Convoy: Physical Context Verification for Vehicle Platoon Admission

Jun Han, Madhumitha Harishankar, Xiao Wang, Albert Jin Chung, and Patrick Tague
Carnegie Mellon University
firstname.lastname@sv.cmu.edu

ABSTRACT

Truck platooning is emerging as a promising solution with many economic incentives. However, securely admitting a new vehicle into a platoon is an extremely important yet difficult task. There is no adequate method today for verifying physical arrangements of vehicles within a platoon formation. Specifically, we address the problem of a *platoon ghost attack* wherein an attacker spoofs presence within a platoon to gain admission and subsequently execute malicious attacks. To address such concerns, we present *Convoy*, a novel autonomous platoon admission scheme which binds the vehicles' digital certificates to their physical context (i.e., locality). *Convoy* exploits the findings that vehicles traveling together experience similar context to prove to each other over time that they are co-present. Specifically, they experience similar road (e.g., bumps and cracks) and traffic (e.g., acceleration and steering) conditions. Our approach is based on the ability for vehicles to capture this context, generate fingerprints to establish shared keys, and later bind these symmetric keys to their public keys. We design and implement the *Convoy* protocol and evaluate it with real-world driving data. Our implementation demonstrates that vehicles traveling in adjacent lanes can be sufficiently distinguished by their context and this can be utilized to thwart platoon ghost attacks and similar misbehavior.

Keywords

Vehicle Platoons; Authentication; Context Verification

1. INTRODUCTION

Amongst the advances in smart vehicles [20, 1] platooning is an emerging one that is achieving considerable traction. Vehicle platooning is a system of coordinated driving, where participating vehicles drive in a single file or platoon; each vehicle strictly follows the preceding vehicle, with the foremost vehicle in the formation as the platoon leader. Truck platooning, in particular, is beneficial in increasing driving safety, convenience, and fuel and road efficiency (due to reduced aerodynamic drags) [14, 22]. These economic incentives are causing platooning to emerge as a real-world solution adopted by many in the commercial trucking industry [5, 4, 7]. To further exploit these benefits, trucks (of different companies) are envisioned to freely join and leave a platoon on the road in an ad-hoc fashion as all members (including the leader vehicle) enjoy the benefits.

Platooning vehicles send control messages containing information about their acceleration, braking and steering to each other for coordinated driving. Specifically, they use Dedicated Short-Range Communications (DSRC) and Wireless Access in Vehicular Environments (WAVE) as the de facto standards for vehicle-to-vehicle (V2V) communications [16, 13]. Because the control messages need to be secured, authenticating each others' public keys is an important step while a new member is joining a platoon. The current DSRC/WAVE model assumes Public Key Infrastructure (PKI) to authenticate using certificates signed by a trusted third party, such as a Certificate Authority (CA).

Unfortunately, this solution is insufficient in a platoon setting. A platooning system that does not account for the verification of relative positions of the vehicles during admission (to ensure that it is indeed in line with the platoon) is critically flawed. The expected single file formation is fundamental to platooning, and any accidental or intentional misbehaviour that tampers with this property can have significant repercussions. The platooning vehicles hence become susceptible to various attacks such as impersonating as non-existing "ghost vehicles" [15, 8] such that the attacker may forge the control messages to induce a collision without physically being in the platoon thereby avoiding the collision. Such attacks are detrimental as they could cause life-threatening accidents, damage to high-value vehicles and cargo, and loss of business.

The root cause of the aforementioned problems is that the vehicles have no way of binding their *locality* information together with the corresponding *physical identity* and *public key*. Verifying the certificate is limited as the certificate merely binds a vehicle's physical identity (e.g., license plate) to a digital public key, but cannot associate this with the relative physical presence of the vehicle. This is exemplified in Figure 1, where *Cars A* and *B* are vehicles in an existing platoon. *Car C* is a vehicle that wishes to join the platoon that has a valid certificate from a trusted CA, and *Car M* is an attacker's car driving in an adjacent lane, also with a

HotMobile '17, February 21-22, 2017, Sonoma, CA, USA

DOI: http://dx.doi.org/10.1145/3032970.3032987

valid certificate from a trusted CA. In this example, *Cars A* and *B* receive certificates of *Car C* and *Car M*, but are not able to associate each certificate with the correct car to the extent of determining that *Car C* is in valid formation and can be admitted to the platoon, while *Car M* is not.

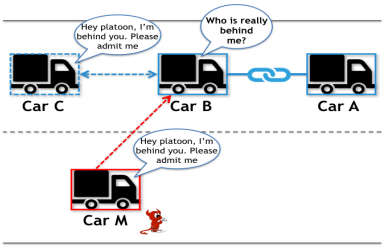

Figure 1: Overview diagram depicting the vulnerabilities of current platooning system to impersonation attacks.

One might posit that relative physical presence can be asserted by GPS information. However, simply leveraging GPS will not help because a remote vehicle cannot verify the validity of the coordinates as GPS information is known to be susceptible to spoofing attacks, in addition to a non-negligible error margin (up to 10 meters) [19, 11].

To address the above problem, we present *Convoy*, an autonomous authentication and verification scheme of platooning vehicles. *Convoy* is based on the findings that vehicles wishing to form a platoon can prove to each other that they are indeed traveling together using context information captured from their sensor data. *Convoy* leverages road and traffic conditions that the platoon is subject to at any time as sources of entropy to establish pairwise symmetric cryptographic key between vehicles, binding to the physical context. The symmetric key is then used to authenticate the vehicles' digital certificates.

We face several challenges in designing the proposed *Convoy* mechanism. First, different vehicles wishing to join a platoon (e.g., *Cars A, B* and *C*) would experience similar but different context, leading to numerically unequal signals. To account for the subtle differences in the signals between the vehicles, *Convoy* makes use of an emerging cryptographic primitive called the *Fuzzy Commitment* [12, 18, 17] that relies on error-correcting codes to establish a shared symmetric key from similar-but-unequal signals. Second, a fundamental requirement in designing *Convoy* is that the context experienced by vehicles in adjacent lanes is sufficiently distinct. This is a factor of the inherent road and traffic conditions. Hence, *Convoy* requires vehicles to repeat the protocol for multiple iterations over time, thereby increasing the probability that vehicles traveling together experience more similar context. We evaluate *Convoy* by mounting an accelerometer on two different cars to capture real-world driving data and demonstrate that we can sufficiently distinguish two cars driving on the same lane (*even with different cars*) as opposed to driving on an adjacent lane (*even with a same car*).

We summarize the following contributions:

- We present *Convoy* to build trust relationships among vehicles wishing to form a platoon by verifying physical context and co-presence. Our approach protects the platoon admission process against potential impersonation attacks.
- We design the *Convoy* protocol to leverage inherently random road and traffic conditions, making it extremely challenging for an attacker to consistently mimic or predict the conditions.
- We implement and evaluate *Convoy* by instrumenting two different cars on a highway to demonstrate the feasibility of distinguishing cars in two adjacent lanes using only accelerometer data.

2. SYSTEM MODELS

In this section, we present our models and assumptions for vehicle platooning and the attacker of interest.

Platoon Model. Platoons are typically set up with a manually-driven lead vehicle with semi-autonomous followers [22]. In our work, assume that a candidate vehicle is only admitted to the platoon after the rear-most platoon vehicle validates the position and identity of the candidate. We suppose that the candidate will initially follow the platoon using Adaptive Cruise Control (ACC) [21] without explicit coordination, until it can be verified and admitted to the platoon. Once admitted, members are declared to be trustworthy and thereby earn the benefits of efficiency and safety offered by platooning [6]. To enable the coordinated acceleration among vehicles, vehicle-to-vehilcle (V2V) communication is employed. As platoons travel amongst other traffic, it is critical for them to communicate securely. We thus assume that all control messages (e.g., acceleration, brake, and steering messages) are encrypted with a group symmetric key known only to the platoon members, though we do not address group key management in this work.

Attacker Model. We consider a ***Platooning Ghost Attack***, where the attacker's goal is to impersonate a non-existing "ghost" vehicle in the platoon. By pretending to be in the platoon formation, and hence gaining admittance to the platoon, the attacker gains knowledge of the control commands (i.e., acceleration, braking, and steering information) from the preceding vehicles relative to the position of the ghost vehicle. The attacker further has control over transmission of the control messages to its succeeding vehicles. Hence, the attacker effectively controls certain aspects of the platoon. The attacker is now capable of launching a variety of attacks as a *platoon insider*, including man-in-the-middle, denial-of-service, and collision induction attacks. More specifically, it may send malicious control messages to its succeeding vehicles to cause it to crash into the rest of the platoon in front. It may prevent admissions of newer members of the platoon, or cause existing succeeding vehicles to brake away from the rest of the platoon.

3. DESIGN AND IMPLEMENTATION

In this section, we first discuss the how *Convoy* leverages entropy sources for its protocol. We then present an overview of the *Convoy* protocol design and implementation.

3.1 Overview

The goal of *Convoy* is to allow a vehicle in a platoon to securely verify that a public key indeed belongs to a vehicle following the platoon instead of an attacking car in another lane. We achieve this goal by binding the public key to the

physical context experienced by the vehicles. Specifically, *Convoy* binds keys to the shared context using observations of highly-variable road and traffic conditions, as seen by embedded accelerometers. The measurements form the basis of a symmetric key established between pairs of vehicles that is then bound to the logical vehicle identity carried by the public key. Given this setup, the simplified goal of *Convoy* is to map sensor measurements to a shared symmetric key that can be verified by the two vehicles.

Sources of Entropy. Since the problem reduces to symmetric key establishment, an important preliminary question points to the entropy that can be extracted from sensors embedded in neighboring vehicles. The key source of entropy that *Convoy* leverages for key agreement is the unique and highly variable *road and traffic conditions*, as observed by the vehicle's sensors. Road conditions observed by a sensor are dynamic and very difficult to predict at the millisecond scale of observation. Road materials and conditions such as patches, bumps, and cracks provide useful data for comparison across vehicle. In addition, traffic conditions are inherently random as they vary when different vehicles on the road travel together, causing the platoon to accelerate, brake, and steer differently. Moreover, traffic often varies across lanes, especially during periods of heavy congestion. Varying traffic patterns further introduce temporal variation in how road conditions are measured, so these two factors combine to increase the resulting measurement entropy. Such variations from road an traffic conditions can be captured by a single axis of an accelerometer.

Cryptographic Protocol using Commitments. From the example illustrated in Figure 1, in order for *Car C* to prove to *Car B* that it is traveling close behind the platoon, it leverages a *fuzzy commitment scheme*. This scheme translates sensor measurements, represented by an extracted fingerprint F, and a secret K into a commitment and decommitment (or opening) pair (μ, o). This is analogous to one-time pad encryption, where F is used as an encryption key, and K is used as the plaintext to be encrypted. μ can only be opened if one has a fingerprint, $\hat{F}$ that is within a few bit errors of F. The fingerprints F_B and F_C extracted by B and C traveling on the same lane would ideally be within a small margin of error, while F_M extracted by M on a different lane would be more error-prone. By applying an error-correcting code operation, vehicle pairs can verify fingerprint similarity, resulting in a shared key. This can be repeated to build confidence over time, ultimately yielding a key with sufficient entropy and corresponding platoon admission. In this work, we rely on a fuzzy commitment scheme similar to that of previous work [12, 18, 17].

3.2 Protocol Design

Convoy protocol consists of five phases – (1) Initialization, (2) Key Agreement, (3) Key Confirmation, (4) Public Key Verification, and (5) Confidence Score Check phases. We describe each phase in detail with the platoon example depicted in Figure 1. The protocol is summarized in Figure 2.

Initialization Phase. To start the initialization phase of *Convoy*, the platoon leader A broadcasts a beacon message $Beacon_A$ containing current platoon member IDs, their (GPS) locations, and a timestamp. When platoon candidate C receives several beacons, it sends a request $JOIN\_RQST$ to join the platoon. Upon receiving the request, A sends a message $INIT\_VERIF$ to B (the trailing vehicle, in general) and C to initialize the verification process; A names C in this message and includes the measurement duration t_F. At this point, B and C commence the key agreement phase.

***Convoy* PROTOCOL**

Phase 1: Initialization

1. $A \xrightarrow{bcast} All$: $Beacon_A{=}ID_A||ID_B||TS||GPS_A||GPS_B$
2. $C \rightarrow A$: $JOIN\_RQST$
3. $A \rightarrow B, C$: $INIT\_VERIF(ID_C, t_F)$

Phase 2: Key Agreement

4. B : Holds F_B, K_B, (μ_B, o_B)
   C : Holds F_C
5. $B \rightarrow C$: $\mu_B||H(K_B)$
6. C : $\hat{o}_B = Open(\mu_B)$
   : $\hat{K}_B = RS_{dec}(\hat{o}_B)$; $H(\hat{K}_B) \stackrel{?}{=} H(K_B)$
   : $\hat{K}_{BC} = KDF(\hat{K}_B)$

Phase 3: Key Confirmation

7. $B \leftrightarrow C$: Key confirmation messages for K_{BC}

Phase 4: Public Key Verification

8. $B \rightarrow C$: $m_{B_1}||M_{K_{BC}}(m_{B_1})$, where $m_{B_1} = K_B^+||K_A^+$
9. $C \rightarrow B$: $m_C||M_{K_{BC}}(m_C)$, where $m_C = K_C^+$
10. $B \rightarrow A$: $m_{B_2}||M_{K_{AB}}(m_{B_2})$, where $m_{B_2} = K_C^+$

Phase 5: Confidence Score Check

11. B, C : Increment CS_{BC}; Check if $CS_{BC}{>}Thr$
    Repeat Steps 4 - 9 until check passes

Figure 2: *Convoy* protocol overview. Upon successful completion of this protocol, *Car C* is securely admitted to the platoon with existing members *Cars A and B*.

Key Agreement Phase. This phase is performed by the trailing platoon member (B in our example) and the candidate vehicle C. When A triggers the $INIT\_VERIF$ messages, B and C collect accelerometer measurements for a duration of t_F seconds. The vehicles then apply a fingerprint extraction function *extractF()*. B computes fingerprint as $F_B = extractF(Acc_B, t_F)$, where C does the same for its measurements Acc_C. We present the details of our fingerprint extraction algorithm in Section 3.3. Subsequently, B generates K_B using a key generation algorithm $KGen$ that outputs keys of length γ (e.g., 128 bits). B's commitment and opening pair (μ_B, o_B) is then computed as $o_B = RS_{enc}(K_B)$ and $\mu_B = F_B \ominus o_B$, where RS_{enc} and $\ominus$ denote Reed-Solomon (RS) encoding and subtraction in a finite field (analogous to an XOR operation), respectively. Finally, B sends μ_B and $H(K_B)$ to C. The hash is sent so that C can locally verify the opening of the commitment. Upon reception of μ_B, C first tries to open the commitment ($Open(\cdot)$) by inverting the operations using its fingerprints F_C in place of B's commitment, obtaining $\hat{o}_B \approx F_C \ominus \mu_B$. As long as $F_B \approx F_C$, the resulting $\hat{o}_B$ will also be similar to o_B. Applying RS decoding operation will yield a key $\hat{K}_B = RS_{dec}(\hat{o}_B)$ that will be equal to K_B if and only if the input fingerprints F_B and F_C are within the error-correction threshold t of the RS code, $\|F_B - F_C\|_1 \leq t$, where $\|\cdot\|_1$ is the Hamming distance (or ℓ_1 norm), counting the number of bit errors between F_B and F_C. Vehicle C then verifies that the acquired $\hat{K}_B$ value matches those computed by B by check-

ing the original hash received from B as $H(\hat{K}_B) \stackrel{?}{=} H(K_B)$. Upon successful verification, C computes a shared symmetric key, K_{BC} using a Key Derivation Function as $KDF(\hat{K}_B)$. B and C then continue to the key confirmation phase.

Key Confirmation Phase. C initiates the key confirmation phase by leveraging the newly computed K_{BC} to challenge B to verify the same key K_{BC} was derived by both parties. To construct the challenge β_1, C computes a Message Authentication Code (MAC) using K_{BC} over a random nonce n_C such that $\beta_1 = n_C || MAC_{K_{BC}}(n_C)$, and sends β_1 to B. Upon receiving the challenge, B similarly computes K_{BC} as $KDF(K_B)$ and verifies the received MAC using its version of K_{BC}. When this verification succeeds, B similarly creates its own challenge α with nonce n_B, such that $\alpha = (n_B || n_C) || MAC_{K_{BC}}(n_B || n_C)$ and sends α to C, who similarly verifies α. Upon successful verification, C transmits a final MAC β_2 over n_B received from B such that $\beta_2 = n_B || MAC_{K_{BC}}(n_B)$. At this point, both B and C have confirmed mutual agreement upon the symmetric key K_{BC}.

Public Key Verification Phase. With a confirmed symmetric key between the platoon trailer B and candidate vehicle C, the platoon provides verifiable public keys of all platoon members. Specifically, B computes a MAC over the public keys K_A^+ and K_B^+ and transmits the public keys and MAC values to C. C mirrors the process and transmits its public key and corresponding MAC to B. If desired, B can share this information internally within the platoon group, using the shared group key, in case B leaves the platoon before C completes the final phase.

Confidence Score Check Phase. After key confirmation and verification, B increments its (or the group's) confidence score CS_{BC} in candidate vehicle C. If CS_{BC} has surpassed a pre-defined threshold Thr, then C is admitted to the platoon and given access to the group key. Otherwise, C remains a candidate and must repeat the process from the key agreement phase until sufficient confidence is achieved. Use of the confidence score minimizes false positives and ensures that over time, B and C must be traveling together in the same lane.

3.3 Fingerprint Extraction Algorithm and Implementation

The *extractF()* function takes raw accelerometer data and the time duration t_F to encode the signal to a fingerprint F of length l_F bits. The algorithm captures abrupt changes in the data and encodes them into *high bits*, mapping the remaining signal to *low bits*. The encoding algorithm includes: (1) pre-processing, (2) moving average and thresholding, and (3) bit translation. These phases are illustrated in Figure 3, beginning with the raw sensor data in Figure 3(a).

Pre-processing. In the pre-processing phase, we perform a noise reduction step to improve signal fidelity. We achieve this by leveraging spectral subtraction, essentially subtracting the spectral noise amplitude [9]. The result after noise reduction is shown in Figure 3(b).

Moving Average and Thresholding From the higher fidelity signal, we compute the absolute value to capture the magnitude of the samples independent of the sign. We then apply a moving average filter to remove high frequency noise, yielding $S[t]$ (Figure 3(c)). We then apply thresholding to capture sudden changes, where the threshold value Thr_{Deriv} is indicated by the dotted line. The resulting binary signal, $S_{binary}[t]$ (Figure 3(d)), is computed as

$$S_{binary}[t] = \begin{cases} 1, & \text{if } S[t] > Thr_{Deriv} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

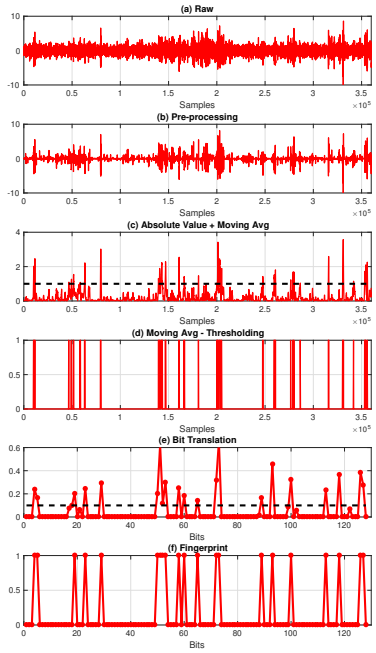

Figure 3: Fingerprint extraction depicting (a) raw data; (b) noise reduction phase; (c) absolute value and moving average; (d) binary signal after thresholding; (e) bit translation phase(total bit length is 128); and (f) extracted fingerprint.

Bit Translation. To convert the binary signal to a fingerprint F, $S_{binary}[t]$ is divided into $l_F = 128$ windows of size $bitWnd$ (i.e., $bitWnd = |S_{binary}[t]|/l_F$). The translated signal is attained as the sum of bits of $S_{binary}[t]$ in each window, as illustrated in Figure 3(e). The final step to extract the fingerprint is to perform an additional thresholding (depicted in dotted black line), yielding binary fingerprint F (Figure 3(f)).

3.4 Entropy Verification

To prevent an attacker from guessing the fingerprint, *Convoy* requires that the fingerprint exceed a certain amount of randomness. We define the *fingerprint weight* $w(F)$ as the fraction of *high bits* in a fingerprint F, capturing the amount of variation in the signal. Hence a fingerprint F with $w(F) = 0.5$ indicates a context that is most unpredictable to guess, as it has equal number of high and low bits. To capture this idea, we define a *fingerprint weight deviation* as $d_w(F)$. The following equations describe how $w(F)$ and $d_w(F)$ are computed.

$$w(F) = \frac{1}{|F|} \sum_i F[i], \qquad d_w(F) = 1 - 2\left|\frac{1}{2} - w(F)\right| \quad (2)$$

Hence a low weight deviation indicates that there are fewer contextual changes, making it easier for the attacker to guess the fingerprint. On the other hand, a high weight deviation

indicates that there are more contextual changes, making it difficult for the attacker to guess. Note that the maximum $d_w(F)$ is 1 when half of the bits are *high*.

Consequently, *Convoy* requires the committing vehicle (B in the example) to compute the fingerprint weight deviation and only transmit the commitment if $d_w(F) > Thr_w$ for a given threashold Thr_w.

4. EVALUATION

We evaluate *Convoy* through experimentation with vehicles in real traffic scenarios. We first describe the experiment setup and then evaluate the effects of road conditions, leaving evaluation of traffic conditions for future work.

4.1 Experiment Setup

We experiment by driving two distinct vehicles (2014 Volkswagen Jetta and a 2012 Subaru Impreza) with trial driving segment spanning over six miles of highway by cruising at 65 mph. We only test the road condition by keeping the traffic condition consistent and delay the traffic condition analysis for future work. Each car was driven in two lanes, with two trials each, yielding a total of 48 miles worth of sensor data. We deployed a triple-axis MEMS accelerometer [10] (with a range of -3 to 3 g sampling at 5KHz) on an Arduino Uno board [3] in the trunk of each car. The z-axis is normal to the road surface to measure road conditions, while the y-axis of the accelerometer points in the direction of travel to measure acceleration due to traffic conditions.

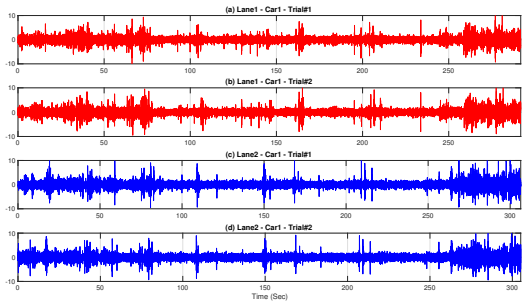

Figure 4: Subsection of accelerometer (Z-Axis) time series data ($\approx$ 5 minutes of drive at 65 mph) of adjacent lanes with two independent trials.

4.2 Fingerprint Similarity

We compare extracted fingerprints from the z-axis accelerometer to evaluate the feasibility of distinguishing between vehicles driving in different lanes, where each fingerprint has a length of 128 bits. Figures 4(a)–(b) and (c)–(d) exemplify the fingerprint similarities between vehicles traveling on the same lane (measured by two trials of the same car). However, comparison across the two pairs depict significant deviance, sufficient to distinguish two adjacent lanes. We discuss our results in three separate cases: similarity between different trials of the same vehicle in the same lane, between different vehicles in the same lane, and between the same vehicle in different lanes. This last case highlights the best-case scenario for an attacker, since the hardware is eliminated as a variable.

Similarity across trials of same vehicle in same lane. We show that the fingerprint pairs created from the same vehicle traveling on same lanes are consistent. We extracted fingerprints from accelerometer data which reflects bumpiness due to imperfection of the road. We repeated this on total of two vehicle models and report the fingerprint similarity of the aggregate result in Figure 5. As the figure shows, high fingerprint similarity is observed in different driving instances of same road with an average of 92.8%. We also note that this result would improve further with usage of lane control modules (such as the Adaptive Cruise Control (ACC)) in a real scenario.

Comparison Pair		p-value
Same Car – Same Lane	Different Car – Same Lane	p=0.60
Same Car – Same Lane	Same Car – Different Lane	p=0.0008
Different Car – Same Lane	Same Car – Different Lane	p=0.003

Table 1: Paired t-test for comparison pairs from Figure 5.

Similarity across vehicles in the same lane. As the same vehicle traveling in the same lane creates consistent fingerprints, we perform additional evaluation to confirm whether the fingerprint similarity is retained as we change vehicles. Again, we use all possible pairs of fingerprints created from accelerometer data. We report the resulting average fingerprint similarity result of 90.6% in the same figure. While the data trends show slight degradation, the fingerprints remain fairly consistent.

Similarity across lanes. We next compare fingerprint similarities for the same vehicle in adjacent lanes. Using the same vehicle minimizes the effect of mechanical variation and reflects a benefit for the attacker. We perform the fingerprint extraction and compare the fingerprint similarity between two different lanes traveled by the same vehicle and report the aggregate result of 81.6% in the same figure.

We also present a set of p-values that compares how fingerprint similarity compares between the Same-Car-Same-Lane (SCSL), Different-Car-Same-Lane (DCSL), and Same-Car-Different-Lane (SCDL) conditions, as depicted in Table 1. The comparison between Same-Lane conditions (SCSL vs. SCDL) yielded p-value of 0.60, showing that these two are not significantly different. However, the comparison between any Same-Lane conditions with Same-Car-Different-Lane condition (SCSL vs. SCDL and DCSL vs. SCDL) yielded 0.0008 and 0.003 respectively, showing significant difference in both comparisons.

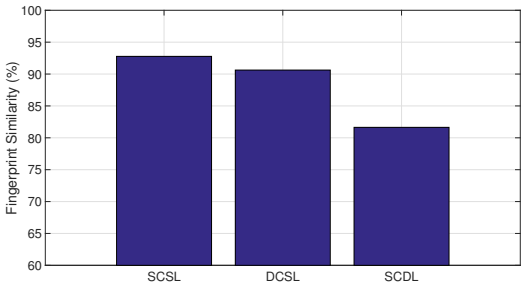

Figure 5: Comparison of fingerprint similarity due to road conditions for ***Same-Car-Same-Lane (SCSL)***, ***Different-Car-Same-Lane (DCSL)***, and ***Same-Car-Different-Lane (SCDL)***

5. DISCUSSION

We now present two main discussion points of *Convoy*.

Road conditions in different cities. The experiments were performed in relatively newer roads in California, which do not have considerable wear and tear. However, given that *Convoy* shows promising performance even with such conditions, we expect to find higher variations from road segments of cities subject to more severe weather conditions.

Sensor placement in trucks. While we report experimental results by driving two sedans, we note that the accelerometer readings from trucks will most likely yield similar results with trivial adjustments such as minor changes to the signal processing algorithm as well as a more careful sensor placement. We note that platooning trucks could place their sensors in locations more sensitive to road conditions and truck movements (e.g., perhaps below the chassis).

Pre-shared keys. One may propose trucks from same vendors to share keys in advance. However, such solutions are not sufficient because of two reasons. First, truck platooning envisions supporting trucks on the road to form a platoon in an ad-hoc fashion regardless of their vendors. Second, even in the extreme case of platoon formation among trucks from same vendors, key pre-sharing approach is inherently vulnerable to insider attack, where a supposedly valid truck turns malicious and launches a ghost attack. *Convoy* addresses such problems because it provides the trucks supplemental guarantee of their physical arrangements.

6. RELATED WORK

We present related work on contextual authentication and secure vehicular networks.

Contextual authentication. Ambient contextual information has been studied for the purpose of authentication. Researchers study secure pairing of devices via a scheme that relies on fingerprinting of similar ambient context by co-present devices. [17, 18]. *Convoy* also leverages contextual information but incorporates a novel method of leveraging traffic and road conditions as sources of entropy.

Secure vehicular networks. Many researchers have proposed using traditional DSRC/WAVE [13, 2] security mechanisms in vehicular systems. They leverage the PKI for authenticating V2V communication, leaving the system vulnerable to spoofing and forging threats such as the Sybil attacks. To mitigate such attacks, researchers propose reputation systems [23]. However, none of these mechanisms consider binding locality information for physical context to the digital certificate.

7. CONCLUSION AND FUTURE WORK

We propose *Convoy* to secure trucks admissions into a platoon by verifying physical context. *Convoy* is novel because it leverages inherent randomness from road and traffic conditions to autonomously bootstrap a shared cryptographic key that is used by vehicles to securely bind physical context, or locality information to digital identifiers, or certificates. We implement and evaluate the *Convoy* fingerprint verification scheme against real-world driving data collected from two different vehicles, and demonstrate the feasibility of sufficiently differentiating between adjacent lanes using only single axis of an accelerometer data. As our future work, we plan to conduct more rigorous experiments covering longer segments with varying traffic conditions. We also plan to provide a robust defense against potential replay attacks on a targeted vehicles on specific road segments.

8. ACKNOWLEDGEMENTS

We thank Kashish Mittal and the reviewers for valuable comments. This research was supported in part by the National Science Foundation under grant CNS-1645759. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of CMU, NSF, or the U.S. Government or any of its agencies.

9. REFERENCES

[1] Google self-driving car project. https://www.google.com/selfdrivingcar/.
[2] Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application. DOT HS 812 014, aug 2014.
[3] Arduino/Genuino UNO. https://www.arduino.cc/en/Main/arduinoBoardUno, 2015.
[4] Daimler Trucks is Connecting Its Trucks With The Internet. http://media.daimler.com/deeplink?cci=2742821, 2016.
[5] European Truck Platooning Challenge – Creating Next Generation Mobility. https://www.eutruckplatooning.com/home/default.aspx, 2016.
[6] Impact of Platooning on Traffic Efficiency. https://trid.trb.org/view.aspx?id=1262546, 2016.
[7] Peloton. http://peloton-tech.com/, 2016.
[8] N. Bissmeyer, J. Njeukam, J. Petit, and K. M. Bayarou. Central misbehavior evaluation for vanets based on mobility data plausibility. In *ACM VANET*, 2012.
[9] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1979.
[10] A. Devices. ADXL335 Datasheet. http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL335.pdf.
[11] N. Drawil, H. Amar, and O. Basir. Gps localization accuracy classification: A context-based approach. *IEEE ITS*, 2013.
[12] A. Juels and M. Sudan. A fuzzy vault scheme. In *IEEE International Symposium on Information Theory*, 2002.
[13] J. Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 2011.
[14] M. P. Lammert, A. Duran, J. Diez, K. Burton, and A. Nicholson. Effect of Platooning on Fuel Consumption of Class 8 Vehicles Over a Range of Speeds, Following Distances, and Mass. SAE Technical Report, 2014.
[15] C. Laurendeau and M. Barbeau. *Ad-Hoc, Mobile, and Wireless Networks: 5th International Conference, ADHOC-NOW*, chapter Threats to Security in DSRC/WAVE. Springer Berlin Heidelberg, 2006.
[16] Y. J. Li. *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, chapter An Overview of the DSRC/WAVE Technology. 2012.
[17] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani. Context-based zero-interaction pairing and key evolution for advanced personal devices. ACM CCS, 2014.
[18] D. Schurmann and S. Sigg. Secure communication based on ambient audio. *IEEE Transactions on Mobile Computing*, 12(2):358–370, Feb. 2013.
[19] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun. On the requirements for successful gps spoofing attacks. ACM CCS, 2011.
[20] Uber. Pittsburgh, your self-driving uber is arriving now. https://newsroom.uber.com/pittsburgh-self-driving-uber/.
[21] B. van Arem, C. J. G. van Driel, and R. Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE ITS*, 2006.
[22] Will Knight. 10-4, Good Computer: Automated System Lets Trucks Convoy as One. MIT Technology Review, May 2014.
[23] J. Zhang. A survey on trust management for vanets. In *Proceedings of the 2011 IEEE International Conference on Advanced Information Networking and Applications*, AINA'11.