Wiggle: Physical Challenge-Response Verification of Vehicle Platooning

Connor Dickey* Christopher Smith* Quentin Johnson*, Jingcheng Li, Ziqi Xu, Loukas Lazos, Ming Li
Bradley University *Stony Brook University* *University of Arizona*
cdickey@bradley.edu chrissmith@stonybrook.edu {quentinj, jli2972, zxu1969, llazos, lim}@arizona.edu

***Abstract*—In this work, we establish a physical access control mechanism for vehicular platoons. The goal is to restrict vehicle-to-vehicle (V2V) communications to platooning members by tying the digital identity of a candidate vehicle requesting to join a platoon to its physical trajectory relative to the platoon. We propose the *Wiggle* protocol that employs a physical challenge-response exchange to prove that a candidate requesting to be admitted into a platoon actually follows it. The protocol name is inspired by the random longitudinal movements that the candidate is challenged to execute. *Wiggle* prevents any remote adversary from joining the platoon and injecting fake V2V messages. Compared to prior works, *Wiggle* is resistant to pre-recording attacks and can verify that the candidate is traveling behind the verifier in the same lane.**

***Keywords*—Vehicular security, access control, authentication.**

I. Introduction

Autonomous platooning refers to a convoy of autonomous vehicles traveling in a single file. The platoon members apply a cooperative adaptive cruise control (CACC) algorithm to maintain a safe distance and react to the surrounding traffic [1]–[3]. Specifically, steering and acceleration are coordinated using vehicle-to-vehicle (V2V) communications and onboard sensors. To secure the V2V message exchange, wireless standards such as the IEEE 1609.2 [4] and the more recent 3GPP TS 33.185 for Cellular Vehicle-to-Everything (C-V2X) [5] recommend the use of public key infrastructure (PKI). Cryptographic methods can authenticate the source and verify the integrity of a V2V message. However, they cannot physically bind the message originator to a trajectory.

The lack of physical trajectory verification opens the door to remote attacks. An adversary could claim to follow a platoon while being at a different location. Communication with the platoon may occur via a long-range transmitter, or via the cellular infrastructure using C-V2X. The adversary may be in possession of valid cryptographic credentials either by compromising the credentials of a valid vehicle or being one. Once authenticated, the adversary can inject fake messages into the platoon and cause accidents. This attack can scale to multiple platoons, as the adversary can impersonate ghost vehicles at various locations at once.

To mitigate remote attacks, several prior works have proposed physical access control mechanisms [6]–[10]. The main idea is to limit platoon access to only those vehicles that can prove they actually follow the platoon. The concept was formalized by Xu *et al.* with the introduction of a Proof-of-Following (PoF) [7], which is demonstrated in Fig. 1. Before being admitted to the platoon, a candidate $\mathcal{C}$ engages with a verifier $\mathcal{V}$ (typically the last vehicle of the platoon) to prove that it follows $\mathcal{V}$ within the designated following distance. *Note that the detection of a vehicle following the platoon is not sufficient as the verifier cannot link the physically-detected vehicle with the V2V messages that it receives.*

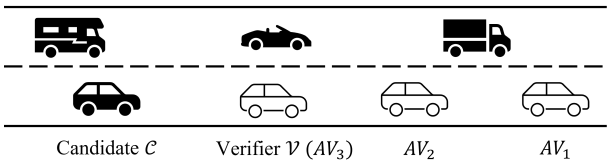


Fig. 1: A platoon of three vehicles. AV_3 acts as a verifier $\mathcal{V}$ for the candidate $\mathcal{C}$ who wishes to be admitted to the platoon. Parties $\mathcal{C}$ and $\mathcal{V}$ engage in a Proof-of-Following protocol.

Limitations of prior methods. The *Convoy* protocol uses the vertical acceleration due to road surface variations to correlate the candidate's and the verifier's trajectories [6]. However, the road surface is static making *Convoy* vulnerable to pre-recording attacks. Our prior work in [7] exploits the large-scale fading effect of ambient cellular transmissions to correlate the candidate-verifier trajectories. This context presents high spatial and temporal entropy, thus resisting pre-recording attacks. However, the protocol cannot verify the relative positioning between the candidate and the verifier, but only bounds the candidate within a radius from the verifier. Moreover, it requires omnidirectional ambient RF signals which are less prevalent with the advent of 5G networks or when traveling in low-coverage areas. Other protocols like [8] are susceptible to pre-recording and MitM attacks, or require tight time synchronization among vehicles [10].

Contributions: We propose *Wiggle,* a new PoF protocol for vehicular platoons which allows for both relative positioning and lane verification. Moreover, *Wiggle* is resistant to pre-recording attacks. The heart of the protocol relies on a series of physical challenges which are designed to bind the digital identity of the candidate to his trajectory. A physical challenge consists of a randomly longitudinal distance perturbation that must be executed by a given deadline.

We analyze the security of *Wiggle* and show that it is resistant to attacks from any candidate that does not follow

* The authors are undergraduate students who equally contributed to this work during a 2021 summer REU at the University of Arizona. This work was supported by NSF grant CCF-1852199, ARO grant W911NF1910050.

the verifier within the following distance, is not on the same lane as the verifier, or is separated by another vehicle. Moreover, by using an adaptive cruise control (ACC) algorithm to execute the challenges, we ensure that the user experiences imperceptible changes to the vehicle's velocity while a PoF is executed. We evaluate the performance of *Wiggle* via the Plexe platooning simulator [11] and show that a PoF verification lasts less than a minute for relevant freeway scenarios.

II. System Model

A. Platooning Model

We consider a vehicular platoon that applies cooperative adaptive cruise control (CACC) [1], [2] to coordinate platooning. The platooning model is shown in Figure 1. Vehicles AV_1, AV_2 and AV_3 form a platoon. Candidate vehicle $\mathcal{C}$ with an ACC system requests to join the platoon claiming to be following AV_3 within the platooning distance. Vehicle AV_3 acts as a *verifier* for $\mathcal{C}$'s trajectory.

Vehicles are equipped with distance measuring sensors such as radar, camera, LIDAR [12] that can measure the distance to the proceeding and following vehicles. To secure the platoon operation, V2V messages are protected using cryptographic primitives. According to the C-V2X communication standard (3GPP TS 33.185 [5]), V2X communication is supported by a PKI that provides each vehicle X with a private/public key pair (pk_X, sk_X) and a digital certificate $cert_X$. These credentials can be used to establish trust among the platoon vehicles and verify the origin of the information.

B. Threat Model

We consider an adversary $\mathcal{M}$ who attempts to join the platoon without following it. The attacker holds a public/private key pair (pk_M, sk_M) and a certificate $cert_M$ issued by a trusted certificate authority. The adversary can communicate with the platoon either via C-V2X communications or directly. A remote adversary is shown in Fig. 2(Top). The adversary may know the platoon's route in advance or in real time. Because the adversary is assumed to be remote, he does not launch attacks against the verifier's ranging sensors. Even if such attacks were launched, a secure-ranging protocol can be used to protect the distance sensing modality [13].

Man-in-the-middle adversary. The adversary can launch a Man-in-the-Middle (MitM) attack to gain admittance to the platoon while a legitimate candidate $\mathcal{C}$ also attempts to join. A MiTM attack is shown in Fig. 2(Bottom). The adversary jams the platoon join request sent from $\mathcal{C}$ and replaces it with his own request. At the same time, $\mathcal{M}$ impersonates the verifier to $\mathcal{C}$. The legitimate verifier challenges $\mathcal{M}$ to prove it follows the platoon by executing a PoF. The adversary relays the same challenge to $\mathcal{C}$ who executes the PoF protocol.

III. The *Wiggle* PoF Protocol

Overview. *Wiggle* is a physical challenge-response protocol that binds the candidate's digital identity with his physical trajectory. The verifier challenges the candidate to execute a series of longitudinal perturbations of its following distance

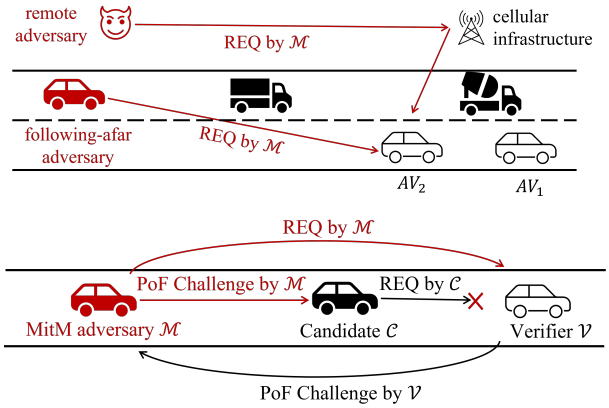


Fig. 2: Top: Remote adversary. Bottom: MiTM adversary.

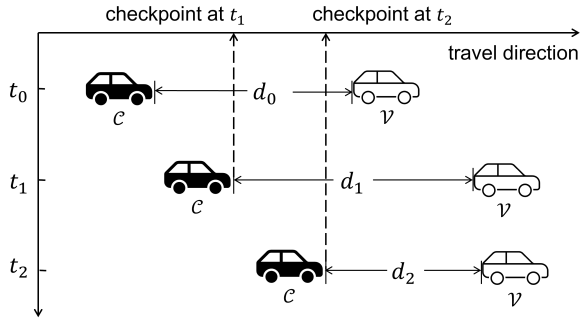


Fig. 3: $\mathcal{V}$ challenges $\mathcal{C}$ to reach randomly-generated checkpoints d_1 and d_2 by t_1 and t_2, respectively.

and measures these perturbations using the ranging modality. Each challenge (d_i, t_i) consists of a desired following distance d_i, referred to as a "checkpoint", and a deadline t_i.

Figure 3 shows the execution of two such challenges. At t_0, $\mathcal{C}$ claims to follow $\mathcal{V}$ at $d_0 = d_{ref}$. The verifier measures d_0 and verifies $\mathcal{C}$'s claim. However, this alone does not constitute proof as another vehicle could happen to follow $\mathcal{V}$. The verifier challenges $\mathcal{C}$ to reach checkpoints d_1 and d_2 by deadlines t_1 and t_2, respectively. The challenges are encrypted by $\mathcal{C}$'s public key. To pass verification, the candidate must reach each checkpoint by the designated deadline, resulting in a "wiggle" motion around d_{ref}. A remote adversary deos not pass verification, as he cannot be present at the checkpoints by the designated deadlines. Furthermore, by pointing the ranging sensor directly behind the verifier, relative ordering verification and lane verification are achieved.

A. The Wiggle Protocol

The protocol consists of three phases: *Digital identity verification*, *Physical challenge-response*, and *Physical verification*.

Digital identity verification phase.

1) The candidate sends a join request REQ to $\mathcal{V}$.

$$m_C(1) \leftarrow ID_V, ID_C, pk_C, cert_C, sig_{sk_C}(\text{REQ}, ID_C, ID_V),$$

where ID_V, ID_C are the verifier's and the candidate's identities, (pk_C, sk_C) are $\mathcal{C}$'s public/private key pair, and $cert_C$ is $\mathcal{C}$'s certificate.

2) The verifier validates the certificate of $\mathcal{C}$ with pk_{CA} the certificate authority's public key pk_{CA} and then verifies the signature with pk_C.

Physical challenge-response phase.

4) $\mathcal{V}$ generates K physical challenges denoted by $\Gamma = \{(d_{ref}, t_0), (d_1, t_1), \cdots, (d_K, t_K), (d_{ref}, t_{K+1})\}$. Each challenge consists of a checkpoint d_i, which is a random longitudinal perturbation of the following distance d_{ref}, and a corresponding deadline t_i by which the checkpoint must be reached. The challenges are signed with sk_V and then encrypted with pk_C. The message also contains the start time t_0 of initiating the response.

$$m_V(1) \leftarrow ID_C, E_{pk_C}[sig_{sk_V}(\Gamma, ID_V, ID_C, t_0), \Gamma, ID_V, ID_C, t_0].$$

5) $\mathcal{C}$ decrypts $m_V(1)$ and verifies the signature of $\mathcal{V}$.
6) $\mathcal{C}$ starts from d_{ref} at time t_0, passes through each checkpoint d_i by deadline t_i and then recovers to d_{ref}.
7) $\mathcal{V}$ measures and records the following distance of the candidate by each deadline. Denote the recorded data set as $\Gamma' = \{(d'_0, t_0), (d'_1, t_1), (d'_2, t_2), \cdots, (d'_K, t_K), (d'_{K+1}, t_{K+1})\}$.

Physical verification phase. In this phase, $\mathcal{V}$ verifies the candidates platooning claim by checking if $\mathcal{C}$ reached the designated checkpoints by the respective deadlines.

6) $\mathcal{V}$ compares each measured distance d'_i with the respective challenge d_i. If each d'_i is within a threshold γ from d_i, the verifier ACCEPTS. Otherwise, the verifier REJECTS.

$$\sum_{k=0}^{K+1} \frac{I(|d_k - d'_k| \leq \gamma)}{K+2} = 1,$$

where $I(\cdot)$ is the indicator function.

B. Parameter Selection

Checkpoint selection. To select each checkpoint d_i, the verifier determines a discrete range $\mathcal{S}$ around the nominal following distance d_{ref}. Using the standard time gap notation to denote following distances, let d_{ref} correspond to a time gap $g_{ref} = d_{ref}/v_V$, where v_V denotes the verifier's velocity. Let also $g_{\min}$ to be the minimum safety time gap between any two vehicles and $g_{\max}$ be the maximum time gap. The verifier computes a continuous range $[g_{\min} \cdot v_V, g_{\max} \cdot v_V]$ for selecting the checkpoints. It then divides this range to equal segments of length 2ρ (twice the radar resolution ρ) and computes a discrete range of M checkpoints $\mathcal{S} = \{s_1, s_2, \ldots, s_M\}$ where

$$M = \lfloor \frac{(g_{\max} - g_{min}) \cdot v_V}{2\rho} \rfloor + 1.$$

The checkpoint for each challenge is randomly selected from $\mathcal{S}$. To demonstrate the checkpoint selection process, consider a verifier traveling at $v_V = 30$m/s, as shown in Fig. 4. Assume $g_{min} = 1s$, $g_{max} = 2s$ and a radar resolution of $\rho = 0.3m$. The verifier computes $M = \frac{60\text{m}-30\text{m}}{2 \cdot 0.3\text{m}} + 1 = 51$ checkpoints between 30m and 60m from itself. The verifier randomly chooses from the 51 checkpoints when populating the K physical challenges for any candidate.

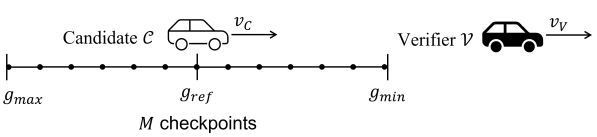


Fig. 4: Setting the checkpoint range for candidate $\mathcal{C}$.

Deadline selection. The deadlines can be selected in any fashion that would allow the candidate to safely move to the designated checkpoints. A straightforward way to select a deadline t_i for checkpoint d_i is to assume some relative velocity differential v_{rel} (positive or negative) to cover the distance difference $||d_i - d_{ref}||$. In this case the deadline becomes $t_i = \frac{||d_i - d_{ref}||}{v_{rel}} + \epsilon$, where ϵ is some tolerance to allow for small variations in the candidate's motion.

However, this simple model ignores the automated nature of platooning and the user experience, as the candidate's velocity is assumed to change instantly rather than smoothly. Alternatively, the verifier can calculate deadlines using an ACC model that accounts for safety and motion smoothness factors. Here, we adopt the ACC control model presented in [14], but any ACC controller can be used. Using this model, the deadline is calculated as follows. Let a challenge d correspond to a gap time of $T = d/\dot{x}_C$ where $\dot{x}_C$ denotes the current speed of the candidate. The algorithm proceeds in steps of duration Δ_t as follows:

(1) The desired acceleration at the n-th step is

$$\ddot{x}_{des}[n] = -\frac{1}{T}(\Delta\dot{x}[n] + \lambda\delta[n]) \quad (1)$$

$$\delta[n] = -d_{act}[n] + d, \quad \Delta\dot{x}[n] = \dot{x}_C[n] - \dot{x}_V[n], \quad (2)$$

where $\Delta\dot{x}[n]$ is the relative velocity between $\mathcal{C}$ and $\mathcal{V}$, $d_{act}[n]$ is the actual following distance, $\delta[n]$ is the distance error to the desired checkpoint d, and $\lambda > 0$ is a design parameter that controls the rate of convergence to d.

(2) Instead of applying $\ddot{x}_{des}[n]$, the acceleration applied involves the input from the previous step:

$$\ddot{x}[n] = \beta \cdot \ddot{x}_{des}[n] + (1-\beta) \cdot \ddot{x}[n-1], \ \beta = \frac{\Delta_t}{\tau + \Delta_t}. \quad (3)$$

Here, τ is a time constant typically set to 0.5s and Δ_t denotes the time gap between the $(n-1)$-st and n-th steps.

(3) The distance gain of $\mathcal{C}$ during Δ_t is computed by

$$l[n] = \dot{x}[n-1] \cdot \Delta_t + \frac{1}{2} \cdot \ddot{x}[n] \cdot {\Delta_t}^2. \quad (4)$$

(4) The distance $\delta[n]$ to the checkpoint d at step n is updated to

$$\delta[n] = \delta[n-1] + l[n] - \dot{x}_V[n] \cdot \Delta_t. \quad (5)$$

(5) Steps 1-4 are iterated through until $|\delta[n]| < \gamma$ where γ is the checkpoint distance tolerance. The deadline t for a checkpoint d is set to $t = \Delta t * n^*$, where n^* is the first value of n for which $|\delta[n]| < \gamma$.

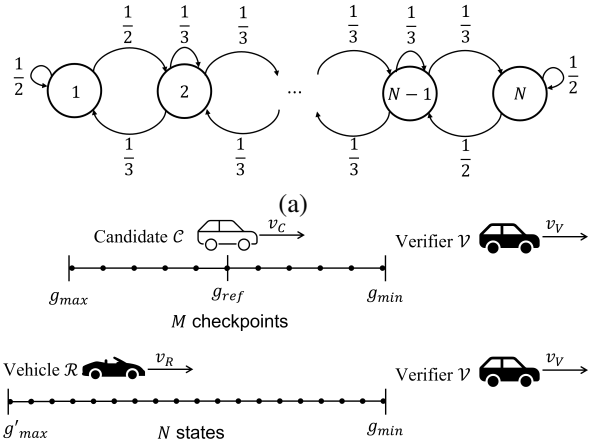


Fig. 5: (a) The Markov chain model for the random walk of vehicle $\mathcal{R}$, (b) the M checkpoints selected by the verifier and the N possible states of vehicle $\mathcal{R}$.

IV. Security Analysis

A. Remote Adversary

We first examine if an adversary that is not platooning can join the platoon. The adversary could be at any location except d_{ref} behind the verifier. For instance, the adversary could be stationary at a remote location, several cars behind the verifier, co-traveling at a different lane, etc. We consider two cases: (a) no other vehicles following the verifier and (b) a vehicle other than the adversary follows the verifier.

(1) No vehicles follow $\mathcal{V}$. Let the adversary $\mathcal{M}$, request to join the platoon by sending $m_M(1)$ to the verifier. The adversary will pass the digital identity verification as he possesses a valid certificate issued by a trusted certificate authority. The verifier will challenge $\mathcal{M}$ with a set of challenges Γ. As no other vehicles follow $\mathcal{V}$, the verifier will be unable to detect a vehicle at the designated checkpoints and the physical verification will fail.

(1) A vehicle follows $\mathcal{V}$. We now consider the case where some vehicle $\mathcal{R}$ other than the adversary follows the verifier. The vehicle $\mathcal{R}$ is not controlled by the adversary, but is in the same lane as the verifier and keeps a safe distance that could be similar to the following distance d_{ref}. The remote adversary requests to join the platoon by sending $m_M(1)$ to the verifier. As mentioned before, the adversary will pass the digital identity verification. The verifier will challenge $\mathcal{M}$ with a set of challenges Γ. The verifier will measure the distance to the following vehicle $\mathcal{R}$ (instead of the remote adversary) at the designated deadlines. The adversary could pass the PoF if $\mathcal{R}$ happens to be at the checkpoints by the respective deadlines.

Modeling $\mathcal{R}$'s trajectory as a random walk: To analyze the probability of passing the PoF, we model the trajectory of $\mathcal{R}$ as a one-dimensional random walk around d_{ref}. The core idea is that $\mathcal{R}$ moves independently of the platoon and may fluctuate its following distance within a limited range while still following. Specifically, the vehicle $\mathcal{R}$ fluctuates its distance to the verifier within a range $[d_{\min}, d_{\max}]$.

The random walk of $\mathcal{R}$ is represented by an N-state Markov chain where states are the candidate positions of $\mathcal{R}$ and state transition probabilities represent the probability of moving to another position within the range after a time step n. We discretize the range $[d_{\min}, d_{\max}]$, by assuming that $\mathcal{R}$ can travel a fixed distance d_{step} within a fixed time step and divide the range by d_{step} to obtain a total of N positions (states). Without loss of generality, the initial state distribution $P^{(0)}$ at time 0 is assumed to be uniform. Moreover the state transition probabilities are given by an $N \times N$ matrix $P = (P_{ij})$ with

$$P_{1,1} = P_{1,2} = P_{N,N} = P_{N,N-1} = 1/2, \tag{6}$$

$$P_{i,i+1} = P_{i,i-1} = P_{i,i} = 1/3, \quad i = 2..N-1, \tag{7}$$

$$P_{i,j} = 0, \quad \text{all other} \quad i, j. \tag{8}$$

The transition state diagram of the random walk is shown in Fig. 5(a). Note that in a typical random walk, there is always a transition to a new state. In our model, we have opted to consider that the vehicle may stay on the same state within a time step. Moreover, given a state, the transition probabilities forward, backward, and at the same state are equiprobable, though any matrix P can be considered. The N candidate states of $\mathcal{R}$ may not necessarily coincide with the M possible checkpoints selected by the verifier. However, we can assume that the M checkpoints are part of the state space of $\mathcal{R}$. Using the random walk model, we now evaluate the probability of passing the PoF verification, considering only the physical challenges and ignoring the initial and final states of d_{ref}.

Proposition 1. *Let the verifier challenge the adversary $\mathcal{M}$ with a set of K challenges $\Gamma = \{(d_1, t_1), (d_2, t_2), \ldots (d_K, t_K)\}$. Each checkpoint is randomly selected from a state space $\mathcal{S}$ of size M. Let some vehicle $\mathcal{R}$ follow the verifier and move in a state space $\mathcal{S}'$ of size N using the random walk model with $\mathcal{S} \subseteq \mathcal{S}'$. The probability that $\mathcal{M}$ passes the PoF verification due to $\mathcal{R}$'s motion is given by*

$$\mathbf{P}_M = \left(\frac{1}{NM}\right)^K \prod_{k=1}^{K} \sum_{i=1}^{M} \sum_{j=1}^{N} P_{j,i}^{\sum_{\ell=1}^{k} n_k}. \tag{9}$$

where P^n indicates the transition probability matrix after n steps and the proof is provided in [15] Sec. 4.2.2.

Lemma 1. *The adversary's passing probability is upper bounded by*

$$P_M \leq \left(\frac{1}{M}\right)^K. \tag{10}$$

The proof can be found in a detailed version [15] Sec. 4.2.2. From Lemma 1, we observe that the passing probability P_M drops *at least* inversely proportional to the cardinality M of the checkpoint space and exponentially with the number of challenges K. By controlling these two parameters, the passing probability can be driven to any desired value at the expense of delay until the PoF verification is completed.

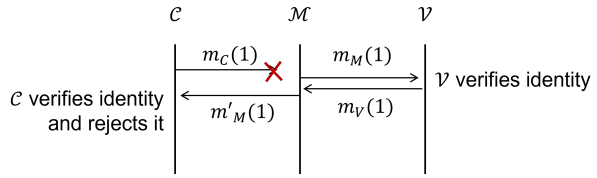


Fig. 6: A MiTM attack. ID_V and pk_V are known to $\mathcal{C}$

B. A MiTM Adversary

In a MiTM attack, the adversary attempts to be admitted to the platoon when a valid candidate initiates a join request with the verifier. We assume the candidate attempts to join a specific platoon with a known verifier identified by his public key pk_V and his certificate $cert_V$.

Let the candidate target a specific platoon identified by verifier with $(ID_V, pk_V, cert_V)$. The steps of a MiTM attack are shown in Fig. 6(a). The candidate initializes the protocol by sending a join request message $m_C(1)$ to $\mathcal{V}$. The request $m_C(1)$ contains the ID_C and ID_V, signed with the candidate's private key. The adversary can attempt to initiate parallel sessions by eliminating $m_C(1)$ (e.g., via jamming) and injecting his own request to join $\mathcal{V}$.

$$m_M(1) \leftarrow ID_V, ID_M, pk_M, cert_M, sig_{sk_M}(\text{REQ}, ID_M, ID_V).$$

Upon receiving $m_M(1)$, the verifier validates the digital identity of $\mathcal{M}$ and challenges $\mathcal{M}$ with Γ. Because the adversary is not following the platoon, the only chance to successfully complete the MiTM attack is for the valid candidate to execute the physical challenges Γ. The adversary can attempt to respond to $\mathcal{C}$'s initial message $m_C(1)$ by sending

$$m'_M(1) \leftarrow ID_C, E_{pk_C}[sig_{sk_M}(\Gamma, ID_M, ID_C, t_0), \Gamma, ID_M, ID_C, t_0],$$

containing the same set of physical challenges Γ, provided by $\mathcal{V}$ to $\mathcal{M}$. However, $\mathcal{C}$ will abort the joining process because the reply is signed by $\mathcal{M}$ and not $\mathcal{V}$. Thus, the MiTM attack fails because $\mathcal{C}$ can only accept challenges signed by $\mathcal{V}$.

V. Evaluation

In this section, we evaluate the security and performance of the *Wiggle* protocol. All platooning experiments were performed in the Plexe simulation environment [16], which is a cooperative driving framework. It features realistic vehicle dynamics and several cruise control models, enabling the analysis of mixed scenarios in traffic.

A. Performance of Wiggle

We first evaluated the performance of *Wiggle* as a function of the protocol parameters. In our simulation, a verifier $\mathcal{V}$ was followed by a candidate $\mathcal{C}$ in a freeway environment. The candidate applied the ACC model presented in Section III-B to control its following distance from $\mathcal{V}$. The simulation parameters are listed in Table I.

Studying the impact of the ACC. The ACC parameters control the deadline for reaching each checkpoint. Parameter λ, in particular, regulates the vehicle acceleration as a function of the distance to the checkpoint. Figure 7 shows the candidate's acceleration, velocity, and following distance when the checkpoint is 3m away from d_{ref}. From Fig. 7(a), we observe that acceleration is gradually decreased, and then the vehicle brakes until the checkpoint is reached. The speed differential hardly exceeds 0.6m/sec (2Km/h), indicating an almost imperceptible transition to the checkpoint. We further observe that when λ is decreased to 0.1, the acceleration and velocity differential decrease at the expense of a longer delay. In the remaining of our simulations, we set $\lambda = 0.4$.

TABLE I: Simulation Parameters

Parameter	Value
Initial velocity of $\mathcal{V}$ (v_V) and $\mathcal{C}$ (v_C)	30m/s
Following distance (d_{ref})	$1.5 \cdot v_C$ (45m)
Checkpoint range	$1 \cdot v_C - 2 \cdot v_C$ (30−60m)
# of checkpoints in range (M)	51
Update step of ACC (Δ_t)	0.1s
ACC parameter λ	0.4
Checkpoint error tolerance (γ)	0.3m

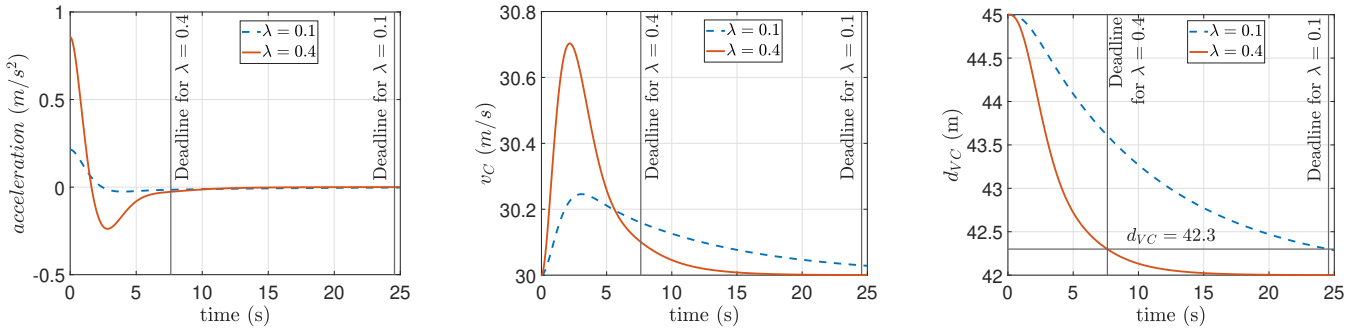
Another important parameter that impacts delay is the distance tolerance γ by which the checkpoint must be reached. Figure 7 shows that the candidate quickly converges in the vicinity of the checkpoint and then fine-tunes its position. By increasing the distance tolerance, the deadline can be shortened. Figure 8(a) indeed shows that the deadline duration is inversely related to γ. We selected $\gamma = 0.3$m, which is close to the typical automotive radar resolution [17].

Finally, in Fig. 8(b), we show the deadline as the function of the distance differential to the checkpoint. The deadline grows with distance but the relationship is not linear. This is justified by the acceleration model of the ACC model. We also note that the deadlines are not symmetric when the same distance has to be covered forward and backward as slightly different accelerations are applied in each direction.

Impact of traffic. So far, the verifier moved at constant velocity when the candidate responded to physical challenges. However, traffic may impact the velocity of the verifier and the way that the candidate's ACC approaches a checkpoint. To study this impact, we simulated a vehicle proceeding the verifier traveling at 27m/s. To maintain a safe distance, $\mathcal{V}$ matches the slower vehicle's velocity while the candidate is attempting to reach a checkpoint. Figure 9 shows the acceleration, velocity, and following distance of the candidate for a checkpoint that is 42m away from $\mathcal{V}$. We observe that the time to reach the checkpoint increased from 7.6sec (according to Fig. 7) to 13.6sec. This indicates that a valid candidate will fail the original deadline if the velocity of the verifier changes.

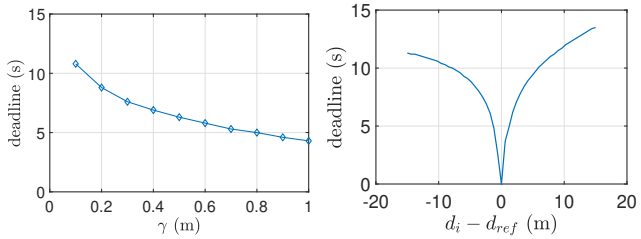
There are two approaches to remedy this problem. The first is to ignore any challenges for which the verifier's velocity changes drastically and repeat them when the velocity stabilizes. The second approach is for the verifier to adjust the deadline based on his own velocity. Given the ACC model, the verifier can re-compute the deadline to allow for the candidate to reach the checkpoint.

Verification time as a function of physical challenges K. The verification time also depends on the number of physical



(a) acceleration, (b) velocity (c) following distance

Fig. 7: The acceleration, velocity, and distance of $\mathcal{C}$ to reach checkpoint $d = 42$m from $d_{ref} = 45$m, when $v_C = 30$m/s and λ is set to 0.1 and 0.4.



(a) (b)

Fig. 8: (a) The deadline duration as a function of the checkpoint distance threshold γ for a checkpoint 3m away from d_{ref} and (b) the deadline duration as a function of the distance covered by checkpoints, when $\gamma = 0.3$m.

challenges issued by the verifier. Indeed, this relationship is expected to be linear as the verification delay is cumulative with every challenge. Variations are due to the variability of the deadlines for randomly selected checkpoints. To study the impact of K, we fixed the checkpoint space to $M = 51$ and varied K while executing *Wiggle*. Figure 10(a) shows the average verification time and its standard deviation as a function of K. We observe the expected linear increase in verification time, with about 10sec overhead per physical challenge. Overall, the verification time is short (less than a minute) relative to the time that the candidate will be platooning with the rest of the platoon. Figure 10(b) shows the average verification time as a function of the number of available checkpoints M, when $K = 5$. As the range of motion of the candidate expands, the verification time increases due to the longer average distance to reach each checkpoint.

B. Security of Wiggle

In Section IV-A, we showed that a remote adversary is unable to pass the PoF verification without performing the physical challenges. The only chance occurs if some independent vehicle $\mathcal{R}$ follows the verifier at the platooning distance. We evaluated the probability that $\mathcal{M}$ passes verification due to $\mathcal{R}$'s motion, as stated in Proposition 1. We simulated $\mathcal{R}$ following a verifier traveling at 30m/sec. The vehicle $\mathcal{R}$ executed a random walk within the checkpoint range (30m - 60m from the verifier) with a step size of 0.3m (i.e., N=100 Markov states). The verifier continuously issued physical challenges with a distance tolerance of $\gamma = 0.3$m. Figure 11(a) shows an instance of $\mathcal{R}$'s following distance to $\mathcal{V}$ as a function of time for five checkpoints. Figure 11(b) shows the distance of $\mathcal{R}$ from each checkpoint at each deadline. We observe that $\mathcal{R}$ is often at a location far away from the respective checkpoint since it does not try to reach it intentionally.

TABLE II: Comparison with related work

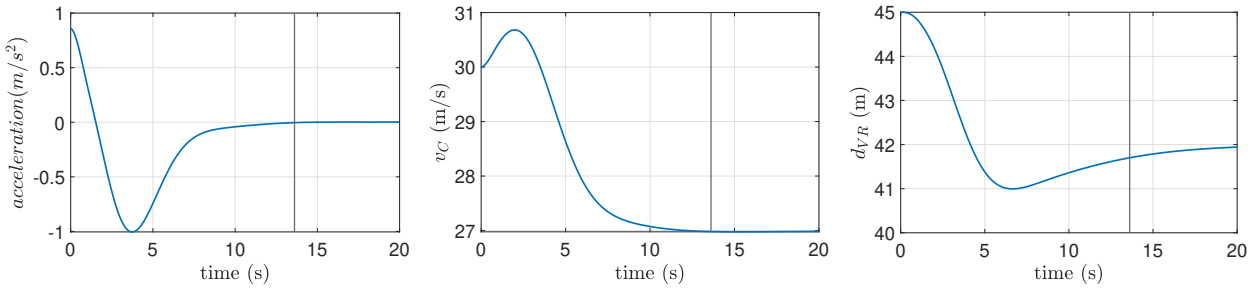
Reference	pre-recording attack	MitM attack	Remote attack	Lane/order verification
Wiggle	✓	✓	✓	✓
[6]	✗	✗	✓	✗
[?]	✓	✓	✓	✗
[8]	✗	✗	✓	✗
[10]	✓	✓	✓	✓

This is further verified in Fig. 12(a) that shows $\mathcal{M}$'s passing rate as a function, calculated at over 2,000 challenges. Note that after $K = 2$, $\mathcal{M}$ did not pass any of the PoFs. For comparison, we also provide P_M when calculated numerically using Proposition 1. A few physical challenges are sufficient to drive the probability of success to very low values. Note that the checkpoint space cardinality M does not affect P_M. This is because $\mathcal{R}$ must reach one specific checkpoint by the deadline. It is fairly straightforward to show that under a random walk, this probability follows the uniform distribution (with slightly higher probabilities for the two boundaries). Therefore, regardless of M, P_M is approximately equal to $(1/N)^K$, as it is also observed in Fig. 10(b).

Finally, Table II compares *Wiggle* with prior methods in terms of security. The check labels indicate protection against a particular attack. Only [10] provides as strong security guarantees as *Wiggle*, but requires multiple verifiers and time synchronization for performing time-of-flight measurements. These are difficult to achieve in practical systems.

VI. Conclusion

We proposed *Wiggle*, a physical challenge-response protocol for controlling physical access to a platoon. *Wiggle* uses random perturbations of the following distance to bind the digital identity of a candidate to his claimed trajectory. We showed that *Wiggle* can verify the following distance of the



(a) acceleration (b) velocity (c) following distance

Fig. 9: The acceleration, velocity, and distance of $\mathcal{C}$ from $d_{ref} = 45$m to reach checkpoint $d = 42$m, when the velocity of the verifier reduces from $30m/s$ to $27m/s$ during the verification.

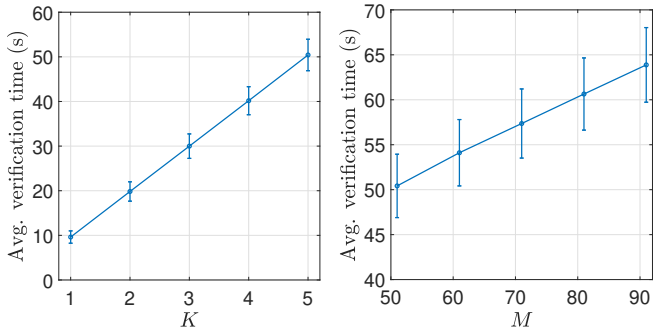
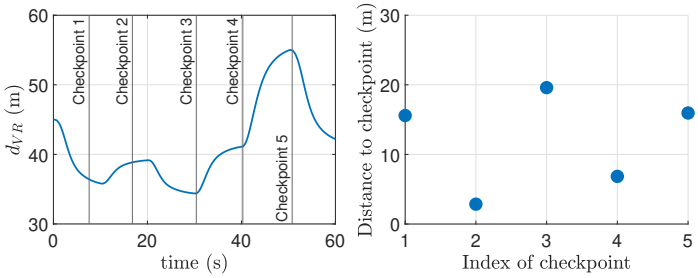


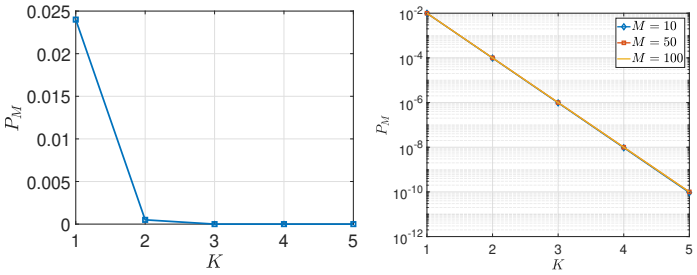
Fig. 10: Verification time as a function of the number of challenges K and available checkpoints M.



(a) (b)

Fig. 11: (a) The distance between $\mathcal{R}$ and $\mathcal{V}$ as a function over five challenges, (b) the distance difference between the vehicle $\mathcal{R}$ and the checkpoints at each deadline.

candidate, the relative positioning of the candidate and the verifier, the candidate's lane, and provide resistance to pre-recording attacks. We evaluated the performance and security of *Wiggle* in the Plexe simulator and showed that a PoF verification lasts less than a minute while inducing almost imperceptible changes to the vehicle's velocity.



(a) simulation (b) numerical

Fig. 12: The simulated and numerical passing probability $\mathbf{P}_M$ as a function of the number of challenges K.

References

[1] V. Turri, B. Besselink, and K. H. Johansson, "Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning," *IEEE TCST 2016*.

[2] N. Lyamin, Q. Deng, and A. Vinel, "Study of the platooning fuel efficiency under ETSI ITS-G5 communications," in *Proc. of IEEE ITSC 2016*.

[3] Z. Wang, G. Wu, and M. J. Barth, "A review on cooperative adaptive cruise control (cacc) systems: Architectures, controls, and applications," in *Proc. of IEEE ITSC 2018*.

[4] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Certificate Management Interfaces for End Entities*, IEEE Std. IEEE 1609.2.1, 2020.

[5] *3GPP:Technical Specification Group Services and System Aspects;Security aspect for LTE support of Vehicle-to-Everything (V2X) services Rel-16, V16.0.0*, 3GPP Std. TS 33.185, Jul. 2020.

[6] J. Han, M. Harishankar, X. Wang, A. J. Chung, and P. Tague, "Convoy: Physical context verification for vehicle platoon admission," in *Proc. of the HotMobile 2017*.

[7] Z. Xu, J. Li, Y. Pan, L. Lazos, M. Li, and N. Ghose, "PoF: Proof-of-following for vehicle platoons," in *Proc. of the NDSS Symposium 2022*. [Online]. Available: https://doi.org/10.147222/ndss.2022.23077

[8] C. Vaas, M. Juuti, N. Asokan, and I. Martinovic, "Get in line: Ongoing co-presence verification of a vehicle formation based on driving trajectories," in *2018 IEEE EuroS&P*.

[9] M. Juuti, C. Vaas, I. Sluganovic *et al.*, "Stash: Securing transparent authentication schemes using prover-side proximity verification," in *2017 SECON*.

[10] A. Studer, M. Luk, and A. Perrig, "Efficient mechanisms to provide convoy member and vehicle sequence authentication in vanets," in *Workshops-SecureComm 2007*.

[11] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "Plexe: A platooning extension for veins," in *2014 VNC*.

[12] D. J. Yeong, G. Velasco-Hernandez, J. Barry, J. Walsh *et al.*, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, 2021.

[13] M. Singh, M. Röschlin, A. Ranganathan, and S. Capkun, "V-range: Enabling secure ranging in 5g wireless networks," in *Proc. of the NDSS Symposium 2022*.

[14] R. Rajamani, *Vehicle dynamics and control*. Springer, 2011.

[15] C. Dickey, C. Smith, Q. Johnson, J. Li, Z. Xu, L. Lazos, and M. Li, "Wiggle: Physical challenge-response verification of vehicle platooning," *arXiv preprint arXiv:2209.00080*, 2022.

[16] M. Segata, R. L. Cigno, T. Hardes *et al.*, "Multi-technology cooperative driving: An analysis based on plexe," *IEEE TMC*, 2022.

[17] C. Waldschmidt, J. Hasch, and W. Menzel, "Automotive radar—from first efforts to future systems," *IEEE Journal of Microwaves 2021*.