VET: Autonomous Vehicular Credential Verification using Trajectory and Motion Vectors

Ebuka Oguchi and Nirnimesh Ghose

School of Computing, University of Nebraska–Lincoln, NE, USA
eoguchi2@huskers.unl.edu, nghose@unl.edu

Abstract. There has been significant progress in autonomous vehicles: autonomous automobiles, unmanned aerial vehicles, and many more are improving our quality of life and making it safer. However, this also opens up a new attack paradigm: now, an adversary can take control of these autonomous systems to cause life-threatening scenarios. It becomes possible due to the broadcast nature of wireless communication, which connects autonomous vehicles in an ad-hoc network. Traditional crypto-algorithms alone cannot tackle the problem as the crypto-credentials can be compromised or even issued to adversarial parties. We propose VET: a framework that verifies the veracity of the crypto-credentials by authenticating them against physical trajectory and motion vectors (TMVs). The verifier implements a location and motion-based authentication to verify the crypto-credentials based on the acceptability of claimed TMVs against randomly estimated TMVs. This prevents any adversary from remotely injecting spoofed messages when it is not physically present. We formally analyze the correctness and robustness of VET using matching conversations. Finally, we attest to the findings of the theoretical analysis using an experimentally analyzed VET on the USRP platform. Our experiments show that VET has 97% true positives when operating without an adversary. Also, VET can detect advanced remote adversaries with 99.9% who is capable of manipulating signals with absolute channel knowledge.

Keywords: Location and Motion based-Authentication · Autonomous VANET · Frequency-of-Arrival · Direct Location Estimation

1 Introduction

Various autonomous systems such as connected autonomous vehicles (CAVs) and [46], unmanned aerial vehicles (UAVs) [32] are inter-connected via ad-hoc networks for efficient implementation. In CAVs, safe and efficient traffic control can be improved through V2X communications [13]. UAV swarm motion [21], efficient control by a ground station [18], geofencing, and midair collision avoidance [58] is efficiently performed when UAVs communicate. Thus, verifying the authenticity and integrity of the CAV and UAV communication is important. For example, a rogue CAV can broadcast spoofed messages, such

as moving slowly, which will cause more traffic to be routed to an alternate route. Or a CAV can broadcast a spoofed emergency braking message due to a fake accident, disrupting safe traffic motion. Similarly, in UAVs, a misbehaving UAV can disrupt swarm motion or break the geofencing to move into no-fly zones. Thus, verifying the message integrity and source authentication of messages injected by the nodes in these ad-hoc networks becomes paramount.

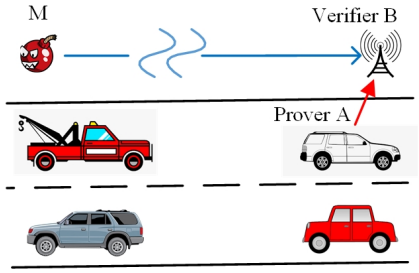


Fig. 1. The verifier B performs verification of a prover A's credentials based on motion state vectors in the presence of an attacker M capable of spoofing trajectories.

Classic cryptographic solutions such as digital signatures or message authentication codes can provide source authentication and message integrity verification [40]. However, this does not prevent a remote adversary from injecting spoofed messages through compromised infrastructure [1, 9, 50] either by utilizing compromised credentials [20] or legitimately issued credentials. For CAVs, a certification authority can certify the public key infrastructure (PKI) credentials for each vehicle registered [20]. UAVs are issued credentials to control and blacklist UAVs; however, these credentials can be compromised to spoof fake motion data to break the geofencing [55]. It is crucial to verify the vehicles' physical veracity and credentials. By this, we mean that *if a vehicle claims to be at a certain location, is it physically there?* This imposes a more stringent restriction on an adversary to be in the physical vicinity where it is attempting to inject messages. This provides security comparable to computational security by imposing strict restrictions on the capability to defeat the system.

Moreover, if an adversary is in the path of traffic, causing disruption, it will also be affected. To perform such verification, we propose to utilize a novel *location and motion-based authentication* architecture. Initially, all the credentials are given limited connectivity until their trajectory and motion claims are authenticated before each session.

Researchers have performed veracity verification of messages by comparing them to estimated data from the wireless physical layer. The state-of-the-art methods for position and velocity estimation utilize time difference of Arrival (TDoA) [6], frequency of arrival (FoA) [34, 43], received signal strength (RSS) [3, 12, 33], or angle of arrival (AoA) [52]. However, they have several limitations, like inaccurate measurement, unreliable, and inefficiency. Further, some of these works considered more than one verifier [34, 52]. If they considered a single verifier, the verification needed fixed reflectors for a rich multipath environment [3, 12, 43]. Also, these are vulnerable to several attacks, especially when the location of the verifier is known [11] where the rogue attacker can claim to be the verifier and receive relevant information from the incoming prover. Different approaches use out-of-band methods for tracking and estimating the position

and velocity of vehicles like the use of light detection and ranging (LiDaR) [26], cameras [54], radar [2], etc. but these out-of-band methods have been known to be very expensive, require extra external hardware, and are easily compromised by a remote attacker [27]. It still leaves a void to develop a method that can perform *reliable veracity verification* with a single verifier and *no assumptions of the wireless environment*. This will make the developed method agnostic to the application and can be implemented in Non-Line-of-sight (NLoS) environments of CAVs and the Line-of-sight (LoS) environment of UAVs.

We propose a framework for credential **VE**rification using **T**rajectory and Motion Vectors (TMVs) for autonomous vehicles. **VET** utilizes a location and motion-based authentication strategy to grant or restrict access to a prover. As compared to the state-of-the-art [3, 34, 43, 52], **VET** is capable of performing verification with a single verifier, which can be either stationary or in motion with no requirements on the environment. This applies to Line-of-Sight (LoS) communication scenarios, such as UAVs. This strategy prevents an adversary from gaining control of the communication system as we do not assume any implicit trust within the communication range of the network.

Moreover, the verifier randomly collects the signals from a prover to estimate the TMVs. Hence making it impossible for an advanced adversary to inject a targeted spoofed trajectory even by performing signal-level manipulation. Consider Fig. 1, a Prover A broadcasts its messages containing claimed TMVs utilizing the secret credential received by the verifier B. The verifier further captures the PHY-layer properties, such as FoA, for estimating the TMVs. We propose to estimate the TMVs for two types of signals, one intended for B and the other intended for other entities within the communication range. It should be noted that the verifier B is not required to communicate with other entities in the range. The verifiers accept A's credentials if the claimed TMVs are within the acceptable range of the estimated TMVs. To the best of our knowledge, this is the first work to incorporate location and motion-based authentication to authorize and authenticate vehicles based on the integrity of TMVs.

Main Contributions: Our major contributions are as follows.

- We develop a location and motion-based authentication for vehicular network security protocol that can prevent the exploitation of valid or compromised credentials from injecting spoofed messages. As compared to state-of-the-art [3, 6, 12, 33, 34, 43, 52], VET, in addition to verifying the credentials efficiently, also verify the veracity of the source's physical location using a single verifier agnostic to wireless channel conditions. Our protocol is immune to advanced signal manipulation attacks and scalable and interoperable.
- We perform the verification by *comparing the claimed TMVs with randomly estimated TMVs* at different times. Thus, an adversary who is unaware of the time to perform a signal manipulation attack for spoofing ghost TMVs is detected. Further, we utilize a frequency of arrival after compensating for wireless channel effects to compute both position and velocity. This makes it applicable to both LoS and NLoS real-world scenarios.

- We formally analyze the security of our protocol against various active adversaries with advanced signal manipulation techniques based on the principles of matching conversations [7]. We prove negligible success probability for an advanced adversary with increasing TMVs.
- We further performed various experimental tests and evaluations of the performance of our protocol on the USRP platform. We showed a high true positive rate for an acceptable false positive rate when no adversary is present and proved the negligible success probability of adversaries with emulated experiments using real-world data.

Paper Organization: In Section 2, we discuss the prior art and contrast them with VET. Further, Section 3 presents the system and threat model with the preliminaries. VET: The protocol is presented in Section 4 with security analysis in Section 5 and experimental analysis in Section 6. Finally, concluding in Section 7.

2 Related Work

Security in ad-hoc autonomous systems ensures that the information has not been tampered with and is transmitted from an authenticated source. Prior works perform additional verification of crypto-credentials can be broadly classified as in-band or out-of-band (OOB). The in-band solutions performs either velocity or location verification [4–6, 10, 17, 19, 28, 30, 37, 38, 42, 43, 47, 52, 56]. Whereas, the OOB methods use Lidar [26], cameras [54], radar [2], etc. for verification. These methods are expensive, require extra external hardware, and are easily compromised by a remote attacker [27]. As VET is an in-band solution, we will discuss the details of in-band solutions.

Velocity Verification: Doppler shift measurement and Frequency Difference of Arrival (FDoA) were used for secure motion verification [38]. However, the solution is limited to a static and single attacker. In contrast, our VET is robust against multiple colluding moving attackers. Another method used an Angle of Arrival (AoA) with Doppler speed for motion verification utilizing a modified, extended Kalman filter framework [42]. However, this requires two verifiers with prior trust to perform the verification. Our framework uses a single verifier, but no trust is required when there are multiple verifiers. In the work by Ghose and Lazos, they used the Doppler spread to verify air traffic navigation, but they placed restrictions on the frequency the attacker can control [17]. However, this also requires very wide band transmissions, which is absent in existing ad-hoc networks in autonomous networks. Another solution relies on a single verifier, which can verify the motion state information of the moving prover [43]. They consider performing the verification based on the similarity of the Doppler shift observed on multipath communication between the prover and the verifier. In contrast, VET can perform verification even with only one Line-of-Sight (LoS) communication. Ad-hoc autonomous systems such as vehicle-to-everything (V2X) networks have physical layer challenges [59] that affect the uplink communications. The authors [45] performed real-world experiments on

physical layer security attacks on mmWave communication where the mmWave-based sensing method was used. In contrast, our method uses a location-based strategy to grant access to the incoming moving entity. Further, UAV is susceptible to various physical layer attacks [44], [23], [22] especially passive and active eavesdropping attacks, jamming attacks, and pilot contamination attacks.

Location Verification: The direct position estimation [4] has been used to estimate the expected position for the moving receiver and stationary transmitter. The differential Doppler is another method used for position determination which uses a two-step method of measuring the position at each receiver and estimating the position based on that measurement. The distance bounding techniques [19], [5], [10] ensure that the RTT(round trip time) for the information exchange is bounded, and it engages in a challenge-response protocol. The problem with this method is the communication is time-sensitive and requires a very accurate system for rapid bit exchange, and also, the communication distance is low [47]. Most other works on localization and secured verification in the vehicular network are passive, as their only objective is to ensure that motion state information is securely communicated to the verifier by guaranteeing the message is truthful. Some other active autonomous vehicle tracking attacks [30] and misclassification attacks [28]. Here, detection was done where they performed the move-in and move-out attacks to hijack a tracker and used Spatiotemporal inconsistencies to detect misclassification of objects. The authors provided a vision-based solution.

Like all existing works, VET prevents message injection attacks with valid credentials. However, in contrast to existing works, VET performs an intermingled verification of velocity and position tied to a valid trajectory. Here, the verifier and as well as the attacker can be either static or in motion. The attacker can also have multiple colluding entities. We do not assume any environmental requirement, which makes VET applicable to both LoS and NLoS communication models. This makes VET applicable for all the ad-hoc networks of autonomous vehicles. In Table 1, we compared VET to other works available for the security of VANET.

Method	**Approach**	**Moving Prover**	**Static/Moving Verifier**	**Static/Moving Colluding Attackers**
PoF [56]	Large-scale RF	✓	×	×
SVM [43]	FoA+AoA	✓	✓	×
Schafer et al. [38]	FoA	✓	×	×
Ghose et al. [17]	FoA	✓	×	×
Zhi et al. [45]	mmWave	✓	×	✓
Wiggle [14]	Challenge-Resp. Prot.	✓	×	×
PEDRO [57]	RTT+GPS	✓	×	×
Rony et al. [24]	FMCW Radar	✓	×	×
Tithi et al. [49]	Friendly Jamming	✓	✓	×
Vaas et al. [51]	Co-presence Verif.	✓	×	✓
Tirer et al. [48]	FoA+Loc. Est.	×	×	×
Sun et al. [41]	RF Based	×	×	×
VET (Ours)	FoA+Loc. Est.	✓	✓	✓

Table 1. Related work summary

3 Models

In this section, we first present the system model followed by the threat model for VET. We present Table 2, which summarizes the frequently used notations in this paper.

3.1 System Model

The Legitimate Prover (A)**:** The prover A has legitimate credentials, which can be either PKI credentials (pK, sK), or symmetric key credentials K. We assume that A uses an omnidirectional antenna to transmit wireless signals.

The Verifier (B)**:** The signal transmitted by A is received by the verifier B, when the prover is within the communication range. The trust is established by performing source authentication of the prover. We assume that there are one or more truthful verifiers X within the communication range. This is a valid assumption for CAVs, and there are other vehicles or roadside units or UAVs with more than one trusted controller and UAVs. However, these *verifiers do not require mutual trust. Each verifier performs VET independently and can broadcast a failure message in case of a failure.* It should be noted that an adversary can exploit to launch a denial-of-service; such an adversary can be manually removed. Also, this is orthogonal to VET presented in this paper.

3.2 Threat Model

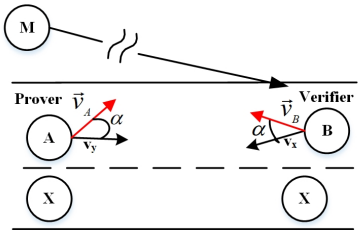


Fig. 2. The prover A attempts to authenticate with the verifier B in the presence of an adversary M and other entities X within the communication range.

We consider a Dolev-Yao attacker [15]. The adversary M has a valid credential, which can be either PKI credential (pk_M, sK_M) or symmetric credential K_M, and injecting its messages to disrupt the acceptable functionalities of a vehicular ad-hoc network. The attacker knows the locations of all the verifiers but does not physically control the verifier. M transmit a message with an intention for the verifier (B) to accept as the legitimate prover. M also knows all the channels between legitimate entities. The adversary can either be within the communication range of B or can compromise static wireless nodes connected to the internet to realize the attack. Finally, we do not put any restrictions on the motion of the adversary. Hence, the adversary can be either static or moving. The attack scenarios for this work are:

Remote Attacker: We consider an attacker located within the verifiers' communication range and attempting to inject his messages without intentional modification of PHY-layer data.

Remote Advanced Attacker: In addition to the capability of a remote attacker, the advanced attacker can intentionally modify the transmitted PHY-layer level wireless signal.

Notation	Description
A	Prover
B	Verifier
X	One or more truthful verifiers within the communication range
M	Adversary
$\mathcal{L}$	Claimed trajectory for k time-ordered locations for initial TMVs verification
$\mathcal{V}$	Claimed motion for k time-ordered locations for initial TMVs verification
$\mathcal{L}'$	Estimated trajectory for k time-ordered locations for initial TMVs verification
$\mathcal{V}'$	Estimated motion for k time-ordered locations for initial TMVs verification
$\tilde{\mathcal{L}}'$	Estimated trajectory for non A-to-B communication for final TMVs verification
$\tilde{\mathcal{V}}'$	Estimated motion for non A-to-B communication for final TMVs verification
$\mathcal{M}$	Set of transmitted k messages $(\{(m(1), t(1)), \ldots, (m(k), t(k)\})$
$\widehat{\mathcal{M}}$	Captured k messages
$\mathcal{F}$	Frequency of Arrival
$\tilde{\mathcal{F}}$	Frequency of Arrival for non A-to-B communication, where $t(i) \neq t'(i)$
ϵ	Acceptable error for location
μ	Acceptable error for velocity
Π_A	Prover oracle
Π_B	Verifier oracle
Π_X	Entity X oracle
Π_M	Adversary oracle
tx	Message transmissions
p^k	Probability of success of Adversary with No -Matching
d_{MB}	Distance between M and B
h_{MB}	Wireless channel between M and B
h_{MX}	Wireless channel between M and X
k	Number of trajectory data points
$RMSE(\cdot)$	Normalized root mean square error function for TMV Verification.

Table 2. Table of Notations

4 VET: Credential Verification using Trajectory and Motion Vectors

We present a secured, in-band vehicular access control method to verify the authenticity and integrity of a set of messages transmitted from a legitimate vehicle A at the verifier. B implements a location based strategy for verification, where B does not trust A in the start of the communication. For the verification, B generates a set of trajectory and motion vectors from the carrier frequencies.

4.1 Vehicular Motion State Verifier

The basic idea is for verifier B to authenticate the claimed trajectory observed for a prover A via a location-based authentication strategy. We exploit the characteristics of the direct position and velocity estimation via the arrival frequency to verify the prover. The protocol is presented as without generality between a prover A and a verifier B. It should be noted that simultaneous runs of the protocol can be initiated between the same prover and different verifiers as the prover A can simultaneously communicate with various entities. First, we describe TMVs utilized to develop VET:

Trajectory and Motion Vectors: The trajectory $\mathcal{L}$ of a moving vehicle is defined as k time-ordered locations

$$\mathcal{L} = \{(\ell(1), t(1)), (\ell(2), t(2)), \ldots, (\ell(k), t(k))\}, \tag{1}$$

where each location $\ell(i) = (x(i), y(i))$ is the geospatial location coordinate at time $t(i)$. Where $1 \leq i \leq k$ for $t(i)$ and $t(i) < t(j)$ for $i < j$. Further, the motion $\mathcal{V}$ is defined as k time-ordered locations in the same epoch

$$\mathcal{V} = \{(\overrightarrow{v}(1), t(1)), (\overrightarrow{v}(2), t(2)), \ldots, (\overrightarrow{v}(k), t(k))\}, \tag{2}$$

where $\overrightarrow{v}(i)$ is the velocity at time $t(i)$. These locations and velocities are obtained using the method described in Appendix A.

The protocol is initiated when the prover A is within the communication range of B. The prover A sends *Request to Authenticate* message with authenticated encryption using issued credentials. An authenticated encryption function $\mathrm{AE}(\cdot)$ utilizing the shared secret K [7]. This will guarantee the source's authenticity, message integrity, and confidentiality. When verifiers share a common secret, $\mathrm{AE}(\cdot)$ can be implemented as an encrypt-then-MAC operation. Whereas for the public key cryptographic scenario, $\mathrm{AE}(\cdot)$ can be implemented as a sign/encrypt/sign (or encrypt/sign/encrypt). Here, the credential can either be the actual one issued by a trusted authority or a pseudonym credential for preserving privacy. The verifier B provides the prover limited connectivity if the credentials are verified.

During the limited connectivity, the verifier B captures the message transmitted to it and extracts the claimed k TMVs $\mathcal{L}$ and $\mathcal{V}$, and estimated TMVs $\mathcal{L}'$ and $\mathcal{V}'$. First, B verifies the claimed and estimated using a root mean square error (RMSE) function. If successful, in the same time epoch, the verifier B captures the frequency of arrival (FoA) $\tilde{\mathcal{F}}$ for the messages transmitted by A but not intended for the B. From these FoA, the verifier B estimates TMVs $\tilde{\mathcal{L}}'$ and $\tilde{\mathcal{V}}'$, which are shifted in time as compared to claimed. Further, B maps the claimed TMVs to the same time as estimated TMVs using kinematic equations. The estimated and claimed TMVs are compared; if these are within the accepted errors, A's messages are accepted and granted full access. Formally, the vehicular motion state verification steps are:

1. **Initial Request :** Once the prover A is within the communication range of the verifier B. A transmits a request to authenticate $\mathrm{AE}_K(RTA)$ to the verifier B to join.
2. **Limited Access Connection:** After verifying the authenticity of A's credential K, B grants it *limited access*. During the limited access B captures k messages transmitted by A as $\mathcal{M} = \{(m(1), t(1)), \ldots, (m(k), t(k)\}$, containing claimed TMVs: velocity vectors $\mathcal{V} = \{(\overrightarrow{v}(1), t(1)), \ldots, (\overrightarrow{v}(k), t(k)\}$ and $\mathcal{L} = \{(\ell(1), t(1)), \ldots, (\ell(k), t(k))\}$. B also records the FoA $\mathcal{F} = \{(f(1), t(1)), \ldots, (f(k), t(k)\}$, and computes TMVs: velocity vectors $\mathcal{V}' = \{(\overrightarrow{v}(1)', t(1)), \ldots, (\overrightarrow{v}(k)', t(k)\}$ and $\mathcal{L}' = \{(\ell(1)', t(1)), \ldots, (\ell(k)', t(k))\}$. It should be emphasized that the verifier has not yet acknowledged any of

the critical directives. It is only used for the verifier to extract the relevant trajectory information of the incoming prover for verification.

3. **Initial TMVs Verification:** B computes the Root Mean Square Error (RMSE) of location as:

$$RMSE(\ell(i), \ell(i)') = \sqrt{\frac{\sum_{i=1}^{k} \left(\frac{\ell(i)-\ell(i)'}{\ell(i)'}\right)^2}{k}}.$$

The RMSE of velocity is as follows;

$$RMSE(\overrightarrow{v}(i), \overrightarrow{v}(i)') = \sqrt{\frac{\sum_{i=1}^{k} \left(\frac{\overrightarrow{v}(i)-\overrightarrow{v}(i)'}{\overrightarrow{v}(i)'}\right)^2}{k}}.$$

B then performs verification:

$$RMSE(\ell(i), \ell(i)') \stackrel{?}{\leq} \epsilon \quad \forall\ 1 \leq i \leq k,$$

$$RMSE(\overrightarrow{v}(i), \overrightarrow{v}(i)') \stackrel{?}{\leq} \mu \quad \forall\ 1 \leq i \leq k,$$

where RMSE$(\cdot)$ is a normalized root mean square error function, and ϵ and μ are the acceptable error. If B passes the check, A grants B partial access and accepts $\mathcal{M}$ as valid. Else, B disregards $\mathcal{M}$ and terminates the connection of A. Also, B broadcasts a signal notifying FAILED authentication of A.

4. **Estimating TMVs for non A-to-B communication:** During the same time epoch, verifier records FoA $\tilde{\mathcal{F}} = \{(\tilde{f}(1), t'(1)), \ldots, (\tilde{f}(k), t'(k)\}$, where $t(i) \neq t'(i)$, from the transmissions (tx) from A not intended for B. Next the verifier B computes corresponding velocity vectors $\tilde{\mathcal{V}}' = \{(\overrightarrow{\tilde{v}'}(1), t'(1)), \ldots, (\overrightarrow{\tilde{v}'}(k), t'(k)\}$, and trajectory vectors $\tilde{\mathcal{L}}' = \{(\tilde{\ell}'(1), t'(1)), \ldots, (\tilde{\ell}'(k), t'(k))\}$.
5. **Interpolating Claimed TMVs:** The estimated TMVs $\overrightarrow{\tilde{v}'}(i)$, $t'(i)$ and $(\tilde{\ell}'(i), t'(i))$ are interpolated to synchronize with claimed TMVs $(\overrightarrow{v}(i), t(i))$ and $(\ell(i), t(i))$ using cubic spline interpolation methods for a timeseries data [29].
6. **Final TMVs Verification:** The RMSE of location is calculated as:

$$RMSE(\ell(i), \tilde{\ell}'(i)) = \sqrt{\frac{\sum_{i=1}^{k} \left(\frac{\ell(i)-\tilde{\ell}'(i)}{\tilde{\ell}'(i)}\right)^2}{k}}.$$

The RMSE of velocity is computed as:

$$RMSE(\overrightarrow{v}(i), \overrightarrow{\tilde{v}'}(i)) = \sqrt{\frac{\sum_{i=1}^{k} \left(\frac{\overrightarrow{v}(i)-\overrightarrow{\tilde{v}'}(i)}{\overrightarrow{\tilde{v}'}(i)}\right)^2}{k}}.$$

Finally, B performs verification:

$$RMSE(\ell(i), \tilde{\ell}'(i)) \stackrel{?}{\leq} \epsilon \quad \forall\ 1 \leq i \leq k,$$

$$RMSE(\overrightarrow{v}(i), \overrightarrow{\tilde{v}'}(i)) \overset{?}{\leq} \mu \quad \forall\ 1 \leq i \leq k,$$

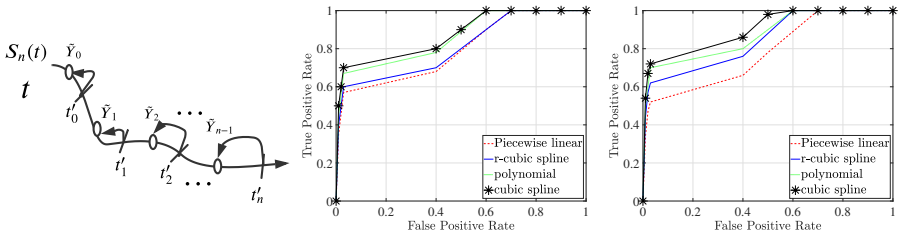
where RMSE$(\cdot)$ is a normalized root mean square error function, and ϵ and μ are the acceptable error. If B passes the check, A grants B full access and accepts $\mathcal{M}$ as valid. Else, B disregards $\mathcal{M}$ and terminates the connection of A. Also, B broadcasts a signal notifying FAILED authentication of A.

	A		B
Initialization:		$\xrightarrow{AE_K(RTA)}$	Grants limited Access
Captures Claimed Messages:	$\mathcal{M} = \{(m(1), t(1)), \ldots, (m(k), t(k)\}$	$\xrightarrow{AE_K(\mathcal{M})}$	$\widehat{\mathcal{M}}$, also capture $\mathcal{F} = \{(f(1), t(1)), \ldots, (f(k), t(k)\}$, extract $\mathcal{V} = \{(\overrightarrow{v}(1), t(1)), \ldots, (\overrightarrow{v}(k), t(k)\}$ $\mathcal{L} = \{(\ell(1), t(1)), \ldots, (\ell(k), t(k))\}$. compute $\mathcal{V}' = \{(\overrightarrow{v}(1)', t(1)), \ldots, (\overrightarrow{v}(k)', t(k)\}$ $\mathcal{L}' = \{(\ell(1)', t(1)), \ldots, (\ell(k)', t(k))\}$.
Initial TMVs Verification:	Partial access.		$RMSE(\overrightarrow{v}(i), \overrightarrow{v}(i)') = \sqrt{\frac{\sum_{i=1}^{k}\left(\frac{\overrightarrow{v}(i) - \overrightarrow{v}(i)'}{\overrightarrow{v}(i)'}\right)^2}{k}}$, $RMSE(\ell(i), \ell(i)') = \sqrt{\frac{\sum_{i=1}^{k}\left(\frac{\ell(i) - \ell(i)'}{\ell(i)'}\right)^2}{k}}$. $RMSE(\overrightarrow{v}(i), \overrightarrow{v}(i)') \overset{?}{\leq} \epsilon$, $RMSE(\ell(i), \ell(i)') \overset{?}{\leq} \mu$,
Computes TMVs's :	tx (Transmission not to B)	$\xrightarrow{Sniff\ Packets}$	$\tilde{\mathcal{F}} = \{(\tilde{f}(1), t'(1)), \ldots, (f\tilde{k}), t'(k)\}$, $t(i) \neq t'(i)$, compute $\tilde{\mathcal{V}}' = \{(\overrightarrow{\tilde{v}}(1), t'(1)), \ldots, (\overrightarrow{\tilde{v}}(k), t'(k)\}$, $\tilde{\mathcal{L}}' = \{(\tilde{\ell}'(1), t'(1)), \ldots, (\tilde{\ell}'(k), t'(k))\}$.
TMVs Interpolation:			From $t(i)'$ to $t(i)$ using cubic spline interpolation: $\mathcal{V} = \{(\overrightarrow{v}(1), t(1)), \ldots, (\overrightarrow{v}(k), t(k)\}$, $\mathcal{L} = \{(\ell(1), t(1)), \ldots, (\ell(k), t(k))\}$.
Final TMVs Verification:			$RMSE(\overrightarrow{v}(i), \overrightarrow{\tilde{v}'}(i)) = \sqrt{\frac{\sum_{i=1}^{k}\left(\frac{\overrightarrow{v}(i) - \overrightarrow{\tilde{v}'}(i)}{\overrightarrow{\tilde{v}'}(i)}\right)^2}{k}}$, $RMSE(\ell(i), \tilde{\ell}'(i)) = \sqrt{\frac{\sum_{i=1}^{k}\left(\frac{\ell(i) - \tilde{\ell}'(i)}{\tilde{\ell}'(i)}\right)^2}{k}}$. $RMSE(\overrightarrow{v}(i), \overrightarrow{\tilde{v}'}(i)) \overset{?}{\leq} \epsilon$, $RMSE(\ell(i), \tilde{\ell}'(i)) \overset{?}{\leq} \mu$,
Decision:	Full access.	$\xleftarrow{Broadcast\ Decision}$	Decision ¡FAIL or SUCCESS¿, Accept $\mathcal{M}$, grant B full access.

Fig. 3. Vehicular Motion Vectors Verifier Protocol.

Figure 3 formally presents the steps of VET. A remote adversary M who cannot modify the physical characteristics of the transmitted signal is detected in Step 3, as the claimed TMVs are for the emulated trajectory while the estimated TMVs are the actual trajectory of M. Further, an advanced adversary M who with the knowledge of channel to the verifier B can craft the FoA to match the emulated and the claimed TMVs. Such as the adversary is detected by Step 6, as in Step 4 the verifier B captures the FoAs when the adversary will be communicating with any other entity present in the vicinity. Such communication can be detected by noting the sender and receiver in the header [8].

We will present a more detailed discussion on the robustness of the protocol in the next section. Here, it is assumed that the advanced adversary is attempting to emulate different trajectories at different verifiers. This is an acceptable assumption as all the verifiers will have different physical locations. Hence, emulating the same physical trajectory will force M to emulate different perceived trajectories at different verifiers.



(a) (b) (c)

Fig. 4. (a) A timeline for the interpolation, (b) ROC curve for location data for various interpolation techniques, (c) ROC curve for velocity data for various interpolation techniques.

4.2 Interpolating TMVs

In Step 5, of VET the estimated TMVs need to be interpolated for synchronizing with claimed TMVs, as shown in Fig. 4(a) such that the comparison can be made between the estimated and claimed trajectories. We perform the interpolation utilizing cubic spline interpolation [29, 39] because of its high accuracy, smoothness, flexibility, robustness, and less noisy interpolation when modeling trajectory motion profiles. Compared to other interpolation techniques like piecewise linear interpolation [25], r-cubic spline [36], and polynomial interpolation [16], cubic spline produces a smoother curve. Piecewise linear interpolation [25] has a high granularity of the TMV data but only does well when the vehicle is moving on a straight line at constant velocity. It is not as robust as cubic-spline for interpolating TMV in the real world. Linear interpolation works well in ideal scenarios, but in our experiments, the vehicles move at changing speeds at different times. Piecewise polynomial interpolation, like quadratic spline, is not the best interpolation technique compared to cubic spline and r-cubic spline regarding accuracy and smoothness, especially for complex scenarios. r-cubic spline [36] is simpler and faster but less accurate because its interpolation is based on simple recurrence equations, unlike the cubic spline which requires solving tri-diagonal matrix-vector equations. In Fig. 4 (b) and (c), we show that cubic spline has the best performing ROC curve as compared to other techniques for location and velocity, respectively, when we account for a trajectory with 90° turn. We used the data we collected for evaluations; please refer to Section 6.1 for more details. Although we do note that cubic spline is more computationally expensive, it is acceptable for our model as a not computationally limited verifier performs

all the computations. Moreover, we need accuracy and smoothness of the curve, especially for irregular data points, rather than speed for VET. The cubic spline interpolation of both the location and velocity is performed using the following equation:

$$\begin{cases} S_0(t) = \tilde{Y}_0 + b_0(t - t'_0) + c_0(t - t'_0)^2 + d_0(t - t'_0)^n \ \forall \ t \in [t'_0, t'_1], \\ \vdots \\ S_n(t) = \tilde{Y}_{n-1} + b_{n-1}(t - t'_n) + c_{n-1}(t - t'_n)^2 + d_{n-1}(t - t'_n)^n \ \forall \ t \in [t'_{n-1}, t'_n]. \end{cases} \tag{3}$$

where $\tilde{Y}$ can be either velocity $\overrightarrow{v}$ or location ℓ of the vehicle at time t', computing the parameters for b, c, d is obtained from solving a system of linear equations and substitution. The result will be a TMV curve that is smooth and more continuous than other forms of interpolations.

5 Security Analysis

In this section, first, we analyze the correctness of VET followed by robustness analysis against the adversary presented in Section 3.2. For the formal analysis of the protocol, we will utilize the idea of matching conversation [8]. The main idea is that two entities can mutually authenticate each other in the presence of an adversary if and only if they have the same chronology of exchanged messages.

5.1 Correctness Analysis

We discuss the correct implementation of VET when there is no adversary present. We consider the prover A and verifier B to be modeled by an Oracle model. We define the protocol transcript at A and B as Π_A and Π_B, respectively as observed by the oracle Π. In the transcripts, the received messages are denoted by a hat notation. The transcripts of the messages exchanged between A and B are:

$$\Pi_A = \{AE_K(RTA); m(1); tx(1); \ldots; m(k); tx(k)\}, \tag{4}$$

$$\Pi_B = \{\widehat{AE_K(RTA)}; \widehat{m(1)}; \widehat{tx(1)}; \ldots; \widehat{m(k)}; \widehat{tx(k)}\}, \tag{5}$$

for ease of depiction, we have skipped the timestamps for the messages. Several communicating oracles are also possible in a distributed way, but each oracle is unique.

The matching conversation is a way of authenticating an entity, which is the prover A. Both A and B will get the same long-lived key K, which would be unknown to anyone else. Once the communication is correct, the verifier B confirms or denies the prover A. That is, at the end of the conversation, the decision (η), from the verifier B is to confirm ($\mathcal{C}$) or reject ($\mathcal{R}$) the prover A ($\eta, \mathcal{C}, \mathcal{R}$). Although rejection can occur before the end of the conversation, confirmation only happens at the end.

The prover oracle (Π_A) sends a message $AE_K(RTA)$, which contains the request to authenticate. The verifier oracle (Π_B) receives the message $\widehat{AE_K(RTA)}$. It decrypts the message to check if the correct key K was used and grants the prover partial access. Next the prover oracle (Π_A) transmits the message $m(1)$ where Π_B extracts trajectory and motion vectors (TMVs) $\ell(1)$ and $v(1)$. This is followed by Π_A transmits a message $tx_A(1)$ not intended for Π_B. From $\widehat{tx_A(1)}$ transmission Π_B records the Frequency of Arrival (FoA) $\tilde{f}(1)$. Now the verifier estimates the velocity $\overrightarrow{\tilde{v}'}(1)$ and location $\tilde{\ell}'(1)$. Finally, compare the estimated and claimed velocities and locations based on the RMSE in Step 6 after interpolating the estimated to synchronize with the claimed in Step 5. It is straightforward to show if the message $m(1)$ and the transmission $tx(1)$ are from the same prover oracle Π_A. The estimated and claimed will be within the acceptable error ϵ for location and μ for velocity. This is repeated for all k transmissions.

5.2 Robustness Analysis

Next, we will analyze the robustness of VET against the threat model we defined in Section 3.2. First, we will analyze VET against a remote attacker who injects messages. This is followed by the remote advanced attacker, who can modify its physical layer envelope in an attempt to force the verifier B to accept the messages.

Remote Attacker: The remote adversary (M) is inside the communication range of the verifier B. Here, in the oracle model, we have an adversary oracle Π_M and the verifier oracle Π_B. The transcripts of the messages exchanged between M and B for VET execution is:

$$\Pi_M = \{AE_{K_M}(RTA); m_M(1); tx_M(1); \ldots; m_M(k); tx_M(k)\}, \tag{6}$$

$$\Pi_B = \{\widehat{AE_{K_M}(RTA)}; \widehat{m_M(1)}; \widehat{tx_M(1)}; \ldots; \widehat{m_M(k)}; \widehat{tx_M(k)}\}, \tag{7}$$

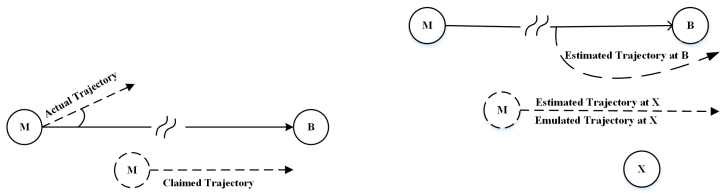
For ease of depiction, we have skipped the timestamps for the messages. For the case of a legitimate prover (A), there can be two different scenarios: (1) A is not present, and M initiates VET, and (2) A is present and M hijacks the execution of VET. In the first case, the prover oracle's transcript is:

$$\Pi_A = \{\emptyset\}. \tag{8}$$

In the second case, the transcript is:

$$\Pi_A = \{AE_K(RTA); m(1); tx(1); \ldots; m(k); tx(k)\}. \tag{9}$$

To prove the robustness of VET against a remote attacker. We aim to prove that the acceptance or authentication at the verifier (B) with non-matching conversations at prover and verifier oracles Π_A and Π_B, respectively, is negligible. Let us dive into the individual messages exchanged between the adversary oracle Π_M and the verifier oracle Π_B, (6) and (7). The first message that is



(a) (b)

Fig. 5. (a) A remote adversary M attempting to authenticate with a spoofed trajectory inside the communication range of the verifier B, and (b) a remote advanced attacker M attempting to authenticate an emulated trajectory to verifier B with other verifiers X in the vicinity.

exchanged between them is $AE_{K_M}(RTA)$, this message is accepted by Π_B even with mismatch with Π_A, (8) and (9). This is because the credential used by Π_M is either issued by a valid Trusted Authority (TA) or compromised from the legitimate prover A. Thus Π_M is able to initiate the session. Now, let us focus on the k messages exchanged for the verification. There are two sets of messages $m_M(i)$ are the messages intended for the verifier oracle Π_B and Π_B estimates the TMVs. And the transmissions $tx_M(i)$ from adversary oracle Π_M intended for other oracles present such as Π_X. Which can be some other verifier in the same vicinity. Such that the verifier oracle Π_B can be a roadside unit and other oracles Π_X can be another vehicle in the vicinity.

For this type of adversary, the claimed and the estimated TMVs because the adversary is present at a remote location, as shown in Fig. 5(a). This will force Π_B to estimate the remote TMVs detected by Step 3 of VET. It should be noted here this applies to both static and moving adversaries. Hence, this type of adversary will be detected and removed from the system. As well as the verifier will notify the presence of an adversary to other entities in the communication range.

Remote Advanced Attacker: Similar to the analysis against a remote attacker, to prove the robustness of VET against a remote advanced attacker. We aim to prove that the acceptance or authentication at the verifier (B) with non-matching conversations at prover and verifier oracles Π_A and Π_B, respectively, is negligible. The individual messages exchanged between the adversary oracle Π_M and the verifier oracle Π_B, (6) and (7). The first message that is exchanged between them is $AE_{K_M}(RTA)$, this message is accepted by Π_B even with mismatch with Π_A, (8) and (9). Here, in addition to injecting the set of claimed TMVs $\mathcal{M}$, the advanced adversary can change the envelope and FoA to emulate a trajectory estimated from sniffed packets indented for Π_X. Hence, an advanced adversary oracle Π_M is capable of emulating a trajectory to Π_B, as shown in Fig. 5(b).

The emulated trajectory of Π_M is accepted at Π_B without matching conversation with Π_A if the RMSE of all the TMVs in Step 6 is within the acceptable range. This cannot happen with certainty as Π_M because even if emulating the same trajectory to Π_B and Π_X. The estimated trajectories will be different; this is because the estimated trajectory $(\mathcal{V}', \mathcal{L}')$ in Step 2 is emulated for Π_B.

Whereas the estimated trajectory $(\tilde{\mathcal{V}}', \tilde{\mathcal{L}}')$ in Step 4 is emulated for Π_X. The adversary does this to pass Step 3, where the claimed and estimated trajectories must match at respective verifiers. Note here that Step 4 for Π_B captures the messages for Step 2 of Π_X. Also, verifiers inform all other entities about the failure of the authentication of any entity. It should be noted here this can be utilized to launch a denial-of-service (DoS) where a legitimate entity is forcibly disconnected. This is orthogonal to the application of VET. This can be trivially tackled by cryptographic verification of the failure broadcast. Next, we show that both types of adversaries have negligible success probability in defeating VET.

Formal Proof: For both the adversary models, we can model the success of the adversary oracle Π_M for claimed TMVs to match the estimated TMVs. Let the probability for Π_M for $\overrightarrow{v}(i)$ and $\ell(i)$ match with $\overrightarrow{\tilde{v}'}(i)$ and $\tilde{\ell}'(i)$, respectively at Π_B be p. This probability depends on the distance of the adversary M from the emulated trajectory. As the wireless channel outdoors decorrelates [56]. We evaluate this probability in the evaluation section. Thus, for k TMVs, the probability of an adversary succeeding with no matching is

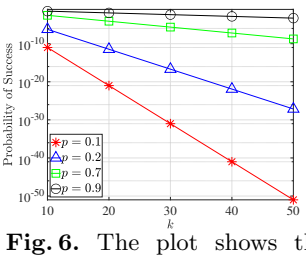


Fig. 6. The plot shows the success probability of M for different numbers of TMVs in the trajectory k.

$$\Pr[B \text{ accept} \wedge \text{No-matching}] = p^k, \tag{10}$$

which is a negligible probability [35], as shown in Fig. 6. Even for a high probability $p_1 = 0.9$, for 50, 40, 30, 20, and 10 TMVs, the success probability is 5×10^{-3}, 1.4×10^{-2}, 4.2×10^{-2}, 1.2×10^{-1}, and 3.5×10^{-1}. Please note here for a single execution of VET, the attacker has only one chance to inject all the TMVs online. Hence, a higher probability of success is acceptable here relative to traditional crypto-algorithm (similar values are acceptable for other online protocols with short authentication strings [31]).

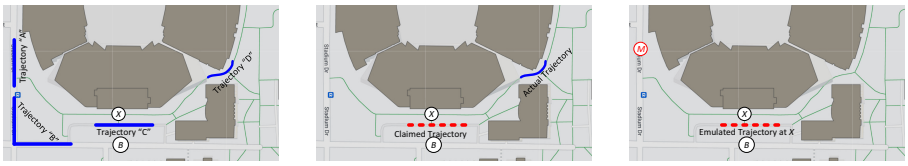
5.3 Discussion on Shortcomings

One of the areas we need to recognize is in the absence of at least two truthful verifiers, a remote advanced adversary can be successful. But it should be noted here that a novice remote adversary who cannot craft the physical layer envelope can be detected with only one verifier. Thus, only detecting an advanced adversary can craft the physical layer envelope with the knowledge of all the channels within the entities. We need more than one truthful verifier, which is not a reasonable requirement for detecting the strongest possible adversary.

6 Experimental Evaluation

In this Section, we evaluate the correctness, robustness, and protocol parameters utilizing a USRP platform with well-defined experiments. First, we describe the experimental setup followed by correctness and robustness analysis.

6.1 Experimental Setup



(a) (b) (c)

Fig. 7. (a) Verifying different trajectories of a legitimate prover A, (b) a remote adversary injecting claimed trajectory, and (c) a remote advanced adversary manipulating the signal's physical properties to emulate trajectory.

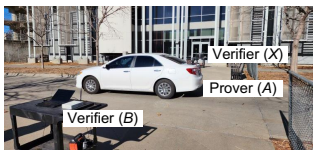


Fig. 8. Experimental setup with Prover car (A) with Verifiers (B) and (X).

Our experimental setup includes a prover (A) vehicle and stationary verifier (B), as shown in Fig. 7(a) and Fig. 8. We have a secondary verifier (X) present in the system for demonstration purposes only; we do not use the data collected at (X) for evaluations. The prover vehicle contains the signal transmitter USRP 2922 inside a car, which continuously broadcasts the BPSK signal at 915MHz using an omnidirectional antenna (VERT-900). The transmitter USRP is connected to a Lenovo ThinkPad T14 running the GNU Radio transmitter code. We choose 915MHz center frequency with a bandwidth, f_0, instead of 2.45GHz, which is in the Wi-Fi band because it is less congested and has a longer range. The verifiers are two stationary USRP 2922s connected to two individual computers placed on the opposite side of the road, which acts as receivers. The center frequency is also set to 915MHz, with a target sampling rate of 32000Sps and an actual sample rate of 195312Sps. The receivers also run GNU Radio code to capture the transmitted data packets from the moving prover. A GPS-enabled phone collects the ground truth of location and velocity data as the prover vehicle drives around the verifiers. We synchronize all three computers and the phone to use the United States Internet Time Server (ITS) of The Network Time Protocol server [53]. The verifier collects timestamped data as the prover drives around at a constant speed.

6.2 Correctness Analysis

First, we focus on evaluating the correct performance of VET. For this, we evaluate the location estimation and velocity estimation individually. The performance of VET is the worst of either of the estimations. We captured the physical layer envelope and frequency of arrival of the signal received from A. We implemented the methods mentioned in Appendix A to estimate velocity and position. Then, we compute the key performance indicator (Receiver operating characteristic) by comparing the estimated velocity and position to the ground truth recorded on the phone kept inside A.

Receiver operating characteristic (ROC) curve: We compute two separate Receiver operating characteristic (ROC) curves for velocity and location

data. We use the ROC curves to evaluate three parameters. First, the acceptable errors to set the thresholds (ϵ, μ) for RMSE of location and velocity, respectively, in Steps 3 and 6. Second, k is the number of trajectory points required to complete the verification with an acceptable true positive rate. Finally, we evaluate the acceptable errors for the straight or turning trajectory of the vehicle.

Figure 9(a) shows the plot between true positive rate (TPR) and false positive rate (FPR) for various ϵ RMSE errors and $k = 3$ for the location data. From the figure, we observe that for $\epsilon = 0.2$, we observe a 0.92 true positive rate for 0.03 false positive rate. In Fig. 9(c), we show the location data ROC curve for various k number of trajectory points for $\epsilon = 0.2$. We observe that for $k = 3$ VET can achieve TPR = 0.96 for FPR or 0.03. Further, in Fig. 9(b) and (d), we plot the velocity ROC curve for various RMSE threshold (μ) and k, respectively. We observe that for velocity, VET achieves a TPR of 0.9 for $\mu = 0.2$ and a TPR of 0.94 for $k = 3$. We also observe that each of the curves are acceptable ROC curve as the TPR goes close to 1 before the FPR reaches 0.05. For the rest of the experimental analysis, we set the values of the thresholds from the ROC curves. Specifically, we fix the $\epsilon = 0.2$ and $\mu = 0.2$ for location RMSE and velocity RMSE, respectively. We selected these values as they achieve optimum TPR for acceptable FPR. Finally, we compute the ROC curves for various trajectories, as shown in Fig. 7(a). The trajectory "A" and "C" are straight line while "B" and "D" involves turns. From the curves in Fig. 9(e) and (f), we observe better performance for straight-line trajectories as compared to ones involving turns. However, all of the TPR and FPR values are acceptable, with TPR reaching 1 for the trajectories involving turns before FPR reaches 0.08.

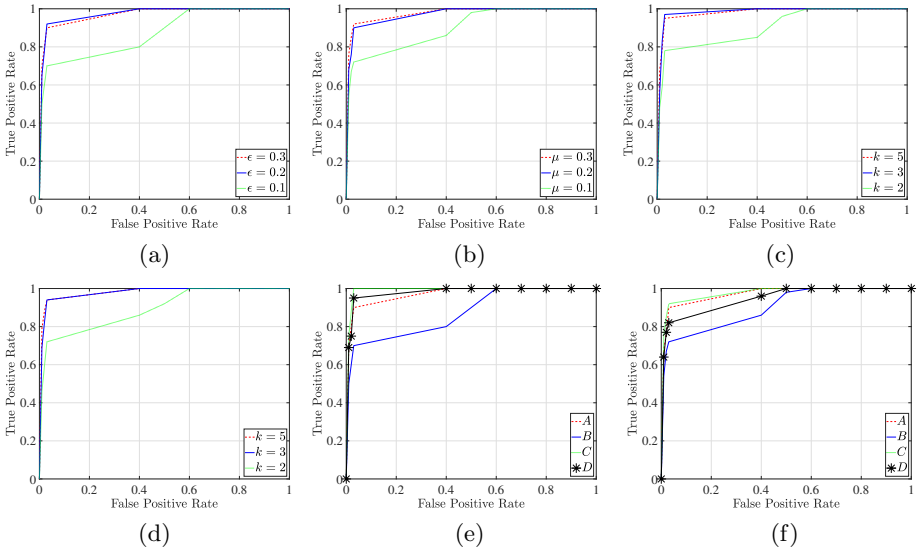


Fig. 9. (a) ROC curve for location data for various RMSE threshold ϵ, (b) ROC curve for velocity data for various RMSE threshold μ, (c) ROC curve for location data for various k the number of trajectory points, (d) ROC curve for velocity data for various k, (e) ROC curve for location data for various trajectories as shown in Fig. 7(a), (f) ROC curve for velocity data for various trajectories as shown in Fig. 7(a).

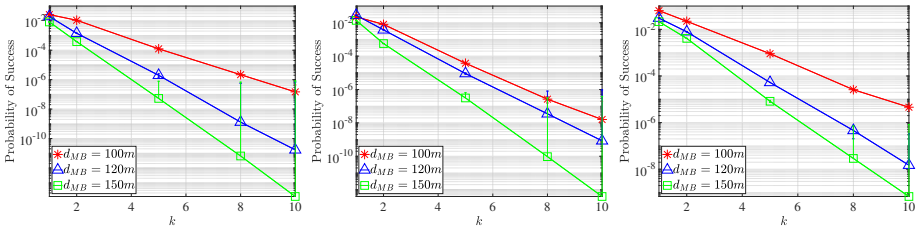


Fig. 10. (a) Probability of success for remote M in defeating velocity verification, (b) probability of success for remote M in defeating location verification, and (c) probability of success for remote advanced M in defeating velocity verification.

6.3 Robustness Analysis

Next, we evaluate the robustness of VET against both adversaries defined in Section 3.2. First, we evaluate the remote attacker who injects a spoofed trajectory as a message. Next, we evaluate the performance of an advanced remote attacker M who can change the physical parameters of the signal to emulate a target trajectory. We compute the success probability using the $RMSE(\cdot)$ function.

Remote Attacker: We utilized the data collected to emulate the remote attacker. Here, the attacker's actual trajectory differed from the claimed trajectory, as shown in Fig. 7(b). Using the data, we plot two graphs for location and velocity. In Fig. 10(a) and Fig. 10(b), we plot the probability of success (p^k) from (10) for the adversary for B to accept velocity and location, respectively against k the number of messages for d_{MB} distances between B and M, varied between 100m and 150m. From the plot, we observe that for $k = 3$ data points of the trajectory, the probability of success for the adversary goes down to the level of 10^{-5} for both the velocity and location. Further, we observe that an adversary farther than 120m from the verifier B has a significantly low success probability in defeating VET. Thus, VET can detect an adversary who might be using compromised infrastructure to inject data. Moreover, VET can detect a remote-moving adversary attempting to inject rogue messages. *This attests to our theoretical finding that the probability of success for the remote adversary is a negligible probability.*

Remote Advanced Attacker: Finally, we performed emulation to evaluate the remote advanced attacker using Matlab. We first computed the wireless channels h_{MB} and h_{MX} between the vehicle and the verifier B, and the vehicle and the second verifier X, respectively. The adversary utilized the knowledge of the channel h_{MX} to emulate the trajectory at X, as shown in Fig. 7(c). This signal is received by B on the h_{MB} emulated by a ray tracing model. B computed the estimated trajectory using the emulated trajectory to compute the probability of success for the adversary.

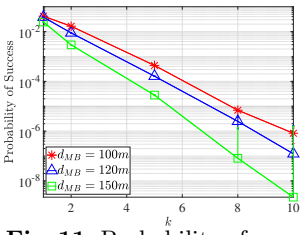


Fig. 11. Probability of success for remote advanced M in defeating location verification.

Figure 10(c) and Fig 11 show the plot between the probability of success (p^k) from (10) for the adversary against k, for the velocity and location, respectively. We observed that an advanced adversary M could defeat VET with a success probability of 10^{-6} for $k = 5$ trajectory data points. Also, here an adversary further than the distance than $d_{MB} \geq 120m$ is detected with probability $(1 - 10^{-6})$, for both velocity and location. *This attests to our theoretical finding that the probability of success for the remote advanced adversary is a negligible probability.* Hence, the advanced adversary M has to be close to B for defeating VET. Even when M is close to B, the adversary can be detected with certainty when more number k of trajectory points are collected for authentication.

7 Conclusion

We proposed VET: a framework that verifies the veracity of the crypto-credentials by authenticating them against physical trajectory and motion vectors (TMVs). The verifier implements a location and motion-based authentication strategy and verifies the crypto-credentials based on the acceptability of claimed TMVs against randomly estimated TMVs. This detects any adversary from remotely injecting spoofed messages when it is not physically present where it claims to be. We formally analyzed the correctness and robustness of VET using matching conversations. Finally, we attested to the findings of theoretical analysis with experimental analysis of VET on the USRP platform. Our experiments showed that VET has 97% true positives when operating without an adversary. We fixed the threshold values for evaluation based on the ROC curve plotted for the location and velocity data. We evaluated both novice and advanced adversarial behaviors. In the experiments, we showed that VET can detect advanced remote adversary with 99.9% who is capable of manipulating signals with absolute channel knowledge. *In the future*, we plan to expand the experimental evaluations on a UAV platform with both moving provers and verifiers.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. We sincerely thank the NIMBUS lab at the University of Nebraska–Lincoln for assisting in setting up the experimentation infrastructure. This research was supported in part by the NCESR Cycle 17 grant. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the NCESR.

References

1. Adebayo, A., Rawat, D.B.: Deceptor-in-the-middle (ditm): Cyber deception for security in wireless network virtualization. In: Proc. of IEEE 17th Annual Consumer Communications & Networking Conference (CCNC). pp. 1–6. IEEE (2020)
2. Alessandretti, G., Broggi, A., Cerri, P.: Vehicle and guard rail detection using radar and vision data fusion. IEEE transactions on intelligent transportation systems **8**(1), 95–105 (2007)

3. Amar, A., Weiss, A.J.: Localization of narrowband radio emitters based on doppler frequency shifts. IEEE Transactions on Signal Processing **56**(11), 5500–5508 (2008)
4. Amar, A., Weiss, A.J.: Direct position determination: A single-step emitter localization approach. In: Proc. of Classical and modern direction-of-arrival estimation, pp. 385–424. Elsevier (2009)
5. Avoine, G., Bingöl, M.A., Boureanu, I., Čapkun, S., Hancke, G., Kardaş, S., Kim, C.H., Lauradoux, C., Martin, B., Munilla, J., et al.: Security of distance-bounding: A survey. ACM Computing Surveys (CSUR) **51**(5), 1–33 (2018)
6. Baker, R., Martinovic, I.: Secure location verification with a mobile receiver. In: Proc. of ACM Workshop on Cyber-Physical Systems Security and Privacy. pp. 35–46. ACM (2016)
7. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Proc. of International Conference on the Theory and Application of Cryptology and Information Security. pp. 531–545. Springer (2000)
8. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Proc. of Annual international cryptology conference. pp. 232–249. Springer (1993)
9. Borgaonkar, R., Jaatun, M.G.: 5G as an enabler for secure iot in the smart grid. In: Proc. of First International Conference on Societal Automation (SA). pp. 1–7. IEEE (2019)
10. Brands, S., Chaum, D.: Distance-bounding protocols. In: Proc. of Workshop on the Theory and Application of of Cryptographic Techniques. pp. 344–359. Springer (1993)
11. Čapkun, S., Rasmussen, K., Cagalj, M., Srivastava, M.: Secure location verification with hidden and mobile base stations. IEEE Transactions on Mobile Computing **7**(4), 470–483 (2008)
12. Chestnut, P.C.: Emitter location accuracy using TDOA and differential doppler. IEEE Transactions on Aerospace and Electronic Systems **AES-18**(2), 214–218 (1982). https://doi.org/10.1109/TAES.1982.309230
13. DeBruhl, B., Weerakkody, S., Sinopoli, B., Tague, P.: Is your commute driving you crazy? a study of misbehavior in vehicular platoons. In: Proc. of ACM Conference on Security & Privacy in Wireless and Mobile Networks. pp. 1–11 (2015)
14. Dickey, C., Smith, C., Johnson, Q., Li, J., Xu, Z., Lazos, L., Li, M.: Wiggle: Physical challenge-response verification of vehicle platooning. In: Proc. of International Conference on Computing, Networking and Communications (ICNC). pp. 54–60. IEEE (2023)
15. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on information theory **29**(2), 198–208 (1983)
16. Gasca, M., Sauer, T.: Polynomial interpolation in several variables. Advances in Computational Mathematics **12**, 377–410 (2000)
17. Ghose, N., Lazos, L.: Verifying ADS-B navigation information through doppler shift measurements. In: Proc. of IEEE/AIAA Digital Avionics Systems Conference (DASC). pp. 4A2–1. IEEE (2015)
18. Guvenc, I., Koohifar, F., Singh, S., Sichitiu, M.L., Matolak, D.: Detection, tracking, and interdiction for amateur drones. IEEE Communications Magazine **56**(4), 75–81 (2018)
19. Hancke, G.P., Kuhn, M.G.: An RFID distance bounding protocol. In: Proc. of SECURECOMM. pp. 67–73. IEEE (2005)
20. Hasrouny, H., Samhat, A.E., Bassil, C., Laouiti, A.: VANET security challenges and solutions: A survey. Vehicular Communications **7**, 7–20 (2017)

21. James, S., Raheb, R., Hudak, A.: UAV swarm path planning. In: Proc. of Integrated Communications Navigation and Surveillance Conference. pp. 2G3–1. IEEE (2020)
22. Kang, H., Chang, X., Mišić, J., Mišić, V.B., Fan, J., Bai, J.: Improving Dual-UAV aided ground-uav bi-directional communication security: Joint uav trajectory and transmit power optimization. IEEE Transactions on Vehicular Technology **71**(10), 10570–10583 (2022)
23. Khan, W.U., Lagunas, E., Ali, Z., Javed, M.A., Ahmed, M., Chatzinotas, S., Ottersten, B., Popovski, P.: Opportunities for physical layer security in UAV communication enhanced with intelligent reflective surfaces. IEEE Wireless Communications **29**(6), 22–28 (2022)
24. Komissarov, R., Wool, A.: Spoofing attacks against vehicular FMCW radar. In: Proc. of the 5th Workshop on Attacks and Solutions in Hardware Security. pp. 91–97 (2021)
25. Lepot, M., Aubin, J.B., Clemens, F.H.: Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment. Water **9**(10), 796 (2017)
26. Li, B., Zhang, T., Xia, T.: Vehicle detection from 3d lidar using fully convolutional network. arXiv preprint arXiv:1608.07916 (2016)
27. Man, Y., Li, M., Gerdes, R.: {GhostImage}: Remote perception attacks against camera-based image classification systems. In: Proc. of International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020). pp. 317–332 (2020)
28. Man, Y., Muller, R., Li, M., Celik, Z.B., Gerdes, R.: That person moves like a car: Misclassification attack detection for autonomous systems using spatiotemporal consistency. In: Proc. of USENIX Security Symposium (2023)
29. McKinley, S., Levine, M.: Cubic spline interpolation. College of the Redwoods **45**(1), 1049–1060 (1998)
30. Muller, R., Man, Y., Celik, Z.B., Li, M., Gerdes, R.: Physical hijacking attacks against object trackers. In: Proc. of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 2309–2322 (2022)
31. Nguyen, L.H., Roscoe, A.W.: Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey. Journal of Computer Security **19**(1), 139–201 (2011)
32. Palmer, A.: Amazon plans to start delivering packages by drone in texas later this year. `https://www.cnbc.com/2022/07/15/amazon-to-start-delivering-packages-by-drone-in-texas-later-this-year.html` (2022)
33. Patwari, N., Hero III, A.O.: Using proximity and quantized rss for sensor localization in wireless networks. In: Proc. of ACM international conference on Wireless sensor networks and applications. pp. 20–29 (2003)
34. Pivato, P., Palopoli, L., Petri, D.: Accuracy of RSS-based centroid localization algorithms in an indoor environment. IEEE Transactions on Instrumentation and Measurement **60**(10), 3451–3460 (2011)
35. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In: Proc. of International conference on the theory and applications of cryptographic techniques. pp. 387–398. Springer (1996)
36. Revesz, P.Z.: A recurrence equation-based solution for the cubic spline interpolation problem. International Journal of Mathematical Models and Methods in Applied Sciences **9**(1), 446–452 (2015)
37. Schäfer, M., Lenders, V., Schmitt, J.: Secure track verification. In: Proc. of IEEE Symposium on Security and Privacy. pp. 199–213. IEEE (2015)

38. Schäfer, M., Leu, P., Lenders, V., Schmitt, J.: Secure motion verification using the doppler effect. In: Proc. of ACM Conference on Security & Privacy in Wireless and Mobile Networks. pp. 135–145 (2016)
39. Schoenberg, I.J.: Contributions to the problem of approximation of equidistant data by analytic functions: Part a.—on the problem of smoothing or graduation. a first class of analytic approximation formulae. IJ Schoenberg Selected Papers pp. 3–57 (1988)
40. Stinson, D.R.: Cryptography: theory and practice. CRC press (2005)
41. Sun, M., Guo, Z., Li, M., Gerdes, R.: Passive drone localization using LTE signals. In: Proc. of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks. pp. 295–297 (2022)
42. Sun, M., Li, M., Gerdes, R.: A data trust framework for vanets enabling false data detection and secure vehicle tracking. In: Proc. of IEEE Conference on Communications and Network Security (CNS). pp. 1–9. IEEE (2017)
43. Sun, M., Man, Y., Li, M., Gerdes, R.: SVM: secure vehicle motion verification with a single wireless receiver. In: Proc. of ACM Conference on Security and Privacy in Wireless and Mobile Networks. pp. 65–76 (2020)
44. Sun, X., Ng, D.W.K., Ding, Z., Xu, Y., Zhong, Z.: Physical layer security in uav systems: Challenges and opportunities. IEEE Wireless Communications **26**(5), 40–47 (2019)
45. Sun, Z., Balakrishnan, S., Su, L., Bhuyan, A., Wang, P., Qiao, C.: Who is in control? practical physical layer attack and defense for mmwave-based sensing in autonomous vehicles. IEEE Transactions on Information Forensics and Security **16**, 3199–3214 (2021)
46. Tesla: Autopilot. `https://www.tesla.com/autopilot` (2022)
47. Tippenhauer, N.O., Luecken, H., Kuhn, M., Capkun, S.: UWB rapid-bit-exchange system for distance bounding. In: Proc. of ACM Conference on Security & Privacy in Wireless and Mobile Networks. pp. 1–12 (2015)
48. Tirer, T., Weiss, A.J.: High resolution localization of narrowband radio emitters based on doppler frequency shifts. Signal Processing **141**, 288–298 (2017)
49. Tithi, T., Deka, B., Gerdes, R.M., Winstead, C., Li, M., Heaslip, K.: Analysis of friendly jamming for secure location verification of vehicles for intelligent highways. IEEE Transactions on Vehicular Technology **67**(8), 7437–7449 (2018)
50. Traynor, P., Butler, K., Enck, W., McDaniel, P., Borders, K.: malnets: large-scale malicious networks via compromised wireless access points. Security and Communication Networks **3**(2-3), 102–113 (2010)
51. Vaas, C., Juuti, M., Asokan, N., Martinovic, I.: Get in line: Ongoing co-presence verification of a vehicle formation based on driving trajectories. In: Proc. of IEEE European Symposium on Security and Privacy (EuroS&P). pp. 199–213. IEEE (2018)
52. Wang, S., Jiang, X., Wymeersch, H.: Cooperative localization in wireless sensor networks with AOA measurements. IEEE Transactions on Wireless Communications (2022)
53. webmaster, N.P.: NTP: The network time protocol. `0.us.pool.ntp.org` (2022)
54. Wender, S., Dietmayer, K.: 3D vehicle detection using a laser scanner and a video camera. IET Intelligent Transport Systems **2**(2), 105–112 (2008)
55. Westerlund, O., Asif, R.: Drone hacking with raspberry-pi 3 and wifi pineapple: Security and privacy threats for the internet-of-things. In: Proc. of 1st International Conference on Unmanned Vehicle Systems-Oman (UVS). pp. 1–10. IEEE (2019)

56. Xu, Z., Li, J., Pan, Y., Lazos, L., Li, M., Ghose, N.: PoF: Proof-of-following for vehicle platoons. In: Proc. of The Network and Distributed System Security Symposium (NDSS 2022), San Diego, CA. pp. 1–18. Internet Society (2022)
57. Yang, Y., Lee, K., Kim, Y., Fawaz, K.: PEDRO: Secure pedestrian mobility verification in v2p communication using commercial off-the-shelf mobile devices. In: Proc. of the 2th Workshop on CPS&IoT Security and Privacy. pp. 41–46 (2021)
58. Zeng, Y., Zhang, R.: Energy-efficient UAV communication with trajectory optimization. IEEE Transactions on Wireless Communications **16**(6), 3747–3760 (2017)
59. Zheng, T.X., Wen, Y., Liu, H.W., Ju, Y., Wang, H.M., Wong, K.K., Yuan, J.: Physical-layer security of uplink mmwave transmissions in cellular v2x networks. IEEE Transactions on Wireless Communications (2022)

Appendix A: Primitives used in VET

Before diving into the details of VET, we present its building blocks. First, we present the method utilized to estimate the velocity. Followed by the method to compute the position and combine both to compute the trajectory and motion vectors (TMVs).

Frequency of Arrival (FoA) for Velocity Estimation: The FoA captures the effect of the velocity on the center frequency. In other words, it is the Doppler effect experienced by the moving verifier B with respect to the moving prover A at speed v. From Fig. 2, the prover vehicle A is within the communication range of the verifier B. The frequency of arrival when the verifier and the prover are moving towards each other, so the Doppler effect experienced by the verifier increases, is given by

$$\mathcal{F} = f_0 \times \frac{c + \overrightarrow{v}_B \; cos \; \alpha}{c - \overrightarrow{v}_A \; sin \; \alpha}, \tag{11}$$

where $\mathcal{F}$ is the Doppler shift on verifier B, c is the propagation speed, f_0 is the prover's center frequency.

From (11), the velocity of the prover at the ith sample is given by

$$\overrightarrow{v}_B(i) = \left[c - \frac{\mathcal{F}(i)(\overrightarrow{v}_A(i) \; sin \; \alpha)}{f_0} \right] cos^{-1} \; \alpha. \tag{12}$$

The Doppler effect of the signal measured by B is dependent on the radial velocity and the center frequency. The relative velocity observed $\overrightarrow{v}(i) = \overrightarrow{v}_B(i) \, cos \, \alpha - \overrightarrow{v}_A(i) \; sin \; \alpha$.

Direct Location Estimation For estimation of the location, it is important to note that the verifier B has more than one antenna. This assumption is valid for vehicular networks as the roadside units are MIMO enabled, and in the case of UAV swarms, multiple single antenna UAVs can collude as the verifier. We used maximum likelihood estimation to directly estimate the position, which maximizes the likelihood for the prover [3] when the prover is broadcasting within

the expected verification range. This is a one-step process that does a 2-D or 3-D grid search of the prover's position.

The location $\ell(i)$ of the prover B is the position that maximizes the log-likelihood function. This position is expressed as

$$\ell(i) = \arg\max_{\ell}\{L_i\}. \tag{13}$$

Here, the log-likelihood function is written as

$$L_i = \sum_{j=1}^{J} \lambda_{max}(Q_j). \tag{14}$$

The log-likelihood function is the summation of all the maximum eigenvalues of the Hermitian matrix Q_j. The matrix contains received signals multiplied by the frequency difference of arrival (FDoA) at the different antennas.

Effect of NLoS on the FoA: Typically, when vehicular wireless signals propagate in the real world, it does that in multipath [43]. The motion claim of the prover reaches the verifier in two or more paths. For simplicity, in our research, we use two path components. The NLoS path exists due to signal reflection before getting to the verifiers. This means there exists a reflection point that is always changing due to the dynamicity of the environment and this change affects the position by some factor δ. Therefore, the prover's position for verifiers is $c - \overrightarrow{v_B} cos\ \alpha + \delta$. This means that for a moving verifier and a moving prover due to NLoS, the Doppler effect will be given by

$$\mathcal{F} = f_0 \times \frac{c + \overrightarrow{v}_B\ cos\ \alpha + \delta}{c - \overrightarrow{v}_A\ sin\ \alpha} + \epsilon, \tag{15}$$

This NLoS component is embedded in the signal path attenuation which maximizes the likelihood [3], at each verifier and the function that contains the unknown provers position and velocity.