A Secure and Efficient Data Aggregation Framework in Vehicular Sensing Networks

著者	DU Suguo, PENG Tian, OTA Kaoru, HAOJIN Zhu
journal or publication title	International journal of distributed sensor networks
volume	2013
page range	298059-298059
year	2013-03-18
URL	http://hdl.handle.net/10258/2728

doi: info:doi/10.1155/2013/298059

A Secure and Efficient Data Aggregation Framework in Vehicular Sensing Networks

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 298059, 10 pages
http://dx.doi.org/10.1155/2013/298059

Research Article

A Secure and Efficient Data Aggregation Framework in Vehicular Sensing Networks

Suguo Du,[1] Peng Tian,[1] Kaoru Ota,[2] and Haojin Zhu[1]

[1] *Shanghai Jiao Tong University, Shanghai 200240, China*
[2] *Muroran Institute of Technology, Muroran 050-8585, Japan*

Correspondence should be addressed to Haojin Zhu; zhu-hj@cs.sjtu.edu.cn

Received 4 February 2013; Accepted 18 March 2013

Academic Editor: Mianxiong Dong

Vehicular ad hoc networks support a wide range of promising applications including vehicular sensing networks, which enable vehicles to cooperatively collect and transmit the aggregated traffic data for the purpose of traffic monitoring. The reported literatures mainly focus on how to achieve the data aggregation in dynamic vehicular environment, while the security issue especially on the authenticity and integrity of aggregation results receives less attention. In this study, we introduce a basic aggregation scheme which could aggregate the data and the message authentication codes by using *syntactic aggregation* and *cryptographic aggregation*. To tolerate duplicate messages and further improve the aggregation performance, we introduce a secure probabilistic data aggregation scheme based on Flajolet-Martin sketch and *sketch proof* technique. We also discuss the tradeoff between the bandwidth efficiency and the estimation accuracy. Extensive simulations and analysis demonstrate the efficiency and effectiveness of the proposed scheme.

1. Introduction

With the advancement of wireless technology, vehicular communication networks, also known as vehicular ad hoc networks (VANETs), are emerging as a promising approach to increase road safety, efficiency, and convenience [1, 2]. Although the primary purpose of vehicular networks is to enable communication-based automotive safety applications, VANETs also allow a wide range of promising applications such as traffic monitoring and data collecting, which are regarded as an important component of future intelligent transportation systems (ITSs). It is also observed that rising popularity of smartphones with onboard sensors (e.g., GPS, compass, accelerometer) and always-on mobile Internet connections sheds light on using smartphones as a platform for large-scale vehicular sensing. Recent reports report that smartphone users have surpassed feature phone users in the USA by 2012. According to figures released by IDC, 207.6 million Android and Apple smart-phones were shipped in the fourth quarter of 2012. This further renders the possibility of vehicular sensing.

As shown in [3–10], Departments of Transportation in the USA must collect various types of data (e.g., average speed or traffic density) for traffic monitoring purposes. Traditionally, these important data are collected by technologies such as inductive loop detectors (ILDs), video detection systems, acoustic tracking systems, or microwave radar sensors. However, these technologies mostly suffer from a high maintenance cost. On the other hand, cooperative data collection and dissemination in VANETs allow the traffic monitoring performed in a more cost-effective way [11]. Specifically, each vehicle collects its own or neighboring information (e.g., its current speed or neighboring traffic) and then transmits it to the remote roadside units (RSUs) via vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. The RSUs can be deployed at various points of interest along the roadway and can be used to collect data from locations up to tens of kilometers away. In this study, we coin the vehicular networks which are designed for traffic sensing and monitoring as the *vehicular sensing networks*.

One of the major challenges of vehicular sensing networks is high overhead of transmitted sensing data. Each sensing result is essentially some spatial-temporal measured values (speed, traffic density), which record the position of vehicles (i.e., a road segment or a small area) and the observation time. Such sensing data is periodically broadcasted. Upon reception of such a broadcast, the intermediate receivers/forwarders incorporate the received data into their local reports and then broadcast them again. Unfortunately, such a periodical broadcast brings on a high traffic load or even *traffic storm*. This problem is more serious in the scenario of high vehicle density, which could be found on multilaned highways in congestion situations. On the other hand, in most cases, drivers or monitors do not need exact individual reports, but only an overview of the general average speed on the road ahead [12]. This motivates the data aggregation issues in vehicular networks, including Flajolet-Martin sketch based probabilistic aggregation [13], fuzzy aggregation [12], and others [14, 15]. However, most of them are mainly focusing on how to achieve the data aggregation in dynamic vehicular environment, while the security issues on the aspect of the authenticity and integrity of aggregation results receive less attention. Since aggregation operation could be made by any intermediate forwarding vehicle, any malicious attacker could easily launch the attacks towards the data aggregation process by modifying the aggregated result or simply inserting invalid sensing data.

Secure data aggregation is a great challenge in vehicular sensing networks due to their unique network characteristics including highly dynamic network topology, intermittent connectivity, and potentially huge numbers of VANET nodes. These unique characteristics make the secure data aggregation in traditional wireless sensor networks such as [16], which always assume either a static network topology or aggregation structure, unsuitable for vehicular sensing networks.

Therefore, to achieve secure and efficient data sensing and collection, in this paper, we present the SAS, a secure data aggregation scheme for vehicular sensing networks which includes the basic scheme and advanced scheme. In the basic scheme, it achieves efficient data and MAC authentication via syntactic aggregation and cryptographic aggregation. However, the basic scheme needs to keep the original sensing data, which prevents a more efficient data aggregation. Further, it cannot work in case of the existence of duplicate messages. Thus, to overcome this problem, we propose an advanced scheme based on Flajolet-Martin sketch and a series of sketch proof techniques. We also discuss the tradeoff between the bandwidth efficiency and the estimation precision. Finally, extensive simulations and analysis demonstrate the efficiency and effectiveness of the proposed scheme.

The remainder of the paper is organized as follows. In Section 2, we introduce the related work. In Section 3, we present the system model and the design goals. In Section 4, we present some preliminaries. In Section 5, we present a secure data aggregation scheme in vehicular sensing networks by using the syntactic aggregation and cryptographic aggregation approach. In Section 6, we propose a probabilistic data aggregation scheme. Performance analysis is given in Section 7, followed by the conclusion in Section 8.

2. Related Work

Vehicular sensing networks represent a promising way to cooperatively collect useful information in order to increase road safety and driver convenience for future intelligent transportation system. By being integrated with the traditional digital map system, vehicular sensing networks provide the functionality of real-time automatic route scheduling [14], decentralized free parking places discovery [15], traffic monitoring [3], and so forth. In these applications, data aggregation is necessary for efficient data propagation and reduced transmission overhead.

There are quite a few research proposals for data aggregation in vehicular sensing networks [14, 15]. Most of them are based on group formulation and vehicle clustering, which can dramatically reduce the communication overhead due to the increased aggregation level. In additional to the above proposals, the structure-free aggregation frameworks are also proposed including Flajolet-Martin sketch-based aggregation [13] and fuzzy aggregation [12] without defining aggregate structures. However, the aforementioned studies focus on the data aggregation itself but do not take the security issues into consideration.

The most related research study for secure data aggregation in VANETs is the voting scheme, including [17, 18], which involves multiple vehicles to collect information towards a specific event (e.g., collision or traffic jam). Each witness (or observer) of this event will submit a message to a group leader. The group leader will take the responsibility of collecting more than a threshold k of proofs from k distinct witnesses to prove the validity of an emergency event by the voting scheme. References [17, 18] discuss how to further improve the aggregation efficiency by exploiting cryptographic tools such as onion signature [18] and aggregate signature [17]. Note that, in this study, we consider a more general data aggregation scenario: collecting data within a certain area and, at the same time, providing security guarantee for the aggregation functionality.

3. System Model and Design Goal

This section describes our system model, attack model, security assumptions, and design goals.

3.1. Network Model. In this paper, we consider a general vehicular sensing network model, which is mainly comprised of three components: traffic monitoring centre (TMC), RSUs, and vehicles. As shown in Figure 1, RSUs could be selectively deployed at some positions (e.g., intersections) to collect the traffic information (e.g., average speed) within a certain area. Due to high maintenance cost, RSUs could be only deployed intermittently to reduce the deployment cost. We assume that each vehicle, which is equipped with an on-board unit (OBU), has the capability of data collecting and reporting. The transmitted sensing data are propagated via V2V and V2I communications to the RSUs, which then forward them to the TMC. SAS is based on the distributed aggregation model similar to [13], which does not require any group/cluster formulation.

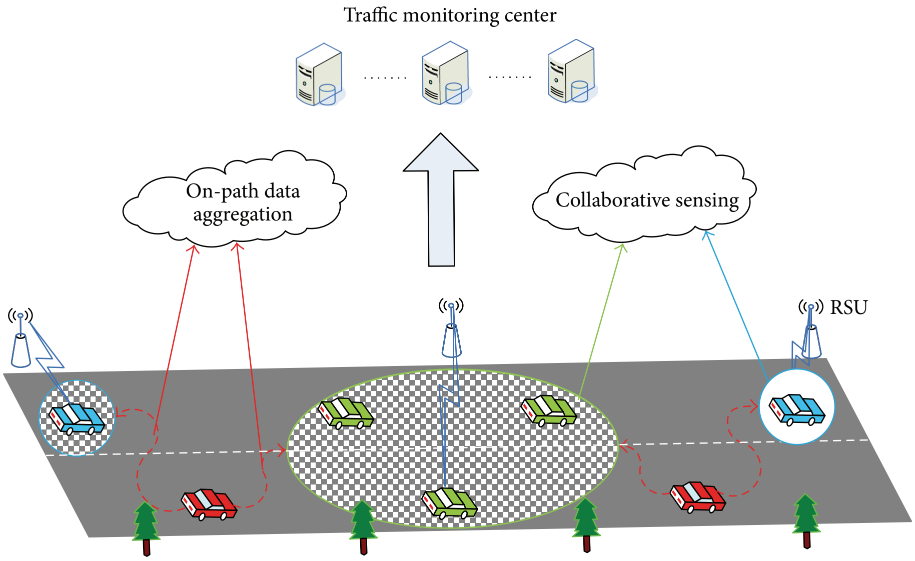

Figure 1: Overview of vehicular sensing network.

3.2. Security Assumptions. We assume that each OBU either shares a distinct secret symmetric key with TMC or obtains a public/private key pair, which is issued by TMC. Whether using shared secret key or public key depends on different system requirements.

3.3. Attack Model. In this study, we assume that the TMC and RSUs are trusted while vehicles (including the sensing vehicles and aggregator vehicles) are potentially malicious and can thus launch various attacks including fabricating, duplicating, and computing the aggregation incorrectly. We do not consider denial-of-service attacks where aggregator vehicles fail to or refuse to provide any acceptable result. A malicious sensor can always report an arbitrary sensing report, which fundamentally cannot be prevented. So we do not aim at preventing such an attack.

3.4. Design Goals

(i) *Security Goal.* The security goal of SAS is to enable the TMC to verify whether an aggregate sensing report is correct or not. Specifically, TMC should accept a reported aggregate report if and only if it is equal to the output of a correct execution of the aggregation function over all of the sensing reports provided by the qualified vehicles in the most recent epoch.

(ii) *Efficiency and Effectiveness Goal.* The efficiency goal of SAS is to minimize the transmission overhead and, at the same time, to ensure a certain sensing accuracy. However, computational cost is not a major concern of this paper since VANET is generally assumed to have unlimited computational capability [17].

4. Preliminaries

4.1. One-Way Chains and MAX Protocols. One-way chain is a widely used cryptographic primitive, which is based on a one-way function F and a secret seed s. The one-way function F is easy to compute but computationally infeasible to invert. The chain has the sequence of values $F(s)$, $F(F(s))$, $F(F(F(s)))$, ... Throughout this paper, we use $F^x()$ to denote recursively applying the function F for x times. Thus, the xth value of the sequence is $F^x(s)$. For example, given two positive integers m and n, where $m < n$, it is easy to compute the $F^n(s)$ by functioning forward the value of $F^m(s)$ for $(n-m)$ times with the function F. However, it is infeasible to compute the value of $F^m(s)$ by functioning backward the value of $F^n(s)$. One-way chain has been widely used in many security topics such as micropayment. Recently, in [16], the authors take advantage of one-way chains to construct a MAX protocol, which could ensure the aggregated maximum message cannot be inflated or deflated. However, MAX protocol is not designed for probabilistic aggregation. Further, the network topology considered in [16] is for sensor networks with statistic network topology. In SAS, what we consider is a dynamic network topology and probabilistic aggregation model.

4.2. Pairing Technique. The proposed basic scheme is based on bilinear pairing which is briefly introduced as below. Let $\mathbb{G}$ be a cyclic additive group and $\mathbb{G}_T$ a cyclic multiplicative group of the same prime-order q; that is, $|\mathbb{G}| = |\mathbb{G}_T| = q$. Let g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ an efficient admissible bilinear map with the following properties:

(i) bilinear: for $a, b \in \mathbb{Z}_q^*$, $e(g^a, g^b) = e(g, g)^{ab}$;

(ii) nondegenerate: $e(g, g) \neq 1$.

4.3. Aggregate Signature and Batch Verification. The major computation cost for authenticating an emergency message comes from verifying a set of supporting signatures issued by different emergency witnesses. The corresponding public key certificates of the signers also need to be verified together. All of them will incur a significant amount of transmission and verification cost. In this study, we use aggregate signature to reduce the transmission cost of supporting signatures, certificates, and batch verification to realize efficient signature verification.

An aggregate signature is a digital signature that supports aggregation of n distinct signatures issued by n distinct signers to a single short signature [19]. This single signature (and the n original messages) will convince the verifier that the n signers indeed sign the n original messages. In addition to the benefit of the reduced transmission size, aggregate signature technique supports batch verification, which enables the receivers to quickly verify a set of digital signatures on different messages by different signers. In this study, we adopt the aggregate signature and batch verification introduced in [20] as our basic cryptographic aggregation technique to improve the aggregation performance.

5. A General Secure Data Aggregation Framework in Vehicular Sensing Networks

In this section, we introduce a general data aggregation framework in vehicular sensing networks by using the syntactic aggregation and cryptographic aggregation approach.

5.1. System Setup. The TMC generates a tuple $(q, g, \mathbb{G}, \mathbb{G}_T, e)$ as the system parameters. The TMC selects a random $sk \in \mathbb{Z}_q^*$ as its secret key and generates its public key $pk = g^{sk}$, by which four hash functions are formed: $H : \{0,1\}^* \rightarrow \mathbb{G}$, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$, $H_2 : \{0,1\}^* \rightarrow \mathbb{G}$, $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q$. The group public key and secret key are $(q, g, \mathbb{G}_1, \mathbb{G}_T, e, pk, H, H_1, H_2, H_3)$ and sk, respectively.

An important task of the setup procedure is to determine the format of emergency report message. In our study, the format of a secure sensing report (SSR) is defined as follows. For a sensed event, the sensor vehicle i will generate an SSR:

$$\mathrm{SSR}_i = (\mathrm{ID}_i, \mathrm{Type\#}, v_i, \mathrm{Loc\#}, \mathrm{epoch\#}, \mathrm{MAC}_i, \mathrm{Cert}_i), \quad (1)$$

where ID_i denotes the identity of the vehicle that generates the claim. Type# denotes the type of SSR reported in this report. v_i denotes the sensing value provided by i. Loc# denotes the sensing area. epoch# denotes the sensing period. MAC_i denotes the message authentication code generated by vehicle i on this SSR. It has two modes: symmetric key mode (Mode I) or public key mode (Mode II). Cert_i denotes the certificate held by vehicle i.

For a specific event, it is reasonable to assume that the relevant SSRs will share the same Type#, Loc#, and epoch#.

5.2. Registration. A vehicle can join the network by performing the following step depending on Mode I or Mode II.

(1) *Private Key Generation for Mode I.* In the symmetric key mode, a vehicle i can randomly choose x_i as its secret key.

(2) *Private/Public Key Generation for Mode II.* In the public key mode, a vehicle can randomly choose $x_i \in \mathbb{Z}_q^*$ as its secret key and generate its public key $X_j = g^{x_j}$. After ensuring the legitimacy of this vehicle, TMC will issue the public key certificate by signing its signature on (i, X_i). Here, the certificate generation process follows a typical Boneh, Lynn, and Shacham signature scheme in [19]. TMC computes $h_i \leftarrow H(i \parallel X_i)$ and $\sigma_i \leftarrow h_i^{x_i}$. $\mathrm{Cert}_i = (i, X_i, \sigma_i)$ is the public key certificate of i. The verification of public key certificate could be as follows. Given a vehicle's public key certificate Cert_i, $h_i \leftarrow H(i \parallel X_i)$ can be computed, and it is accepted if $e(\sigma_i, g) = e(h_i, pk)$.

5.3. SSR Generation and Broadcasting. Once an event is sensed by one or multiple vehicles and the observation is (Type#, Loc#, epoch#), the sensing vehicles $i \mid i = 1, 2, \ldots$ may independently generate their SSRs as follows.

(1) *Mode I SSR Generation.* In terms of Mode I SSR generation, given the type and observation time of the emergency message $\mathrm{TL} = \mathrm{Type\#} \parallel \mathrm{epoch\#}$ as well as the location information $\ell = \mathrm{Loc\#}$, a witness vehicle i with its private key x_i could compute message authentication code as follows:

$$\mathrm{MAC}_i = H(x_i, i, \mathrm{Type\#}, \mathrm{Loc\#}, \mathrm{epoch\#}). \quad (2)$$

Thus, $(i, \mathrm{Type\#}, \mathrm{Loc\#}, \mathrm{epoch\#}, \mathrm{MAC}_i)$ constitutes an SSR claim generated by vehicle i towards the sensing event. After that, i will broadcast this SSR to its neighbors.

(2) *Mode II SSR Generation.* For Mode II SSR, given the type and observation time of the emergency message $\mathrm{TL} = \mathrm{Type\#} \parallel \mathrm{epoch\#}$ as well as the location information $\ell = \mathrm{Loc\#}$, a witness vehicle with its public and private key pairs (X_j, x_j) can compute $w_i \leftarrow H_3(\mathrm{TL} \parallel \ell)$, $a \leftarrow H_1(\ell)$, $b \leftarrow H_2(\ell)$ and generate the signature $\mathrm{MAC}_i = a^{x_i} b^{x_i w_i}$. Thus, $(i, \mathrm{Type\#}, \mathrm{Loc\#}, \mathrm{epoch\#}, \mathrm{MAC}_i, \mathrm{Cert}_i)$ constitutes an SSR claim generated by vehicle i towards the sensing event. After that, i will broadcast this SSR to its neighbors.

A single SSR verification can be performed as follows: given $\mathrm{SSR} = (i, \mathrm{Type\#}, \mathrm{Loc\#}, \mathrm{epoch\#}, \mathrm{MAC}_i, \mathrm{Cert}_i)$, the verifier will first check the validity of certificate included in this SSR. After that, it can check the validity of supporting signature by computing $w_i \leftarrow H_3(\mathrm{TL} \parallel \ell)$, $a \leftarrow H_1(\ell)$, $b \leftarrow H_2(\ell)$. It is accepted if $\mathrm{MAC}_i = a^{x_i} b^{x_i w_i}$.

5.4. SSR Opportunistic Forwarding. In VANETs, the network topology could be very dynamic and diversified in shape from time to time, even sometimes sparse and frequently partitioned. The communication between vehicles is expected to be performed in an opportunistic manner. This means a vehicle can carry packets when routes do not exist but forward the packets to the new receivers when they move into its vicinity [21]. To enable the opportunistic data propagation, vehicles that are within a range r and maintain connectivity for a minimum time t with each other can be arranged to form a cluster. The detailed discussion on cluster creation and maintenance can be found in [21]. We refer to the node at the head of every cluster as header, which is responsible for forwarding the data to the next cluster in a typical opportunistic data forwarding algorithm such as [21, 22]. The messages will be buffered at the header until they are forwarded to the next cluster, which is also referred to as the "*Carry and forward*" strategy. In this study, it is considered that the header can also play the role of emergency message aggregator because of the following two reasons.

(1) If taking a header of a cluster as the aggregator, the aggregation process will be merged into a part of data forwarding process. Therefore, there is no need to elect another cluster head to perform the data aggregation operations.

(2) The process of message propagation between two clusters is referred to as a catch-up process, where a message traverses along with its carrying vehicles until it reaches within the radio range of the vehicle at the end of another cluster, which obviously presents a considerable propagation interval depending on the speed of vehicles and the gap between clusters. Therefore, we can use such an interval to aggregate the related emergency messages to minimize the aggregation latency.

In the following sections, a cluster head will be taken as the aggregator of the cluster, which will perform the following SSR aggregated authentication algorithm.

5.5. SSR Secure Aggregation. For any specific emergency event, each aggregator maintains two local message lists, which keep the forwarded SSRs and ReadytoForward SSRs, respectively. The forwarded message list, denoted as $\mathscr{F}$, contains all the SSRs which have been forwarded by this vehicle before, while the ReadytoForward message list, denoted as $\mathscr{R}$, stores messages which have not been transmitted but can be forwarded some time later. The SSRs set $\mathscr{F} \cup \mathscr{R}$ includes all the SSRs related to a specific event. Whenever receiving an SSR, the aggregator should check if this SSR is a duplicate. If yes, such an SSR will be dropped; otherwise it will be put into the message list R. Before the forwarded propagation, the aggregator will perform the SSR aggregation (or *Aggregate_SSR*) and SSR batch verification (*BatchVerify_SSR*) operations as follows.

5.5.1. SSR Aggregation. Aggregate_SSR is used to aggregate multiple SSRs into a single SSR, which includes two steps: *syntactic aggregation* step and *cryptographic aggregation* step.

(i) *Syntactic Aggregation.* For a specific event, given n SSRs $(i, \text{Type\#}, \text{Loc\#}, \text{epoch\#}, \text{MAC}_i, \text{Cert}_i)$ by vehicles $1, \ldots, n$, we can obtain syntactically aggregated SSR as $\text{SSR}_{\text{agg}} = (1, \ldots, n, \text{Type\#}, \text{Loc\#}, \text{epoch\#}, \text{MAC}_1, \ldots, \text{MAC}_n, \text{Cert}_1, \ldots, \text{Cert}_n)$.

(ii) *MAC Aggregation.* It is used to aggregate multiple MACs into a single MAC, which includes the following two modes: Mode I and Mode II.

(1) *Mode I Aggregation.* Mode I aggregation is

$$\text{MAC}_{\text{agg}} = H(x_1, 1, \text{Type\#}, \text{Loc\#}, \text{epoch\#}) \otimes \cdots \otimes H(x_n, n, \text{Type\#}, \text{Loc\#}, \text{epoch\#}), \tag{3}$$

where $\otimes$ can be XOR operation.

(2) *Mode II Aggregation.* Mode II aggregation includes the certificate aggregation $\text{Cert}_{\text{agg}} \leftarrow (i, X_i, \sigma_{\text{agg}})$ and MAC aggregation $\sigma_{\text{agg}} \leftarrow \prod_{i=1}^{n} \text{Cert}_i$. $\text{MAC}_{\text{agg}} \leftarrow \prod_{i=1}^{n} \text{MAC}_i$.

After syntactic aggregation and cryptographic aggregation, we can obtain the aggregated SER as $\text{SSR}_{\text{agg}} = (1, \ldots, n, \text{Type\#}, \text{Loc\#}, \text{epoch\#}, \text{MAC}_{\text{agg}}, \text{Cert}_{\text{agg}})$.

5.5.2. SSR Batch Verification. In this section, we exploit batch verification to further reduce the computational cost.

(i) *Mode I Verification.* For Mode I verification, TMC could verify the sensing reports by verifying the following equations:

$$\text{MAC}_{\text{agg}} = H(x_1, 1, \text{Type\#}, \text{Loc\#}, \text{epoch\#}) \otimes \cdots \otimes H(x_n, n, \text{Type\#}, \text{Loc\#}, \text{epoch\#}). \tag{4}$$

(ii) *Mode II Verification.* For Mode II verification, TMC could perform the certificate batch verification as well as signature batch verification.

(1) *Certificate Batch Verification.* Given an aggregated certificate $\text{Cert}_{\text{agg}} \leftarrow (i, X_i, \sigma_{\text{agg}})$, the verifier accepts if $e(\prod_{i=1}^{n} \sigma_i, g) = e(\prod_{i=1}^{n} h_i, pk)$ holds.

(2) *Signature Batch Verification.* Given MAC_{agg}, the message set $\text{SSR}_i \mid 1 \leq i \leq n$ and public keys $X_i \mid\leq i \leq n$ for all the vehicles in set $\mathscr{V}$ accept if $e(\text{MAC}_{\text{agg}}, g) = e(a, \prod_{i=1}^{n} X_i) \times e(b, \prod_{i=1}^{n} X_i^{w_i})$.

If the batch verification holds, the aggregator will accept SSRs in list $\mathscr{R}$ as valid SSRs. Then the aggregated SSR in $\mathscr{R}$ will be forward propagated. Meanwhile, the aggregator will put all the SSRs in $\mathscr{R}$ to message list $\mathscr{F}$.

However, the previous proposed solution may face the following two problems. Firstly, it need to carry the original input of each sensing node for future verification. This is because MACs authentication requires the original input. Secondly, the duplicated message should be carefully removed from the aggregation; otherwise many of them will be aggregated for several times. This point is difficult to prevent in the context of VANET, which is a typically dynamic and distributed environment. In the next section, we will introduce a probabilistic data aggregation scheme which could automatically filter duplicate messages.

6. A Probabilistic Data Aggregation Scheme for Vehicular Sensing Networks

In this section, we firstly introduce the concept of FM sketch, which is the foundation of probabilistic data aggregation in vehicular networks. We then propose a secure data aggregation scheme based on our proposed sketch proof technique.

6.1. FM Sketches-Based Data Aggregation in VANETs. A Flajolet-Martin sketch (or "FM sketch") is a data structure for probabilistic counting of distinct elements that has been introduced in [23]. FM sketch represents an approximation of a positive integer by a bit field $s = s_1, \ldots, s_w$ of length w, where $w \geq 1$. The bit field is initialized to zero at all positions. To add an element x to the sketch, it is hashed by a hash function h with geometrically distributed positive integer output, where $P(h(x) = i) = 2^{-i}$. The entry $s_{h(x)}$ is then set to one. After processing all objects, FM finds the first bit of the sketch that is still 0. Let the position of this bit be k; then the number of distinct objects is estimated as $n = 1.29 \times 2^k$.

The variance of n is quite significant [13], and thus, the approximation is not very accurate. To overcome this, instead of using only one sketch, a set of sketches can be used to represent a single value to achieve trade-off between the accuracy and memory. The respective technique is called probabilistic counting with stochastic averaging (PCSA) in [23]. With PCSA, each added element is first mapped to one of the sketches by using an equally distributed hash function, and it is then added there. If m sketches are used, denoted by $S_1, \ldots, S_m$, let $a_1, a_2, \ldots, a_m$ be the positions of the first 0 in the m sketches, respectively; the estimate for the total number of distinct items added is then given by $n = 1.29 \times 2^{k_a}$, where $k_a = (1/m)\sum_{i=1}^{m}(a_i)$.

Sketches can be merged to obtain the total number of distinct elements added to any of them by a simple bitwise OR. Important here is that, by their construction, repeatedly combining the same sketches or adding already present elements again does not change the results, no matter how often or in which order these operations occur. FM sketch summaries are naturally composable: simply OR-ing independently built bitmaps (e.g., over data sets a_1 and a_2) for the same hash function gives precisely the sketch of the union of the underlying sets (i.e., $a_1 \cup a_2$). This makes FM sketches ideally suited for VANET aggregation.

For the purpose of discussion, let us consider a specific application. Assume that we are interested in monitoring the average speed within a certain area. As the first step, we use a sketch for each road segment and approximate the sum of speeds of vehicles within this road segment. For the second step, we will calculate the average speed by dividing the speed sum by the number of vehicles involved. In the following sections, we will discuss how to generate the sketch proof and secure sketch aggregation.

6.2. Sketch Proof Generation. According to the FM sketch definition, given the ID i and speed v_i, a vehicle may add the tuples $(i, 1), \ldots, (i, v_i)$ to the sketch by hashing them and setting the respective bit position $h(i, 1), \ldots, h(i, v_i)$ to 1. The malicious attackers may launch two kinds of attacks towards the FM sketch: inflation attack and deflation attack.

We start from three basic pieces of information that each sensor generates in our protocol. Let $\Lambda^i = \{\ell_1, \ldots, \ell_{v_i}\}$ denote v_i 1-bit positions generated by i. Given that ψ_i is the position of first 0-bit, Λ^i could be represented as the union of two subsets $\Lambda^i_{\psi_i} = \{1, \ldots, \psi_i - 1\}$ and $\overline{\Lambda^i_{\psi_i}} = \{\ell_{\psi_i}, \ldots, \ell_{v_i}\}$, where ℓ_{ψ_i} represents the first 1-bit larger than ψ_i. Thus, each vehicle i generates

(1) $s_i^+ = \{i, \psi_i, \text{Loc\#}, \text{epoch\#}, \text{MAC}_{K_i}(\omega \,||\, \text{Loc\#} \,||\, \text{epoch\#}) \mid \omega \in \Lambda^i_{\psi_i}\}$, which is called vehicle i's *inflation-free proof*. Here, Loc# and epoch# refer to the road segment number and time slot number, respectively.

(2) $s_i^- = \text{MAC}_{K_i}(\text{Loc\#} \,||\, \text{epoch\#})$, which is called vehicle i's *deflation-free proof*. This is basically the authentication code generated by the vehicle on the common information Loc#, epoch#.

(3) $s_i^\times = \{\overline{\Lambda^i_{\psi_i}}, \text{MAC}_{K_i}(\omega \,||\, \text{Loc\#} \,||\, \text{epoch\#}) \mid \omega \in \overline{\Lambda^i_{\psi_i}}\}$, which is called vehicle i's *supplement security proof*.

In the following, we will introduce these three security proofs one by one.

6.2.1. Inflation-Free Proof. Inflation-free proof is basically the authentication code generated by the vehicles on the 1-bit positions, which are smaller than the position of first 0. To prevent the inflation attacks, it is sufficient to require that each 1-bit, whose position is less than ψ_i, should be authenticated by a single signed value from one of the sensing vehicles that turn it on. We define two extra operations for inflation-free proofs.

(i) *Merging Operation* $\oplus$. Consider two sketches Λ^i and Λ^j (for simplicity of presentation, we assume $\psi_i > \psi_j$). Let ψ_m be the globally maximum value of first 0-bit after sketch merging, which corresponds to the new $\Lambda_{\psi_m} = \{1, \ldots, \psi_m - 1\}$ and $\overline{\Lambda_{\psi_m}} = \Lambda^i \cup \Lambda^j \setminus \Lambda_{\psi_m}$. We define

$$\oplus_{\omega=i,j} s^+_{\psi_w} = s^+_{\psi_i} \cup s_i^\times \left(\Lambda_{\psi_m}\right) \cup s_j^\times \left(\Lambda_{\psi_m}\right), \qquad (5)$$

where $s_i^\times(\Lambda_{\psi_m})$ is the operation that picks up all the supplement security proof whose positions are less than ψ_m. In other words, to generate inflation-free proof for the merged sketches, the aggregator could first pick up the inflation-free proof $s^+_{\psi_i}$ of the sketch with a higher 0-bit position ψ_i. For the remaining 1-bit positions $\psi_i, \ldots, \psi_m - 1$, the aggregator could pick up the inflation-free proofs either from $s_i^\times$ or $s_j^\times$. Note that, if a 1-bit is authenticated by multiple MACs generated by multiple vehicles, aggregators could choose inflation-free proof of vehicles with a lower ID.

(ii) *Aggregation Operation* $\otimes$. The MACs of s_i^+ could be further aggregated. For example, if MAC is generated by symmetric key-based hash function (e.g., MD5 or SHA-1), then ⊗ can be simple XOR; if MAC is signatures, ⊗ could be achieved by using aggregate signature technique such as [19].

With merging operation and aggregation operation, size of inflation-free proof could be minimized to $|\mathrm{ID}| * N_{1-\mathrm{bit}} + |\mathrm{MAC}|$, where |ID| and |MAC| refer to the size of vehicle ID and MAC, respectively, and $N_{1-\mathrm{bit}}$ denotes the number of 1-bits.

6.2.2. Deflation-Free Proof. Deflation attack is defined as that the malicious aggregators may try to turn 1-bits into 0-bits, removing the corresponding MACs from the security proofs. To prevent deflation attack, SAS adopts the hash-chain-based MAX protocol, which is introduced in [16]. The basic idea is to construct one-way chains whose seeds are all the s_i^-. Specifically, given the one-way function $F()$, vehicle node i reports to the aggregator $F^{\Psi_0}(s_i^-)$. In a case of multiple sketch aggregation, let ψ_m be the maximum positions observed by the aggregator. The aggregator can obtain $F^{\Psi_m}(s_i^-)$ by performing hash operations on $F^{\Psi_0}(s_i^-)$ by $\psi_m - \psi_0$ times. After obtaining all the $F^{\Psi_m}(s_i^-)$, a new operation is introduced in [16] to reduce the transmission cost, which is shown as follows.

(i) *Hash Chain Folding Operation* ⊙. The aggregator could use the folding function ⊙ to fold all the hash chains into a single one $\odot F^{\Psi_m}(s_i^-)$. Obviously, due to adoption of one-way function, it is impossible for the attackers to generate a new security proof for $\psi_i < \psi_m$, which prevents the deflation attack.

Note that one-way chains should be rolled forward even after they have been folded together with an operation like ⊙. Therefore, it requires the one-way function to achieve homomorphic property in that $F(x_1 \odot x_2) = F(x_1) \odot F(x_2)$. There is a wide range of cryptographic tools such as RSA encryption that could support such kind of homomorphic property. In this case, ⊙ could be defined as modular multiplication.

The size of deflation-free proof is a constant number $|F()|$, which represents the size of one-way function output. If choosing RSA as the cryptographic tool, $|F()| = 1024$.

6.2.3. Supplement Security Proof. Supplement security proof enables the aggregator to derive the new inflation-free proof when ψ_0 changes because of the merge of sketches. Therefore, SAS records all 1-bits whose positions are larger than ψ_m and their corresponding MACs as the supplement security proof. Since they are not continuous, supplement security proof cannot be aggregated. Further, we denote $s_i^{\times}(\overline{\Lambda_{\psi_m}})$ as the set of all the supplement security proofs whose positions are not less than ψ_m.

6.3. Sketch Proof Aggregation. As shown in Figure 2, multiple sketches could be aggregated during their propagation process, and sketch proofs could be aggregated along with sketches merging. Without loss of generality, we discuss aggregation algorithm only for two sketch proofs, and more than two sketch aggregations can be aggregated by applying it for multiple times.

Consider two sketches Λ^i and Λ^j and their corresponding sketch proofs $s_i^+, s_i^-, s_i^{\times}$ and $s_j^+, s_j^-, s_j^{\times}$. Let ψ_m be the globally maximum value of first 0-bit after sketch merging. The sketch proofs could be aggregated by performing the following steps:

(i) inflation-free proof aggregation: $\otimes(\oplus_{\omega=i,j} s_{\psi_w}^{+})$;

(ii) deflation-free proof aggregation: $\odot_{\omega=i,j} F^{\Psi_m}(s_\omega^-)$;

(iii) supplement security proof updating:

$$s_i^{\times}\left(\overline{\Lambda_{\psi_m}}\right) \cup s_j^{\times}\left(\overline{\Lambda_{\psi_m}}\right). \tag{6}$$

Note that such a sketch proof aggregation process could be performed fully distributed, which means it naturally supports hierarchical aggregation, while it does not require any aggregation architecture.

6.4. Sketch Proof Verification and Average Calculation. After the aggregation results and the security proof arrive at the TMC, TMC should verify the correctness of the inflation-free proof and deflation-free proof. To check the validity of inflation-free proof, TMC should perform the following operations in different MAC modes.

(i) *Symmetric Key Mode*. In this mode, TMC should recalculate the MAC of each 1-bit and then aggregate them into a single one. After that, TMC should check if the obtained result is equal to the received one.

(ii) *Signature Mode*. In this mode, TMC could batch verify the aggregated signatures by performing batch verification technique [19].

To verify the correctness of deflation-free proof, it needs to compute all individual s_ω^- and fold them together to create the $\odot_{\omega=1,2,\ldots} F^{\Psi_m}(s_\omega^-)$. The answer is accepted if and only if the calculated result is equal to the received one. Finally, by obtaining the ψ_m, the average speed could be computed as follows:

$$\mathrm{speed}_{\mathrm{average}} = 1.29 \times \frac{2^{\Psi_m}}{N_{\mathrm{ID}}}, \tag{7}$$

where N_{ID} refers to the number of vehicles involved. Similar to the original FM sketch, the accuracy of this average speed estimation could be further improved by introducing multiple sketches.

6.5. Further Discussion. In this subsection, we give an extended discussion on some issues closely related to the proposed SAS protocol.

6.5.1. Symmetric Key versus Asymmetric Key. As we have mentioned in Section 3, MAC in this study represents two modes: symmetric key-based mode and asymmetric key- (or signature-) based mode. Generally speaking, different MAC modes have different advantages as well as disadvantages.

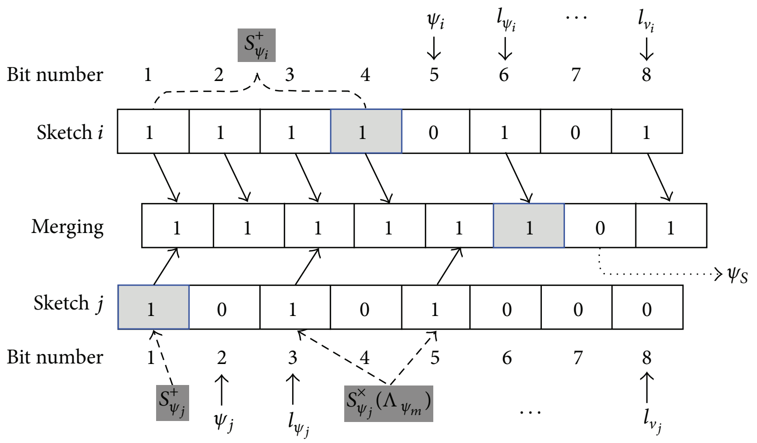

Figure 2: Sketch generation and sketch proof.

From the performance point of view, symmetric key-based MAC has the advantage on asymmetric key-based approach in that it has shorter size and will not introduce the computational expensive operations. Symmetric key-based MAC is expected to play an important role in the vehicular sensing applications where sensing information is directly sent to the TMC since they could be processed faster than signature-based approach and also introduce less transmission overhead. However, on-path vehicles cannot verify an MAC's authenticity since only TMC shared the key with MAC generator. On the other hand, signature-based approach could provide many extra features such as nonrepudiation and public authentication. In the context of vehicular sensing networks, it means the aggregated information could be verified by any on-path vehicles, which allows the drivers to have fast access to the authenticated traffic information instead of waiting for the response of the RSUs.

6.5.2. Size of Sketch Proofs. There are three kinds of sketch proofs for SAS. The first two sketch proofs including inflation-free proof and deflation-free proof could be aggregated and thus introduce a minimized transmission overhead. The third sketch proof, supplement security proof, does not support proof aggregation since they will be merged with inflation-free proof in the future. This means supplement security proof may incur a higher transmission overhead. However, we argue that size of supplement security proof is still acceptable in that, during the aggregation process, size of supplement security proof will decrease along with the increase of first 0-bit position ψ_m. In the performance evaluation part, we will give a more detailed discussion on the size of sketch proofs.

7. Performance Evaluations

In this section, we evaluate the performance of the proposed SAS in terms of the resultant communication cost and approximate accuracy. To demonstrate the superiority of SAS, we also compare SAS with nonaggregation transmission approach. In this part, we consider SHA-1 as the building blocks of MAC. Note that asymmetric key-based MAC mode will have a similar communication cost if we choose short aggregate signature as the building blocks.

Table 1: The size of each component of SAS (bytes).

	No SAS	SAS
T&L	$8 \times n$	8
ID	$8 \times n$	$8 \times n$
Data v	$8 \times n$	0
Sketch$_i$	0	$8 \times \log_2(v_{\max} \times n)$
Sketch proofs	$8 \times n$	$8 \times \log_2(v_{\max} \times n) + 136$
Total size	$32 \times n$	$8 \times n + 16 \times \log_2(v_{\max} \times n) + 144$

7.1. Transmission Overhead. One of the major advantages of SAS is the reduction of its transmission cost. The communication cost is determined by the size of aggregated security proof including inflation-free proof, deflation-free proof, and supplement security proof. Note that, since MAC in this study represents two modes: symmetric key-based mode and asymmetric key- (or signature-) based mode, here we only discuss the symmetric key-based MAC due to page limitation. As a typical example, we choose the 64-bits SHA-1 as the basic MAC technique and RSA-1028 as the basic one-way function tool. Table 1 summarizes the size of different components as well as the overall transmission overhead for nonaggregation transmission and SAS transmission. Here, we consider the worst case of our aggregation in that the size of supplement security proofs is bounded by $\log_2(v_{\max} \times n)$ [13], where $v_{\max}$ is the maximum speed for this road segment while n is maximum number of vehicles in this area. However, it is important to point out that, in practice, the size for supplement security proof should be much less than this bound since it will decrease along with the aggregation.

By choosing different number of sketches, we obtain the different communication cost of SAS under different vehicle numbers as well as different sketch numbers, which has been shown in Figure 3. It is observed that the probabilistic aggregation does not show its advantage when the

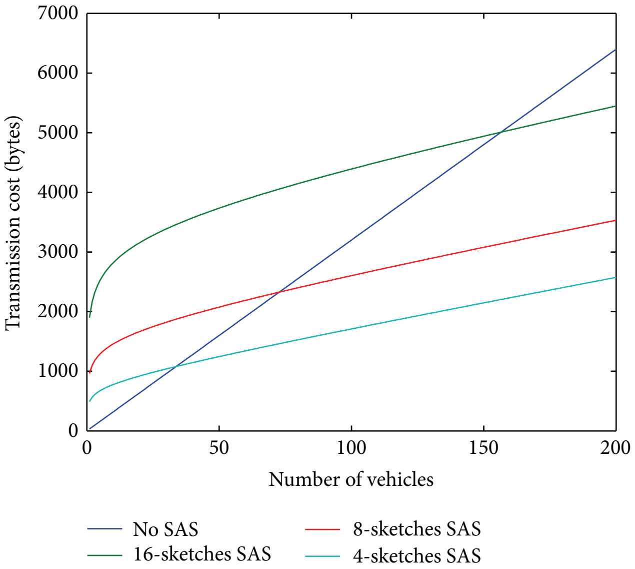

Figure 3: Transmission overhead of various secure FM sketches.

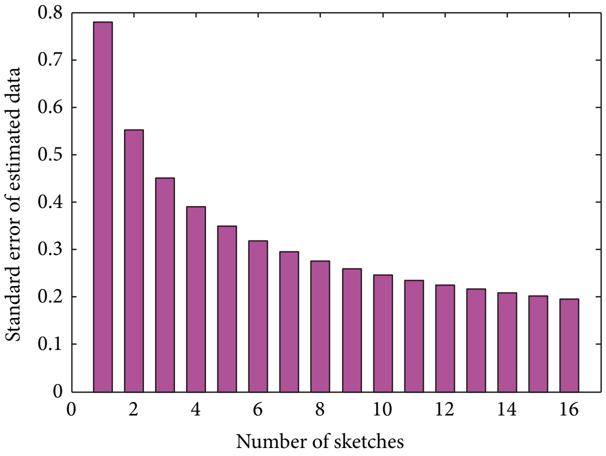

Figure 4: Standard error of SAS secure sketch.

number of vehicles is small. However, when the number of vehicles grows, the proposed SAS aggregation scheme could dramatically reduce the communication cost when the sketch number is small. It is also observed that the number of sketches plays an important role for the overall system performance in that a small sketch number such as 4 makes the proposed SAS have a better performance while, when the sketches number is large such as 16, the advantage is not so obvious. Therefore, if an acceptable accuracy is guaranteed, the number of sketches should be as small as possible to achieve a better performance. In the next section, we will discuss the tradeoff of accuracy and the number of sketches.

7.2. Tradeoff of the Accuracy and Number of Sketches. According to [13], PCSA yields a standard error of approximately $0.78/\sqrt{m}$. By choosing different sketch numbers, we can obtain the corresponding standard error, which has been plotted in Figure 4. It is observed that the standard error decreases dramatically along with the increase of number of sketches in the beginning while it stays relatively stable after a specific threshold (e.g., 4 in Figure 4). However, as we pointed out in the previous section, in the vehicular sensing networks, a small number of sketches (e.g., 4) guarantee an acceptable standard error (e.g., 0.39). This further demonstrates the effectiveness of the proposed SAS.

8. Conclusion and Future Work

Vehicular sensing networks have been envisioned to play an important role for future traffic monitoring applications. In this study, we propose a secure and efficient aggregation method based on FM sketch and security proofs techniques. The extensive performance evaluations have demonstrated the efficiency and effectiveness of the proposed scheme. Our future work includes implementing SAS in a specific application scenario and evaluating its performance with more realistic simulations or even experiments.

Acknowledgments

The correspondence author of this paper is H. Zhu. This research is supported by the National Natural Science Foundation of China (Grant no. 61003218, 70971086, 61272444, 61161140320, and 61033014), the Doctoral Fund of Ministry of Education of China (Grant no. 20100073120065), the JSPS A3 Foresight Program, and the NEC C&C Foundation.

References

[1] H. Zhu, X. Lin, M. Shi, P. H. Ho, and X. Shen, "PPAB: a privacy-preserving authentication and billing architecture for metropolitan area sharing networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2529–2543, 2009.

[2] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, "A probabilistic misbehavior detection scheme towards efficient trust establishment in delay-tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, 2013.

[3] M. Fontaine, "Traffic monitoring," in *Vehicular Networks from Theory to Practice*, CRC Press, 2009.

[4] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "SMART: a secure multilayer credit-based incentive scheme for delay-tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4628–4639, 2009.

[5] H. Zhu, R. Lu, X. Shen, and X. Lin, "Security in service-oriented vehicular networks," *IEEE Wireless Communications*, vol. 16, no. 4, pp. 16–22, 2009.

[6] H. Zhu, X. Lin, R. Lu, P. H. Ho, and X. S. Shen, "SLAB: a secure localized authentication and billing scheme for wireless mesh networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3858–3868, 2008.

[7] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen, "Secure and energy-efficient disjoint multipath routing for WSNs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 7, pp. 3255–3265, 2012.

[8] A. Liu, X. Jin, G. Cui, and Z. Chen, "Deployment guidelines for achieving maximal lifetime and avoiding energy holes in sensor network," *Information Sciences*, vol. 230, pp. 197–226, 2013.

[9] A. Liu, J. Ren, X. Li, Z. Chen, and X. Shen, “Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks,” *Computer Networks*, vol. 56, no. 7, pp. 1951–1967, 2012.

[10] A. Liu, P. Zhang, and Z. Chen, “Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 71, no. 10, pp. 1327–1355, 2011.

[11] M. H. Arbabi and M. Weigle, “Using vehicular networks to collect common traffic data,” in *Proceedings of the 6th ACM International Workshop on Vehicular Ad Hoc Networks (VANET ’09)*, pp. 117–118, Beijing, China, 2009.

[12] S. Dietzel, B. Bako, E. Schoch, and F. Kargl, “A fuzzy logic based approach for structure-free aggregation in vehicular adhoc networks,” in *Proceedings of the 6th ACM International Workshop on Vehicular Ad Hoc Networks (VANET ’09)*, pp. 79–88, Beijing, China, 2009.

[13] C. Lochert, B. Scheuermann, and M. Mauve, “Probabilistic aggregation for data dissemination in VANETs,” in *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET ’07)*, pp. 1–8, September 2007.

[14] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, “TrafficView: traffic data dissemination using car-to-car communication,” *Mobile Computing and Communications Review*, vol. 8, no. 3, pp. 6–19, 2004.

[15] M. Caliskan, D. Graupner, and M. Mauve, “Decentralized discovery of free parking places,” in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET ’06)*, pp. 30–39, ACM, New York, NY, USA, September 2006.

[16] S. Nath, H. Yu, and H. Chan, “Secure outsourced aggregation via one-way chains,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD ’09)*, pp. 31–44, Providence, RI, USA, 2009.

[17] H. Zhu, X. Lin, R. Lu, P. H. Ho, and X. Shen, “AEMA: an aggregated emergency message authentication scheme for enhancing the security of vehicular ad hoc networks,” in *Proceedings of IEEE International Conference on Communications (ICC ’08)*, pp. 1436–1440, Beijing, China, May 2008.

[18] M. Raya, A. Aziz, and J. P. Hubaux, “Efficient secure aggregation in VANETs,” in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET ’06)*, pp. 67–75, September 2006.

[19] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weilpairing,” *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[20] J. Camenisch, S. Hohenberger, and M. Pedersen, “Batch verification of short signatures,” in *Advances in Cryptology (EUROCRYPT ’07)*, vol. 4515 of *Lecture Notes in Computer Science*, pp. 246–263, Springer, New York, NY, USA, 2007.

[21] T. D. C. Little and A. Agarwal, “An information propagation scheme for VANETs,” in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pp. 155–160, September 2005.

[22] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, “MDDV: a mobility-centric data dissemination algorithm for vehicular networks,” in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET ’04)*, pp. 47–56, October 2004.

[23] P. Flajolet and G. N. Martin, “Probabilistic counting algorithms for data base applications,” *Journal of Computer and System Sciences*, vol. 31, no. 2, pp. 182–209, 1985.