PEDRO: Secure Pedestrian Mobility Verification in V2P Communication using Commercial Off-the-shelf Mobile Devices

Yucheng Yang, Kyuin Lee*
University of Wisconsin–Madison
yang552,kyuin.lee@wisc.edu

Younghyun Kim
University of Wisconsin–Madison
younghyun.kim@wisc.edu

Kassem Fawaz
University of Wisconsin–Madison
kfawaz@wisc.edu

ABSTRACT

Vehicle-to-Pedestrian (V2P) communication enables numerous safety benefits such as real-time collision detection and alert, but poses new security challenges. An imminent and probable scenario is where a malicious node claiming to be a legitimate pedestrian within the network broadcasts false observations or phenomena on the roads (e.g., traffic load, road hazard, and false road crossing alarms) in order to impede traffic flow, erode user's trust in alert messages, or even cause traffic accidents. Therefore, it is crucial to identify legitimate road users against adversaries pretending to be one. In this work, we propose PEDRO, a **PED**est**R**ian m**O**bility verification mechanism for pedestrians using commodity hardware, where only legitimate mobile pedestrians can be admitted to the ad hoc network consisting of trustworthy vehicles and pedestrians. We leverage the round-trip time (RTT) of wireless signal between vehicle and pedestrian's devices, and verify only moving (mobile) ones while rejecting stationary ones, based on the realistic assumption that the adversaries are likely to remotely launch attacks through static malicious devices. Through an extensive analysis based on simulation as well as real-world experiments, we show that PEDRO's verification takes under 8 s while achieving an 8.5% Equal Error Rate (EER) under regular road environments.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Security and privacy** → *Authentication.*

KEYWORDS

secure location verification; vehicle-to-pedestrian verification

---

*Both authors contributed equally to this research.

---

1 INTRODUCTION

Vehicle-to-Pedestrian (V2P) communication is a networking paradigm that involves direct communication between a vehicle and pedestrians within its vicinity. It promises to improve pedestrian safety and reduce roadway fatalities and injuries through exchanging advanced warning messages between pedestrians and drivers [1]. These messages enable the different road users to cooperatively prevent collisions based on their exchanged position, speed, and direction information ahead of time [8, 11]. Systems that use V2P to improve road safety share a common working principle: detect a road crossing event from the pedestrian's behavior and alert nearby vehicles through wireless messages [19].

Despite its safety benefits, security is one of the biggest barriers to the adoption of V2P communication in safety-critical applications [3, 7, 18]. In particular, false data injection attacks, by which an attacker broadcasts fake/false warning messages, hinder the safety of road users and undermine the trust in the V2P system [3, 6, 10]. In this paper, we are concerned with one relevant attack scenario that we plan to generalize later: an adversary can deploy a wireless node to broadcast false messages about pedestrians disregarding traffic rules (i.e., jaywalking); these messages will cause trailing vehicles to be continuously congested, trying to heed the misguided collision warnings. To prevent such an attack, it is essential to verify the authenticity of the claims of every pedestrian before its messages are accepted by vehicles.

One solution for this problem is deploying a trust infrastructure to develop and maintain trust relationships between pedestrians and vehicles in the V2P system [10]. A road user enrolls in the V2P system by registering and authenticating their identities [6] to the trust infrastructure. While seemingly straightforward, this solution suffers from scalability and usability shortcomings because: 1) the deployment of the trust verification infrastructure is costly, and 2) it requires time-consuming user involvement for registration, especially when visiting new locations [10]. In this paper, we advocate for an alternative direction that does not rely on any pre-existing infrastructure and thus does not require explicit registration. We propose a novel approach to *verify the claimed behavior of a pedestrian*, where a vehicle rejects messages until it can conclusively verify that the movement pattern of the sender matches that of a pedestrian.

Existing movement or location verification schemes are limited in terms of their practical V2P use cases. For example, Schafer et al. [15] proposed to verify the sender mobility by leveraging Doppler shift measurements from multiple verifier nodes. Other methods verify the sender's location by utilizing the received signal power in an outdoor environment or a combination of GPS, ad hoc location, or dead reckoning [2, 4, 13]. However, these mechanisms apply only in limited scenarios as they require carefully positioned

verifiers, which might not be readily available in most realistic V2P scenarios [17]. They also need specialized hardware capabilities that are not available in commercial off-the-shelf (COTS) devices. A practical mobility verification scheme should avoid the above shortcomings by meeting the following conditions. First, it should utilize readily available hardware with no or minimal modification. Second, it should operate without pre-distributed verifiers that are costly to deploy and manage.

In this work, we propose PEDRO, a new mobility verification mechanism that uses commodity smartphones without requiring pre-distributed verifiers. In PEDRO, the vehicle receiving a message from a potential pedestrian utilizes the round-trip time (RTT) of a wireless signal to verify the movement of the sender. Realizing PEDRO without specialized hardware or verifiers is a challenging proposition. First, measurements about the sender's location using round-trip time (RTT) tend to be inaccurate from non-specialized hardware. Second, the lack of pre-distributed verifiers makes it hard to obtain measurements from different anchor points. PEDRO addresses these challenges through a novel concept of tracking the sender's possible region. In particular, the vehicle keeps track of the sender's possible location or *region* by mapping the measured (and often inaccurate) RTT into a disk region. Tracking these regions over a period of time allows the vehicle to conclusively verify whether the sender has moved along the side of the road, which implicitly implies that the sender is a moving pedestrian, not a stationary device. We prove that under a set of very realistic conditions, a stationary attacker cannot mimic the behavior of a moving pedestrian. Our analysis and real-world experiments show that this mechanism is simple, quick, and tolerant to noisy GPS and RTT measurements.

2 SYSTEM AND THREAT MODELS

We consider an urban vehicular ad hoc network environment where different road users (vehicles and pedestrians) exchange messages to enhance road safety and efficiency. In this paper, we adopt the notion of *mobility verification* as a proxy for the authenticity of the sender as a moving pedestrian. We consider a device as belonging to a legitimate pedestrian only when it is moving, and any non-moving device is regarded as a stationary pedestrian or malicious attacker, both of which should not send warning messages. As we will prove in Section 4.2.2, a stationary attacker is not able to impersonate a moving pedestrian, and the only way to be able to send an accepted message is to actually move. This constraint significantly raises the bar for broadcasting fake safety transmissions.

Problem Definition: We define the *pedestrian mobility verification* problem as following: a set of moving vehicles, known as *verifiers*, $V = \{V_1, V_2, ...V_j\}$, where j denotes vehicle number, aims to verify the mobility of a *prover* (pedestrian) P. Once its mobility has been verified, the vehicles accept and process safety broadcasts from P, which is shared with all other verifiers.

2.1 System Model

Figure 1 illustrates the system model of PEDRO. As one or more verifiers on the road (V_1 and V_2) pass by a prover (P_1 or P_2), multiple RTT measurements are taken over time to estimate the distance between the verifiers and the prover. We assume all vehicles in V
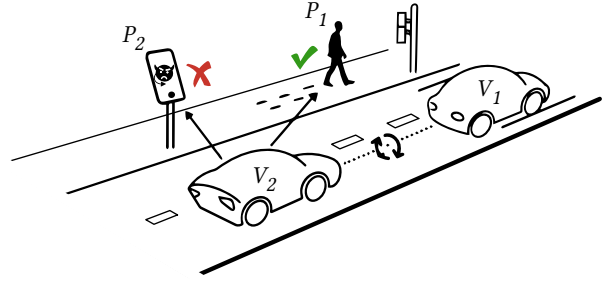
Figure 1: Traveling verifiers on the road (V_1 and V_2) attempts to verify the mobility of the provers (P_1 and P_2) and verify only moving pedestrian P_1.

to be already authenticated as trusted vehicles based on existing state-of-the-art V2V authentication protocols [9, 16]. Thus, any information relayed among V is treated as trustworthy.

The hardware requirements for PEDRO are minimal; all entities, V and P, utilize off-the-shelf mobile devices without further hardware modifications. More specifically, both P and V have the WiFi capability[1], and V is additionally equipped with GPS. Furthermore, each entity has a public and private key pair. An entity uses a private key to sign its messages so that other entities can track its messages and establish trust relations over time using the public key. As such, the entity's public key serves as its pseudonym and can be reset, which requires re-establishing the trust relationship with other entities.

For all wireless communications, we assume verifiers and the prover can measure RTT using line-of-sight measurements. Additionally, PEDRO is only concerned with verifying the mobility of provers within its own wireless range. Without loss of generality, we consider the 2-D Cartesian coordinate system for simplicity; our method can be easily extended to 3-D cases.

2.2 Threat Model

We envision an attack scenario where an adversary deploys a malicious node on the roads or sidewalks (without any location restrictions) to broadcast false messages. An adversarial node has at least the same capabilities as a true pedestrian, except for *mobility*, i.e., the adversarial node is physically immobile. The adversary is fully aware of the verification protocol, and we do not impose any restrictions on their knowledge. It can also accurately measure the location and velocity of V at any given time instant. However, we assume that the different adversarial nodes do not collude.

The adversary can launch passive and active attacks. In the passive attack setting, the adversary is stationary and conforms with the protocol; it does not alter the RTT measurements. In the active attack setting, the adversary is also stationary and can arbitrarily alter RTT measurements aiming to be recognized as a mobile pedestrian. Under both attack scenarios, the adversary is not able to impersonate a trusted prover because it would require accessing their private key used for signing the safety broadcasts.

---

[1]We use WiFi that is most widely available on mobile devices, but PEDRO is not limited to a specific wireless technology.
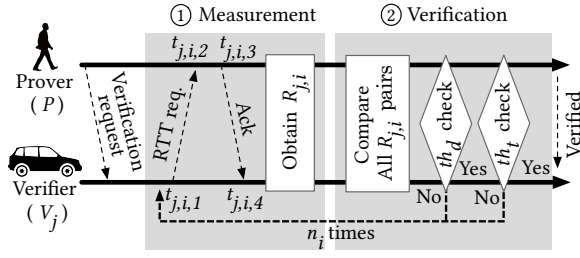
Figure 2: PEDRO's two stage verification protocol.

3 PEDRO PROTOCOL

This section presents the protocol of PEDRO. As illustrated in Figure 2, P initiates the verification request as V enters its wireless range. The protocol consists of a sequence of cycles; each cycle involves the Measurement and Verification stages between a single prover P and a single verifier V_j. The prover, P, repeats the cycle of two stages or verification instance (denoted as i) n_i times until the verification succeeds, or until V_j exits its wireless range. If P fails to be verified by V_j, then it initiates the verification again with the next passing verifier V_{j+1}. The remainder of this section details each stage of PEDRO's protocol.

3.1 Measurement

In the Measurement stage, V estimates the location region of P by measuring the distance between itself and P by measuring the RTT of a WiFi message. As illustrated in Figure 2, V first initiates the RTT measurement by sending the RTT request message to P at time $t_{j,i,1}$. As soon as P receives this message at $t_{j,i,2}$, it acknowledges back to V at time $t_{j,i,3}$, which is then received back by V at $t_{j,i,4}$. Then, the RTT is defined as the time difference measured by V, i.e., $RTT = t_{j,i,4} - t_{j,i,1}$. Note that this RTT measurement includes the time taken by P to respond to the RTT request, i.e., $t_{j,i,3} - t_{j,i,2}$, which is not proportional to the distance between V and P and thus ideally should be removed by immediately acknowledging the RTT request or measuring the response time and reporting it to V. However, since P has not been verified yet, V cannot rely on the response time reported by P. In addition, since smartphones are not equipped with a real-time operating system, time measurement of P would not be precise either, even if it is a legitimate prover. Our experiments in Section 4.1 show that modern smartphones are capable of almost immediately respond to the RTT request unless they intentionally delay their response. Therefore, for practicality reasons, we assume $t_{j,i,3} - t_{j,i,2} \approx 0$.

Based on the measurement, the distance between the two entities, denoted as $d_{j,i}$, can be calculated as $d_{j,i} = RTT/2 \times c$, where c is the speed of light constant. Assuming no RTT and GPS errors, if P immediately responded to the RTT request, we can conclude that P is on the rim of a circle with a radius of $d_{j,i}$ that is centered at $o_{j,i}$, which is the location of V obtained from its GPS readings. On the other hand, if P arbitrarily delays the time-to-respond to send the acknowledgement, i.e., $t_{j,i,3} - t_{j,i,2} > 0$, P can be located anywhere within the circle. Considering both cases, we consider that P can be anywhere inside the region, and we define this circle as P's *constrained region*, or $R_{j,i}$. Note that the location of P at $t_{j,i}$

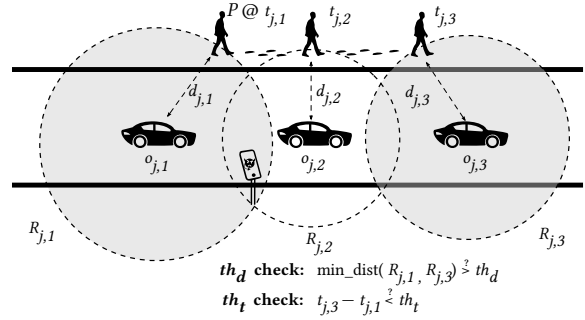
Figure 3: Verification stage of PEDRO with $n_i = 3$. The prover must meet two requirements to be verified: th_d and th_t check.

is bounded by $R_{j,i}$, because P cannot arbitrarily shorten $d_{j,i}$ due to the physical constraint of the wireless signal (P cannot reduce $t_{j,i,3} - t_{j,i,2}$ to below 0). During this measurement stage, when V_j obtains a constrained region at time $t_{j,i}$, the center, radius, and timestamp information is shared with all the nearby verifiers in V for the following Verification stage.

3.2 Verification

After completing the Measurement stage, V proceeds to the Verification stage for verifying the mobility of P. Figure 3 illustrates the verification process with $n_i = 3$ as V passes P, which is moving within its wireless range. At every time instance, i, V_j obtains $d_{j,i}$ from the measurement stage. It uses this value to construct a constrained region $R_{j,i}$, resulting in a series of $R_{j,i}$ for each P. The core idea underlying the verification protocol is that if two $R_{j,i}$ (not necessarily consecutive) do not overlap, then P must have moved between the two measurement instances.

Recall that an adversary can make RTT arbitrary longer, but it is impossible to make it shorter than the ground-truth value; by manipulating the RTT value, the attacker can make a certain $R_{j,i}$ only larger. If two the regions overlap (e.g., $R_{j,1}$ and $R_{j,2}$), even if their centers are far apart, it is a possible that P is a stationary node located in their overlapping area (i.e., stationary node in Figure 3). On the other hand, if the two regions do not overlap, (e.g., $R_{j,1}$ and $R_{j,3}$), it implies that P is a mobile node because the regions represent the maximum perimeter that P can be located at each time point since P cannot intentionally reduce the radius of $R_{j,i}$ as previously mentioned. Therefore, P must have moved to be included in two non-overlapping boundaries at different times. Leveraging this physical constraint sufficiently verifies the mobility of P.

This non-overlap condition, however, is not enough to ensure mobility in the real-world due to possible measurement errors, especially in the verifier's location; the measured region $R_{j,i}$ might be smaller than the ground truth, which results in false positive verification instances. We address this problem by two introducing two conditions for non-overlap. First, the timestamps of two consecutive $R_{j,i}$ must be less than the time threshold th_t, (i.e., $t_{j,k} - t_{j,i} < th_t$) where $k > i$. This condition prevents stale measurements from being used in the verification. Second, for P to be verified, it must have at least one $R_{j,i}$ pair with its minimum distance greater than

the distance threshold, th_d. In other words, between the $R_{j,i}$ and $R_{j,k}$ with $k > i$, the minimum distance (i.e., min_dist($R_{j,i},R_{j,k}$)) must be greater than th_d:

$$\sqrt{(o_{j,i,x} - o_{j,k,x})^2 + (o_{j,i,y} - o_{j,k,y})^2} - d_{j,i} - d_{j,k} > th_d, \quad (1)$$

where $o_{j,i,x}$ and $o_{j,i,x}$ represents the x and y coordinates of $o_{j,i}$, respectively. The two conditions are checked every time new $R_{j,i}$ is obtained, and P is verified only if it meets the above requirements. Imposing these allows the verifiers to only verify recently moved pedestrians under the RTT as well as GPS measurement errors.

Generally, PEDRO can thwart potential attacks from passive and active attackers, described in Section 2.2, with a high success rate. In some rare cases, however, decreased $d_{j,1}$ or $d_{j,3}$ due to RTT error or enlarged Euclidean distance due to GPS error in Equation 1 may cause the minimum distance to exceed the distance threshold th_d, even when passive attackers report RTT measurements without altering. As for active attacks, altered RTT measurements $d_{j,i}$ cannot be smaller than the actual (true) distance because the active attacker can make RTT longer by delaying its response to verifier's RTT request, but cannot make it shorter, which will require impractically accurate prediction of the arrival of RTT requests. Therefore, for any $R_{j,i}$ pairs, the measured minimum distance after altering is always smaller than that without altering, meaning that active attacks achieve a lower attack success rate than passive attacks. We will evaluate the robustness of PEDRO against the passive attacks in Section 4.2.2.

4 EXPERIMENT AND EVALUATION

In this section, we evaluate the overall performance of PEDRO as well as its robustness against the attack scenarios of Section 2.2. The experiments answer the following questions:

(1) *How reliable is RTT-based ranging using COTS devices for distance estimation?*
    We report the real-world distribution of the RTT errors between a moving pedestrian and a moving vehicle using Pixel Android phones. Our results show a mean error of 0.21 m and a standard deviation of 1.87 m of the RTT-based distance compared to the ground truth distance. This small error demonstrates the reliability of using COTS devices in distance estimation.

(2) *What are the optimal thresholds for PEDRO that balance usability and security properties?*
    We developed a simulator to investigate the impact of different road factors on the time it takes for moving prover to obtain two minimum constrained regions. We use this result to determine the time threshold, th_t, that maximizes the robustness of our verification protocol. Based on the result, we evaluate the overall security of the verification process and find optimal th_d, by obtaining the *Equal Error Rate (EER)*, which represents the intersection between FAR (False Acceptance Rate) and FRR (False Rejection Rate) intersects.

(3) *What is the real-world performance of PEDRO when employing the optimal thresholds?*
    We assess the usability of PEDRO using a real-world case study between two moving nodes with two Pixel 2 devices.
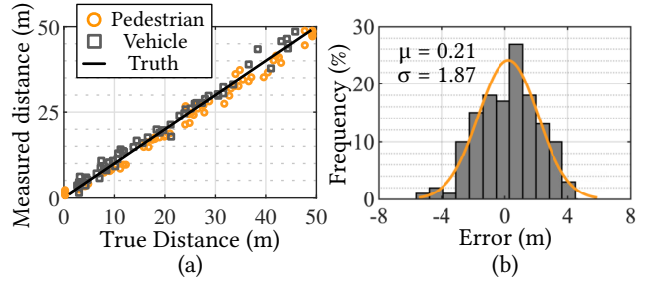
Figure 4: (a) RTT based distance measurement of moving pedestrian and vehicle. (b) Error distribution and its Gaussian fitted model with mean (μ) of 0.21 m and standard deviation (σ) of 1.87 m.

Our results show that the verification is successful with one verifier within 8 seconds. Further, the results from the case study are consistent with the simulation results for the same conditions. This finding suggests that the simulation platform is representative of the real-world performance of PEDRO.

4.1 RTT Error Model

Because PEDRO uses COTS mobile devices, RTT errors dominate its verification performance as well as the tightness of threshold values, th_d and th_t. To empirically investigate the RTT errors, we conduct real-world experiments to measure RTT-based distance on moving devices with different relative speeds representing pedestrian and vehicle. We implement the Measurement stage as an Android application on: Google Pixel 2 (Android 9.0 on a 2.35-GHz processor) and Pixel 3 (Android 9.0 on a 2.5-GHz processor). The application leverages the Wi-Fi Aware protocol; the prover acts as an active publisher while the verifier establishes the connection as a passive subscriber. We measure the ground truth distance between the two devices using a BOSCH GLM400CL laser range finder. We perform the measurements with the devices moving at the speed representing a pedestrian (1.5 m/s) and a vehicle (6.7 m/s). Figure 4(a) illustrates the measured RTT-based distance with respect to the ground truth distance. The result shows that the error of the RTT-based distance measurement is within 3 m. We observe that the error values remain similar regardless of the distance separating the nodes. This distribution of this error be modeled as a Gaussian fitted model with a mean (μ) of 0.21 m and standard deviation (σ) of 1.87 m, as shown in Figure 4(b). We use this error model in the following experiments.

4.2 Distance and Time Thresholds

We developed a simulation framework to model the performance of PEDRO under different conditions. This framework incorporates the RTT error model from the previous section and represents the GPS error as a Gaussian distribution with standard deviation (σ) of 1.2 m [12]. We use this framework to estimate the time required to verify the node mobility, which we use to derive the time threshold th_t. Next, we model an attacker within the platform to derive the distance threshold, th_d, which minimizes the EER.
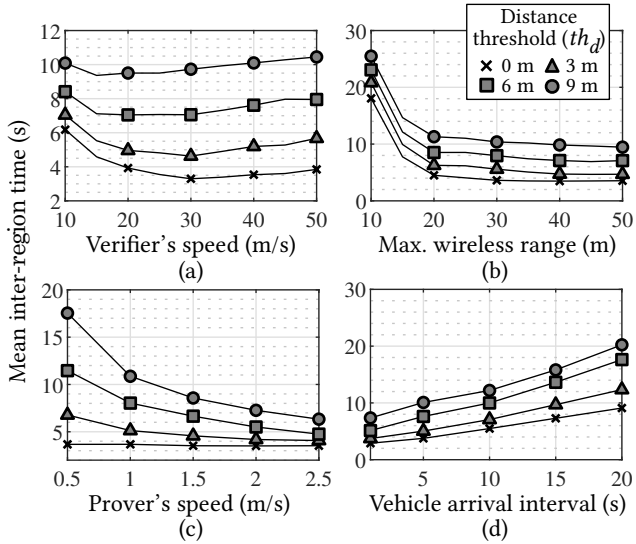
Figure 5: Mean inter-region time under varying (a) verifier's speed, (b) maximum wireless range, (c) prover's speed and (d) vehicle arrival interval with different th_d.

4.2.1 Verification Time.

We first investigate how different road conditions (i.e., verifier and prover's moving speed, maximum wireless range, etc.) affect the verification stage of a mobile prover. We quantify the performance of the verification through the *inter-region time*, defined as the time for a moving prover to obtain a pair of constrained regions with a minimum distance greater than th_d. We simulate a straight two-lane road while the prover (located within 2–5 m away from the road) is either traveling with its direction along or against the verifier. Additionally, the verification instance, i, is set to 1 second.

As illustrated in four plots in Figure 5, generally, as distance threshold th_d increases, the mean inter-region time increases due to the greater distance that the prover has to move to get verified. Also, we observe that the inter-region time decreases when the prover (1) moves faster, (2) is seen more frequently, and (3) is visible from further distances. In these circumstances, the verification becomes easier by taking short time since the prover will be more visible. As the verifier's speed varies from 10 m/s to 30 m/s, the inter-region time decreases as shown in Figure 5(a). Specifically, when $th_d = 0$, the mean inter-region time reduces from 6.1 s to 3.3 s because faster verifier speed leads to more rapidly generated constrained regions. However, starting at 30 m/s, the inter-region time starts to increase because the verifier passes by the prover too quickly and is unable to obtain enough number of regions needed for verification. In terms of varying maximum wireless range, a higher range reduces the inter-region time ranging from 25.4 s down to 3.4 s as shown in Figure 5(b). This is because as the wireless range becomes higher, verifiers can obtain faster as well as greater number of constrained regions compared to lower range. Figure 5(c) illustrates the impact of different prover's speed ranging from 0.5 m/s to 2.5 m/s. When $th_d = 0$, varying the prover's moving speed does not significantly affect the inter-region time due to the shorter distance that the prover has to move. However, as th_d increases, the slower prover
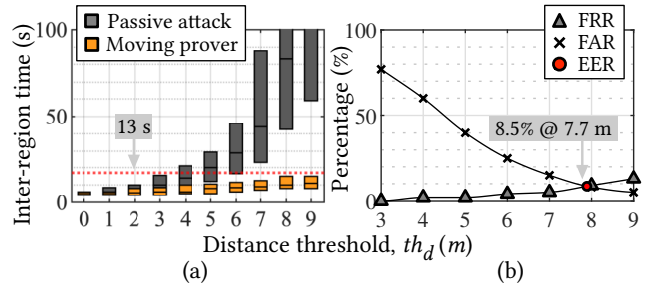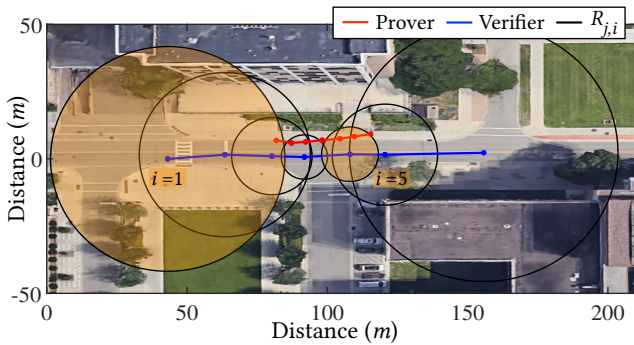
exhibits greater mean inter-region time due to the long distance it has to travel. In Figure 5(d), we illustrate the effect of verifier arrival interval. When interval is set to 1 s, which means verifiers are approaching the prover every 1 s, the prover's inter-region time exhibits less than 10 s. Comparatively, higher arrival interval leads to higher time due to less number of verifiers within same period of time, leading to less number of constrained regions. Specifically, at 20 s interval, the prover's inter-region time exhibits 20.4 s when $th_d = 9$. Nevertheless, under different road factors, the moving verifiers can obtain a pair of regions exceeding distance threshold of 9 m under 26 s.

4.2.2 Attack Robustness.

We evaluate the robustness of PEDRO against the passive attack scenario. In the passive attack, the adversary is fixed to a stationary location and attempts to be verified while complying with the protocol (does not add any arbitrary timing delay during the Measurement stage). We first simulate this attack scenario 1000 times under different road factors and obtain its inter-region time. This will allow us to derive the time threshold th_t by comparing it against the inter-region of the moving prover presented in Section 4.2.1. Figure 6(a) illustrates the distribution of the inter-region times of the passive attacker and mobile prover with respect to varying th_d. When th_d is less than 5 m, the two distributions do not exhibit much difference in their inter-region times due to GPS and RTT errors that allow adversarial prover to quickly obtain pair of constrained region within small th_d. If th_d increases above 6 m, the two distributions exhibit more significant differences because the adversarial prover requires greater number of verifiers as well as constrained regions to leverage the noise/errors in its favor. From this inter-region time, we can choose th_t, which effectively distinguishes between the moving prover against the passive attacker. We experimentally choose 13 s and plot the FAR as well as FRR to obtain EER as shown in Figure 6(b). Generally, low ERR represents higher accuracy of distinguishing legitimate pedestrians over adversarial nodes. When $th_t = 13$, the EER of 8.5% is achieved with th_d of 7.7 m.

4.3 Real-world Case Study

Using the derived thresholds, we conduct a case study under real road conditions to evaluate the usability and feasibility of PEDRO. We utilize two Pixel 2 devices where one device is carried by moving pedestrian (prover) and the other is deployed in the vehicle (verifier). The Measurement stage is performed using the same Android

Figure 6: (a) Distribution of inter-region time of moving prover and passive attacker. (b) EER of verification with $th_t = 13$.

Figure 7: Real-world experiment result. Constrained region pairs with $i = 1$ and $i = 5$ meets the two verification requirements and the prover is verified.

application as in Section 4.1. The maximum wireless range is set to 50 m and the time between each verification instance, i, is set to 1 s. We conduct five experiments where the pedestrian moves at 1.2 m/s, and the vehicle travels at the speed of 8.4 m/s on average. Figure 7 illustrates one attempt of the verification. As the verifier and the prover move along the indicated paths (GPS-obtained) in the same direction, the verifier obtains total of 7 constrained regions ($n_i = 7$). As each region is obtained, the verifiers check the two threshold requirements on all pairs of boundaries. In this case, the verifier verifies the prover by observing a region pair with $i = 1$ and 5; the minimum distance between the two exhibits 13.1 m, which exceeds the th_d of 7.7 m and the differences in their timestamps falls under $th_t = 13$ s. In all five cases, the moving pedestrian was all verified through single verifier. The average verification time is 7.4 s, which shows that the PEDRO is able to quickly verify the moving pedestrian even under road conditions where there are not many readily available verifiers. Furthermore, we simulate this scenario (identical road factors) 1000 times and obtained closely matching verification time of 7.9 s. This suggests that our simulation platform represents the real-world performance of PEDRO. Note that the result does not imply that a single verifier should be around the prover for more than 7 s as the decision can be collectively made by more than one verifier.

5 RELATED WORK

Several prior studies have proposed various methods to verify the location or motion of the sender (prover) to be used in many safety-critical scenarios. In order to prevent location spoofing, at least one node needs to be constantly moving, while the prover has no knowledge of the moving verifier. Similar to our work, Capkun et al. [5] proposes to leverage RTT distance measurement from at least three verifiers to cooperatively verify location claims. In motion verification domain, verifiers verifies the location, speed and direction of a moving prover or a sequence of claimed locations. For example, Schafer et al. [15] verifies the claimed movements of a mobile prover by measuring wireless signal's frequency shifts in verifier's side caused by the Doppler effect. With at least three verifiers in different locations, they can uniquely identify the position, speed and direction of any prover that cannot be faked. Based on the difference in time of arrival (DToA) from a mobile prover to a

static verifier, Schafer [14] verifies a sequence of one mobile node's location claims with at least three static verifiers.

6 CONCLUSION

In this work, we propose PEDRO, a mobility verification protocol for pedestrians, utilizing commodity devices. Without tight clock synchronization, we leverage RTT of ubiquitously available wireless signal to obtain sequence of constrained regions of the prover over time. Under RTT and GPS errors, PEDRO's two threshold requirements only verify the moving prover while effectively rejecting stationary attacks. With realistic simulation framework as well as through real-world case study, we show that PEDRO can achieve 8.5% EER against malicious attackers while maintaining relatively usable verification time of under 8 seconds.

REFERENCES

[1] Fabio Arena et al. 2020. V2X Communications Applied to Safety of Pedestrians and Vehicles. *Journal of Sensor and Actuator Networks* 9, 1 (2020).
[2] Richard Baker and Ivan Martinovic. 2016. Secure location verification with a mobile receiver. In *ACM Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC)*. 35–46.
[3] K. Bian et al. 2017. Security in Use Cases of Vehicle-to-Everything Communications. In *IEEE Vehicular Technology Conference (VTC-Fall)*. 1–5.
[4] Srdjan Capkun et al. 2008. Secure location verification with hidden and mobile base stations. *IEEE Transactions on Mobile Computing* 7, 4 (2008), 470–483.
[5] Srdjan Capkun and J-P Hubaux. 2005. Secure positioning of wireless devices with application to sensor networks. In *IEEE Computer and Communications Societies (INFOCOM)*, Vol. 3. 1917–1928.
[6] Jin Cui et al. 2019. A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. *Ad Hoc Networks* 90 (2019), 101823.
[7] B. M. ElHalawany et al. 2019. Physical-Layer Security and Privacy for Vehicle-to-Everything. *IEEE Communications Magazine* 57, 10 (2019), 84–90.
[8] E. C. Eze et al. 2014. Vehicular ad hoc networks (VANETs): Current state, challenges, potentials and way forward. In *International Conference on Automation and Computing (ICAC)*. 176–181.
[9] Jun Han et al. 2017. Convoy: Physical context verification for vehicle platoon admission. In *International Workshop on Mobile Computing Systems and Applications (HotMobile)*. 73–78.
[10] Monowar Hasan et al. 2020. Securing Vehicle-to-Everything (V2X) Communication Platforms. *IEEE Transactions on Intelligent Vehicles* 5, 4 (2020), 693–713.
[11] C. Li et al. 2018. V2PSense: Enabling Cellular-Based V2P Collision Warning Service through Mobile Sensing. In *IEEE International Conference on Communications (ICC)*. 1–6.
[12] Kaisen Lin et al. 2010. Energy-Accuracy Trade-off for Continuous Mobile Device Location. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 285–298.
[13] Kasper Rasmussen et al. 2010. Realization of RF Distance Bounding. In *USENIX Security Symposium*. 389–402.
[14] Matthias Schäfer et al. 2015. Secure track verification. In *IEEE Symposium on Security and Privacy (S&P)*. 199–213.
[15] Matthias Schäfer et al. 2016. Secure motion verification using the doppler effect. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*. 135–145.
[16] Mingshun Sun et al. 2017. A data trust framework for VANETs enabling false data detection and secure vehicle tracking. In *IEEE Conference on Communications and Network Security (CNS)*. 1–9.
[17] Mingshun Sun et al. 2020. SVM: secure vehicle motion verification with a single wireless receiver. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*. 65–76.
[18] Harsha Vasudev et al. 2020. Secure message propagation protocols for IoVs communication components. *Computers & Electrical Engineering* 82 (2020), 106555.
[19] X. Wu et al. 2014. Cars Talk to Phones: A DSRC Based Vehicle-Pedestrian Safety System. In *IEEE Vehicular Technology Conference (VTC-Fall)*. 1–7.