Efficient and Secure Threshold-based Event Validation for VANETs*

Hsu-Chun Hsiao† Ahren Studer† Rituik Dubey§ Elaine Shi◦ Adrian Perrig†

† CyLab/CMU
{hchsiao, astuder, perrig}@cmu.edu

§ Cisco
ritdubey@cisco.com

◦ PARC/UC Berkeley
elaines@eecs.berkeley.edu

ABSTRACT

Determining whether the number of vehicles reporting an event is above a threshold is an important mechanism for VANETs, because many applications rely on a threshold number of notifications to reach agreement among vehicles, to determine the validity of an event, or to prevent the abuse of emergency alarms. We present the first efficient and secure threshold-based event validation protocol for VANETs. Quite counter-intuitively, we found that the z-smallest approach offers the best tradeoff between security and efficiency since other approaches perform better for probabilistic counting. Analysis and simulation shows that our protocol provides $> 99\%$ accuracy despite the presence of attackers, collection and distribution of alerts in less than 1 second, and negligible impact on network performance.

General Terms: Algorithms, Design, Security
Categories and Subject Descriptors: C.2.0 [Computer – Communication Networks]: General – *security and protection*; C.2.1 [Computer – Communication Networks]: Network Architecture and Design – *Wireless communication*
Authors Keywords: VANETs, threshold-based event validation, multi-hop communication

1. INTRODUCTION

In Vehicular Ad-hoc NETworks (VANETs), vehicles' On-Board Units (OBUs) broadcast information, such as location, time, speed, and congestion level over the wireless channel for a variety of safety and convenience applications [2]. For example, the Emergency Electronic Brake Light (EEBL) application enables a vehicle performing emergency braking to broadcast a warning message to any following vehicles. Similarly, the Road Hazard Condition Notification (RHCN) application enables a vehicle to detect ice or obstacles on the road and alert the vehicles approaching the hazard zone. Such notification-based VANET applications can be classified into two types, single-hop-relevant and multi-hop-relevant, based on the number of alerts that are generated and the distance these alerts need to be propagated. In single-hop-relevant applications (such as emergency braking or lane change alerts), only one or a few vehicles – those vehicles involved in the event – will send out a notification to nearby vehicles. Such traffic information is irrelevant to vehicles multiple network hops away. In multi-hop-relevant applications (such as road hazard and congestion notification systems) a large number of vehicles are involved and report the event to vehicles that are potentially multiple network hops away, so recipients can respond appropriately. For example, congested road notifications may be transmitted several kilometers so drivers can find another route (e.g., take an exit to avoid a congested part of a highway).

Despite the great potential of VANET applications, security has long been a concern [10, 18, 24, 25, 28, 31], and thus it is imperative to provide functionality to validate an event reported by vehicles in both types of applications. Although the IEEE 1609.2 [14] standard is proposed to secure VANETs using digital signatures and certificates to prevent attacks (e.g., impersonation), the standard fails to address event falsification. For example, a selfish driver can still generate a false alert about congestion on a road segment, but other drivers will believe the alert since it is digitally signed using valid cryptographic credentials. As a result, these drivers will avoid this road, providing the selfish driver with an improved driving experience.

Counting the number of vehicles that report an event allows a recipient to evaluate the validity of a VANET event [13, 16, 29]. For example, a traffic jam reported by 2 vehicles is likely to be fake (or just started), but alerts from 50 vehicles is a strong indicator of road congestion. Particularly, we focus on **threshold-based event validation** where an event is considered valid if the number of reports exceeds a certain threshold. In contrast to counting alerts for single-hop-relevant applications, which can be done without collecting

*This research was supported by CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and W911NF-09-1-0273, and by MURI W 911 NF 0710287 from the Army Research Office, and by support from General Motors through the GM-CMU Collaborative Research Laboratory. 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 ARO, CMU, GM, or the U.S. Government or any of its agencies.

WiSec'11, June 14–17, 2011, Hamburg, Germany.

alerts from vehicles several hops away, correctly counting the number of reporting vehicles in **multi-hop-relevant (MH-relevant) applications** is challenging; it is crucial that the counting mechanism satisfies efficiency and security *at the same time*. Rebroadcasting all signed messages with certificates associated with an event is secure, yet causes network contention [23]. Messages without the signatures and certificates may elevate efficiency, but a vehicle could claim that an arbitrary number of vehicles have observed an event. Cooperative rebroadcasts have been proposed to reduce network contention [35], but it is still inefficient when combined with signatures and certificates. Hence, the major challenge is to securely and accurately estimate the number of vehicles that report an alert without requiring all of the associated data.

Prior work has proposed schemes for probabilistic counting to estimate the total number of items (e.g., unique elements in a database) based on a single pass over the data while requiring significantly less space [1, 3, 9]. In this paper, we leverage such counting schemes to perform *probabilistic threshold-based event validation* where a vehicle that receives a small subset of alerts can distinguish between a small number of potentially malicious alerts and a large number of alerts for a legitimate event. To reduce space requirements, probabilistic counting assumes that items follow a distribution. Based on this distribution, the reception of different items has (with high probability) different implications about the total number of items in existence. For example, item A may be so rare that receiving A implies there are 100 items. Probabilistic counting yields an estimate of the number of alerts, whereas threshold-based event validation only needs to indicate if the number of alerts is above a fixed threshold. By focusing on estimating a binary condition, i.e., whether the count is over or under a fixed threshold, rather than on the numerical value of the count itself, probabilistic threshold-based validation can sacrifice accuracy of the underlying counting schemes in order to further improve efficiency.

Current schemes for probabilistic counting assume the absence of malicious parties. Unfortunately, a malicious party can generate different variants of a single alert (e.g., by making small changes to the time, location, or randomness in the signature) until it acquires a rare enough alert instance that the scheme indicates the threshold was passed — a *decision changing attack*. To prevent such attacks, the scheme must limit the number of alerts one sender can generate for an event. We propose an event description format that uses coarse-grained event, time, and location descriptions to achieve this goal. We perform threshold-based validation on the event description and source of an alert while keeping the associated signature and certificate to verify the source that generated the alert. This combination of signatures and threshold-based validation based on messages of our format provides an efficient means to prevent malicious parties from abusing VANET applications.

Contributions. The main contributions of this work are:
1) We prove that threshold-based validation requires much less accuracy in counting than probabilistic counting does.
2) We propose a secure and efficient probabilistic threshold-based event validation protocol with an event description format to prevent decision changing attacks.
3) We design a message exchange protocol enabling timely collection and distribution of multi-hop alerts.
4) The evaluation shows that vehicles can accurately validate an event by storing and forwarding only 15 alerts while incurring limited packet loss due to bandwidth consumption associated with VANET applications.

2. BACKGROUND ON PROBABILISTIC COUNTING

In this paper, we propose a protocol for efficient and secure threshold-based event validation, building on probabilistic counting schemes. In this section, we provide an overview of probabilistic counting, one example of a specific probabilistic counting scheme, and a discussion of probabilistic counting schemes' trade-offs and limitations.

Probabilistic counting selects several representative elements, or a *synopsis* [22], as an estimator for the total number of distinct elements [1, 3, 9]. The synopsis summarizes the entire element set and thus permits estimation of the total size. Probabilistic counting provides a trade-off between synopsis size and accuracy: the more elements in the synopsis, the more accurate the count. The extreme trade-off points are to either keep all elements (achieving perfect accuracy) or to store only minimal statistical information. For example, storing only the lexicographically smallest element enables estimation of the total number of elements, because assuming uniformly distributed elements, the unbiased estimator for the total number is $(e_1 - e_0)/(e - e_0)$, where e represents the value of the smallest observed element, and e_0 and e_1 the minimal and maximal value, respectively.

Generally a probabilistic counting scheme provides three functions on synopses: *Generation*, *Fusion*, and *Evaluation* [22]. A *Generation* function selects the representative items from the input set I to use as a synopsis $\mathbb{S}$. In this paper, we consider a class of probabilistic counting schemes whose *Fusion* function prevents double counting and *Evaluation* function provides an error guarantee on its approximation $\tilde{n}$, such that we have high confidence $(1 - \delta)$ on a probabilistic statement that $\tilde{n}$ deviates from the real count n by only a small amount. Formally, each scheme provides the following functions:

***Generation*: SG(.)** $\mathbb{S} = SG(I)$, where $\mathbb{S} \subseteq I$.

***Fusion*: SF(.,.)** $SF(\mathbb{S}_1, \mathbb{S}_2) = SG(I_1 \cup I_2)$ when $\mathbb{S}_1 = SG(I_1)$ and $\mathbb{S}_2 = SG(I_2)$.

***Evaluation*: SE(.)** $\tilde{n} = SE(\mathbb{S})$.

$$Pr[B_L(n) \leq \tilde{n} \leq B_U(n)] > 1 - \delta, \tag{1}$$

where δ is in $[0, 1]$, and $B_L(\cdot)$ and $B_U(\cdot)$ are monotonically increasing functions that indicate the lower bound and the upper bound of $\tilde{n}$, respectively. This probability is taken over the space of random items, not over the entire distribution of n, i.e., n is taken as given.

In this paper, we consider four error-bounded probabilistic counting schemes (KeepAll, AMS [1], FM sketch [9],

Table 1: Error bounded probabilistic counting schemes. $\epsilon < 1$ for z-smallest and FM sketch. $w > 4$ for AMS. The right most column shows the approximate size of a synopsis when $n = 10000$, $\epsilon = 0.1$, $\delta = 0.05$, $w = 5$.

scheme	$B_L(n)$	$B_U(n)$	synopsis size	
KeepAll	n	n	n	10000
z-smallest	$n(1-\epsilon)$	$n(1+\epsilon)$	$O(\frac{\ln(1/\delta)}{\epsilon^2})$	128
AMS	n/w	wn	$\frac{\ln(1/\delta)}{2(1/2-2/w)^2}$	150
FM sketch	$n(1-\epsilon)$	$n(1+\epsilon)$	$O(\frac{\ln 1/\delta \ln n}{\epsilon^2})$	1700

and z-smallest [3]) which satisfy such requirements as examples for theoretical analysis and simulation. KeepAll is the approach where every unique item is part of the synopsis. Due to space limitations, we only provide a summary of z-smallest below, and refer readers to the original publications for more details [1, 3, 9]. After the example and a discussion of the accuracy and efficiency trade-off for probabilistic counting schemes, we discuss how maliciously crafted inputs can cause probabilistic counting schemes to produce unrealistically large estimates.

Probabilistic Counting Example. Bar-Yossef et al. [3] proposed using the z^{th}-smallest hash value (v_z) as an estimator of the number of distinct elements (n). The intuition is that if the hashes of the elements are uniformly distributed in $[0, 1]$, the expected number of hashes falling into $[0, v_z]$ is $v_z n$. Hence, the estimator is $\tilde{n} = z/v_z$. For example, if the resulted hash set is $\{0.05, 0.1, 0.15, 0.2, \ldots\}$, with elements perfectly uniformly distributed in $[0, 1]$, the total number of elements can be estimated by the 2^{nd}-smallest value (v_2): $\tilde{n} = 2/v_2 = 2/0.1 \approx 20$.

Accuracy and Efficiency Trade-off. Probabilistic counting schemes provide a trade-off between efficiency and accuracy. For example, KeepAll sacrifices efficiency to provide perfect accuracy. Other probabilistic counting schemes selectively store a subset of the data to shrink the synopsis while maintaining an accurate estimate. As the error bound ($B_U(n) - B_L(n)$) and the probability of an inaccurate estimate (δ) decrease, probabilistic counting schemes must increase the synopsis size. Table 1 provides a summary of these parameters for the four schemes we consider.

Vulnerability to Maliciously Crafted Inputs. Probabilistic counting schemes were originally designed to operate in environments without malicious behavior. However, when an attacker controls the inputs to the *Generation* function, the attacker can craft inputs to bias the output of the estimator. Such manipulation of inputs is known as an *inflation attack*. Secure threshold-based event validation is unable to prevent minor inflation. However, our goal is to prevent decision changing attacks, where the threshold comparison output changes.

3. PROBLEM DEFINITION

To obtain high certainty for a MH-relevant event, vehicles rely on a threshold number of vehicles to report that event before alerting the driver. The core challenge in threshold-based event validation for VANETs is to create an efficient mechanism to combine and distribute event alerts with a low error rate in the presence of malicious entities.

3.1 Application Model

Fig. 1 provides an example of threshold-based validation for a congestion notification application. *Witnesses* (vehicles that observe the event directly and report the event) work collaboratively to collect alerts. If the number of witnesses (n) exceeds a threshold (τ), the witnesses generate a compact *event proof* proving that $n \geq \tau$, and distribute the event proof to vehicles multiple hops away. A vehicle that did not observe the event itself can verify the event proof to ensure that $n \geq \tau$. In our example, timely multi-hop distribution allows vehicles to avoid the congestion by taking another route. Next we provide more details about the Collection and Distribution phases of the applications.

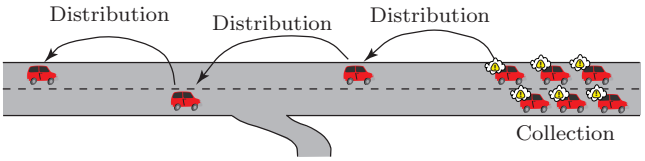

Figure 1: Example of road congestion. vehicles in the traffic jam collect alerts and distribute an event proof to warn vehicles behind.

Collection phase: Once a vehicle observes an event, that vehicle begins broadcasting alerts about the event and starts to collect other vehicles' alerts pertaining to the event. Specifically, a witness vehicle broadcasts a triple $\langle \mathcal{E}, \sigma, cert \rangle$, where $\mathcal{E}$ is an event description, σ is a signature on $\mathcal{E}$, and *cert* is a public-key certificate. To reduce communication overhead in the Collection phase, a witness only keeps a *synopsis*, a subset of alerts providing a rough estimate of number of alerts ($\tilde{n}$). The witness vehicles exchange synopses with each other using the *Message Exchange Protocol*. The Collection phase is finished when the *threshold-based validation algorithm* determines that the vehicle has collected sufficient alerts to generate an event proof (a synopsis showing $\tilde{n} \geq \tau$), or when the event expires. If $\tilde{n} \geq \tau$, the witnesses transit to the Distribution phase to spread the synopsis.

Distribution phase: After receiving an event proof that indicates $n \geq \tau$, vehicles rebroadcast the event proof to alert vehicles further away. Similar to in the Collection phase, in the Distribution phase, the rebroadcast frequency and message payload is determined by the *message exchange protocol*. By verifying an event proof, a vehicle away from the event scene can be assured that the total number of alerts exceeds a certain threshold value ($n \geq \tau$) without hearing all of the n alerts.

Figure 2 outlines the phase transitions in threshold-based applications. During the Standby phase, there is no active MH-relevant event. In the occurrence of multiple concurrent events, the applications maintain per-event phase and synopsis, but broadcast their synopses in the same beacon.

We detail the Threshold-based Validation Algorithm in Section 4 and the Message Exchange Protocol in Section 5.

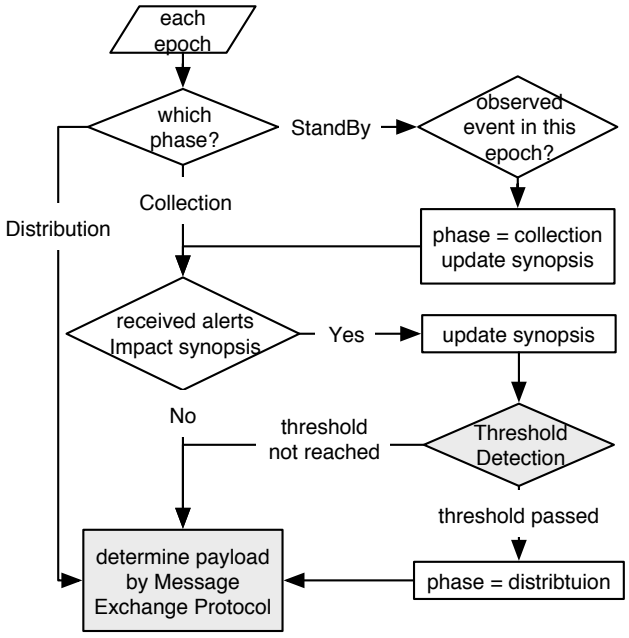

Figure 2: The phase transitions and operations in threshold-based applications.

In this work, we consider RSU-free collection and distribution. Roadside Units (RSUs) are immobile base stations that often play the role of a resource-abundant and trusted authority in many VANET proposals [2, 24, 34, 37]. In practice, however, it is difficult and costly to deploy RSUs along all roads and ensure their integrity. Our design allows vehicles to collect and distribute messages collaboratively, and thus no RSU is involved.

3.2 Problem Formulation

Successful operation of MH-relevant applications requires a threshold-based validation algorithm D, which outputs 1 when at least a threshold number of vehicles (τ) report an event and 0 otherwise. In the presence of adversaries, a threshold-based validation scheme may produce the wrong output. The error rate of D is expressed through false positive rate δ_1 and false negative rate δ_0. We define a *positive* as when the threshold-based validation algorithm outputs 1, and a *negative* when it outputs 0. Consequently, in a *false positive* (FP) D outputs 1 when less than τ vehicles report an event. In a *false negative* (FN), D outputs 0 when more than τ vehicles report an event. A *spurious* alert reports an event that did not occur. A *legitimate* alert reports an event that occurred. If receivers can verify the signature in an alert using the included public key and certificate, the alert is *valid*. Spurious and legitimate alerts can be valid.

More formally, in a setting with n_0 spurious alerts that try to report a fake event $\mathcal{E}$:

$$Pr[D(\mathcal{E}) = 1 | n_0 < \tau] \leq \delta_1 \quad (2)$$

If n_1 legitimate alerts report a real event $\mathcal{E}$:

$$Pr[D(\mathcal{E}) = 0 | n_1 \geq \tau] \leq \delta_0 \quad (3)$$

3.3 Evaluation Metrics

We evaluate the performance of a threshold-based event validation protocol based on the following metrics.

Overhead: The bandwidth associated with transmission of a synopsis provides a way to evaluate the efficiency of a threshold-based validation protocol. Because communication is limited, an efficient threshold-based validation protocol should consume a sub-linear amount of bandwidth with respect to the number of total alerts.

FP and FN rate: A secure threshold-based validation protocol should provide low FP and FN rates.

Delay: The time from vehicles' first alert until the reception of the event proof represents the delay, assuming that τ or more vehicles report the event.

3.4 Assumptions

PKI. We assume that a Public Key Infrastructure (PKI) exists, where each vehicle possesses one (and only one) valid public key and private key pair at a time.[1] For example, auto manufactures can act as certificate authorities to generate and sign key pairs. Each key pair will then be stored in an OBU, with tamper-resistant protection to protect the private key from compromise.

Bimodal distribution of number of alerts. We assume the number of alerts associated with events follows a bimodal distribution such that the number of spurious alerts during a fake event (n_0) is significantly smaller than the number of legitimate alerts (n_1) during a real event. That is, we assume that the majority of vehicles that participate in alert collection and distribution are honest. A honest participant complies with all VANET protocols and reports correct information. A temporary, localized dishonest majority may exist [21] (e.g., 7 out of 10 vehicles in one block are dishonest). However, such a small-scale dishonest majority has a limited impact on MH-relevant applications because the number of malicious entities is too small to successfully cause a decision changing attack. This disparity between n_0 and n_1 ensures that with high probability a large number of alerts represents a legitimate event while a small number of alerts, in the steady state, indicates a fake event. The actual values of n_0 and n_1 may vary based on the current circumstance (such as road capacity, speed, and number of spurious alerts that we want to tolerate). For instance, for a congestion notification application, we may have $n_1 = 100$ on a highway, but $n_1 = 50$ on a narrow local street. However, the mechanism to determine proper values is outside the scope of this paper. We assume the system knows a priori what values are appropriate for a given scenario.

Time and location information. Time and location information is required in each event description $\mathcal{E}$. The information can be provided by the Global Positioning Sys-

[1]VANETs can leverage multiple keys per vehicle to provide privacy [28, 33]. However, only one key pair is valid at any given time to prevent Sybil attacks [8] where one vehicle poses as many vehicles.

tem (GPS), which is available in many vehicles nowadays and necessary for VANET safety applications. We do not require secure positioning, and thus we tolerate vehicles lying about their location. So long as the majority are honest, a threshold-based application can limit the influence of fake reports.

Event detection. We assume a mechanism for event detection, either through human input or automatic detection through vehicle kinematics. A human observer may trigger an alert by pressing a button and selecting an event type. Automatic detection may rely on sensors (e.g., wheel slip to detect ice) or vehicle kinematics (e.g., vehicle standing on highway indicates congestion or danger) to detect an event and automatically send out an alert. After witness vehicles use the aforementioned mechanism(s) to detect the event, the vehicles can broadcast an alert reporting the event. However, our secure event validation protocol does not require such event detection mechanisms to be secure.

3.5 Attacker Model

In general, the attacker's goal is to bias other vehicles' views, i.e., cause a threshold-based validation algorithm to return an incorrect result. In particular, we consider *decision changing attacks*, where an attacker can make vehicles believe an inflated number of alerts such that a detection algorithm outputs "threshold detected" while in fact $n < \tau$ (a large FP rate of $P[D = 1|n < \tau] > \delta_1$).

We assume jamming and denial-of-service attacks can be mitigated by techniques such as spread spectrum [6], channel switching [28] or adaptive authentication [30]; providing reliable wireless communication is outside the scope of this paper. We do not consider deflation attacks, where an attacker covers up the occurrence of an event by dropping alerts or jamming the wireless channels because the attacker has difficulty to persistently (compared to the protocol execution time) isolate one group of vehicles from the other.

We assume the attacker targets an event or a set of similar events, all of them satisfy a specific intention. For example, the attacker intends to reduce her commute time when she goes to work in the early morning. Hence, any fake congestion event that falls in such a time frame (early morning) and space window (home to office) can serve the purpose of reducing commute time by misleading other drivers to take different paths. We do not consider an aimless attacker who just wants to cause trouble somewhere, e.g., any location within the US, because in most cases it is impractical for the attacker to ship and deploy a wireless device broadcasting fake alerts at that location, which may be far away. Note that attackers cannot distribute fake alerts over the Internet and rebroadcast by WiFi devices because WiFi operates in the 2.4GHz radio band while VANETs in 5.9GHz. Attackers could collude over the Internet by posting their fake alerts on a message board, from which others can download a message that successfully launches an attack. However, law enforcement can find such illegal sites and try to shut them down. Moreover, malicious vehicles would be easy to detect, because two messages would appear in a short timescale during which it would have been impossible to get from one location to the other.

4. EFFICIENT AND SECURE THRESHOLD-BASED VALIDATION

Multiple-hop-relevant VANET applications require a threshold number of alerts to validate an event. Witnesses to the event collect a subset of alerts, a synopsis, and distribute the subset to vehicles further away. The synopsis allows other vehicles to determine if the total number of alerts surpasses the threshold. Our goal is a small synopsis which provides an accurate threshold-based validation, because collecting and relaying every alert, digital signature, and certificate would cause severe link-layer contention. Moreover, such a synopsis should be secure against malicious manipulation that impacts the applications, i.e., a decision changing attack. This section describes how our proposed protocol achieves each of those goals.

Reducing the size of a synopsis. In this section, we formally *prove that threshold-based validation based on error bounded probabilistic counting can be efficient in MH-relevant applications*, where the expected number of legitimate alerts is much larger than the number of spurious alerts (Section 4.1). In contrast to a probabilistic counting scheme which requires a large synopsis for an accurate estimation of the number of alerts, a threshold-based validation scheme requires much less overhead to accurately detect a threshold number of alerts. We introduce a notion of *noise zone* to characterize the bimodal distribution of number of alerts in a MH-relevant application. The noise zone represents the value range from the *anticipated number of colluding attackers* to the *minimum number of legitimate witnesses*. When the actual count fails outside the noise zone, our threshold-based validation algorithm will return an accurate decision with high probability.

Securing synopses against manipulation. Every vehicle adds a digital signature (σ) and certificate ($cert$) to its alert to secure the threshold-based validation result. Certificates and signatures prevent an attacker from posing as a large number of vehicles reporting a fake event. An attacker, however, could subvert the decision of threshold-based validation by a single special message that represents a high count in a probabilistic threshold-based validation scheme. The attacker can obtain the special message by brute force search in a number of distinctly constructed alerts with equivalent meaning. To thwart such a decision changing attack, we propose a message description format that *specifies every event by a pre-defined structure and granularity* (Section 4.2). Such a format prevents the attacker from generating a large number of alerts by making small changes to the message (e.g., changing the longitude by a few meters).

4.1 Efficient threshold-based validation

We observe that a MH-relevant VANET application can be characterized by a *noise zone*, which is a value interval $[a, b)$ satisfying the following condition:

$$Pr[n \in [0, a] \cup [b, \infty)] > 1 - \eta, \tag{4}$$

where η is close to zero. In other words, the number of alerts in a steady state (e.g., the state where no new alerts

are observed for a certain amount of time) falls outside the noise zone with high probability. We give a formal definition:

DEFINITION 1. A threshold-based validation algorithm D is (τ, a, b, δ)-*guaranteed* if for a threshold τ and a noise zone $[a, b)$, D can output a decision with false positive and false negative rates less than δ when $n \notin [a, b)$.

Combining Definition 1 and (4) directly gives us Theorem 1, a probabilistic bound on a threshold-based validation algorithm over all inputs n.

THEOREM 1. *A (τ, a, b, δ)-guaranteed threshold-based validation scheme can output a correct decision with probability at least $(1-\delta)(1-\eta)$.*

Theorem 2 shows the relation between a noise zone $[a, b)$ and a threshold τ. We show that a threshold-based validation scheme guarantees an accurate decision when the number of alerts (n) is outside its noise zone; otherwise, the decision is interfered by "noise". Precisely, it can distinguish between a fake event and a real event with high probability, when at most a spurious alerts report a fake event or at least b legitimate alerts report a real event.

THEOREM 2. *Let ρ be a (B_L, B_U, δ) probabilistic counting scheme (i.e., satisfying (1), $Pr[B_L(n) \leq \tilde{n} \leq B_U(n)] > 1-\delta$). a, b, and τ are values that satisfy the equation*

$$B_U(a) < \tau \leq B_L(b). \tag{5}$$

Let $\mathcal{D}$ be the probabilistic threshold-based validation algorithm that runs ρ to receive an estimate $\tilde{n}$ of n, and outputs 0 when $\tilde{n} < \tau$ and 1 when $\tilde{n} \geq \tau$. Then $\mathcal{D}$ is a (τ, a, b, δ)-guaranteed probabilistic threshold-based validation algorithm.

Proof of Theorem 2: When $n \geq b$,

$$\begin{aligned} Pr[\tilde{n} \geq \tau] &\geq Pr[\tilde{n} \geq B_L(n) \geq \tau] \\ &= Pr[\tilde{n} \geq B_L(n) \text{ and } B_L(n) \geq \tau] \\ &= Pr[\tilde{n} \geq B_L(n) | B_L(n) \geq \tau] Pr[B_L(n) \geq \tau] \\ &> (1-\delta) Pr[B_L(n) \geq \tau] \\ &\geq (1-\delta) Pr[B_L(b) \geq \tau] \end{aligned}$$

$$\Rightarrow Pr[\tilde{n} < \tau] < \delta.$$

We replace $Pr[\tilde{n} \geq B_L(n)]$ by $1-\delta$ based on (1), which holds unconditionally of n. Finally, we replace n with b because $B_L(\cdot)$ is a non-decreasing function.

Similarly, when $n \leq a$, $Pr[\tilde{n} \geq \tau] < \delta$. ■

Theorem 2 shows that to achieve (τ, a, b, δ) guarantee, threshold-based validation algorithm should satisfy both (1) and (5), and output 1 when $\tilde{n} \geq \tau$ and output 0 when $\tilde{n} < \tau$.

4.1.1 Discussion

According to (5) and the $B_L(n)$ and $B_U(n)$ in Table 1 we can express the noise zone in terms of the threshold τ. For example, the D_z scheme has to satisfy

$$B_L(b) = b(1-\epsilon) < \tau \leq B_U(a) = a(1+\epsilon)$$

and thus $[a, b) = [\frac{\tau}{1+\epsilon}, \frac{\tau}{1-\epsilon})$, where ϵ is an adjustment parameter whose increment reduces the synopsis size but extends the noise zone. Note that probabilistic counting requires ϵ to be close to zero (e.g., 0.05) to have an accurate count, whereas in threshold-based validation ϵ can be much higher (e.g., 0.5) thus greatly reducing the communication overhead caused by synopsis exchange.

Table 2: Comparison of four instantiations threshold-based validation.

scheme	$\lvert\mathbb{S}\rvert$	$[a, b)$
D_{KA}	$O(\tau)$	N/A
D_z	$O(\frac{\ln(1/\delta)}{\epsilon^2})$	$[\frac{\tau}{1+\epsilon}, \frac{\tau}{1-\epsilon})$
D_{AMS}	$O(\frac{\ln(1/\delta)}{2(1/2-2/w)^2})$	$[\tau/w, \tau w)$
D_{FM}	$O(\frac{\ln 1/\delta \ln \tau}{\epsilon^2})$	$[\frac{\tau}{1+\epsilon}, \frac{\tau}{1-\epsilon})$

Table 2 summarizes four (τ, a, b, δ)-guaranteed threshold-based validation algorithms, D_z, D_{FM}, and D_{AMS}, based on z-smallest, FM, and AMS sketch, respectively. D_{KA} represents a naive threshold-based validation scheme which keeps all alerts until τ alerts are stored.

Given a noise zone $[a, b)$ and a required false positive (negative) rate δ, an application can determine proper values of τ and ϵ and thus $|\mathbb{S}|$ based on Table 2. For example, when $[a, b) = [40, 90)$, we can set $\tau = 56$ and $\epsilon \leq 5/13$ for D_z.

In D_z, D_{FM}, and D_{AMS}, a wider $[a, b)$ implies a larger ϵ (or smaller w, the adjustment parameter for D_{AMS}) thus reducing the synopsis size. In Section 6, we analyze and simulate the schemes to determine the impact of synopsis size on false positives, false negatives and network performance. We find that D_z causes the lowest overhead among all schemes given the same error rates.

4.2 Event Description Format

To prevent a decision changing attack, we require that the valid message space is bounded. In other words, a valid event description $\mathcal{E}$ needs to conform to a prescribed format:

```
[emergency type] [time epoch] [location]
```

Both the time epoch and location are *coarse-grained*. For example, time epochs have the granularity of 10 minutes, and location is approximated to the nearest intersection or the previous highway exit. The approach limits the attacker to a single description for a given event, thereby preventing a decision changing attack.

Given every witness will generate the same $\mathcal{E}$, we hash the event descriptor along with the signer's public key as the input to a probabilistic counting scheme. Hence, each public key acts as an unique identifier of an alert, and allows our scheme to detect a threshold number of vehicles by estimating the number of distinct alerts. In VANETs, authorities assign key pairs to vehicles [28]. This prevents an attacker from selecting a specific public key as part of a decision changing attack; vehicles are limited to the public keys assigned to them. The advantage of hashing the above rather than signatures is that signatures are often randomized, and one can produce many signatures for the same message by supplying different random bits which would enable an attack. One way to address this is to use a deterministic signature scheme. However, if we hash the signer's public key along with the message, our design becomes independent of the underlying signature schemes.

Without our description format, the message field has high entropy and thus there are numerous equivalent messages indicating the same event. The attacker can thus find special messages to significantly inflate the estimation of the number of alerts with almost no delay. However, our description format slows down such an attack because it limits the entropy in the message field.

4.2.1 Discussion

In addition to our coarse-grained event format, the limitations on time and location help prevent decision changing attacks. Equation (6) models the relation between these limitations. A threshold-based validation scheme satisfying (6) is secure against decision changing attacks because an attacker can only launch such attacks with low probability.

Time limit. In VANETs, vehicles change public keys periodically (e.g., every 5 minutes) to prevent long-term location tracing. When vehicles are unable to connect to keying authorities on a frequent basis, the vehicles are allowed to preload multiple key pairs [28]. Let T_{PK} be the average time length between a public key is known by its owner and the key is being used. For example, $T_{PK} = 6$ months when vehicles download a year worth key pairs for the next year during annual inspection. To launch an effective decision changing attack, the attacker has to find a special description that causes significant inflation within T_{PK}.

Location limit. In most cases, an honest vehicle is unlikely to report events far away from each other in a short timescale, in contrast to an aimless attacker who would look for forgeable events regardless of location. Though such aimless attacks can be detected by law enforcement as explained in the previous section, law enforcement can further deter aimless attacks by running a posterior analysis on collected event proofs to detect such location inconsistency or proofs that indicate a single vehicle was in two places at once.

Coarse-grained event description. We denote N_E as the number of events available per time. For example, consider a time granularity of ten minutes and a location granularity of one square kilometer, and an attacker who wants to falsely report a congestion event occurs between her home at location (x, y) and office at $(x + 100\text{km}, y + 100\text{km})$ between 7 am to 9 am, $N_E = 1.2 * 10^5$ per day.

Hence, our scheme is secure against a decision changing attack if the average time in finding a special description that triggers a decision changing attack, T_{attack}, is larger than the available time of public keys. The security condition holds when:

$$T_{attack} = 1/(P_{DC} * N_E) > T_{PK} \quad (6)$$

where P_{DC} is the probability of a decision changing attack against one event. We derive formulas for P_{DC} in Section 6.1. P_{DC} is determined by the number of colluding attackers, and T_{PK} by the public key management mechanism in VANETs. An application can select a good tradeoff value of N_E to satisfy this condition. For example, $T_{attack} = 8.3 * 10^2$ (days) $> T_{PK}$ when $N_E = 1.2 * 10^5$ (events per day), $P_{DC} = 10^{-8}$ per event (based on the analysis in Section 6), and $T_{PK} = 365$ days.

5. MESSAGE EXCHANGE PROTOCOL

Even with a smaller synopsis, unorganized collection and rebroadcasting of messages in the ad hoc network can cause severe channel contention [23]. In this section, we describe a message exchange protocol (MEP) to efficiently collect and distribute synopses in threshold-based validation scheme.

5.1 Protocol Overview

According to the IEEE 1609.2 specification [14], each vehicle sends a beacon every 100 ms. The beacon is a signed message that authenticates the sender's information (location, speed, etc.). Therefore, a vehicle can piggyback its current synopsis in a beacon. A synopsis of an event $\mathcal{E}$ is a set of representative alerts $\{A_1, A_2, \cdots, A_{|\mathbb{S}|}\}$ reporting that event, where $A_i = \langle \mathcal{E}, \sigma_i, cert_i \rangle$. Note that σ_i is a signature on $\mathcal{E}$ so we can represent a synopsis in a compressed form, i.e., $\{\mathcal{E}, \{\sigma_1, cert_1\}, \cdots, \{\sigma_{|\mathbb{S}|}, cert_{|\mathbb{S}|}\}\}$, without losing information by discarding other data in witnesses' beacons.

Our scheme relies on broadcast communication to deliver an event proof to vehicles multiple hops away. However, multihop broadcast may cause a broadcast storm [23] — severe link-layer contention and collision due to an excessive number of replicated messages. Various techniques have been proposed to alleviate the broadcast storm problem in general [17, 23, 32, 35]. Built upon existing broadcast storm solutions, we describe a customized message exchange protocol that can further reduce the bandwidth overhead by suppressing redundant broadcasts of synopses. For example, a vehicle only broadcasts its synopsis if the vehicle hears a different set of alerts from vehicles within its communication range.

5.1.1 Synopsis Advertisement

During synopsis advertisement, a vehicle advertises a digest of its current synopses. Hence, receivers can determine if they have the same information as the sender.

At any point in time, a total of K emergency events are active. This means that each vehicle maintains a total of K synopses/sets. We denote the K sets as $\mathbb{S}(\mathcal{E}_1)$, ..., $\mathbb{S}(\mathcal{E}_K)$. Each vehicle attaches a digest to its beacon:

$$\mathsf{digest} = h(\mathcal{E}_1, \ldots, \mathcal{E}_K, \mathbb{S}(\mathcal{E}_1), \ldots, \mathbb{S}(\mathcal{E}_K))$$

where h is a hash function. Each alert in $\mathbb{S}$ is ordered based on the public keys.

A vehicle overhears the beacon of nearby vehicles, and checks if the digest matches its own. If the hashes differ, the vehicle verifies the signature on the digest, and if the signature is valid, it adds the other vehicle's public key to a list $\mathcal{N}$ that it maintains. The list $\mathcal{N}$ stores nearby vehicles whose views are different.

5.1.2 Synopsis Update

Whenever the list $\mathcal{N}$ becomes non-empty, a vehicle waits r beacons, where r is uniformly drawn from an interval (e.g., $[0, 10]$), before broadcasting its K sets. If vehicle V hears from V_s a new synopsis set that results in an updated digest, V's next beacon will act as an implicit acknowledgment, such that vehicles that hear this beacon with a now matching digest will delete V from their $\mathcal{N}$ list, and cancel any pending broadcast dedicated for V.

An attacker who keeps advertising different random strings as digests may trigger contention because none of her neighbors have the same digest and thus will broadcast their synopsis sets. To prevent such an abuse, we require every vehicle to maintain a blacklist of vehicles that have been added to $\mathcal{N}$ frequently. Advertisements from blacklisted vehicles will be dropped. Also law enforcement can track down the attacker by the blacklists.

Optimization. In the message exchange protocol, a vehicle suppresses its synopsis update when every received digest is the same as the vehicle's digest. A vehicle broadcasts its synopsis set when receiving a different digest, because seeing a different digest indicates that the vehicle may know alerts unknown to others. Nevertheless, the synopsis set may also include alerts that are already known to others. To avoid transmitting such redundant alerts and thus further optimize the message exchange protocol, we instead use a Bloom filter [4] as the digest. A Bloom filter allows constant time membership queries. Hence, the vehicle can reduce bandwidth usage by identifying absent alerts in the sender's synopsis set, and only broadcast those alerts. Specifically, a Bloom filter requires $1.44 \log_2(1/(1-0.999)) \approx 1.75$ bytes per alert to identify 99.9% of the absent alerts [4], rather than redundantly rebroadcasting all 181 bytes associated with each alert (64-byte Elliptic Curve DSA signature along with a 117-byte certificate [14]).

5.2 Discussion

Effective interval. To avoid an explosion of the number of events, a vehicle only stores alerts for recent events occurred in a nearby area. Specifically, a vehicle keeps track of an event occurring in L at T if

$$|L_{cur} - L| \leq \Delta L \text{ and } T_{cur} - T \leq \Delta T,$$

where L_{cur} is the current location of the vehicle and T_{cur} the current time, and ΔL and ΔT represent the acceptable location and time differences, respectively.

Collection delay. Our scheme provides accurate decision when the total number of alerts n is outside a certain noise zone $[a, b)$. However, alerts do not arrive in bursts. When first collecting alerts for an event, it is possible that only a few vehicles have observed the event, even if the event is occurring. To avoid such a false negative due to early evaluation, an witness vehicle keeps evaluating an event until the time $\mathcal{E}$ expires. Hence the vehicles can guarantee low false negatives while minimizing the collection delay (the time from the first alert reporting the event till the generation of an event proof) to enable timely reception of an event proof at the distant vehicles.

6. EVALUATION

Section 4 provides a summary of the asymptotic behavior of our scheme based on probabilistic counting, which was designed to work with large datasets (several thousands). In this section, we examine the behavior with hundreds of vehicles based on mathematical analysis and simulation. Our evaluation confirms that our scheme, with a reasonable error rate, can largely reduce the overhead compared to the baseline scheme, D_{KA}, which keeps all distinct alerts received by the vehicle.

6.1 Analysis of Threshold-based Validation Algorithms

We analyze three probabilistic threshold-based validation algorithms, D_z, D_{FM}, D_{AMS}, built on z-smallest, FM sketch, AMS probabilistic counting, respectively, and compare them to the D_{KA} scheme. To facilitate our analysis, we derive the probability that the estimate of number of vehicles ($\tilde{n}$) is larger than a given threshold value (τ). We denote the probability as $P_{\tilde{n} \geq \tau}$.

D_{KA}: $P_{\tilde{n} \geq \tau} = 0$ if $n < \tau$. Otherwise $P_{\tilde{n} \geq \tau} = 1$. The synopsis size is $|\mathbb{S}| = \tau$. D_{KA} keeps a threshold number of alerts to achieve perfect accuracy.

The probabilistic counting schemes run C copies of an algorithm, and take median in D_z and D_{AMS}, but mean in D_{FM} to increase the accuracy. Though FM sketch is proven to be asymptotic to a normal distribution when n is large, to our knowledge, there is no such asymptotic bound for AMS or z-smallest. On the other hand, using median in D_{FM} outputs a similar result as in D_{AMS}, where the estimate is limited to certain values, e.g., the power of 2.

D_z: First we consider one copy of the z-smallest algorithm storing z elements. The probability the estimate of n is larger than the threshold (τ) is: $p = 1 - \sum_{i=0}^{z-1} \binom{n}{i} (z/\tau)^i (1 - z/\tau)^{n-i}$. When C copies of the probabilistic counting algorithms are used, the probability that the median of these C estimates exceeds the threshold is:

$$P_{\tilde{n} \geq \tau} = \sum_{j=\lceil C/2 \rceil}^{C} \binom{C}{j} p^j (1-p)^{C-j} \tag{7}$$

The size of a synopsis is $|\mathbb{S}| = Cz$.

D_{FM}: $p_{0,i} = (1 - 1/2^i)^\tau$. $p_i = p_{0,i} \prod_{j=1}^{i-1} (1 - p_{0,i})$. $u = C \log_2(0.77351\tau)$. x_i are integers $\forall i$. $|\mathbb{S}| \leq u$.

$$P_{\tilde{n} \geq \tau} = 1 - \sum_{(\sum_{i=1}^{C} x_i) < u} \left(\prod_{i=1}^{C} p_{x_i} \right) \tag{8}$$

D_{AMS}: $p = 1 - (1 - 1/\tau_1)^n$, where $\tau_1 = 2^{\lceil \log_2 \tau \rceil}$. $P_{\tilde{n} \geq \tau}$ can be derived from (7) as well. $|\mathbb{S}| = C$.

6.1.1 Configuring Parameters

We study the relations among τ (threshold value), n_0 (number of alerts reporting a fake event), n_1 (number of alerts reporting a real event), ER (error rate) and $\mathbb{S}$ (communication overhead in terms of synopsis size). We define ER as the summation of the false positive and false negative rates. Our default setting is $n_1 = 100$, $n_0 = 0.2n_1$, $|\mathbb{S}| \approx 15$. We set $[a, b) = [2n_0, 0.5n_1)$ to ensure n falls into the noise zone with low probability. Based on Table 2, we set the threshold value as $\tau = \lceil a(1 + \frac{b-a}{b+a}) \rceil = 45$ for D_z and D_{FM}, and $\tau = \lceil \sqrt{ab} \rceil = 45$ for D_{AMS}. Note that threshold-based validation schemes are compromised when the number of malicious vehicles surpasses the threshold (i.e., $n_0 = \tau$); in other word, our default setting is highly adversarial ($n_0/\tau = 0.44$).

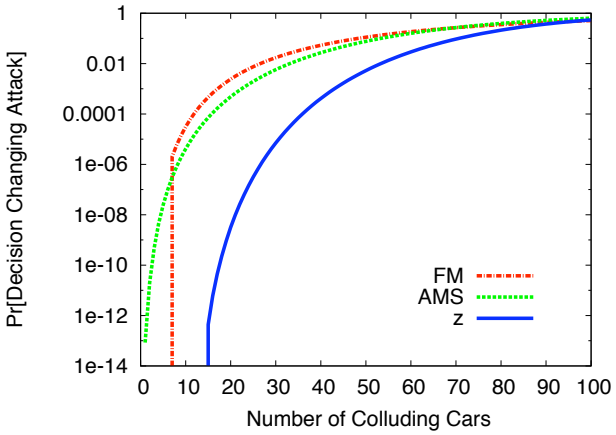

Figure 3: $\tau = 100$. $|\mathbb{S}| \approx 15$.

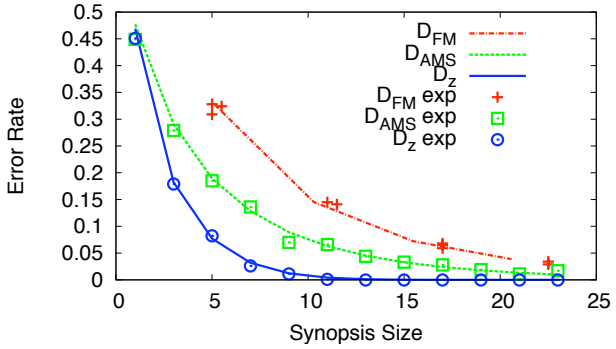

Figure 4: $n_1 = 100$, $n_0 = 20$.

6.1.2 *Probability of decision changing attacks*

A larger number of colluding attackers (n_0) is more likely to successfully claim a fake event. In an extreme situation where no malicious parties are present, the false positive rate is zero. In the presence of τ colluding attackers, the false positive rate is close to 0.5 because probabilistic threshold-based validation schemes have difficulty in distinguishing τ colluding attackers from τ honest participants.

Fig. 3 shows the probability of a decision changing attack (P_{DC}) vs. the number of colluding attackers when threshold value is 100 and the synopsis size is around 15 for a fair comparison among the different schemes. With such a constraint on the synopsis size, D_{KA} outputs "threshold detected" when the number of kept alerts passes the threshold or the size of the synopsis.

$P_{DC} = P_{\tilde{n} \geq \tau}$ when $n < \tau$. In contrast to D_{KA}, whose P_{DC} raises to 1 sharply as soon as $n \geq |\mathbb{S}|$, P_{DC} for other schemes gracefully increases as the number of colluding attackers increases. D_{KA} only works when the threshold number is small, for example 15. However, because the number of colluding attackers may be slightly larger, we require schemes for probabilistic threshold-based validation.

Given the same synopsis size, D_z is more secure (less chance of a decision changing attack) than the other schemes for any number of colluding vehicles. In the remainder of this analysis section, we focus on the three probabilistic threshold-based validation algorithms because this result shows that probabilistic counting largely reduces the synopsis size at the cost of slightly degraded accuracy.

6.1.3 *Error Rate vs. Synopsis Size*

Fig. 4 shows the error rate vs. communication overhead, expressed by the synopsis size. The error rate can be computed by $ER = P_{\tilde{n}_0 \geq \tau} + (1 - P_{\tilde{n}_1 \geq \tau})$. For each threshold-based validation, we simulate the decision process and records the error rate and synopsis size for a given threshold and number of vehicles. In the experiment, we obtain $P_{\tilde{n}_0 \geq \tau}$ and $P_{\tilde{n}_1 \geq \tau}$ by the percentage of false positives and true positives out of 1000 runs. The experimental result validates the correctness of our analytical result. We represent the analytical and experimental results by lines and points, respectively.

The graph confirms that we can improve the confidence on the output at the cost of communication overhead. The improvement is non-linear; storing more than 10–15 signatures has little advantage. For the same overhead, D_z has lowest (best) error rate while D_{FM} has the highest (worst).

6.1.4 *Error Rate vs. Number of alerts*

Fig. 5 shows the error rate vs. the number alerts reporting an event. We focus on the D_z scheme because it provided the best tradeoff in the previous two analyses. Fig. 5(a) shows that the error rate rate is lower when the number of alerts (n) falls outside the noise zone. Fig. 5(b) shows that given the same synopsis size ($|\mathbb{S}|$) and error rate (δ), increasing the threshold τ also increases the size of the noise zone.

In summary, the analysis shows that the D_z threshold-based validation algorithm provides the lowest error rates and requires the smallest synopsis size. These results make D_z most suitable for MH-relevant VANET applications.

6.2 Simulation

We use the NS-2 simulator to measure the impact of threshold-based validation algorithms and message exchange protocol (MEP) on network performance and the delay associated with distributing an event proof. We summarize our NS-2 simulation settings and implementation, and present the results with respect to the packet reception rate and the delay for event proof collection and distribution. The results show that the MEP protocol, which rebroadcasts synopses intelligently, can distribute a proof of congestion to vehicles 4.5 kilometers away from the congestion area in less than 1 second with little impact on network performance.

6.2.1 *Simulation Environment*

The vehicles are represented as mobile nodes in the simulation. Every 0.1 seconds an vehicle sends out a beacon that contains the safety information and any MH-relevant application data. We use IEEE 802.11p with parameters to reflect VANET wireless conditions [15]. Without any MH-relevant application data, each beacon is 368 bytes [14]. When broadcasting a synopsis, each certificate is 117 bytes, each signature is 64 bytes, and each $\mathcal{E}$ is 136 bits (8 bit event type, 64 bit time, 32 bit longitude, and 32 bit latitude).

We simulate a straight road with traffic in two directions. One direction has three distinct regions. The first region

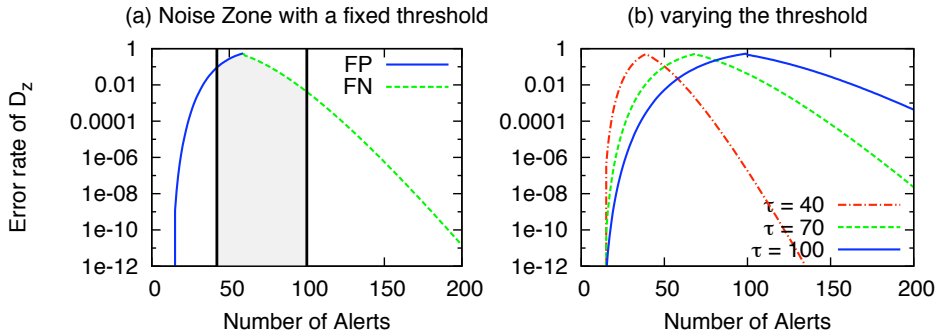

Figure 5: (a) Given $\tau = 60$ and $|\mathbb{S}| = 15$. When $\epsilon = 0.4$, the noise zone is $[a, b) = [42, 100)$ (indicated by the grey area) and $\delta = 0.076$. If we increase the ϵ value to 0.6, the noise zone becomes [37, 150) and $\delta = 0.024$. Hence increasing the noise zone decreases the error rate when n is outside of the noise zone. (b) Given $\epsilon = 0.4$, $|\mathbb{S}| = 15$ and $\delta \approx 0.1$. Increasing τ from 40, 70 to 100 expands the noise zone from $[28, 67)$, $[50, 117)$, to $[72, 166)$.

(R_1) is 7.5 kilometers long and has 300 vehicles at a density of 1 vehicle per 25 meters. This is followed by a region (R_2) 3 kilometers long with 300 vehicles at a density of 1 vehicle per 10 meters. The last region (R_3) is 1.5 kilometers long and has 60 vehicles with a density of 1 vehicle per 25 meters. Travelling in the opposite direction of the three regions are vehicles with a density of 1 vehicle per 25 meters. R_2 represents a congested region while the other regions are non-congested. Vehicles in R_3 do not witness the congestion, but can utilize an event proof from vehicles in R_2 to notify the driver and avoid the congestion ahead. R_1 and oncoming traffic are included to simulate the wireless communication from nearby vehicles.

6.2.2 Simulation Details

At a fixed time, the first 100 vehicles in R_2 start sending out a congestion alert corresponding to a single event. The vehicles hearing the alerts will retain a synopsis to generate an event proof that at least 50 vehicles are reporting the event ($\tau = 50$). For D_z, the synopsis size is 15 alerts. In simulations without MEP, a vehicle rebroadcasts its current synopsis in every beacon.

To implement our message exchange protocol described in Section 5, each vehicle sends a beacon every 100ms and can be in one of the four states which dictate what MH-relevant application data is included in the next message: 1) include a synopsis advertisement, 2) include the synopsis, 3) wait some number of epochs (randomly selected from 1 to 10) before including the synopsis (only include the advertisement) 4) include no MH-relevant application data (the vehicle lacks knowledge of the event). The reception of a message from another vehicle triggers the transition from one state to another. The content of the received message and the current state determines the next state.

6.2.3 Results

Figure 6(a) presents the normalized packet reception rate per vehicle vs. the distance from the beginning of the congestion. We define normalized packet reception rate to be the number of successfully received packets with the MH-relevant application enabled divided by the number of successfully received packets with MH-relevant applications disabled. This quantifies our protocol's impact on the network performance. The reception rates are lower in the congested area (0 to 3000 meters) and increase for vehicles away from the congestion. Without our MEP protocol, both D_{KA} and D_z lose on average 40% of packets. With MEP, D_z has a normalized packet reception rate close to 1 and greater reception in congested areas compared to D_{KA}. These results show smaller synopsis size and intelligent rebroadcast is needed to limit network degradation.

Figure 6(b) presents the collection and distribution delays vs. the distance from the beginning of the congestion. Despite the random backoff, MEP allows dissemination of an event proof within 1 second of the witnesses' original broadcasts. D_{KA} has a longer delay than D_z for both cases with and without MEP because D_{KA} experiences higher packet loss, which retards alert collection and distribution. Because beacons are not sent in synchrony, a message can spread more than one hop in less than 0.1 seconds, as shown in the distribution area of all four cases.

7. RELATED WORK

We are not aware of prior work on either threshold-based event validation or secure threshold detection. We thus discuss work in related topics: VANET event validation, secure aggregation, and probabilistic counting (detailed discussion on probabilistic counting is provided in Section 2).

VANET event validation. The number of alerts from nearby vehicles is a strong indicator of the validity of an event [13, 16, 29]. However, prior work either focuses on one-hop-relevant applications where only one-hop alerts are counted or assumes all alerts are available for analysis regardless how the alert distribution works. Dietzel et al. adopts the notion of data-centric trust [29] for event validation [7]. However, their scheme results in high dissemination delay. In contrast, our protocol enables a bandwidth-efficient solution to promptly distribute alerts and provides the event validity indicator for multi-hop-relevant applications.

Secure count aggregation. Work to secure the count aggregation problem uses cryptographic solutions [5, 11, 26,

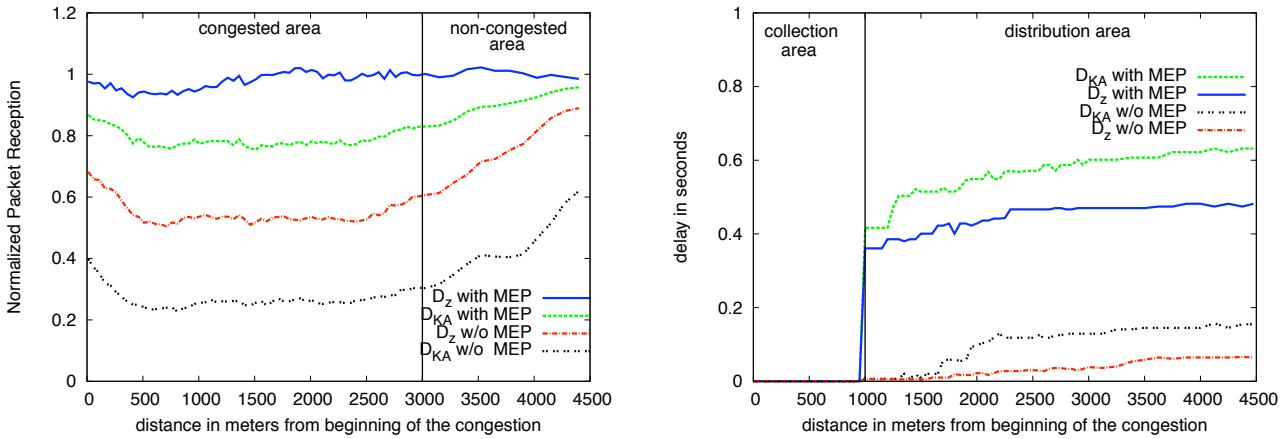

Figure 6: (a) Normalized packet reception rate at different distances. (b) Collection and distribution delays at different distances.

36] to defend against attacks, but their assumption of known network topology conflicts with vehicle mobility. Probabilistic counting has been proposed for efficient data dissemination in VANETs [20]. However, to secure probabilistic counting, most schemes need to store hundreds of signatures [12, 19]; such overhead is impractical for VANET.

Aggregate signatures. Aggregate signatures have been widely studied for reducing the signature size for multiple signers but still require broadcasting one certificate per signer for verification. In VANETs, Raya et al. propose a sequential aggregation scheme to reduce the communication overhead for signature broadcast [27]. However, this scheme does not scale to hundreds of signatures because sequential aggregation is sensitive to topology change and duplication.

8. CONCLUSION

So far, security approaches for VANETs have mostly only focused on basic primitives and mechanisms, e.g., by simply adding a digital signature to messages. Unfortunately, digital signatures alone are woefully inadequate because most applications need specialized security properties. In this paper, we propose a secure and efficient threshold-based event validation protocol for MH-relevant applications. We convert probabilistic counting to threshold-based validation, and show that threshold-based validation schemes yield significant savings compared to just counting accurately and comparing to the threshold, because threshold-based validation schemes can output an accurate decision based on an inaccurate estimate. Since VANETs are expected to be deployed within five years, we hope that the research community will embrace these important research challenges to ensure that we have secure and reliable VANET applications ready upon deployment.

Acknowledgements

We gratefully thank Fan Bai, Bhargav Bellur, and Aravind Iyer for their insightful suggestions, as well as the anonymous reviewers for their valuable comments.

9. REFERENCES

[1] Alon, N., Matias, Y., and Szegedy, M. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci. 58*, 1 (1999), 137–147.

[2] Bai, F., Krishnan, H., Sadekar, V., Holland, G., and Elbatt, T. Towards characterizing and classifying communication-based automotive applications from a wireless networking perspective. In *Proceedings of IEEE AutoNet* (2006).

[3] Bar-Yossef, Z., Jayram, T. S., Kumar, R., Sivakumar, D., and Trevisan, L. Counting distinct elements in a data stream. In *Proceedings of RANDOM* (2002).

[4] Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM 13*, 7 (1970), 422–426.

[5] Chan, H., Perrig, A., and Song, D. X. Secure hierarchical in-network aggregation in sensor networks. In *Proceedings of ACM CCS* (2006).

[6] Chiang, J. T., and Hu, Y.-C. dynamic jamming mitigation for wireless broadcast networks. In *Proceedings of IEEE INFOCOM* (2008).

[7] Dietzel, S., Schoch, E., Könings, B., Weber, M., and Kargl, F. Resilient secure aggregation for vehicular networks. *Netwrk. Mag. of Global Internetwkg. 24* (2010), 26–31.

[8] Douceur, J. R. The sybil attack. In *Proceedings of International Workshop on Peer-to-Peer Systems* (2002).

[9] Flajolet, P., and Martin, G. N. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci. 31*, 2 (1985), 182–209.

[10] Francillon, A., Danev, B., and Capkun, S. Relay attacks on passive keyless entry and start systems in modern cars. In *Proceedings of NDSS* (2011).

[11] Frikken, K. B., and Joseph A. Dougherty, I. An efficient integrity-preserving scheme for hierarchical sensor aggregation. In *Proceedings of ACM WiSec* (2008).

[12] Garofalakis, M. N., Hellerstein, J. M., and Maniatis, P. Proof sketches: Verifiable in-network aggregation. In *Proceedings of IEEE ICDE* (2007).

[13] Golle, P., Greene, D., and Staddon, J. Detecting and correcting malicious data in vanets. In *Proceedings of ACM VANET* (2004).

[14] IEEE. 1609.2: Trial-use standard for wireless access in vehicular environments-security services for applications and management messages. IEEE Standards, 2006.

[15] Jiang, D., Chen, Q., and Delgrossi, L. Optimal data rate selection for vehicle safety communications. In *Proceedings of ACM VANET* (2008).

[16] Kim, T. H.-J., Studer, A., Zhang, X., Dubey, R., Perrig, A., Bai, F., Bellur, B., and Iyer, A. Vanet alert endorsement using multi-source filters. In *Proceedings of ACM VANET* (2010).

[17] Korkmaz, G., Ekici, E., Özgüner, F., and Özgüner, U. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Proceedings of ACM VANET* (2004).

[18] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., and Savage, S. Experimental security analysis of a modern automobile. In *Proceedings of IEEE Symposium on Security and Privacy* (2010).

[19] Kuhn, M. G. Probabilistic counting of large digital signature collections. In *Proceedings of USENIX Security Symposium* (2000).

[20] Lochert, C., Scheuermann, B., and Mauve, M. Probabilistic aggregation for data dissemination in vanets. In *Proceedings of ACM VANET* (2007).

[21] Moore, T., Clulow, J., Papadimitratos, P., Anderson, R., and pierre Hubaux, J. Fast exclusion of errant devices from vehicular networks. In *Proceedings of IEEE SECON* (2008).

[22] Nath, S., Gibbons, P. B., Seshan, S., and Anderson, Z. R. Synopsis diffusion for robust aggregation in sensor networks. *ACM Transactions on Sensor Networks 4*, 2 (2008).

[23] Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S., and Sheu, J.-P. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of ACM MobiCom* (1999).

[24] Papadimitratos, P., Gligor, V., and Hubaux, J.-P. Securing Vehicular Communications - Assumptions, Requirements, and Principles. In *Proceedings of Workshop on Embedded Security in Cars* (2006).

[25] Parno, B., and Perrig, A. Challenges in securing vehicular networks. In *Proceedings of ACM HotNets* (2005).

[26] Przydatek, B., Song, D. X., and Perrig, A. SIA: secure information aggregation in sensor networks. In *Proceedings of ACM SenSys* (2003).

[27] Raya, M., Aziz, A., and Hubaux, J.-P. Efficient secure aggregation in vanets. In *Proceedings of ACM VANET* (2006).

[28] Raya, M., and Hubaux, J.-P. Securing vehicular ad hoc networks. *J. Comput. Secur. 15* (2007), 39–68.

[29] Raya, M., Papadimitratos, P., Gligor, V. D., and pierre Hubaux, J. On data centric trust establishment in ephemeral ad hoc networks. In *Proceedings of IEEE INFOCOM* (2008).

[30] Ristanovic, N., Papadimitratos, P., Theodorakopoulos, G., Hubaux, J.-P., and Leboudec, J.-Y. Adaptive message authentication for vehicular networks. In *Proceedings of ACM VANET* (2009).

[31] Rouf, I., Miller, R., Mustafa, H., Taylor, T., Oh, S., Xu, W., Gruteser, M., Trappe, W., and Seskar, I. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In *Proceedings of USENIX Security Symposium* (2010).

[32] Stojmenovic, I., Seddigh, M., and Zunic, J. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems 13*, 1 (2002), 14–25.

[33] Studer, A., Shi, E., Bai, F., and Perrig, A. TACKing together efficient authentication, revocation, and privacy in vanets. In *Proceedings of IEEE SECON* (2009).

[34] Wisitpongphan, N., Bai, F., Mudalige, P., and Tonguz, O. On the routing problem in disconnected vehicular ad-hoc networks. In *Proceedings of IEEE INFOCOM* (2007).

[35] Wisitpongphan, N., Tonguz, O. K., Parikh, J. S., Mudalige, P., Bai, F., and Sadekar, V. Broadcast storm mitigation techniques in vehicular ad hoc networks. *Wireless Communications, IEEE 14*, 6 (2007), 84–94.

[36] Yang, Y., Wang, X., Zhu, S., and Cao, G. SDAP: a secure hop-by-hop data aggregation protocol for sensor networks. In *Proceedings of ACM MobiHoc* (2006).

[37] Zhang, C., Lu, R., Lin, X., Ho, P.-H., and Shen, X. An efficient identity-based batch verification scheme for vehicular sensor networks. In *Proceedings of IEEE INFOCOM* (2008).