FastTrust: Fast and Anonymous Spatial-Temporal Trust for Connected Cars on Expressways

Chen Lyu*†, Amit Pande‡, Yuanyuan Zhang†, Dawu Gu†, Prasant Mohapatra‡
*Department of Computer Science and Technology, Shanghai University of Finance and Economics, Shanghai, China
Email: lyu.chen@mail.shufe.edu.cn
† Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
Email: {chen_lv, yyjess, dwgu}@sjtu.edu.cn
‡Department of Computer Science, University of California, Davis, CA
Email:{pande, pmohapatra}@ucdavis.edu

***Abstract*—Connected cars have received massive attention in Intelligent Transportation System. Many potential services, especially safety-related ones, rely on spatial-temporal messages periodically broadcast by cars. Without a secure authentication algorithm, malicious cars may send out invalid spatial-temporal messages and then deny creating them. Meanwhile, a lot of private information may be disclosed from these spatial-temporal messages. Since cars move on expressways at high speed, any authentication must be performed in real-time to prevent crashes. In this paper, we propose a Fast and Anonymous Spatial-Temporal Trust (FastTrust) mechanism to ensure these properties. In contrast to most authentication protocols which rely on fixed infrastructures, FastTrust is distributed and mostly designed on symmetric-key cryptography and an entropy-based commitment, and is able to fast authenticate spatial-temporal messages. FastTrust also ensures the anonymity and unlinkability of spatial-temporal messages by developing a pseudonym-varying scheduling scheme on cars. We provide both analytical and simulation evaluations to show that FastTrust achieves the security and privacy properties. FastTrust is low-cost in terms of communication and computational resources, authenticating 20 times faster than existing Elliptic Curve Digital Signature Algorithm.**

I. Introduction

In recent years, connected cars or connected vehicles have become immensely popular due to their potential in enhancing users' safety and convenience, and avoiding traffic accidents and congestions. It is also regarded as a technology to provide more data to improve the better performance of autonomous driving technologies [1]. By obtaining real-time data through communication, a connected car could accurately and rapidly recognize another car. Meanwhile, the connected car could make motion planning and avoid obstacles, improving safety and traffic efficiency.

In the past year, two significant connected cars have been launched: 2017 E-Class developed by Mercedes-Benz [2] and 2017 CTS Sedan developed by General Motors [3]. Both of them are able to connect with not only Internet but also another car by wireless channel in practice. This opens opportunities for a range of connected vehicles' applications. For example, Forward Collision Warning (FCW) application enables a car to keep a safe distance from cars in front by exchanging the speed and location information on highways. The Intersection Collision Warning (ICW) application alerts the driver of approaching nearby cars which might otherwise be invisible due to visibility issues or sharp turns, to avoid the potential traffic accidents at the road's intersection [4].

These cooperative applications rely on cars to frequently broadcast spatial-temporal messages (STMs) including the current time, speed, position and direction information, and process the incoming messages. Recent works [5]–[7] have mentioned such a message as beacon message or basic safety message. In this paper, we consider these terms as interchangeable. We prefer "STM" as it indicates that such a message mainly contains spatial and temporal information. For users' safety on expressways, each connected car is recommended to broadcast an STM with a high frequency (i.e.,1Hz or 10 Hz) [8].

Malicious cars may broadcast invalid STMs to disrupt the normal operation of transportation systems. It poses not only financial loss but also a potential threat to users' lives [7]. Therefore, we must address the problem of broadcast authentication, which ensures that the STMs are sent by valid cars and not modified during connections. Although the IEEE 1609.2 implements broadcast authentication by using Elliptic Curve Digital Signature Algorithm (ECDSA) [8], it is vulnerable to signature flooding attack that verifying signatures of the frequent STMs incurs excessive computational overhead on cars [5], [6]. Therefore, a fast and low-cost broadcast authentication is mandatory for an STM-broadcast system in order to defend against signature flooding attack. Another issue of broadcasting STMs is privacy leakage of connected cars. Frequently exchanging spatial-temporal information among cars could reveal a lot of personal information, such as cars' identities, driving routes and users' personal habits. Hence, a solution to preserve cars' location privacy and anonymity should be another main design goal. However, for additional computational cost introduced by a privacy-preserving scheme, there is an inherent conflict between fast broadcast authentication and privacy. Finding a solution that achieves both fast broadcast authentication and privacy for connected cars is a major challenge in such a system [9]. Especially when cars are moving at high speed on expressways, it becomes more challenging to support frequent real-time STM exchange.

There have been plenty of research that investigated authentication techniques against signature flooding attack in vehicular ad-hoc networks (VANETs). However, their solutions either depend too much on pre-deployed infrastructures in VANETs

incurring high costs, or neglect the privacy requirements. Other pieces of research works which discussed privacy model for STMs did not consider the security requirements. In this work, we propose a Fast and Anonymous Spatial-Temporal Trust (FastTrust) mechanism, trying to address the problem of "fast broadcast authentication with privacy" for fast-moving cars. To the best of our knowledge, our work is the first attempt for a fast and low-cost broadcast authentication mechanism while preserving the desired privacy in the context of car-to-car connections.

To target a wide range of applications, FastTrust is distributed, enabling cars to verify signatures of STMs on their own instead of involving a third-party entity, such as an infrastructure or another car. In contrast to most existing authentication schemes which are built on public-key cryptography, we design the framework of our authentication with hash chains to enable low-cost authentication. In order to provide real-time and fast verification of STMs, FastTrust employs symmetric keys and an entropy-based commitment constructed by Huffman Hash Trees (HHTs) [10] for authentication. The mechanism also guarantees cars' privacy. Multiple pseudonymous certificates are utilized by each car to reveal different valid identity to other cars. We examine pseudonym linkability attack, and develop a pseudonym-varying scheduling scheme to satisfy the anonymity and unlinkability requirements while also supporting fast authentication of STMs.

The main contributions of our work are as follows:

- A distributed FastTrust mechanism is introduced to achieve fast broadcast authentication and privacy for connected cars. No additional third parties, i.e, infrastructures or cars, are required to be involved in our system.
- A fast authentication protocol is mainly designed on hash chains and symmetric keys to mitigate signature flooding attack and achieve secure authentication. To support real-time verification, an entropy-based commitment is constructed with HHTs by exploiting the predictability of STMs in our protocol.
- A pseudonym-varying scheduling scheme is developed to protect users' privacy and support fast broadcast authentication. Pseudonyms generation time interval follows Poisson distribution making it difficult for attackers.
- Analysis and validations are done to demonstrate that FastTrust achieves security objectives. FastTrust is able to significantly preserve cars' privacy against pseudonym linkability attack. The average verification time is about 1.2 milliseconds, 20 times faster than existing ECDSA.

The rest of the paper is organized as follows. Section II discusses the related works. In Section III, we introduce our system model, security requirements, and threat model. Section IV gives the details of our FastTrust mechanism. The security and privacy analysis of FastTrust are presented in Section V. In Section VI and Section VII, we provide our evaluation results. Finally, Section VIII concludes our paper.

II. Related Works

Broadcast authentication techniques have been extensively studied to mitigate signature flooding attack in VANETs. Most

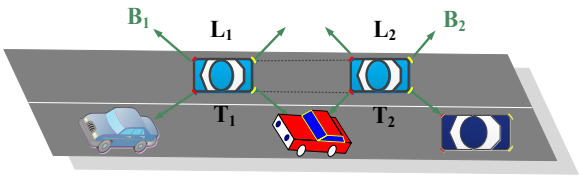


Fig. 1. An illustration of system model.

of the solutions use batch signature verification or aggregate signature scheme based on public-key cryptography, where multiple signatures of STMs are simultaneously verified by pre-deployed or fixed infrastructures [5], [11], [12]. In these schemes, infrastructures equipped with high computational power are required to do a few pairing operations and point multiplication operations over the elliptic curve [13]. This cannot be afforded by cars as their hardware for computation is restricted due to auto manufacturers' cost constraints. Hence, these schemes are unsuitable for cars. In recent years, selective verification is studied as a method to prune out attacks and then verify STMs only sent by valid cars [14]. However, it can be very limited since signature flooding attack may be triggered on busy roads even in absence of malicious attackers.

Without relying on any fixed infrastructure, TESLA [15] is a fast authentication architecture designed using symmetric-key cryptography for lossy multicast data streams. In the context of vehicular networks, VAST [16] is proposed to secure cars' data messages by combining both TESLA and basic ECDSA scheme. However, these mechanisms have a common drawback that STMs cannot be immediately verified by cars. Hsiao et al. [6] propose a lightweight broadcast authentication protocol which resists signature flooding attack by developing a one-time signature scheme. However, as the verification of one STM signature relies on other previous STMs' information, their scheme is vulnerable to packet losses. To reduce the storage overhead due to signature flooding attack, PBA [17] constructs a secure scheme using shortened Message Authentication Codes (MACs) of signatures for broadcast authentication in VAENTs. Our work proposes a novel fast broadcast authentication protocol specifically for connected cars and both real-time authentication and packet loss resilience are supported.

There have been several works to consider the location privacy issue for connected vehicles. Sampigethaya et al. [18] propose a privacy preserving scheme named CARAVAN, which relies on a group of vehicles with the use of silence periods to provide the property of location unlinkability. This needs vehicles to form and maintain a group at all time to defend against location tracking attack, which is a strict demand for cars. Lu et al. [19] present a strategy of changing pseudonyms at social hot spots to protect the location privacy. However, a vehicle's trajectory could be tracked by comparing the accurate spatial-temporal information in successive STMs. Guo et al. [20] design an anonymous authentication protocol using pairwise keys to encrypt real certificate information. Such scheme is not suitable for broadcast authentication and is limited in usage to two parties.

None of these solutions considered the high need of fast authentication and privacy of STMs broadcast by cars during inter-vehicle communications.

III. System and Security Model

A. System Model

As illustrated in Fig. 1, a sender car on an expressway broadcasts spatial-temporal messages at a high frequency. STMs include the sender's current kinetic information, i.e., position, time, direction and velocity. Upon receiving STMs, a receiver should verify and then act upon them before STMs' deadline. The hardware for connecting cars is divided into built-in and brought-in connection systems [4], and we do not specify which one to use in our work. For example, cars equipped with wireless On-Board Units (OBUs) could periodically send out STMs and process the incoming STMs during car-to-car connections.

Many applications especially safety-related ones rely on the spatial-temporal information in STMs. For instance, in the FCW application, cars constantly monitor nearby cars' current position and velocity in order to warn drivers of potential accidents. The spatial-temporal information can be acquired from on-board devices. For example, GPS on the car can support positioning accuracy of meter level and timing accuracy of nanosecond level [6]. As specified by ETSI, cars may broadcast STMs from 1 Hz to 10 Hz based on the channel environment [21]. In this paper, we consider cars broadcast STMs 10 times per second for safety on expressways. We denote B_i as an STM broadcast by a car at the time T_i.

According to the standard of IEEE 1609.2, each car has a set of ECDSA key pairs: public keys for verification and private keys for signing STMs to secure message transmissions. A trusted Certificate Authority (CA) will certify these public keys as valid identities of one car. In a real deployment, regional Ministry of Transportation or auto manufacturers may act as the role of CA. We assume that the key pairs are stored in each car's hardware, with the tamper-resistant property to defend against compromising attacks. Since users have strong incentives not to give up their security completely, we assume each car never gives or releases his private keys to another entity.

B. Security Requirements

In this work, our target is to design signature schemes providing effective and fast authentication, non-repudiation and privacy protection for fast-moving cars. We define privacy for inter-vehicle communications in terms of anonymity and unlinkability of STMs.

1. Fast verification: Before acting on an STM, a receiver car must be capable of verifying the authenticity of STM broadcast by one valid car. In particular, fast verification of STMs is mandatory for cars on expressways, where receivers need to act instantly based on STMs.

2. Non-repudiation: Non-repudiation is an extremely important requirement that usually indicates authentication. It enables a receiver car to prove to a third peer that a sender car is responsible for one broadcast STM. With this property, receivers could identify the sender of any STM and then report malicious participants to legal authorities.

3. Packet loss resilience: As studied in prior work [22], packet loss rate can be as high as 30% in a favorable situation and 60% in a congested road. Even when one STM has been lost due to the poor connection, it should not disable authentication of other subsequent STMs.

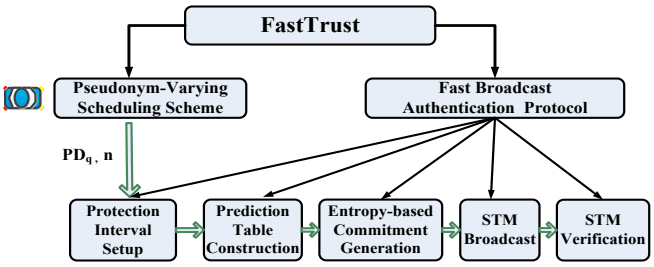


Fig. 2. An illustration of FastTrust.

4. Anonymity: Disclosing both spatial-temporal and identity information to an untrusted peer poses privacy threats to one user. Although it's necessary to exchange STMs with a high frequency, the identity of a sender should be hidden from receivers.

5. Unlinkability: The unlinkability property requires that no car can profile and analyze another car's trace through observation of STMs. Especially, when multiple identities or pseudonyms are used by a car, an adversary should not be able to link them or reveal the identity of the car.

C. Threat Model

We assume each car registers with CA by preloading z public/private key pairs: K_q^+/ K_q^-, $q \in \{1, 2, ...z\}$. A public key K_q^+ can be served as one pseudonym of the car, which has been certified by CA. Therefore, each car locally stores these pseudonymous certificates, i.e., $Cert_1, Cert_2, \cdots, Cert_z$, issued by CA. A private key K_q^- is used by a sender to sign an STM. A receiver could ensure the authenticity of the STM with the corresponding public key K_q^+.

To disrupt an STM-broadcast system, a malicious car seeks to send bogus STMs, including creating fake STMs by himself, lying to others about his spatial-temporal information, and pretending to be another valid car. If another party reports him to CA or other legal authorities, he will try to repudiate STMs that have been generated by him. Moreover, a malicious car will seek to analyze and profile another car based on STMs exchanged between them. That includes linking another car's K_q^+, and using the spatial-temporal information in STMs to reconstruct driving trajectory.

An attacker may tamper with STMs broadcast by a car. If signatures for a number of STMs are only generated and broadcast in last few STMs, an attacker may suppress the authentication of STMs by capturing signatures during the connections. An attacker with high computational and storage resources attempts to collect spatial-temporal information as much as possible by eavesdropping on cars' connections. It may profile a car based on statistical analysis and acquire his real identity information.

We assume signature flooding attack caused by a few colluding attackers sending useless signatures or a large number of legitimate cars nearby broadcasting valid message signatures. Jamming attack enhances the effect of packet losses, which can be addressed by the mechanism in [23]. Other

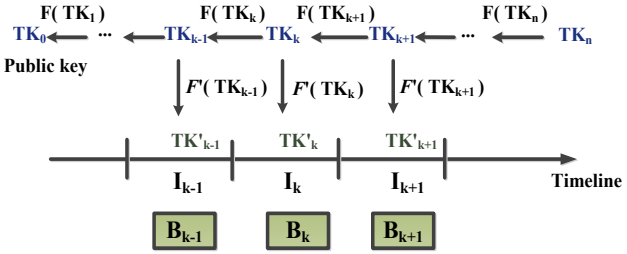


Fig. 3. Protection interval setup.

TABLE I. LIST OF NOTATIONS

$H(m)$	One-way hashing of m
K_q^+	One public key of a car
K_q^-	One private key of a car
CT	Commitment of a hash tree
$Cert_q$	One pseudonymous certificate of a car
$m_1 \| m_2$	Concatenations of message m_1 and m_2
$F(m), F'(m)$	One-way function of m
TK_k, TK'_k:	Trusted private key for time interval I_k
$MAC(K, m)$	Message Authentication Code of message m with key K

attacks in the physical layer and wormhole attack are out of the scope of this paper.

IV. THE FASTTRUST MECHANISM

In this section, we present our FastTrust mechanism, including a fast broadcast authentication protocol and a pseudonym-varying scheduling scheme. Fig. 2 gives an overview of the major steps and the work flow of our FastTrust. All used cryptographic notations are listed in Table I.

A. Fast Broadcast Authentication

We mainly deploy hash chains and symmetric keys to design our fast broadcast authentication protocol. In addition, we construct an entropy-based commitment to support real-time and faster verification. To secure our protocol, loose time synchronization is required in the system. Nevertheless, it could be supported naturally since STMs sent from cars with GPS are timestamped with the accuracy of nanosecond. Our protocol consists of five phases: *protection interval setup*, *prediction table construction*, *entropy-based commitment generation*, *STM broadcast* and *STM verification*.

1) ***Protection Interval Setup****:* To prevent from a long-term tracking, each car first divides the timeline into a number of protection intervals. In one protection interval, the same pseudonym (e.g., PD_q) is used as an identity of the sender. In each protection interval, there are a sequence of STM events, such as $B_0, B_1, ..., B_n$ broadcast in the interval $I_0, I_1, ..., I_n$. The length of each STM interval is decided by the frequency of STMs broadcast by cars. The cutting of protection interval and the value of n are determined by our privacy-preserving scheme. We will elaborate it in Section IV-B.

At the start of a protection interval, a sender generates n chained trusted private keys (i.e., $TK_1, TK_2, \cdots, TK_n$) for the next n STMs, which is similar to the scheme of TESLA. To build this hash chain, the last key TK_n is first selected randomly, and other keys are then derived by performing a one-way hash function: $TK_k = F(TK_{k+1})_{\forall k \in \{0,...,n-1\}}$, as shown in Fig. 3. TK_0 is served as the public key of the chain, and allows anyone to check the authenticity of the following keys. Then, the sender uses a second one-way hash function F' to obtain TK'_k: $TK'_k = F'(TK_k)$. With TK'_k, a MAC of a message m_k can be computed as the signature of m_k, i.e., $MAC(TK'_k, m_k)$. In our protocol, one interval worth of private key TK_k $(1 \leq k \leq n)$ will be used to authenticate B_k broadcast in the interval I_k.

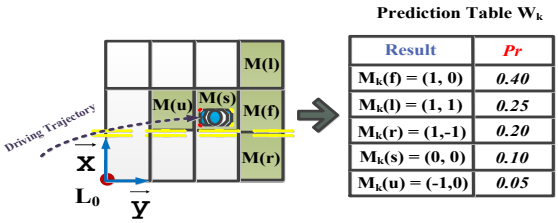


Fig. 4. Prediction table construction for STM interval I_k.

2) ***Prediction Table Construction****:* By comparing with previous STMs, we find that an STM's information except position is almost deterministic. In other words, the entropy of STMs is very low from a sender car's perspective. For example, a car driving at 80 mph would have specific probabilities of his physical movement in 100 ms, since his movement is restricted by the length of STM interval and mobility speed. Especially, cars mostly go along the road rather than making a U-turn. Based on these observations, we study how to predict a car's future positions. As the speed and direction information are also related to the position, we omit them in STMs for ease of description.

To compress the amount of movement information, we consider a car's relative position instead of entire position [6], [24]. The beginning position L_0 at one protection interval is set as the reference point. Moreover, the car needs to choose a pair of orthogonal vectors (i.e., $\vec{x}$ and $\vec{y}$), the scalar of which could be picked according to the positioning accuracy. For instance, both $|\vec{x}|$ and $|\vec{y}|$ could be set 2 meters with the accuracy of GPS. In this protection interval, a car's future position in I_k could be presented as: $L_k = L_0 + \alpha_k \vec{x} + \beta_k \vec{y}$, where α_k and β_k are rounded to integers. Therefore, the movement of the car made between two consecutive STMs, i.e., B_{k-1} and B_k:

$$M_k = L_k - L_{k-1} = (\alpha_k - \alpha_{k-1})\vec{x} + (\beta_k - \beta_{k-1})\vec{y}, \quad (1)$$

is encoded by a pair of integers $(\alpha_k - \alpha_{k-1}, \beta_k - \beta_{k-1})$.

A prediction table W_k collects all the results of M_k, and then maps M_k to a probability P_r of making such a movement, as shown in Fig. 4. In our example, $M_k(f)$ represents the sender is going to locate at $M_k(f) + L_{k-1}$ with probability 0.4. Here, we do not consider how to build an accurate prediction model, which is orthogonal to our work. If the car could obtain well-analyzed traffic statistics from cooperative applications, his mobility can be modeled more accurately and therefore our mechanism has better performance.

3) ***Entropy-based Commitment Generation****:* Given the prediction table W_k, the car starts to generate a commitment CT_k using the structure of a hash tree for the STM interval I_k. Merkle Hash Tree (MHT) is one of most common approaches. According to the entropy of movements in the prediction table, we suggest using a Huffman tree.

Similar to a Merkle tree, a Huffman tree is another kind of binary tree. Each leaf in the tree is related to a probability of

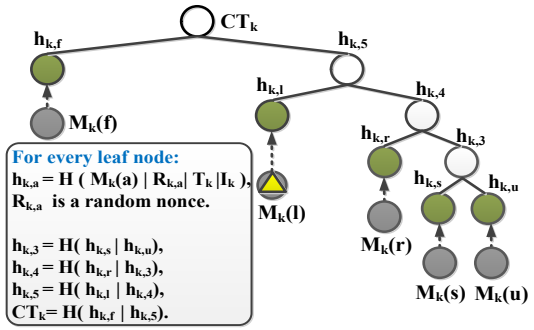


Fig. 5. Commitment generation with HHT.

one message. The probabilities of all the levies could add up to one. In the structure of a Huffman tree, the leaf associated with a high probability does not have a deeper location than the leaf with a low probability. Therefore, the leaves in Huffman trees have the quality of minimal expected depth.

Hence, we use a structure of HHT to link all the results of M_k together and then generate one single commitment for the prediction table W_k. Each leaf node is given a hash value h and a probability Pr of M_k. The organization of the tree can be determined by Huffman coding. Each inner node is constructed by hashing of its two children. The root of the HHT is the commitment CT_k.

As illustrated in Fig. 5, one entry $\{M_k(a), Pr\}$ in the W_k is associated with a leaf node $h_{k,a}$: $h_{k,a} = H(M_k(a)|R_{k,a}|T_k|I_k|)$, where $R_{k,a}$ is a random nonce and T_k is the time stamp to prevent replay attacks. The inner node is obtained by hashing its two children, e.g., $h_{k,5} = H(h_{k,l}|h_{k,4})$. Finally, the commitment CT_k is obtained for the interval I_k.

*4) **STM Broadcast**:* Suppose the sender is now being at the beginning of one protection interval T_0, whose length is n STM intervals. His pseudonym identity PD_q is related with one of his pseudonymous certificates, i.e., $Cert_q$.

After constructing W_1 and generating the commitment CT_1, the sender now constructs his first STM (denoted as B_0), which includes a message body m_0, the ECDSA signature $S(m_0)$, and the car's pseudonymous certificate $Cert_q$. The first STM is used to securely boot our fast broadcast authentication. Besides his spatial-temporal information, the sender should also put the public key of chained keys (i.e., TK_0), the commitment (i.e., CT_1) for I_1 and other local parameters into m_0:

$$m_0 = \{PD_q, T_0, I_0, L_0, TK_0, n, \vec{x}, \vec{y}, CT_1\}. \quad (2)$$

When being at location L_k at time T_k, the sender creates CT_{k+1} by performing the steps of *prediction table construction* and *entropy-based commitment generation* for the interval I_{k+1}. To construct the message body m_k, the sender will position his movement M_k on the leaf of the HHT, and then extract the necessary information (denoted as U_k) from the HHT. It includes a random nonce assigned for M_k and off-path nodes of the leaf to the root of HHT. Since U_k enables receivers to perform real-time verification, the sender should put it in m_k. Fig. 5 illustrates which values a car should extract from the HHT. A triangle shows the located leaf. The car now moves to $L_k = L_{k-1} + M_k(l)$, related with $h_{k,l}$ in the HHT.

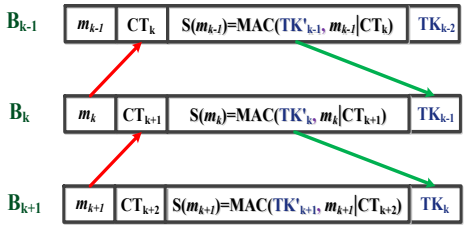


Fig. 6. Real-time and fast verification of successive STMs: B_{k-1}, B_k and B_{k+1}.

Therefore, $U_k = \{R_{k,l}, h_{k,4}, h_{k,f}\}$. Thus, for interval I_k, the sender generates the message as follows:

$$m_k = \{PD_q, T_k, I_k, L_k, U_k\}, 1 \leq k \leq n. \quad (3)$$

After producing CT_{k+1} and m_k, the sender starts to generate the signature using the trusted private key TK_k, which is predetermined and used only for interval I_k: $S(m_k) = MAC(TK'_k, m_k|CT_{k+1})$. Finally, the broadcast STM B_k is created as follows:

$$B_k = \{m_k, CT_{k+1}, S(m_k), TK_{k-1}\}, 1 \leq k \leq n, \quad (4)$$

where TK_{k-1} is the disclosure private key for receivers to securely verify STMs.

Since the public key TK_0 is only broadcast with B_0 at the beginning of a protection interval, a receiver car may be unable to verify the sender's STMs especially if he missed B_0 on expressways. Therefore, we consider each sender signs its STM and key parameters (e.g., TK_0) by ECDSA every few STMs, such as 10 STMs. After getting an ECDSA signature, the receiver could start to authenticate STMs.

*5) **STM Verification**:* Upon receiving the first STM B_0 in one protection interval, a receiver car first checks the pseudonymous certificate $Cert_q$ to validate the public key of the sender. If it is valid, he performs ECDSA verification to ensure the authenticity of m_0. Once it passes the verification, the receiver stores the public key TK_0 and other parameters included in m_0 to verify the following STMs.

To verify any B_k signed by one private key, the receiver first checks if it satisfied the security condition of using symmetric keys for authentication, which is computed on the time difference between the local receiving time of B_k and T_k [15]. Then, he performs the following steps :

- **Private key verification:** The validity of TK_{k-1} is checked by repeatedly using the hash function to recover TK_0 broadcast in the STM B_0.
- **Commitment verification:** The receiver recalculates the value of M_k based on B_{k-1} and B_k. Then, with M_k and U_k included in m_k, the receiver recomputes the root of the HHT, and then examines whether the root matches to CT_k broadcast in B_{k-1} or not.
- **Signature verification:** After the above two steps, the receiver checks if $S(m_{k-1})$ agrees with m_{k-1} and CT_k with TK_{k-1}.

We use Fig. 5 and Fig. 6 to illustrate the process of signature verification. When the receiver gets two subsequent STMs, i.e., B_{k-1} and B_k, he validates the private key TK_{k-1} from B_k first, and extracts the tree root CT_k from B_{k-1}. To verify

m_k, $M_k(l)$ is reconstructed from the location information (e.g., L_{k-1} and L_k) in the STMs. With $M_k(l)$, the receiver then calculates the leaf node $h_{k,l}$. Using $h_{k,l}$ and off-path nodes $\{h_{k,4}, h_{k,f}\}$ from U_k, the receiver is able to reconstruct the root of HHT by performing $H(H(h_{k,l}|h_{k,4})|h_{k,f})$. If the root matches CT_k, the receiver uses the private key TK_{k-1} to check the signature $S(m_{k-1})$ is valid or not. If B_k passes the above verification process, the sender can convince the receiver that he has moved M_k from T_{k-1} to T_k and is at the position $L_k = L_{k-1} + M_k$.

If the receiver missed one previous STM, e.g., B_{k-1}, during the connections, we may not accomplish real-time verification for lacking the commitment. However, based on $S(m_k)$, we can verify m_k with private key TK_k. If TK_k arrives at the receiver side after interval I_k (e.g., I_{k+1}) before the deadline of m_k, the receiver could verify the signature and then act on the message m_k.

B. Pseudonym-Varying Scheduling

Although a certain degree of privacy has been provided by pseudonyms, it is possible for an attacker to obtain complete coverage and reveal cars' location information throughout the entire system, by obtaining and analyzing all history STMs. The attacker may examine the content of STMs, which include a car's identity and spatial-temporal information. Suppose each car has multiple pseudonymous identities $PD_1, PD_2, \cdots, PD_z$, and changes pseudonyms periodically during communications. However, it is still possible for an attacker to correlate multiple pseudonyms by statistical traffic analysis if they vary at regular time or rate.

Based on the above discussion, we first give our definition of pseudonym unlinkability.

Definition 1. For all the pseudonyms $PD_1, PD_2, \cdots, PD_z$ belonged to one car, if an attacker runs a deterministic polynomial-time algorithm on observation of long-term STM records O_b, there exists a non-negligible parameter ξ such that, $\forall i, j, O_b$, $|p(PD_i|O_b) - p(PD_j|O_b)| > \xi$, $1 \leq i, j \leq z$, $i \neq j$. We call this mechanism has the property of pseudonym unlinkability.

To provide this property, one possible approach is to design a probabilistic solution to determine the time intervals between two pseudonyms. As we described before, a car divides the timeline into a sequence of protection intervals, and the same pseudonym PD_q is used in one protection interval. Another pseudonym can be assigned to a different protection interval. Each protection interval contains a sequence of STM events, such as $B_0, B_1, \cdots, B_n$. The length of protection interval n follows a probabilistic distribution.

Ideally, we may use a distinct distribution for each n, when a car changes his pseudonym from one to another. However, in terms of burst events, a car may not be able to initialize our fast broadcast authentication immediately in one protection interval. As studied in [25], if the distribution of n_i for PD_i and n_j for PD_j ($1 \leq i, j \leq z$, $i \neq j$) follows Poisson distribution with different parameters, an adversary performing a statistical test cannot identify and correlate them. To provide the property of pseudonym unlinkability, we use this strategy to design our pseudonym-varying scheduling scheme.

To further avoid spatial and temporal correlation of STMs sent by a car, pseudonyms should be changed during silent periods [18], regions where attackers cannot cover, or hot regions when cooperating with other cars [19]. In our model, as pseudonyms are varied now and then based on the privacy requirement, silent periods are more favorable. In our system, the maximum silent period for cars' connections is one STM interval. When pseudonyms are varied during a silent period, there exists a mix of location and time, which makes the attacker confused by comparing the spatial-temporal information of two subsequent STMs.

Without loss of generality, we consider a car's pseudonyms are varied in the order of $PD_1, PD_2 \cdots, PD_z$ circularly. Each car generates z distinct parameters for these pseudonyms, such that $\lambda = \sum_{q=0}^{z} \lambda_q$, where $1 \leq q \leq z$. For each pseudonym PD_q, the car first determines the length of a protection interval n, which follows the Poisson distribution with λ_q. Second, for any protection interval, the car randomly picks a silent period from zero to one STM interval. The beginning time of a protection interval is delayed a silent period. Finally, we start our fast broadcast authentication protocol based on a set of PD_q and n in the protection interval, as illustrated in Fig. 2.

With a fixed λ, increasing z could improve the privacy level, since it decreases the possibility for attackers to identify a car from a large set of pseudonyms. However, the number of z might be limited due to cars' constrained resources. For each pseudonym PD_q, a number of approaches can be used to pick a distinct λ_q, and we do not specify how cars pick such a value. We will evaluate the impact of the standard deviation of λ_q (denoted as δ), z and λ on FastTrust in Section VI.

V. Security Analysis

In this section, we discuss that FastTrust could achieve the mentioned security and privacy properties.

Proposition 1 *FastTrust provides a negligible probability that a valid authenticated message could be forged by an attacker.*

Assuming the underlying cryptographic functions are assumed to be secure, the goal of an attacker is to pretend to be a valid sender by generating a new message m'_k instead of m_k broadcast by the sender. There are three situations to consider.

First, an attacker tries to find the undisclosed private key TK_k in order to create any valid message and signature pair. For a given public key TK_0, the attacker succeeds in finding TK_k only if the hash function does not have the one-way property.

Second, an attacker may look for a different commitment CT'_k for m'_k, leading to the same signature as the original one: $S(m_{k-1}) = MAC(TK'_{k-1}, m_{k-1}|CT'_k)$. The attacker could produce such a commitment only if the MAC function is forgeable under an adaptive chosen-message attack.

At last, an attacker intends to forge some new spatial-temporal information in m'_k, which produces a different leaf node in the HHT but $CT'_k = CT_k$. If the attacker could successfully produce such a message, it means that the hash function is not collision resistance.

Proposition 2 *A car cannot repudiate his own STM broadcast.*

TABLE II. DEFAULT PARAMETERS FOR PRIVACY

Parameter	Value
Poisson parameter λ	1000
Standard deviation δ	30
Number of pseudonyms z	10
Number of STM events	10000
Length of STM interval $\lvert I_B \rvert$	100 ms

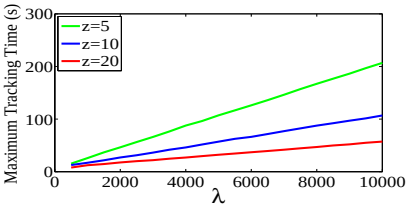


Fig. 7. Maximum tracking time under different Poisson parameter λ.

In one protection interval, the first STM B_0 is signed by ECDSA. Based on the assumption that a car's private keys are never disclosed to others, the signature $S(m_0)$ provides the property of non-repudiation of B_0.

As the public key TK_0 for our fast broadcast authentication is also included in B_0, ECDSA signature also enables a receiver to ensure that the sender is responsible for using private keys to sign the subsequent STMs. Therefore, through correlating these private keys to TK_0, the sender cannot deny an STM signed by him.

Proposition 3 *A car can verify STMs in presence of packet losses.*

In our fast broadcast authentication protocol, private keys for signatures are generated by one-way hash chains, and disclosed in an inverse order. Although a previous STM B_{k-1} with the commitment CT_k has been lost during connections, a receiver is able to verify B_k's signature with the private key TK_k broadcast in B_{k+1}. Even if B_{k+1} is also missed by the receiver, B_k could be verified as long as one of later private keys is received. With a later private key, it is feasible for the receiver to obtain TK_k by doing a few of hash operations.

Proposition 4 *A car cannot obtain another car's real identity information.*

Every time a car sends an STM, the pseudonym PD is included in the message body to replace the real identity. A car owns multiple pseudonyms and changes them periodically. Based on the characteristics of the pseudonym mechanism, any information about the car's real identity cannot be inferred from these pseudonyms.

Proposition 5 *A car cannot link multiple pseudonyms of another car used in different protection intervals.*

For a different protection interval, a car may choose a different pseudonym to broadcast STMs. Our pseudonym-varying scheduling scheme makes PD owned by the same car different for different protection intervals. Receivers or attackers cannot find out any linkage between these pseudonyms due to the same distribution of PD with different parameter [25]. By analyzing different protection intervals, an attacker cannot use T_0 to link the used pseudonyms from the same car as well, because the randomness is introduced by a silent period.

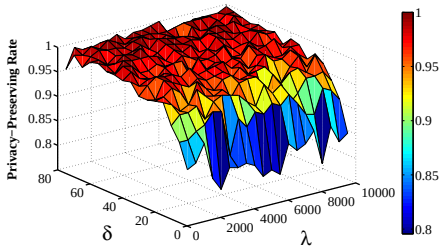


Fig. 8. Privacy-preserving rate under different Poisson parameter λ and standard deviation δ.

VI. PRIVACY EVALUATION

A. Setup

We implemented our pseudonym-varying scheduling scheme with C++. In our test, 10 cars are moving with a random trajectory for the purpose of simulation in the network of 3 km $\times$ 3 km. Each car is equipped with z pairs of 256-bit public/private keys. Thus, there are z pseudonyms used in our scheme. We use a Poisson distribution with parameter λ to determine when we change these pseudonyms. Based on the number of pseudonyms, λ is divided into z distinct values such that $\lambda = \sum_{q=1}^{z} \lambda_q$. Each pseudonym PD_q is assigned a Poisson distribution parameter of λ_q, and δ measures the standard deviation of λ_q.

Each car sends STMs every 100 ms in the system. We consider an attacker who has high storage and computational resources can collect all STMs sent by cars, and then analyze them by traffic monitoring and statistic testing. The attacker attempts to link two pseudonyms by comparing the distributions of their protection intervals. Carrying out a number of Kolmogorov-Smirnov (K-S) tests [26], the attacker intends to distinguish whether two pseudonyms are owned by the same car or not. After an attacker has made a decision, the False Negative (FN) rate is the percentage of instances where two pseudonyms of a car are not recognized as belonging to the same car.

The performance of FastTrust is evaluated on *maximum tracking time* and *privacy-preserving rate*. The maximum tracking time is defined as the average maximum time for an attacker to trace a source car. The privacy-preserving rate is defined as the FN rate for classification algorithms. In our simulation, each data point is run on 100 times based on 10000 STM events.

B. Evaluation Results

First, we want to study the maximum tracking time in terms of different number of λ. From Fig. 7, we can see that the maximum tracking time monotonically increases as the value of λ increases. This indicates that a fast rate of pseudonym-varying can improve cars' location privacy. Our scheme is more resistant to attacks with a higher value of z. When a larger set of pseudonyms are used by a car, we could see a lower value of maximum tracking time as expected.

Another important aspect to investigate is the privacy-preserving rate of our FastTrust mechanism. With different λ and δ, our test results are shown in Fig. 8. A high δ could improve the privacy-preserving rate, since an attacker is hard to

TABLE III. SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
Hash, MAC operation	1 μs	Hash, MAC size	20 Bytes
ECDSA generation	7 ms	ECDSA verification	22 ms
ECDSA key size	32 Bytes	STM size	328 Bytes
STM's lifetime	1 s	Number of cars	30
Packet loss rate p	0.3		

7	6	5	6	7
6	4	3	3	5
5	3	1	1	5
6	4	3	4	5
7	6	5	6	7

Fig. 9. The default prediction table for HHT. The sender is located at the center. Each number in the block is the off-path nodes for the leaf node.

differentiate two Poisson distributions from a large deviation. It is observed that our mechanism could resist pseudonym linkability attack, which significantly preserves cars' privacy. When $\lambda > 500$ and $\delta > 10$, the privacy-preserving rate can be very high ($> 95\%$).

VII. PROTOCOL SIMULATION

A. Setup

FastTrust was simulated using a discrete-event network simulator NS-3. To measure the performance against signature flooding attack, we consider 30 cars with OBUs broadcast STMs every 100 ms, and the lifetime of STMs to be one second. The mobility pattern of each car on an expressway is generated by SUMO in our simulation, and we denote the packet loss rate as p in our network. To simulate the exchange of STMs among cars, we use IEEE 802.11p in MAC layer and Nakagami model in the physical layer.

Table III lists our parameter settings and the sample values commonly used by vehicular networks [6], [20]. In our FastTrust, a car needs a prediction table to determine his future positions. We consider the speed limit of vehicles to be 80 mph, and the scalar of each orthogonal vector to be 2 meters. Based on prior work [6], we build a simple prediction table using a default prediction model as well (See Fig. 9), which was tested to outperform a trained prediction model based on some real traffic traces. In the future, well-analyzed traffic data could be applied by car suppliers or some vehicular applications to construct our prediction table. Based on the default prediction table, we consider each car constructs the commitment with both the structure of MHT and HHT.

Our FastTrust will make use of both Table II and Table III as default settings. We evaluate FastTrust's performance with the following metrics: (1) signature generation time; (2) signature verification time; (3) ratio of communication, defined as the ratio of FastTrust's communication overhead

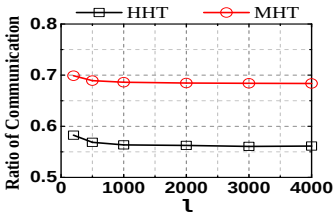


Fig. 10. The communication overhead of HHT and MHT compared to ECDSA.

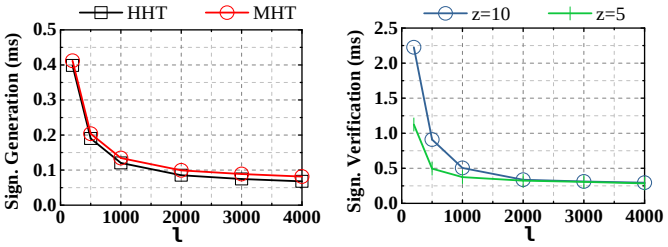


Fig. 11. Signature generation time and signature verification time with different privacy parameters.

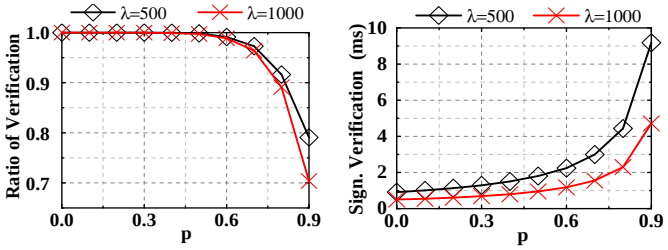


Fig. 12. Impact of packet losses on FastTrust.

to ECDSA; (4) verification speed, defined as the average time to accomplish one STM verification; (5) ratio of verification, defined as the number of verified STMs to received STMs. Each of the simulation results is based on 10 runnings.

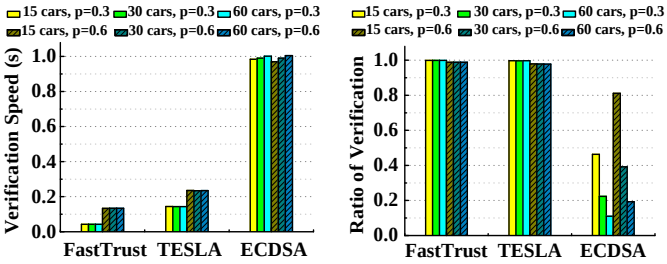
B. Simulation Results

1) Performance of HHT: We first evaluate FastTrust with both the structure of HHT and MHT. As shown in Fig. 10, both of them have less communication cost than ECDSA. The communication radio of FastTrust with HHT is about 56%, performing much better than MHT as we expected.

2) Impact of Privacy: Fig. 11 shows the performance of FastTrust with different privacy parameters. Based on the results, our FastTrust is shown to be extremely efficient and privacy-preserving for car-to-car connections. When $\lambda > 500$ and $\delta = 30$, the average time of signature generation is less than 0.2 millisecond, and the verification time for STMs is less than 1 millisecond, which is more than 20 times faster than ECDSA.

With a larger z, we find that FastTrust needs more time to verify a signature of an STM. Based on the results in Fig. 7 and Fig. 8, cars' privacy could be improved with a low λ and a large z. It can be seen that such setting slightly increases the authentication time of STMs. Hence, these parameters should be chosen carefully based on a specific security and privacy requirement for practical applications.

3) Impact of Packet Loss: We want to examine the impact of packet loss rate on FastTrust, and our result is shown in Fig. 12. As p increases, receivers' computational time increases and the ratio of verification reduces gracefully. This is because receivers would wait for the later private key to verify STMs when there exist wireless losses during inter-car connections. Although the performance of FastTrust degrades in terms of packet losses, our FastTrust still provides significant advantages even when p is particularly high. When $p = 0.6$ and $\lambda = 1000$, FastTrust is able to process 98% of STMs, and the verification time of STMs is about 1.2 milliseconds.



(a) verification speed (b) ratio of verification

Fig. 13. Performance comparison.

4) Performance Comparison: Fig. 13 shows the verification speed and ratio of verification of FastTrust, ECDSA and TESLA under different p and car density (the number of cars in the radio range). We can see that the performance of FastTrust and TESLA is not affected by the car density. They could resist signature flooding attack due to lightweight computational cost for authentication. Meanwhile, as we deploy an entropy-based commitment to verify STMs, FastTrust's verification speed is much faster than TESLA. It is also seen that ECDSA cannot authenticate more than 80% of STMs under a heavy load (i.e., 60 cars). From the simulation results, FastTrust is indicated to perform best with the fastest verification speed.

VIII. Conclusion

Fast broadcast authentication and maintaining device privacy are two desirable goals for inter-car communications and often have conflicting requirements. In this work, we propose FastTrust to achieve these goals. First, we design a fast broadcast authentication protocol based on symmetric-key cryptography to mitigate signature flooding attack. To provide real-time and faster authentication, an entropy-based commitment is constructed with the structure of Huffman Hash Trees in our protocol. Furthermore, we develop a pseudonym-varying scheduling scheme to protect users' privacy while also supporting fast broadcast authentication.

Security analysis demonstrates that FastTrust is able to achieve the security and privacy objectives. Our simulation results indicate that FastTrust could achieve a high privacy-preserving rate ($> 95\%$), and fast authenticate STMs with low computational and communication cost. In our future work, we will consider the issue of revocation of pseudonymous certificates. In addition, we will deploy our mechanism into real vehicular applications and carry out more intensive evaluations on security protection of connected cars.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 61702319) and Shanghai Sailing Program (Grant No. 17YF1405500).

References

[1] S. P. Wood, J. Chang, T. Healy, and J. Wood,"The potential regulatory challenges of increasingly autonomous motor vehicles," *Santa Clara L. Rev.*,vol. 52, no. 4, pp. 1423–1502, Dec. 2012.

[2] "2017 Mercedes E Class." http://jalopnik.com/new-mercedes-e-classes-can-talk-to-each-other-now-1752242141.

[3] "2017 CTS Sedan." http://www.wheels.ca/news/cadillac-adds-v2v-tech-cts-sedans/.

[4] E.Uhlemann, "Introducing connected vehicles [connected vehicles]," *IEEE Vehicular Technology Magazine*,vol. 10, no. 1, pp. 23–31, Feb. 2015.

[5] C. Zhang, R. Lu, X. Lin, P. H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, pp. 816–824, 2008.

[6] H. C. Hsiao, A. Studer, C. Chen, A. Perrig, F. Bai, B. Bellur, and A. Iyer, "Flooding-resilient broadcast authentication for vanets," in *Proc. ACM MOBICOM*, pp. 193–204, Sep. 2011.

[7] M. Amoozadeh, A. Raghuramu, C. N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, Jun. 2015.

[8] IEEE Std 1609.2-2016 - IEEE standard for wireless access in vehicular environments - Security services for applications and management messages, Mar. 2016.

[9] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, Apr. 2015.

[10] J. Katz and Y. Lindell, " Introduction to modern cryptography," CRC press, 2014.

[11] J. L. Huang, L. Y. Yeh, and H. Y. Chien, "ABAKA: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks, " *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 248–262, Jan. 2011.

[12] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: A robust signature scheme for vehicular networks using binary authentication tree," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1974–1983, Apr. 2009.

[13] T. Unterluggauer and E. Wenger, "Efficient pairings and ecc for embedded systems," in *Proc. CHES*, pp. 298–315, 2014.

[14] S. Khanna, S. S. Venkatesh, O. Fatemieh, F. Khan, and C. A. Gunter, "Adaptive selective verification: an efficient adaptive countermeasure to thwart dos attacks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 715–728, Jun. 2012.

[15] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proc. NDSS* , pp. 35–46, 2001.

[16] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient vanet authentication," *Journal of Communications and Networks*, vol. 11, no. 6, pp. 574–588, Dec. 2009.

[17] C. Lyu, D. Gu, Y. Zeng, and P. Mohapatra, "PBA: Prediction-based authentication for vehicle-to-vehicle communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 71–83, Jan.-Feb. 2016.

[18] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "CARAVAN: Providing location privacy for vanet," in *Proc. Embedded Security Cars*, 2005.

[19] R. Lu, X. Lin,T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: an effective strategy for location privacy in vanets," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 86–96, Jan. 2012.

[20] S. Guo, D. Zeng, and Y. Xiang, "Chameleon hashing for secure and privacy-preserving vehicular communications" *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2794–2803, Nov. 2014.

[21] ETSI EN 302 637-2-Intelligent Transport Systems (ITS), "Specification of cooperative awareness basic service", Sep. 2014.

[22] F. Bai, D. D. Stancil, and H. Krishnan, "Toward understanding characteristics of dedicated short range communications from a perspective of vehicular network engineers," in *Proc. ACM Mobicom*, pp. 329–340, 2010.

[23] W. Shen, P. Ning, X. He, and H. Dai, "Ally friendly jamming: How to jam your enemy and maintain your own wireless connectivity at the same time," in *Proc. IEEE S&P*, pp. 174–188, 2013.

[24] C. Lyu, A. Pande, X. O. Wang, J. Zhu, D. Gu, and P. Mohapatra, "CLIP: Continuous location integrity and provenance for mobile phones," in *Proc. IEEE MASS*, pp. 172–180, 2015.

[25] Z. Zhu and G. Cao, "Toward privacy preserving and collusion resistance in a location proof updating system," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 51–64, Jan. 2013.

[26] F. J. Massey Jr., "The kolmogorov-smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.