent} for each node. Fig. 4 also shows D_{cent}'s CDF taken by the AP when a MAC address exchange took place (line with diamonds). In this figure we can see that if the attacker chooses $\mathcal{TH}_{attacker} = 15$, only 4% of MAC address exchanges are detected (see the line with diamonds). At the same time, the AP has a false positive rate of 0% (see the line with circles, in which all D_{cent} values obtained in the absence of MAC address exchange are found under this line). This $\mathcal{TH}_{attacker}$ value minimizes the false positive rate, although the identity exchange detection rate is very small. On the opposite extreme, if the AP chose $\mathcal{TH}_{attacker} = 1$, this would imply that the AP can detect a MAC address exchange event with 100% accuracy (see line with diamonds), but at the same time the AP has a 64% false positive rate (see line with circles). This means that in 64% of all the events tracked by the AP (considering there might be dozens of nodes connected to the AP at any time), the AP cannot discern which ones are truly MAC address exchange events. Table 3 shows different $\mathcal{TH}_{attacker}$ values and the percentage of false positives. We consider that $\mathcal{TH}_{attacker} = 5$ minimizes the false positive rate while it maximizes the number of correct MAC exchange detections. For this $\mathcal{TH}_{attacker}$ value the attacker detects 37% of real MAC address exchanges with only a 3% false positive rate. Following the discussion and best settings proposed for the attacker, a mobile node implementing the Safe-zone algorithm can also select its threshold ($\mathcal{TH}_{Safe-zone}$) using Table 3.

To evaluate the effectiveness of the Safe-zone algorithm, we defined the false positive rate as the percentage of cases in which the Safe-zone algorithm erroneously believed that the MAC address exchange would not be detected by the attackers. To quantify the false positive rate, we resorted to experiment 2, in which mobile nodes A and B roam within a single AP's coverage. We set node A as the initiator requesting a MAC exchange and node B as the only available candidate. During the MAC address exchange, the initiator computes $P_r\{\Delta P_{Rx} > \mathcal{TH}_{Safe-zone}\}$, and the attacker computes D_{cent}. For the purpose of this experiment, we collected 1×10^3 MAC address exchanges, and fixed $\mathcal{TH}_{Safe-zone}$ at 5. Fig. 5 shows how the false positive rate varies according to Pr_{max} and $\mathcal{TH}_{attacker}$. In this figure, we observed that the higher the value of Pr_{max}, the higher the false positive rate. For instance, suppose the attacker fixed $\mathcal{TH}_{attacker} = 4$ and the Safe-zone algorithm chose $Pr_{max} = 0.2$, then the attacker would detect 29.14% of the MAC address exchanges (see Fig. 5). Now, suppose that the attacker and the initiator have the same threshold ($\mathcal{TH}_{Safe-zone} = \mathcal{TH}_{attacker} = 5$), Fig. 5 shows that when $Pr_{max} = 0.2$, only 5.2% of the MAC ad-

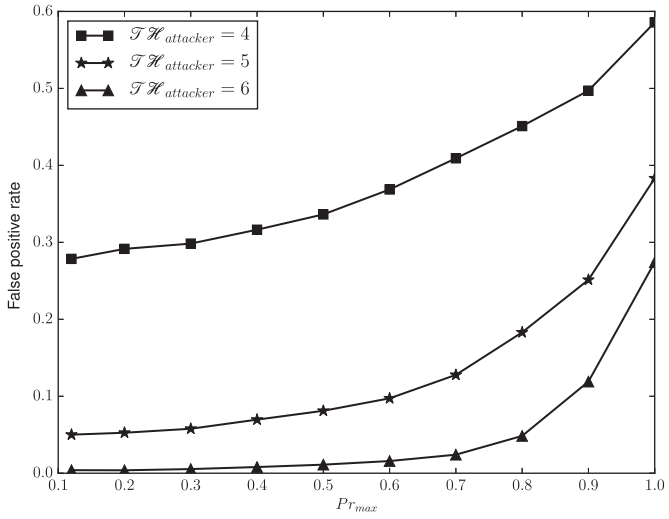


Fig. 5. False positive rate vs. Pr_{max}.

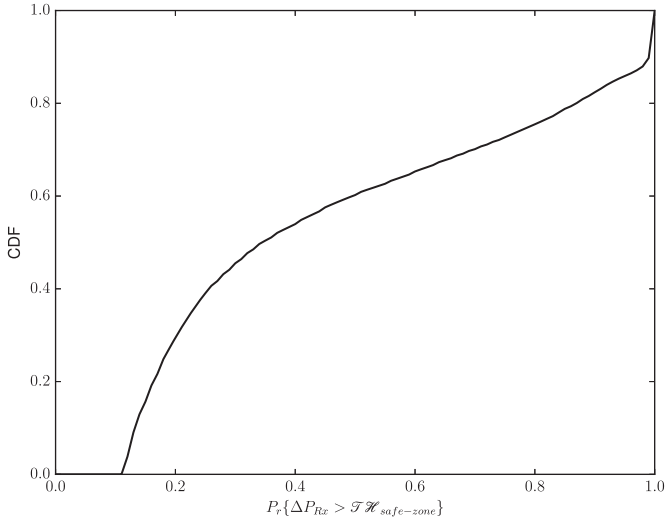


Fig. 6. CDF of P_r values in experiment 2.

dress exchanges are detected. We can also see in this figure that the false positive rate decreases when $\mathcal{TH}_{attacker} > \mathcal{TH}_{Safe-zone}$.

Fig. 6 shows the CDF of collected P_r values for experiment 2. In this figure, we can see that 29.43% of the cases fall below $Pr_{max} = 0.2$. In other words, a mobile node roaming within the AP's coverage area has a probability of 29.43% of finding a candidate with a $P_r \leq 0.2$. In contrast, by fixing $Pr_{max} = 0.4$, the user has a 53.95% probability of finding a candidate. However, Fig. 5 shows that if Safe-zone selects $Pr_{max} = 0.4$ and $\mathcal{TH}_{Safe-zone} = 5$, and the attacker chooses $\mathcal{TH}_{attacker} = 4$, the Safe-zone effectiveness will be 68.37% . In the following experiments, we used $Pr_{max} = 0.2$, since according to the results in Figs. 5 and 6, with this setting Safe-zone has only 5% of false positives while it guarantees there is a 30% probability of finding a suitable candidate to carry out the MAC address exchange.

Using collected results from experiment 2, Fig. 7(a) demonstrates the candidates' probability (P_r) computed when the initiator was located 15 m away from the AP. We can see in this figure that according to Safe-zone, the best candidates are located between 14–19 m and 22–26 m away from the AP, since their P_r is smaller than $Pr_{max} = 0.2$, when $\mathcal{TH}_{Safe-zone} = 5$. Moreover, if the attacker should choose $\mathcal{TH}_{attacker} = 5$, nodes located between 8–10 m and 15–28 m performing a MAC address exchange will not be identified by attacker's detectors (see Fig. 7(b)). Similarly, Fig. 8(a) shows the candidates' probability (P_r) when the initiator is located 50 m away from the AP. We can see in this figure that according to Safe-zone, the best candidates are now located between 35 and 75 m away from the AP since their P_r is smaller than $Pr_{max} = 0.2$, when $\mathcal{TH}_{Safe-zone} = 5$. If the attacker should choose $\mathcal{TH}_{attacker} = 5$, the candidate nodes selected by Safe-zone will not be identified by the attackers since their D_{cent} distance is smaller than $\mathcal{TH}_{attacker}$ (see Fig. 8(b)). In contrast, if the attacker should choose $\mathcal{TH}_{attacker} = 2$, any MAC address exchange will not be detected. However, the attacker will observe 32% false positives (see Table 3).

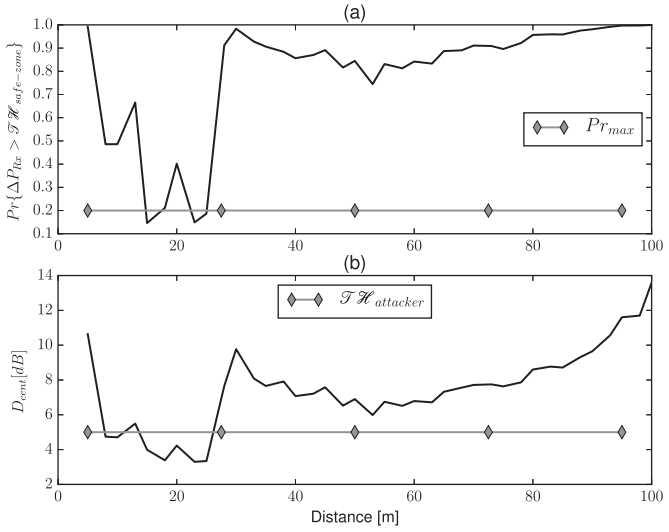


Fig. 7. Safe-zone's potential candidates when initiator is located 15 m away from the AP.

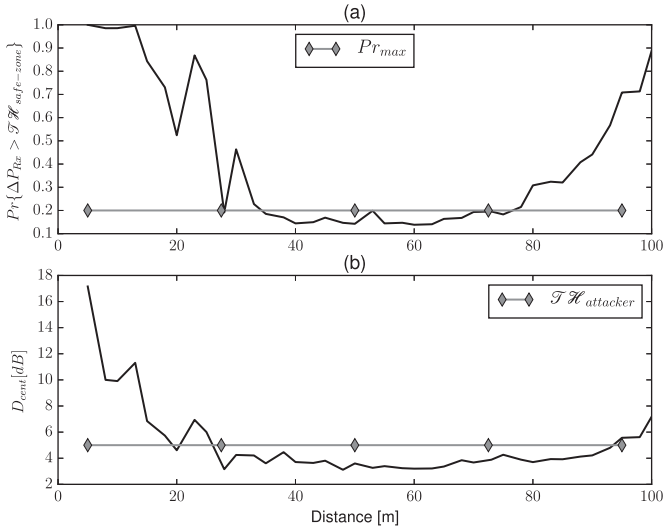


Fig. 8. Safe-zone's potential candidates when initiator is located 50 m away from the AP.

Finally, we combined Safe-zone and VIME to evaluate MSP's full potential as an effective location privacy tool in a test-bed scenario. We presented a scenario in which two mobile nodes swapped their MAC addresses with and without using MSP. Fig. 9(a) shows RSSI values measured by the AP from two mobile nodes roaming within its coverage area. At $time = 1957$ s, both nodes swap their MAC addresses without using MSP. Fig. 9(b) shows how the sequence number of both mobile nodes went back to zero. Moreover, Fig. 9(c) shows how the physical layer detector raised an alarm since D_{cent} exceeded $\mathcal{TH}_{attacker} = 5$, for both nodes. Fig. 9(d) shows

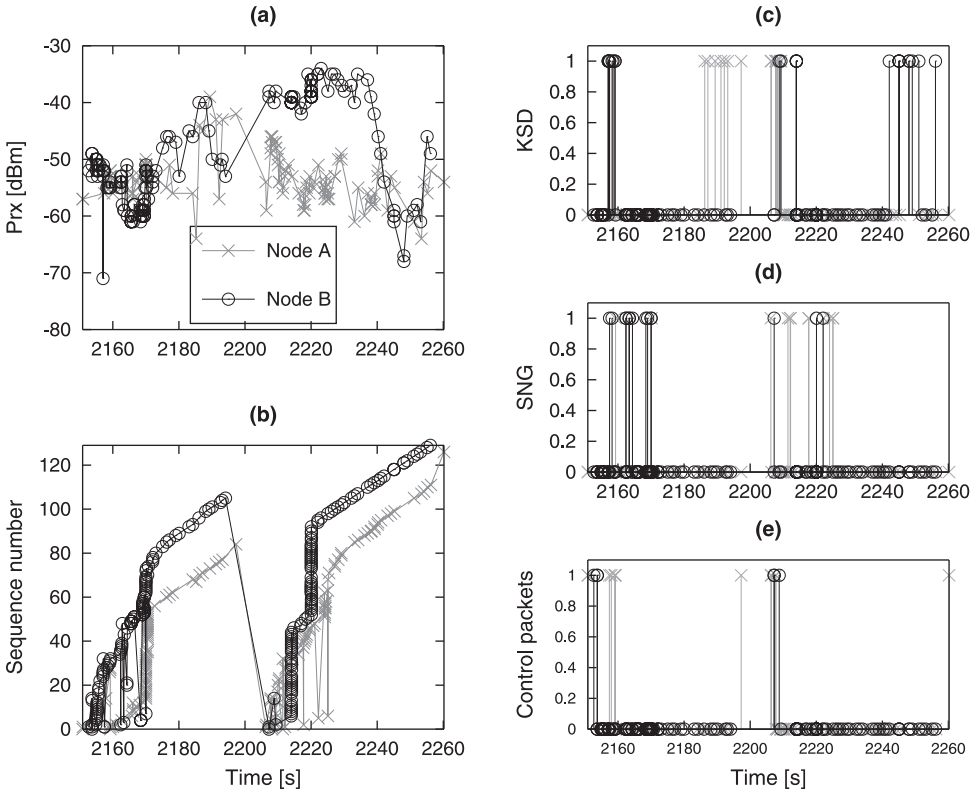


Fig. 9. MAC address exchange without MSP.

Table 4
Combination of physical and MAC layer detectors.

KSD	SNG	Control packets	Description
0	0	0	Normal operation
0	0	1	Abnormal behavior
0	1	0	Packet loss or MAC spoofing
0	1	1	Interface reboot or possible identity exchange
1	0	0	RSSI fluctuations or MAC spoofing
1	0	1	Abnormal behavior
1	1	0	MAC Spoofing
1	1	1	MAC Address identity exchange

how the sequence number gap exceeded the 25 threshold according to Eq. 5. Finally, Fig. 9(e) shows the presence of control packets due to the re-association process with the AP after the nodes acquired a new MAC address. This scenario exemplifies a MAC address exchange that triggered all alarms since no countermeasures were considered.

In order to reduce false positives, the attackers could combine physical and MAC layer detectors, as shown in Table 4. For instance, when the sequence number detector and the control packets detector are both equal to one but RSSI detector is zero, this probably implies that an interface reboot event occurred. Combining the three detectors in this way decreases the attackers' false positive rate, thus increasing the probability of detecting a real MAC address exchange event.

Fig. 10 shows a MAC address exchange performed by two mobile nodes using MSP. We can see in this figure that the sequence number associated to each MAC address increases monotonically without raising any alarm. Moreover, the sequence number detector at the MAC layer did not trigger any alarm at $time = 1448$ s when a MAC address exchange was carried out between the two mobile nodes. For this experiment, the initiator was located 30 m away from the AP, while the candidate was located 50 m away from the AP. The initiator fixed $Pr_{max} = 0.2$ and $\mathcal{TH}_{Safe-zone} = 5$. This scenario exemplifies a MAC address exchange that triggered no alarms even though the information from both the physical and MAC layer detectors were combined.

Table 5
MSP vs attackers.

Layer	Attacker	MSP's protection
PHY	Trilateration [3]	Since MSP dissociates the user's location from his identity, trilateration algorithms will track the wrong user.
PHY	MAC spoofing [18–21]	Since the safe-zone algorithm selects the best candidate node by looking for other nodes with similar RSSI values, from the attacker's perspective, it eliminates any abnormal behavior at the physical layer (i.e., RSSI gap).
MAC	MAC spoofing [18,23]	Since VIME can modify the 802.11 packet headers (source MAC address, sequence number, etc.) between the candidate and initiator nodes, the identity exchange raises no abnormal behavior at the MAC layer (i.e., the sequence number gap).

4.1. Security analysis

This section describes the MSP security analysis against a general attacker model having access to both physical and MAC layer information. Table 5 lists known attackers designed to operate at

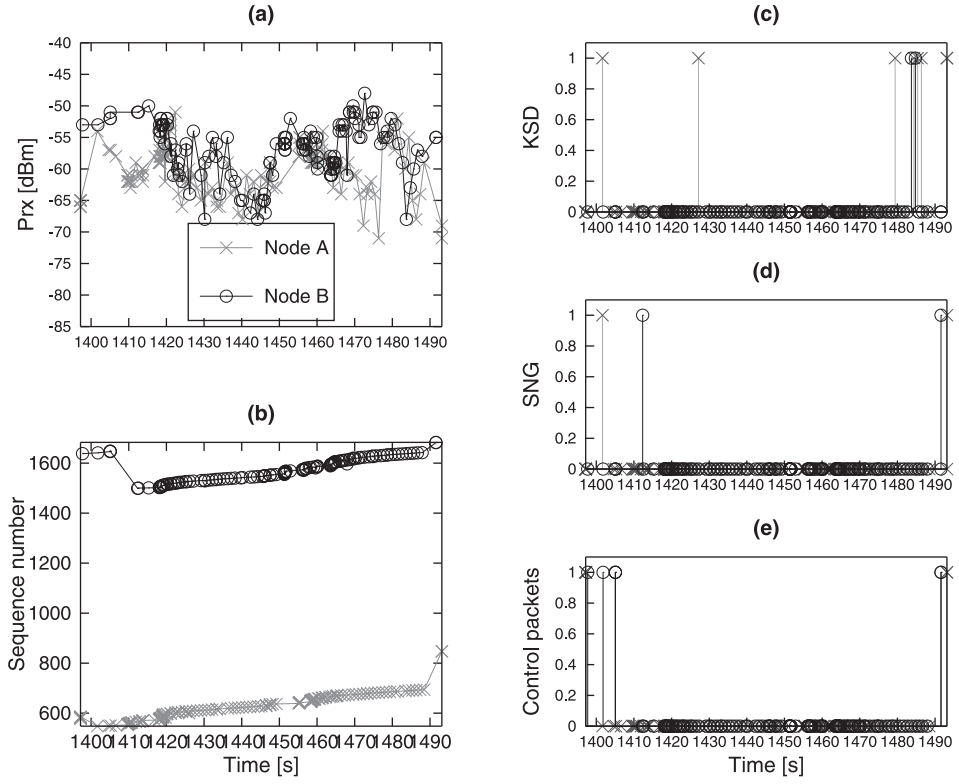


Fig. 10. MAC address exchange with MSP.

the physical or MAC layers as well as a discussion about whether or not MSP was able to defeat them. As the table shows, no known attacker was able to detect a MAC address exchange using MSP. Moreover, in case the access point (attacker) also pretended to be a candidate or initiator in a MAC address exchange, the Safe-zone algorithm would discard such user, since users located near the access point always have a P_r above the $\mathcal{TH}_{Safe-zone}$, see Fig. 7(a), and Fig. 8(a) in which the value of P_r is closest to one when the candidate is located around the AP. On the other hand, if an attacker became another mobile node, it is possible that the user exchanged his MAC address with the attacker, in which case the user's identity becomes temporarily compromised. Even in this situation, MSP still guarantees the user's location privacy since the attacker is unaware that the MAC address exchanged by the user is already another node's address (i.e., obtained in a previous exchange). Finally, the attacker can only use the user's reported identity for as long as the age of the user's identifier lasts.

4.2. Performance comparison

This section first analyzes a comparison of MSP versus similar schemes, and subsequently presents test-bed experiments conducted to evaluate VIME's overhead.

In the literature only the works in [5,6,13] operate in LE scenarios. These works protect the mobile nodes' location privacy at the physical or MAC layers. As Table 6 shows, most schemes can withstand location privacy attacks only in the layer in which they were designed to operate. However, they are rendered useless once an attacker uses information from the other layer. For example, TPC is able to protect a user's location privacy from attackers implementing trilateration techniques (see the check-mark), but, it cannot protect the user from spoofing attacks (see cross marks). This table also shows that MSP is able to deceive attackers having access to information from both layers simultaneously (see check-marks).

Table 6
Performance comparison.

Layer	Algorithm	PHY		MAC
		Trilat	MACSpoof	MACSpoof
PHY	TPC [13]	✓	×	×
MAC	MAC exchange [5,6]	×	×	✓
BOTH	MSP	✓	✓	✓

In order to evaluate VIME's overhead, we conducted two experiments. In the first experiment, we measured throughput performance using the Iperf tool across two Linux devices equipped with a 100 Mbps NIC. We varied the packet rate from 1 Mbps to 108 Mbps and ran each test 100 times to acquire average values. We carried out throughput experiments with and without VIME. Results showed that both experiments presented the same number of packets received.

In the second experiment, we measured the time overhead added by VIME's operation. We used the ping tool to measure round-trip times between the two Linux devices. We varied the packet size from 56 bytes to 1400 bytes and ran each test 100 times to obtain average results. Once again, we carried out the experiments in the absence of VIME and with VIME. Results showed that VIMEs operation added about 100 μs of processing time for each packet on average.

5. Conclusions

In this paper, we propose the MAC Swapping Protocol (MSP), a strategy that allows two mobile users to swap their MAC addresses, avoiding the attacker's detection in LE scenarios. This strategy provides location privacy for mobile users against potential eavesdroppers having access to physical and MAC layer information in WLAN scenarios. In contrast to the majority of related

proposals in which MAC address exchanges require at least k mobile users to intervene, MSP needs only two mobile users to carry out the exchange in order to deceive potential eavesdroppers. We propose an algorithm named *Safe-zone* in order to solve the problem related to physical layer detection when two mobile users swap their MAC addresses. We also propose an algorithm based on virtual interfaces (*VIME*) in order to solve problems related to MAC layer detection (MAC spoofing detectors). Combining both algorithms provides an integral solution to mobile users that prevents potential eavesdroppers from detecting a MAC address exchange at the physical and MAC layers. This in turn dissociates a mobile user's identity from its actual location, seen from the attackers' viewpoint. We supported the validity of MSP through test-bed experiments implementing standard IEEE 802.11. We demonstrated that when the attacker and the Safe-zone algorithm select similar thresholds, Safe-zone has 95% effectiveness when an identity exchange takes place. Moreover, when $\mathcal{TH}_{attacker}$ is greater than $\mathcal{TH}_{Safe-zone}$, Safe-zone effectiveness is nearly a 100% . On the other hand, results showed that VIME achieved 100% effectiveness at the MAC layer. The combination of both algorithms is sufficient to protect user's location privacy. In addition, our test-bed experiments also showed that VIME requires about 100 μs on average to process each packet before delivering it to the wireless NIC. Nevertheless, MSP operation presented no throughput degradation.

Acknowledgment

This work was supported in part by research funds from PAPIIT GRANT (IN116316/IN117017).

References

[1] A.R. Beresford, F. Stajano, Location privacy in pervasive computing, IEEE Pervasive Comput. 2 (1) (2003) 46–55.
[2] M. Duckham, L. Kulik, A formal model of obfuscation and negotiation for location privacy, in: International Conference on Pervasive Computing, Springer, 2005, pp. 152–170.
[3] L. Lin, H.C. So, Y.T. Chan, Accurate and simple source localization using differential received signal strength, Digit. Signal Process. 23 (3) (2013) 736–743.
[4] F. Garcia, et al., LEA: an algorithm to estimate the level of location exposure in infrastructure-based wireless networks, Mob. Inf. Syst. (2017).
[5] M. Gruteser, D. Grunwald, Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis, Mob. Netw. Appl. 10 (3) (2005) 315–325.
[6] M. Lei, X. Hong, S. Vrbsky, Protecting location privacy with dynamic MAC address exchanging in wireless networks, in: IEEE Global Telecommunications Conference GLOBECOM'07, 2007, pp. 49–53.
[7] P. Wightman, et al., Evaluation of location obfuscation techniques for privacy in location based information systems, in: IEEE Latin-American Conference on Communications (LATINCOM), 2011, pp. 1–6.
[8] J. Krumm, Inference attacks on location tracks, Pervasive Comput. 4480 (2007) 127–143.
[9] C.A. Ardagna, et al., An obfuscation-based approach for protecting location privacy, IEEE Trans. Depend. Secur. Comput. 8 (1) (2011) 13–27.
[10] C. Bettini, et al., Privacy in Location-based Applications: Research Issues and Emerging Trends, 5599, Springer Science & Business Media, 2009.
[11] L. Asmaa, K.A. Hatim, M. Abdelaaziz, Localization algorithms research in wireless sensor network based on multilateration and trilateration techniques, in: 3rd IEEE International Colloquium in Information Science and Technology (CIST), 2014, pp. 415–419.
[12] G. Mao, B. Fidan, B.D.O. Anderson, Wireless sensor network localization techniques, Comput. Netw. 51 (10) (2007) 2529–2553.
[13] T. Jiang, H.J. Wang, Y. Hu, Preserving location privacy in wireless lans, in: Proceedings of the 5th International Conference on Mobile Systems Applications and Services, 2007, pp. 246–257.
[14] T. Wang, Y. Yang, Location privacy protection from RSS localization system using antenna pattern synthesis, in: IEEE Proceedings of INFOCOM, 2011, pp. 2408–2416.
[15] K. Bauer, et al., Using wireless physical layer information to construct implicit identifiers, in: 1st Hot Topics in Privacy Enhancing Technologies, 2(2), 2008, pp. 3–6.
[16] J. Freudiger, et al., On non-cooperative location privacy: a game-theoretic analysis, in: 16th Proceedings of the ACM Conference on Computer and Communications Security, 2009, pp. 324–337.
[17] J. Freudiger, et al., On the age of pseudonyms in mobile ad hoc networks, in: IEEE Proceedings of INFOCOM, 2010, pp. 1–9.
[18] D. Madory, New Methods of Spoof Detection in 802.11 b Wireless Networking, Dartmouth College, 2006 Ph.D. thesis.
[19] Y. Chen, et al., Detecting and localizing identity-based attacks in wireless and sensor networks, IEEE Trans. Veh. Technol. 59 (5) (2010) 2418–2434.
[20] Q. Li, W. Trappe, Light-weight detection of spoofing attacks in wireless networks, in: IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), 2006, pp. 845–851.
[21] Y. Chen, et al., Detecting and localizing wireless spoofing attacks, in: Securing Emerging Wireless Systems, 2009, pp. 1–18.
[22] IEEE 802.11 Standard, http://standards.ieee.org/getieee802/download/802.11-2012.pdf, 2012.
[23] F. Guo, T. Chiueh, Sequence number-based MAC address spoof detection, in: International Workshop on Recent Advances in Intrusion Detection, 2005, pp. 309–329.
[24] P. Bahl, V.N. Padmanabhan, A. Balachandran, Enhancements to the radar user location and tracking system, Microsoft Res. 2 (MSR-TR-2000-12) (2000) 775–784.
[25] O. Arana, Técnicas de privacidad geográfica en redes móviles, Universidad Nacional Autónoma de México, 2011 Master's thesis.
[26] F. Garcia, et al., Ghost: Voronoi-based tracking in sparse wireless networks using virtual nodes, Telecommun. Syst. 61 (2) (2016) 387–401.
[27] A. Goldsmith, Wireless Communications, Cambridge University Press, 2005.
[28] T. Sarkar, et al., A survey of various propagation models for mobile communication, IEEE Antenna. Propag. Mag. 45 (3) (2003) 51–82.
[29] J. Yang, et al., Detection and localization of multiple spoofing attackers in wireless networks, IEEE Trans. Parallel Distrib. Syst. 24 (1) (2013) 44–58.
[30] J. Lubacz, W. Mazurczyk, K. Szczypiorski, Principles and overview of network steganography, IEEE Commun. Mag. 52 (5) (2014) 225–229.
[31] Q. Li, W. Trappe, Detecting spoofing and anomalous traffic in wireless networks via forge-resistant relationships, IEEE Trans. Inf. Foren. Secur. 2 (4) (2007) 793–808.
[32] M. Krasnyansky, Universal TUN/TAP driver, 2017, (Online).
[33] J. Petit, et al., Pseudonym schemes in vehicular networks: a survey, IEEE Commun. Surv. Tutor. 17 (1) (2015) 228–255.
[34] E.C. Pons, G. Baldini, D. Geneiatakis, A wireless propagation analysis for the frequency of the pseudonym changes to support privacy in VANETs, in: 40th Jubilee International ICT Convention, MIPRO, 2017, pp. 1485–1490.
[35] J. Miranda, et al., Path loss exponent analysis in wireless sensor networks: experimental evaluation, in: 11th IEEE International Conference on Industrial Informatics (INDIN), 2013, pp. 54–58.

Oscar Arana received his BS degree in Telecommunications Engineering in 2009 from the National Autonomous University of Mexico (UNAM) and his M.Eng. with honors in Telecommunications Engineering in 2011 from the National Autonomous University of Mexico. He has worked as programmer. His areas of interest are location privacy, localization, machine learning techniques. He is currently a Ph.D. student at UNAM.

Francisco Garcia received his B.Sc. with honors in Electrical Engineering in 2005 from the National Autonomous University of Mexico (UNAM) and his M.Sc. with honors in Computer Science from the National Autonomous University of Mexico (UNAM-IIMAS). He has worked as Servers and Network Manager and Programmer Senior. His areas of interest are computational geometry, tracking and localization techniques. He is currently a full time professor at the Department of Telecommunications Engineering, School of Engineering (UNAM).

Javier Gomez received the BS degree with honors in Electrical Engineering in 1993 from the National Autonomous University of Mexico (UNAM) and the MS and Ph.D. degrees in Electrical Engineering in 1996 and 2002, respectively, from Columbia University and its COMET Group. During his Ph.D. studies at Columbia University, he collaborated and worked on several occasions at the IBM T.J. Watson Research Center, Hawthorne, New York. His research interests cover routing, QoS, and MAC design for wireless ad hoc, sensor, and mesh networks. Dr. Gomez is currently a full time professor at the Department of Telecommunications Engineering, School of Engineering (UNAM). Javier Gomez is member of the SNI (level II) since 2016.

Victor Rangel Licea obtained his Bachelor Degree in Computer Engineering from the National Autonomous University of Mexico (UNAM). He obtained his Master Degree in Data Communication Systems and his doctoral degree in Telecommunications Engineering from the Centre for Mobile Communications Research, The University of Sheffield (England). His Ph.D. thesis focused on the modeling and analysis of Cable TV networks supporting broadband Internet traffic. Dr. Rangel is currently a full time professor at the Department of Telecommunications Engineering, School of Engineering (UNAM).