SeDyA: Secure Dynamic Aggregation in VANETs

Rens W. van der Heijden, Stefan Dietzel
University of Ulm
Institute of Distributed Systems
rens.vanderheijden@uni-ulm.de
stefan.dietzel@uni-ulm.de

Frank Kargl
University of Ulm, Inst. of Distributed Systems
University of Twente, DIES Group
frank.kargl@uni-ulm.de

ABSTRACT

In vehicular ad-hoc networks (VANETs), a use case for mobile ad-hoc networks (MANETs), the ultimate goal is to let vehicles communicate using wireless message exchange to provide safety, traffic efficiency, and entertainment applications. Especially traffic efficiency applications benefit from wide-area message dissemination, and aggregation of information is an important tool to reduce bandwidth requirements and enable dissemination in large areas. The core idea is to exchange high quality summaries of the current status rather than forwarding all individual messages. Securing aggregation schemes is important, because they may be used for decisions about traffic management, as well as traffic statistics used in political decisions concerning road safety and availability. The most important challenge for security is that aggregation removes redundancy and the option to directly verify signatures on atomic messages. Existing proposals are limited, because they require roads to be segmented into small fixed-size regions, beyond which aggregation cannot be performed. In this paper, we introduce SeDyA, a scheme that allows more dynamic aggregation compared to existing work, while also providing stronger security guarantees. We evaluate SeDyA against existing proposals to show the benefits in terms of information accuracy, bandwidth usage, and resilience against attacks.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection (e.g., firewalls)*; C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms; Design; Security

Keywords

VANETs; secure aggregation; multi-hop communication

©ACM, (2013). This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version is set to be published in the Proceedings of the ACM WiSec Conference, 2013, where it was accepted as a full paper.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

1. INTRODUCTION

In the past decade, research on vehicular ad-hoc networks (VANETs) has developed from a challenging research application of mobile ad-hoc networks (MANETs) to a viable type of networks that is set to be deployed in the coming decade. A number of standards have been developed, including IEEE 802.11p-2010 Amendment, which was recently incorporated into 802.11-2012 [19], the IEEE 1609 draft standards [20], and ETSI TS 102 637-* [7–9] to facilitate this deployment. These standards provide specifications for on-board units (OBUs) with which vehicles will be equipped and the communication they will use to create a VANET. Communication will potentially be supported by a network of road-side units (RSUs) that provide connection to a back-end infrastructure and the Internet. However, RSU coverage is likely to be sparse, especially on highways. The core research challenges of VANETs include the highly dynamic network topology and short reaction times, which provide strict bandwidth limitations.

For some envisioned applications for VANETs, it suffices to provide high-frequency periodic one-hop communication (beaconing) [8] between vehicles, to exchange current position, speed, and other environmental data. However, many next-generation applications, especially traffic efficiency applications, require multi-hop communication, creating a demand for even more bandwidth. To improve traffic efficiency, knowledge about the area beyond the direct neighborhood of the vehicle is necessary, typically in the order of several kilometers. Such knowledge allows for the detection of traffic jams and alternative routes. However, it is not feasible to simply forward messages generated by beaconing in large areas, due to the well-studied broadcast storm problem [27]. However, applications such as traffic information systems often do not require exact information for their decisions. Information about average speed in certain regions is enough to enable efficient routing. Hence, many authors have proposed to apply in-network aggregation to VANETs [2, 21, 22, 26, 29]. While aggregation can be applied to many different applications, including traffic information systems, road conditions, temperature, and parking spot availability, we will focus on traffic information systems as the main use case for the remainder of the paper. However, we note that our scheme is flexible enough to be applied to other domains, as well. In traffic information systems, aggregation does not only involve the computation of a sum, count or average, but also the dynamic selection of the area over which the aggregate should be computed.

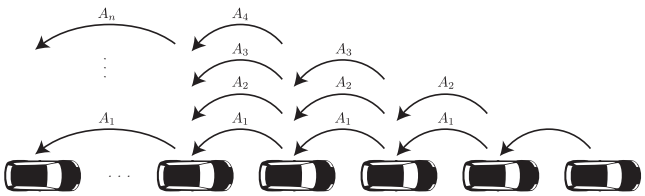
Besides bandwidth efficiency, security is a highly challeng-

ing goal for VANETs, due to the potential impact of a successful attack, which could cause crashes or traffic jams [3]. To provide security, the IEEE 1609.2 standard [18] provides certification for each vehicle, requiring signatures on all the messages that are sent. For privacy protection, short-term certificates, typically called pseudonyms, are used. While signatures and certificates provide reasonable sender attribute authentication, hence preventing attacks with commodity hardware, it is likely that insider attackers will be able to extract key material from cars they own and use it for attacks. Therefore, additional security mechanisms are required, both to detect misbehavior of nodes, as well as to maintain a minimum level of information accuracy. For aggregation, security is even more challenging, because vehicles merge information received from several other vehicles and remove redundancy, as well as the original signatures and certificates. Then, only the aggregates information is disseminated further. Thus, unstructured aggregation can allow an attacker to claim his sensor readings are supported by hundreds of vehicles, while these vehicles never existed. Moreover, legitimate nodes may transmit invalid data if they aggregate a set of messages that includes messages from an attacker. Conventional cryptographic signatures, as provided by IEEE 1609.2 [18], cannot solve these two challenges, because they do not foresee advanced cryptographic protocols beyond basic signatures and certificates.

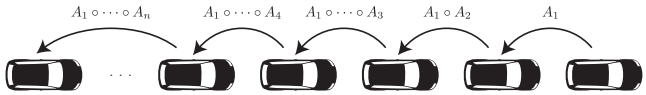
To address these shortcomings, researchers have proposed several new aggregation mechanisms that explicitly address security. Most of these mechanisms are based on a specific underlying aggregation mechanism, which assumes a fixed segmentation of the road. Such aggregation mechanisms are known to not scale well to large area dissemination. Moreover, most existing schemes employ probabilistic counting techniques, such as Flajolet-Martin (FM) sketches [12] as their underlying data structure. Due to the probabilistic nature of the sketches, the resulting schemes that allow aggregation of discrete values still allow attackers to influence aggregated values within certain error bounds.

In this paper, we introduce a new scheme, called Secure Dynamic Aggregation (SeDyA), to provide a stronger and more flexible security mechanism, while still remaining feasible in terms of bandwidth requirements. In particular, our mechanism uses aggregation with flexible road segmentation to allow for good scalability. Like existing schemes, we use secured FM sketches as a basis for our aggregation mechanism, but add to the basic signature-based security mechanisms with a combination of plausibility checks and advanced cryptography, in particular multisignatures and identity-based signatures, to provide stronger guarantees. We evaluate our mechanism against multi-hop beaconing, as well as existing secure aggregation schemes to assess the bandwidth efficiency, accuracy, and security against attacks. In addition, a variant of our scheme that applies only our novel security mechanism, without the security on the FM sketches themselves, provides an improved trade-off between security and bandwidth consumption.

The remainder of the paper is structured as follows: first we introduce the requirements and attacker model in Section 2, followed by a discussion of related work in Section 3. We discuss our new scheme in Section 4 and evaluate it in Section 5. Section 6 concludes the paper with a discussion of open issues and future work.



(a) Atomic message dissemination



(b) Aggregation

Figure 1: Comparison of atomic message dissemination and aggregation.

2. SYSTEM MODEL

To foster a better understanding of our proposed security mechanisms, we will introduce a generic model for data aggregation mechanisms, and outline the specific attacker model we assume.

2.1 Network Model

VANETs are a form of mobile ad-hoc network, which is formed by vehicles on the road. Each vehicle is equipped with a wireless communication device operating at 5.9 GHz according to the IEEE 802.11p-2010 Amendment now incorporated in 802.11-2012 [19]. Wireless communication operates similar to the well-known 802.11a-1999 Amendment in 802.11-2012 [19], but on dedicated frequencies. The expected communication range is approximately 250 meters, depending on shadowing effects. Some papers foresee that the networks are supported by roadside infrastructure, which could relay information from and to back-end systems. Additionally, cellular networks could be envisioned to provide additional information dissemination means. However, both kinds of infrastructure are unlikely to be deployed on a large scale. Especially in highway scenarios, the availability of infrastructure is unlikely in the near future. Thus, communication protocols should ideally rely solely on communication between vehicles, possibly spanning multiple hops. The main challenge for multi-hop protocols is to overcome the ephemeral nature of vehicular communication due to high vehicle mobility.

Both periodic single hop data dissemination and multi-hop dissemination of event notifications have recently been standardized in Europe [8, 9] and the US [20]. However, no mechanisms have been standardized yet to support periodic multi-hop dissemination of environmental data, such as current traffic situation. One of the reasons for missing standardization is that simple relaying of information is too bandwidth inefficient to be used for large scale dissemination of messages, as shown in Figure 1.

Hence, a number of protocols have been proposed (cf. Section 3.1) that employ information aggregation. The basic idea is simple: whenever a vehicle receives information from a neighbor, it decides whether the received information can be merged with already known information, and only the merged information is disseminated further. Further vehicles do the same, resulting in an aggregated view of a stretch of road. For instance, a traffic information system will form

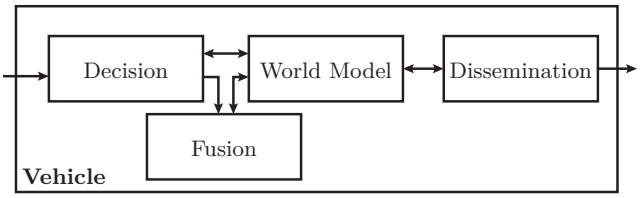


Figure 2: Basic architecture model of aggregation mechanisms.

aggregates like "there is a traffic jam from kilometer X to kilometer Y on road Z", which can be encoded efficiently and disseminated in large regions.

Due to the lack of infrastructure, the aggregated view is created as a collaborative effort of all vehicles on the road, and each vehicle operates as an equal peer. While variations are possible, most aggregation mechanisms follow the basic structure shown in Figure 2 [4]. Four main components define the aggregation process. A vehicle receives information from local sensors or remote vehicles.

- The **decision** component compares newly-received information with other information already contained in the *world model* and decides whether two items of information are similar enough to be aggregated.
- The **fusion** component takes two items of information and merges them to form an aggregate, for instance by averaging the speed contained.
- The **world model** represents a vehicle's current knowledge about the surroundings.
- Periodically, the **dissemination** component selects a, possibly further summarized, subset of the *world model* for dissemination to other nodes.

The main challenge of the different components is to create a bandwidth-efficient yet non-biased summary of the real world situation that is still accurate enough to support applications such as advanced navigation systems.

2.2 Attacker Model

We assume an insider attacker that aims to disrupt traffic, typically for his own gain, by creating false traffic jam reports or hiding real traffic jams. Besides such targeted attacks, we also assume attackers interested in disrupting the aggregation mechanisms, exploiting denial of service attacks, and reducing the quality of disseminated information by as much as possible.

We assume that the entire implementation of the aggregation mechanism and all its parameters are known to the attacker, allowing the computation of a maximum impact given sufficient information about other vehicles on the road. Moreover, the attacker has complete control of the OBU of any vehicle he physically owns, including key material. However, the attacker cannot obtain private key material from other cars remotely. Like most other works, we do not consider denial of service attacks that simply disrupt the network by jamming globally, or by violating MAC protocol parameters to prevent any transmissions from occurring.

Although the attacker may control multiple vehicles, it is assumed that the majority of the network participants is honest. Due to the relatively large aggregation areas that SeDyA will use, this assumption also implies that the aggregation area contains an honest majority. The assumption of an honest majority is motivated by the price of a vehicle. The attacker cannot obtain many pseudonyms for the same vehicle in the same timespan, because we assume that the pseudonym mechanism is secure. This means that the pseudonyms provided to the vehicle are bounded to use in a relatively short timespan, on the order of a few minutes.

However, for location privacy, it has been shown that it is not sufficient to have one pseudonym at any time. To solve this issue, pseudonym exchange schemes like Mix-Zones [15] use several pseudonyms in the exchange period, requiring a period in which an attacker can have more than one pseudonym. There are many pseudonym mechanisms in related work, but a discussion of these is out of scope for this paper. For the purpose of this work, we assume that the attacker is bounded to at most three pseudonyms at any time, which allows for the complete overlap of two pseudonym change periods.

3. RELATED WORK

Data aggregation is a well-researched field with applications in different domains that require data summarization to achieve higher efficiency. Before the advent of VANETs, aggregation has been widely researched in the domain of wireless sensor networks (WSNs). Several mechanisms for data aggregation [11] and corresponding integrity protection mechanisms [25] have been proposed. In contrast to VANETs, WSNs typically consist of low-cost sensor devices, which are deployed in the field to collect a set of data. Data aggregation is used to combine data from several nodes, which is then forwarded to central back-end systems. Aggregation mechanisms developed for WSNs are typically not applicable to VANETs, because they assume limited node mobility, hierarchic structures, and few data sinks interested in the aggregation result.

Hence, a different set of mechanisms, which is optimized for high node mobility and a large set of interested vehicles, has been proposed for aggregating data in VANETs. We will first survey a number of aggregation schemes that do not consider security specifically to highlight a number of challenges when designing an efficient aggregation schemes for VANETs. Following, we introduce existing work on secure aggregation.

3.1 VANET Aggregation

One of the first mechanisms proposed is SOTIS [29]. Here, the road is divided into segments of fixed size, which correspond to wireless range. All vehicles continuously send beacons containing their current velocity, and beacons from the same segment are combined by calculating the average speed. Only the average speed per segment is disseminated in a larger area. While SOTIS reduces the communication overhead by only forwarding averages per segment, it does not scale to larger areas [26]. Since segment size is fixed, the total bandwidth reduction achieved is constant, as opposed to schemes which use larger segment sizes to disseminate information about areas further away. Moreover, fixed segments can fail to correctly depict small traffic jams at segment borders, and waste bandwidth in case of larger phenomena spanning multiple segments, which could be aggregated further. Hence, newer aggregation schemes use flex-

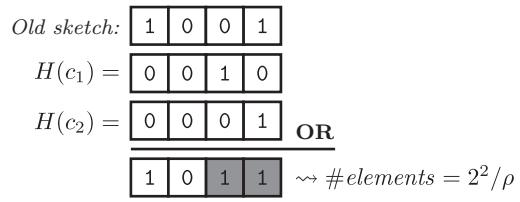


Figure 3: Two vehicles with identities c_1, c_2 add their count to an existing sketch. The initial sequence of uninterrupted ones, here $l = 2$, is used to approximate the total number of values in the sketch.

ible road segmentation [2, 3, 22] to better adapt to the real situation on the road.

The aforementioned schemes suffer from the problem of duplicate reports. If a vehicle's velocity is counted multiple times in the same aggregate, the resulting average will be biased. Lochert et al. [21] propose to use an enhanced version of Flajolet Martin (FM) sketches [12] for aggregating the number of free parking spots in certain city regions with automatic duplicate elimination. We will shortly introduce the FM sketch data structure, because several secure aggregation mechanisms build on the idea of securing FM sketches against malicious manipulation.

Originally used for large databases, FM sketches are a counting method that can be used to approximate the total number of distinct unique values with low storage overhead. The data structure used is a sequence of m bits, which is initially set to all zeros. To increase the sketch's count by 1, a hash $H(c_i)$ of an element's unique id c_i is calculated where H is a geometric hash function. Using a geometric hash function, the probability that the output is n is $1/2^n$. Then, the $h(i)$-th bit in the sketch is set to 1, as shown in Figure 3. Because the hash is based on the unique id, the same element can be added multiple times without setting more than one bit to 1. The number of elements in the sketch can be approximated using the length l of the initial uninterrupted sequence of one bits in the sketch as:

$$\#elements = 2^l/\rho$$

with $\rho \approx 0.775351$. To increase the approximation's accuracy, multiple sketches can be used and their results averaged. The approach using multiple sketches is called Probabilistic Counting with Stochastic Averaging (PCSA) [12].

In addition to simple counts, sketches can be adapted to represent sums and averages. Due to their efficient representation of duplicate insensitive merging functions, they have been widely adopted for aggregation schemes.

3.2 Secure Aggregation

The aggregation mechanisms discussed so far do not consider security, most notably integrity, of aggregates explicitly. Hence, it is possible for a malicious attacker to modify the reported values. Raya et al. [24] propose a security mechanism that can be applied to fixed segments aggregation schemes, such as SOTIS. Once all vehicles within a segment have agreed on an average value, the goal is to secure the average against further modification. The paper discusses three different signature schemes to achieve integrity protection. Simply attaching all participating vehicles' signatures as a list achieves the lowest computational overhead while requiring a lot of bandwidth to accommodate all signatures. Onion signatures, meaning that each vehicle resigns the aggregate's existing signature instead of adding its own signature to a list, reduce the bandwidth usage. On the other hand, they increase the computational overhead, because each receiving vehicle needs to re-calculate all signatures. The biggest limitation of the scheme is that it can only be applied to fixed segments aggregation, which has been proven to not scale well [26].

Instead of first agreeing on an aggregate value and then signing it, Dietzel et al. [5] add signed atomic reports, which have been used in an aggregate's calculation, as attestation meta-data. This meta-data is then used to verify that the atomic values have been correctly aggregated. To save bandwidth, only a subset of of all atomic values is added. The subset is chosen such that the atomic reports' locations are equally distributed throughout the aggregate area. Still, the resulting mechanism can only provide a heuristic for the aggregates' correctness. Because not all signed atomic reports are added, an attacker can still modify the resulting aggregate within certain limits. However, the scheme can be applied to arbitrary dynamic aggregation schemes, because no fixed values need to be agreed before calculating the attestation meta-data.

Picconi et al. [23] follow a different idea for achieving probabilistic integrity protection. Their scheme borrows from interactive commitment schemes where a sender calculates the aggregated value and a cryptographic commitment on the value and sends both to the receiver who then asks for one or more atomic reports used in the calculation. To avoid interactivity, Picconi et al. use trusted hardware in the sender's vehicle, which challenges the commitments instead of the actual receiver. In case of misuse, the role of the trusted hardware is to guarantee that a proof of the failed challenge is sent to allow other vehicles to detect the attacker. Beyond the reliance on trusted hardware, one of the scheme's problems is that it can still only provide a probabilistic detection of attackers.

A first attempt to secure an FM-sketch-based aggregation scheme is made by Garofalakis et al. [13]. The authors only consider aggregation schemes, which count the number of witnesses of a binary event, such as "there is an accident at position X." Each vehicle that agrees to the event hashes its id into the corresponding FM sketch. For each bit set to 1, the vehicle id, bit position in the sketch, and a signature on these values is added as proof data. Hence the sketch is protected against inflation by an attacker. However, a signature and corresponding certificate have to be kept for each bit set to 1. To protect against deflation, a second sketch is kept that represents the inverse count c', that is, $c' = N - c$ if c is the actual count and N the maximum expected value.

Similarly, [17] counts witnesses on binary events, but reduces the overhead of signatures and certificates needed. Instead of FM sketches, z-smallest is used as underlying probabilistic counting method. The idea of z-smallest is that, given n elements uniformly distributed between 0 and 1, the z-smallest element gives an approximation of n by calculating z/c where c is the value of the z-smallest element. To protect against inflation, the authors exploit the fact that it is difficult for an attacker to forge z signed reports with a value smaller than or equal to c.

Han et al. present SAS [16], a scheme to protect average

values rather than binary events. The authors assume that the underlying aggregation scheme uses fixed segments to calculate the average values. Moreover, the authors assume that vehicle only report aggregates to traffic management centers (TMC), which allows then to use symmetric cryptography and shared keys between each car and the TMC. As counting method, FM sketches are used. However, a number of enhancements over Garofalakis' scheme are proposed.

For inflation protection, again signatures for each 1 bit are generated. However, the signatures on those 1 bits that are part of the initial uninterrupted sequence of ones are merged. Only the signatures of the additional 1 bits are kept separate. To protect against deflation, the authors propose to use a hash chain that represents the length of the initial sequence of one bits. Because the hash cannot be inverted, attackers cannot remove bits from the sequence. The size of the hash chain is constant, as is the combined signature on the initial sequence of ones. Only the extra signatures on one bits, which are not part of the initial sequence, use extra space.

SAS eliminates the limitation to binary events and offers a number of bandwidth improvements over other secure aggregation schemes. In addition, SAS provides good protection against inflation and deflation of sketch values. However, the scheme only allows central infrastructure to check the integrity of aggregates. More importantly, the scheme is limited to fixed segments and consequently suffers from limited scalability.

3.3 Open Issues

Having discussed related work, we now point out some key remaining issues, which are the goals that our scheme, SeDyA will aim to solve. The challenge is twofold: dynamic aggregation and secure aggregation.

First, an issue that all secure aggregation schemes have in common is that they are bounded to a predefined aggregation area. This area is defined either completely in advance (fixed segments) or determined when the aggregate is first generated and not changed afterwards. On the other hand, when we examine most VANET aggregation schemes that do not consider security, we observe that the aggregation is performed in a dynamically defined region. This key distinction prohibits secure aggregation schemes from achieving the same accuracy and scalability as aggregation schemes that do not consider security.

Second, there is the challenge of security. A core challenge of secure aggregation is that a receiver of an aggregate must be able to verify that the aggregation process was preformed correctly. One solution that allows such verification is the set of mechanisms applied by SAS [16]. However, SAS does not fit our requirements, as our goal is to directly disseminate information in the network rather than reporting summarized to a centralized back-end. Thus, the goal for SeDyA is to provide a way to allow for dynamic aggregation and address the requirements of typical VANET aggregation scenarios, as discussed in the previous sections. Beyond dynamic segments, the removal of the TMC allows for potential vulnerabilities, as the hash chains that SAS uses can no longer be employed. This is due to the fact that the hash chains are bound to the usage of a central authority (the TMC). We will address these vulnerabilities in the construction of our new scheme, SeDyA.

4. SEDYA

SeDyA provides improvements on related work in two important areas: security and dynamic aggregation. The mechanism is divided into three phases, as shown in Figure 4: the *aggregation phase* (Phase 1) where vehicles collaboratively agree on an aggregate, the *finalization phase* (Phase 2) where additional signatures are added to attest aggregate correctness, and the *dissemination phase* (Phase 3) where the finalized aggregates are disseminated in larger regions. We will provide a short overview of all three phases before explaining each in more detail.

For example, consider a traffic jam of 6 kilometers length. The traffic jam can be described by indicating the 6 km section as the area and providing an aggregated average speed that is close to zero. Because the length of the traffic jam is unknown to the first vehicles initiating the aggregation process, the *aggregation phase* allows a dynamic description of the area. The aggregation phase ends when a minimum duration has expired and a node detects the edge of an area of homogeneous speed. This allows for accurate aggregates with relatively low standard deviation between the contained atomic speed values and a good approximation of the real world situation. The aggregation phase also contains an optional security mechanism, which simplifies earlier detection of malicious aggregates at the cost of additional bandwidth usage.

When the edge of a homogeneous speed area is detected, the *finalization phase* is initiated, which provides additional security: in this phase, the aggregate message is transmitted from the node on the edge back through the aggregation area. As the message is forwarded, each forwarding node checks the contents of the message against its observed values. If the message is accurate, the forwarder attaches a certificate and signature and forwards the message; otherwise, it discards the message. Instead of using regular ECDSA signatures and public key certificates, an identity-based multisignature scheme is used to reduce the signature size. In order to focus on the conceptual aspects of our scheme, we omit details of the cryptographic mechanisms used in this section. However, the underlying cryptography is discussed in Appendix A.

Eventually, the forwarding process will reach the other end of the aggregation area. Once a node detects that its location is not contained within the aggregation area, it initiates the *dissemination phase*. In this phase, the message is disseminated to potentially interested vehicles further away. Each receiving node can compare the amount of attached certificates to the estimated amount of participants to determine the quality of the aggregate. In addition, SeDyA's multisignature provides a method to verify the amount of vehicles that agree with the aggregate. This can be compared to the amount of vehicles that the aggregation mechanism indicates, when performing plausibility checks and the resolution of conflicting information.

4.1 Aggregation Phase

In the aggregation phase, the goal is to agree on the contents of the aggregate. That is, the aggregation area must be determined, and the average speed must be measured. First, a vehicle (e.g., v_1 in Figure 4) encodes the average speed into an aggregate A_1. In addition, the – initially small – aggregation area needs to be encoded. We use two types of probabilistic counting here: FM sketches and LC sketches.

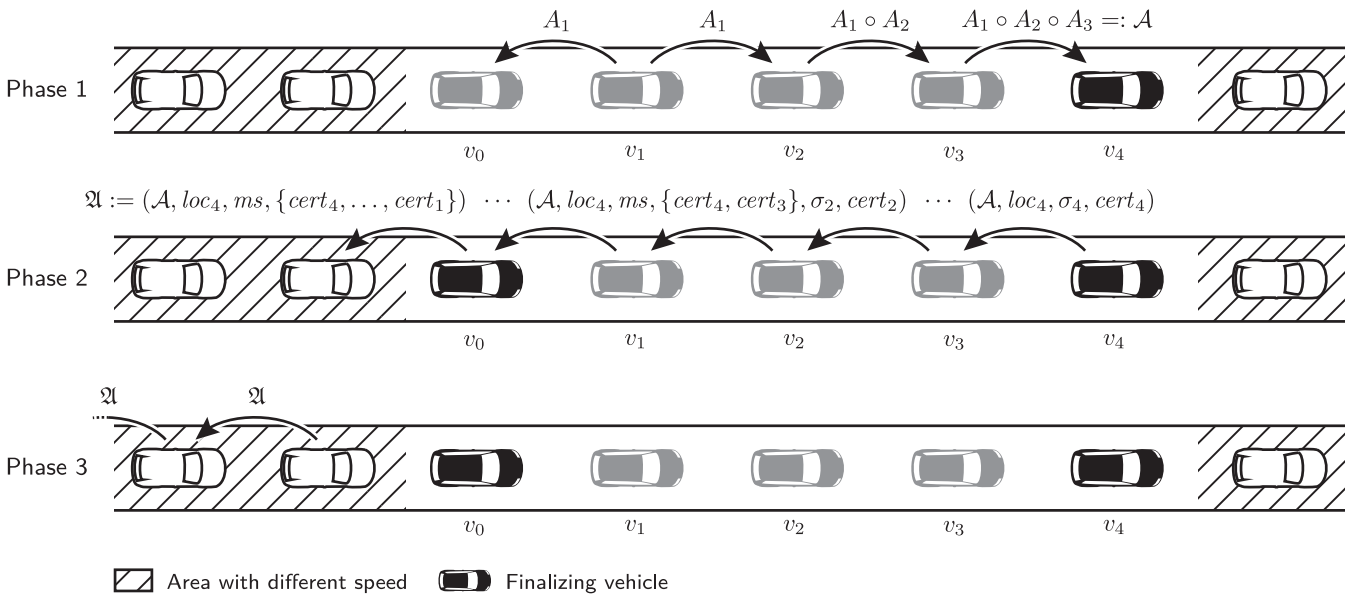
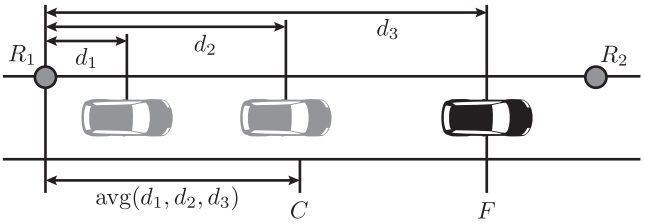


Figure 4: This figure illustrates the different phases of SeDyA. In particular, it shows the aggregate-and-verify steps in the first and second phase.

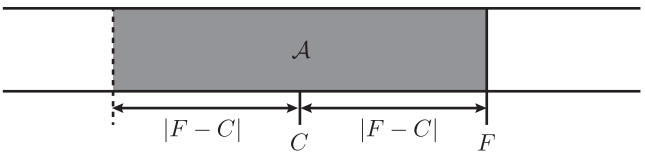
LC sketches are a variation on FM sketches that is designed to be more accurate, but they cannot represent large values as well as FM sketches can. Their functionality is described and compared to FM sketches in [10]. We use LC sketches to count the amount of participants in the aggregation area, while we use FM sketches to describe the bounds of the area and to describe the measured average speed itself. We apply PCSA over several FM sketches to increase the accuracy of FM sketches at the cost of additional hash operations and bandwidth.

Encoding the average speed is a straight-forward application of FM sketches. To create a dynamic aggregation scheme based on FM sketches, we eliminate fixed segments used by related work and encode the aggregation area as follows. First, the road is marked with a set of fixed *reference points.* Instead of storing its absolute location, each vehicle calculates its relative distance to the last passed reference point, marked as d_i in Figure 5a. We use an FM sketch to store the average of these relative distances as the center of the aggregate's area (C in the figure). Thus, when v_1 creates an aggregate A_1, it will approximate $R_1 - v_1$; when v_2 receives A_1, it computes a new aggregate that will approximate $avg(R_1 - v_1, R_2 - v_2)$, and so on. Whenever a vehicle contributes to the aggregate, the average center is updated by adding its relative distance to the FM sketch and increasing the vehicle count. In the aggregation phase, the area is not yet well-defined, because it can still expand. Once a vehicle determines that it is on the edge of an area of homogeneous speed, it enters the finalization phase, and the area is defined by the location of this edge vehicle (F in Figure 5b) and the center of the area C. The assumption here is that, due to the homogeneity of the vehicles' speeds, the vehicle distribution is mostly uniform within the aggregate area.

During the aggregation phase, aggregates are disseminated as follows. Whenever a new aggregate is started, it is broadcast to direct neighbors. Each vehicle will merge the received aggregates with that of other vehicles and its own observations, by merging the associated FM and LC sketches. The vehicles then forward the message after a random waiting time. This forwarding is bounded, to protect against a typical broadcast storm problem. The amount of messages a node may transmit every second is set to a contention-dependent value between 1 and 10, inversely dependent on the size of the neighbor table. At each merge operation, each vehicle also checks whether it is at the edge of an aggregation area. To detect this, we use the vehicle's sensors and neighbor table to detect a change in average speeds. Each vehicle compares its velocity with the neighbors in front and



(a) Calculation of aggregate dimensions.



(b) Interpolation of aggregate area.

Figure 5: During the aggregation process, the aggregate's center C is calculated as the average relative distance between participating vehicles and a reference point (R_1). Assuming uniform distribution of vehicles, the aggregate area $\mathcal{A}$ is interpolated as $2 \cdot |F - C|$ where F is the position of the finalizing vehicle.

behind. If the speed difference is higher than a threshold, the aggregate area edge is reached, and the aggregate can be finalized.

Before we discuss the finalization phase in detail, we address the security of the aggregation phase. We consider the security mechanism to be optional in this phase: while it improves the scheme through early detection of malicious aggregates, the main security contribution of SeDyA is in the next phase (the *finalization phase*). The security mechanism in the aggregation phase is inspired by two mechanisms from related work; SAS [16] and AM-FM sketches [13]. We create a signature for each bit in the FM or LC sketch when the bit is set to 1. The values signed are the position of the respective bit, the current time slot, and the reference point with respect to which the added distance is computed. Note that the reference point is the same for every participating vehicle, a prerequisite for the signatures to be verifiable by other vehicles. This approach to securing FM sketches is comparable to the mechanism that SAS and AM-FM apply.

As previously discussed, we use FM sketches to compute averages: this means that each vehicle can sign more than one bit in each sketch. This is analogous to the method used in SAS [16]. In AM-FM sketches [13], this mechanism is not used; instead, multiple rounds of communication are executed to produce an average. As multiple rounds of communication are not feasible in our setting, because we do not aggregate towards a single sink node, using FM sketches to compute averages is the only feasible approach.

However, this approach leaves an open attack vector. An attacker can exploit the fact that he possesses key material to modify an aggregate as desired and then sign the necessary bits. Such manipulation is also possible when attacking schemes in related work, but more challenging in proposals that use a central authority. Although this attack on SeDyA can often be detected and traced back to the attacker through plausibility checks, an intelligent attacker could use the approach to influence the output while remaining hidden. Therefore, we provide an additional security mechanism in our finalization phase. The key difference is that the mechanism in the finalization phase only operates after the aggregation phase; an attack will thus be detected later than by using security in the aggregation phase. We note that due to the significant overhead involved in the security mechanism in the aggregation phase, it may be more efficient to only employ the security mechanism in the finalization phase, despite the delayed detection of an attack.

4.2 Finalization Phase

In this phase, the goal is to provide vehicles within the aggregation area the opportunity to verify the contents of the aggregate. The first step is to produce the finalized aggregate message. This message consists of the aggregate from the previous phase, the current location of the finalizing vehicle, a signature, and a certificate. The finalization location is stored to allow receivers of the message to reconstruct the aggregation area, as described in the previous section. The signature of the finalizing vehicle signifies that it agrees with the contents of the aggregate. Here "'agreeing"' means that the contained average does not deviate by more than a predefined threshold from the vehicle's own speed readings. The certificate is attached to enable receivers to verify the signature.

Because more than one vehicle may detect the edge of the aggregate at the same time, there is a short waiting time followed by a repetition of the same message to agree on one finalized aggregate to be disseminated further. After waiting, the finalized message of the vehicle with the lowest ID is selected for further dissemination. At this point, vehicles within the aggregation area may initiate the forwarding protocol. The aim of this protocol is to allow each vehicle to add its signature and certificate to the finalized message. Due to the high bandwidth requirements for implementing this signature process naively, we apply multisignatures to reduce the amount of signatures. Using multisignatures, several signatures can be merged, thus saving bandwidth. However, a list of all corresponding certificates is still necessary to verify the signers' attributes. Hence, we use identity-based signatures to reduce public key size compared to regular certificates. The details of these cryptographic mechanisms are discussed in Appendix A.

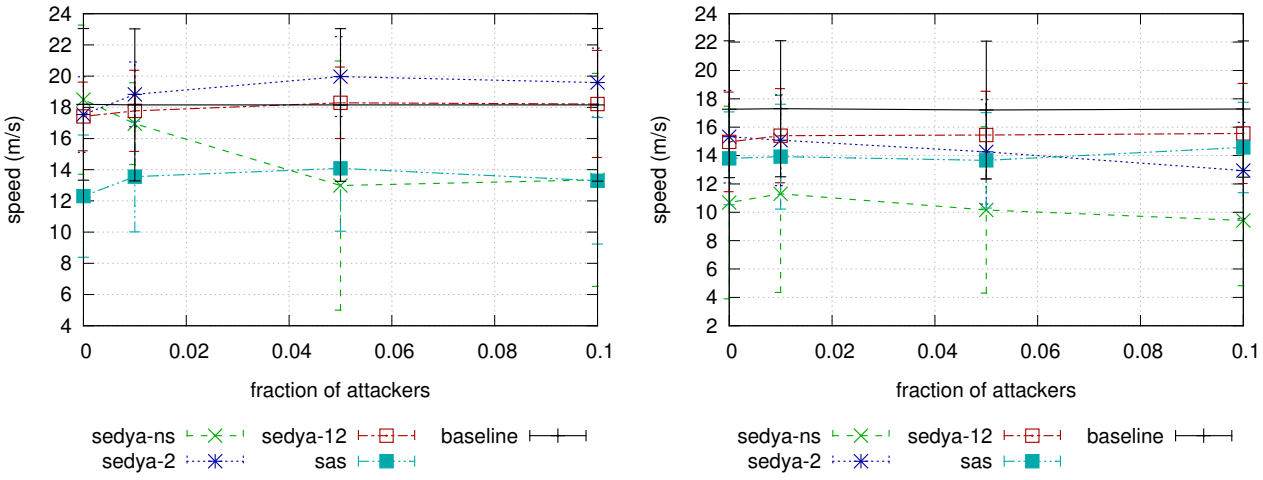
When a vehicle receives a signed message in the finalization phase, it will only forward it if it is a correct and legitimate message. This removes the majority of errors and naive attacks. The decision of whether the message is legitimate is a purely data-centric one: the vehicle checks whether it is inside the aggregation area, and then compares its own sensor readings with the contents of the message. If the message aligns with the vehicle's sensor readings, the vehicle signs the message and forwards it. To protect the network from a broadcast storm problem, we bound the maximum amount of messages a vehicle is allowed to broadcast using the same mechanism as discussed in the previous phase. Due to this redundant, flooding-based message dissemination, it is unlikely that an attacker can selectively drop messages to prevent the transmission of an aggregate.

Because of the non-deterministic forwarding mechanism, it is possible that a vehicle receives multiple versions of the aggregate at the same time, but with different, overlapping sets of previous signers. The overlap can be detected because of the attached identity sets, but the multisignature cannot be decomposed. Hence, the merging vehicle would need to create a message with duplicate identities in the resulting identity set. To resolve this, we introduce a specific message format, where each forwarded message consists of the aggregate, the multisignature, the set of certificates, and a separately-kept signature and certificate of the current sender. This allows to resolve duplicate identities when merging aggregates; details of the mechanism are discussed in Appendix B.

Finally, we stress that the goal of the finalization phase is not just to produce a valid multisignature. The multisignature also transfers the information that sufficient nodes in the aggregation area agree with the aggregated data. This fact allows the receiver of such a multisigned message to estimate the amount of vehicles that agree with the message, compared to the amount of participants in the aggregation process. We thus exploit the properties of digital signatures to transfer information about the results of plausibility checks of the signers. As a result, we can transfer trust in the aggregate correctness over multiple hops, allowing for dynamic and large aggregation areas.

4.3 Dissemination Phase

Finally, the dissemination phase is where finalized aggregates are distributed throughout the network. The dissemination phase is initiated when a message reaches the op-



(a) Low density

(b) High density

Figure 6: This figure shows the average speed over one period for the low and high density scenarios, with a varied amount of attackers, for each of the studied schemes.

posite edge of the aggregation area, as shown in Figure 4. Once, the aggregate is received by vehicles that are not in the area, they switch to the dissemination phase. Each vehicle simply forwards the messages it receives using any generic efficient dissemination scheme, such as advanced adaptive geocast [1]. To compute the confidence value for a message, receivers determine the amount of participants in the multisignature and compare it to the reported amount of participants in the aggregation process, as described in the previous section. This check allows receivers to identify whether a message can be trusted for navigation decisions. It is possible to build more complex systems to resolve conflicts in data, as has been done in previous work, for instance, using the generic aggregation model from [5]. For such mechanisms, SeDyA provides reliable input concerning the amount of participants.

Finally, we note that the computed confidence values, in addition to data quality metrics, can be used to determine forwarding priorities. This provides a way to allow high quality messages to persist in the network, while low quality messages will be filtered. We consider developing and analyzing a dissemination scheme that uses such metrics in detail an interesting challenge for future work.

5. EVALUATION

In this section, we aim to show that SeDyA achieves the intended goals and is feasible to implement. To do this, we have implemented and simulated SeDyA in a network simulator. The implementation is available from the authors on request.

5.1 Simulation

The network simulator we use is an updated version of the JiST/SWANS simulator [28], which includes enhancements by the University of Ulm, published in 2008, which we have maintained internally[1].

Our simulations are performed on a highway setting, with two lanes and a length of 5 kilometers. Initially, the vehicles are randomly distributed over the road; each vehicle uses a simple mobility model that contains car-following and overtaking. This model is based on the highway model in the Ulm JiST/SWANS distribution. The density of the network is either high (800 vehicles) or low (200 vehicles). The beacon frequency is set to 10 Hz with a beacon size of 209 byte (including security overhead). We use default IP and 802.11p implementations available in the JiST/SWANS distribution. On the physical layer, we use the Ray-Leigh fading model, the TwoRay path loss model, a transmit power of 10.9 dB and an additive noise model. The mobility model is configured for a maximum speed of 36 m/s with an acceleration capability of 1 m/s (5 m/s for breaking) and a target average of 25 m/s. Our simulation ran over a period of 20 seconds, with a single 9 Mbit/s channel for all communication, including beacons.

SeDyA itself is configured to use 64 bits for the LC sketch that counts participants, 4 sketches of 8 bits each for describing the location and 8 sketches of the same size for describing the payload (i.e., the speed measured). In all of these, the MD5 hash is used to hash elements; for PCSA, a second, independent hash is required, for which we use SHA-1. Both of these are implemented using the standard Java API. We remark that it is not required for the security of our scheme that these hashes are cryptographically secure, so MD5 and SHA-1 are sufficient. To simulate pseudonyms, we require that each period in which an aggregate is produced, the vehicle in question signs with a different identifier.

5.2 Attack Implementation

To prove that our scheme works, we need to show that the it is secure. Proving security against external attackers is typically done using security proofs. However, we rely on existing mechanisms that have been shown to be secure against these types of attacks. The novel part of our scheme is designed to protect specifically against insider attacks. In

[1]Due to licensing constraints, this source code is not public; please contact us if you require the source code for your research.

such an attack, a node possess the necessary key material to be a legitimate participant. An attacker typically has control over his own vehicle, and can therefore always manipulate the sensor readings, even when a perfect hardware security module is used. Hence, we concentrate on attacks where malicious nodes try to influence the aggregation mechanism by crafting spoofed aggregates.

For these reasons, we have chosen to provide an attack implementation that abstracts from any particular attack on the scheme. Instead, we provide the attacker with tools to selectively increase or decrease aggregates. In this paper, we show our analysis for the most interesting attack: creating a fake traffic jam. We argue that the converse, attempting to hide a traffic jam, can be prevented analogously. Moreover, active safety mechanisms implemented using CAM messages can prevent accidents in case traffic jams are hidden by an attacker.

We typically assume that a single insider attacker tries to influence the aggregation result. The reason is that attacks on aggregation are especially interesting for attackers that only possess a single or few vehicles. Due to the aggregation, high impact can be achieved with little resources. In addition, we simulate scenarios where multiple independent attackers try to achieve the same goal. These attackers are randomly selected from the set of all vehicles.

Finally, we note that the attacker is assumed to be aware of the security mechanism and plausibility mechanisms in SeDyA. He will thus attempt to maximize the impact of an attack by working within the bounds of the plausibility mechanism that SeDyA provides.

5.3 Results

In this section, we present the results of our evaluation. As metric for performance, we use the speed measured by the aggregation mechanism in addition to a plot with the real speed that the mobility model provides. The speed values produced by the mobility model are the baseline: the best possible scheme would produce exactly this graph. By examining the average speed produced, in addition to the standard deviation, we can see how the scheme performs in the presence of attackers. The standard deviation is particularly important: a high standard deviation compared to the baseline indicates that a significant amount of messages contains higher or lower values. This means that either the aggregation mechanism is particularly poor for some messages, or the attacker is successful for some messages. In either of these cases, the scheme creates uncertainty for the receiver: she can no longer rely on a message, because there is no way to verify which message is good and which message is bad.

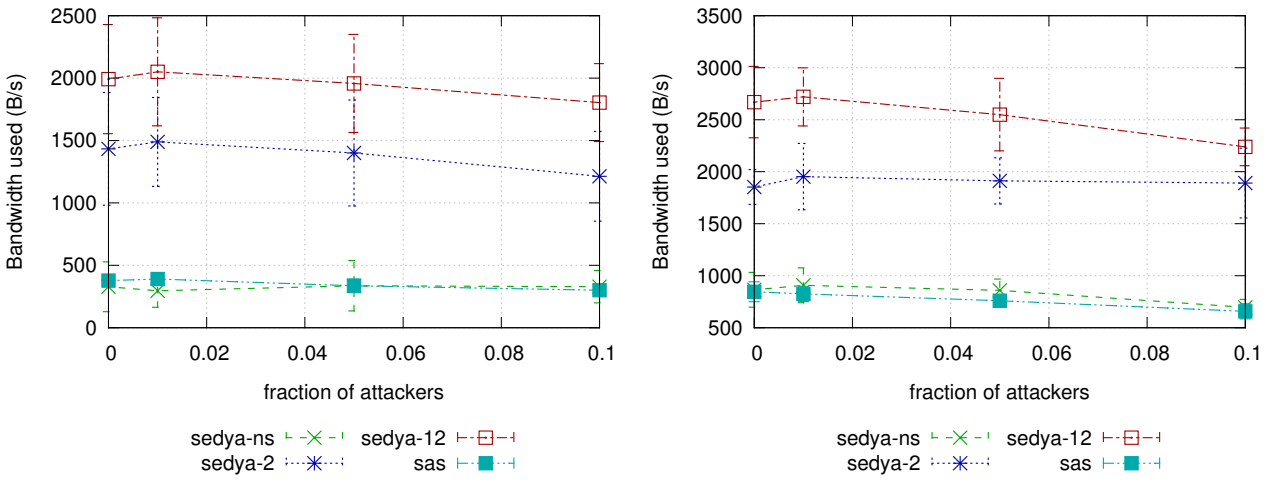
We have varied our simulations over several parameters: the density of the network, type of security mechanism and fraction of attackers. The density of the network is varied by setting the amount of vehicles on the road to 200 (for a density of 0.04 vehicles/meter) or 800 (for a density of 0.16 vehicles/meter).

To assess SeDyA's performance, we compare SeDyA against an improved version of SAS [16]. We take SAS as a baseline and add plausibility checks similar to those used in SeDyA's finalization phase (cf. Section 4.2) in order to make the two schemes more comparable. Also for comparability, we exchanged SAS' symmetric cryptography mechanisms with asymmetric ones, as suggested by the SAS authors in the original paper. The corresponding graphs are labeled `sas`. For SeDyA itself, we evaluate a version of the underlying aggregation mechanism without SeDyA's security added (labeled `sedya-ns`), the full version of SeDyA (`sedya-12`), and a variant where only the security mechanisms presented in the finalization phase are used (`sedya-2`). In addition, we note that `sedya-ns` is not equipped with plausibility checks, while all others (including `sas`) are.

In addition to these communication-based schemes, our graphs include the baseline, which is supplied by the mobility model. The baseline indicates the natural variation in speed on the road; an ideal scheme should result in similar aggregated values. Finally, the fraction of attackers indicates the fraction of vehicles on the road that is malicious, which is varied from 0% to indicate no attacks at all, to 10%, which indicates a high amount of attackers. This fraction is computed over all simulated vehicles: it is possible for the attackers to form a local majority, as the attacker-controlled vehicles are randomly selected from all participants.

In the graphs in Figures 6a and 6b, we show the results for a low and a high density network. Each set of parameters has been repeated eight times. In particular for the low density setting, performance of SeDyA is good: showing an average speed of 17.4 ± 2.2, 17.5 ± 2.5 and 18.5 ± 4.8 m/s for the case without attackers using `sedya-12`, `sedya-2` and `sedya-ns`, respectively. SAS, on the other hand, underestimates the speed at 12.3 ± 3.9 m/s. The `sedya-ns` line clearly shows the impact of an attack, even with few attackers. Both the lower speed and the increased standard deviation indicate that the impact of attacks varies per message, creating high uncertainty for the receiver of any such message. On the other hand, we note that plausibility checks for all the schemes with security avoid high-impact attacks, as in both Figure 6a and Figure 6b, the graphs remain consistent as the amount of attackers increases. We note that for `sedya-12` and `sedya-2`, the standard deviation for each of these points is between 2.0 and 3.7 m/s; for Figure 6a, the standard deviation is lower, while for 6b all standard deviations are greater than 3. The cause for these standard deviations is in part due to the inherent inaccuracy of FM sketches, and due to the natural variation in the traffic, some vehicles will not have accurate information. We note that in SAS, the standard deviation for Figure 6a is significantly higher than SeDyA's, with values between 3.5 and 4m/s, while for 6b it is comparable. SeDyA's reduced performance in the high density scenario can be explained by the fact that the determination of edges is based on average speed reported by neighbors. This leads to larger aggregation areas for higher densities, which has an impact on accuracy.

In addition to the performance of the security mechanism, it is important to consider the bandwidth usage. Thus, we have computed the bandwidth consumed by the aggregation mechanism per vehicle per second in bytes. Clearly, SAS is more efficient in terms of bandwidth due to the flooding-based mechanisms that SeDyA employs in the finalization and dissemination phases. However, SeDyA's overhead is still reasonable due to the fact that our simulations were bounded to a single channel of 9 Mbit/s. In reality, the assignment of channels is still unclear; however, [6] defines 6 Mbit/s for safety applications and an additional two channels (6 Mbit/s and 12 Mbit/s) for traffic efficiency applications. Thus, SeDyA's bandwidth usage, which is on the



(a) Low density

(b) High density

Figure 7: This figure shows the consumed bandwidth in bytes per vehicle per second in the low and high density scenarios.

order of 2 kilobytes per second per vehicle in a high density scenario, is well within the limits of the available bandwidth. When no spatial re-use is assumed, that is, if all vehicles are in each others transmission range, it would take 750 vehicles to use the complete 12 Mbit/s channel. Thus, we conclude that SeDyA is capable of producing improved results, particularly in low density settings, while maintaining a reasonable level of bandwidth consumption.

We remark that our bandwidth requirements are indeed much higher than those claimed by other secure aggregation schemes. However, our scheme is generally applicable to aggregation mechanism, rather than restricting the aggregation process to a particular implementation. In addition to this increased flexibility, we show our scheme with secure FM sketches (`sedya-12`), which provides a higher level of security in exchange for higher bandwidth requirements. Therefore, we illustrate that there is a fundamental trade-off between security and bandwidth efficiency, a trade-off that we have explored using the different schemes we have implemented.

Although the very high level of security that `sedya-12` provides is not always the best trade-off, we claim that it provides a good insight of what is fundamentally possible when faced with insider attackers. `Sedya-2` provides a more balanced trade-off, sacrificing some security for increased efficiency, while still providing good performance. As noted, future work should focus on applying our security mechanism to other aggregation schemes, and the incorporation of SeDyA's multisignature information into a misbehavior detection scheme.

6. CONCLUSION

We have introduced a new secure aggregation scheme that satisfies the unique requirements posed by VANETs. Our main improvements over related work are supporting dynamic aggregation mechanisms and offering additional security guarantees that prevent insider attacks from influencing the aggregation results. Our simulations show that the impact of generically implemented attacks is significantly less than for related work. However, our scheme's bandwidth usage is higher. We also show that our scheme is feasible, despite the increased bandwidth requirements, by running it in a limited bandwidth channel, which also accommodates secure beaconing. Currently, we are further exploring the design space and trade-offs between high levels of protection and bandwidth efficiency. We envision that adaptive schemes could dynamically react to changing network situations and adapt the security overhead accordingly. Moreover, we are investigating a more general attack detection framework that uses information offered by schemes like SeDyA to detect misbehaving nodes in VANETs. In addition, we believe it is possible to apply SeDyA to existing aggregation mechanisms, by replacing the sketches in our first phase with the existing mechanisms. Thus, SeDyA can provide a generically applicable method for securing aggregation schemes that permit finalization and dependable aggregates. Our results using a custom-tailored aggregation scheme show that dynamic aggregation and proper security mechanisms can be combined and significantly reduce the attack vectors for insider attackers.

7. REFERENCES

[1] B. Bako, F. Kargl, E. Schoch, and M. Weber. Advanced adaptive gossiping using 2-hop neighborhood information. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008.*, pages 1–6. IEEE, 2008.

[2] S. Dashtinezhad, T. Nadeem, B. Dorohonceanu, C. Borcea, P. Kang, and L. Iftode. TrafficView: a driver assistant device for traffic monitoring based on car-to-car communication. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 5, pages 2946 – 2950 Vol.5, may 2004.

[3] S. Dietzel, B. Bako, E. Schoch, and F. Kargl. A fuzzy logic based approach for structure-free aggregation in vehicular ad-hoc networks. In *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking - VANET '09*, page 79, 2009.

[4] S. Dietzel, F. Kargl, G. Heijenk, and S. Florian. On the poptential of generic modeling for vanet data

aggregation protocols. In *IEEE Vehicular Networking Conference (VNC)*, pages 78–85, 2010.

[5] S. Dietzel, E. Schoch, B. Konings, M. Weber, and F. Kargl. Resilient secure aggregation for vehicular networks. *IEEE Network*, 24(1):26, 2010.

[6] ETSI. Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band, November 2009. ETSI ES 202 663, final draft, version 1.1.0.

[7] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Functional Requirements, 2010.

[8] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, 2010.

[9] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service, 2011.

[10] Y.-C. Fan and A. L. Chen. Efficient and robust sensor data aggregation using linear counting sketches. In *2008 IEEE International Symposium on Parallel and Distributed Processing*, page 1, 2008.

[11] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 14(2):70, 2007.

[12] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31:182–209, October 1985.

[13] M. Garofalakis, J. M. Hellerstein, and P. Maniatis. Proof sketches: Verifiable in-network aggregation. In *2007 IEEE 23rd International Conference on Data Engineering*, page 996, 2007.

[14] C. Gentry and Z. Ramzan. Identity-based aggregate signatures. In *Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography*, pages 257–273, 2006.

[15] M. Gerlach. Assessing and improving privacy in VANETs. *ESCAR Embedded Security in Cars*, 2006.

[16] Q. Han, S. Du, D. Ren, and H. Zhu. SAS: A secure data aggregation scheme in vehicular sensing networks. In *2010 IEEE International Conference on Communications*, page 1, 2010.

[17] H.-C. Hsiao, A. Studer, R. Dubey, E. Shi, and A. Perrig. Efficient and secure threshold-based event validation for VANETs. In *WISEC'11*, pages 163–174, 2011.

[18] IEEE. IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages, 2006. IEEE Std 1609.2-2006.

[19] IEEE. IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007), March 29 2012.

[20] IEEE. IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE). IEEE Standard 1069.*, 2012.

[21] C. Lochert, B. Scheuermann, and M. Mauve. Probabilistic aggregation for data dissemination in VANETs. In *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks - VANET '07*, page 1, 2007.

[22] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode. Trafficview: a scalable traffic monitoring system. In *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on*, pages 13 – 26, 2004.

[23] F. Picconi, N. Ravi, M. Gruteser, and L. Iftode. Probabilistic validation of aggregated data in vehicular ad-hoc networks. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks - VANET '06*, page 76, New York, New York, USA, 2006. ACM Press.

[24] M. Raya, A. Aziz, and J.-P. Hubaux. Efficient secure aggregation in VANETs. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, VANET '06, pages 67–75. ACM, 2006.

[25] Y. Sang, H. Shen, Y. Inoguchi, Y. Tan, and N. Xiong. Secure data aggregation in wireless sensor networks: A survey. In *PDCAT'06*, pages 315–320, 2006.

[26] B. Scheuermann, C. Lochert, J. Rybicki, and M. Mauve. A fundamental scalability criterion for data aggregation in VANETs. In *Proceedings of the 15th annual international conference on Mobile computing and networking - MobiCom '09*, page 285, 2009.

[27] Y.-c. Tseng, S.-Y. Ni, Y.-s. Chen, and J.-P. Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. *Wireless Networks*, 8(2-3):153–167, 2002.

[28] University of Ulm. vanet.info simulation portal. `http://www.vanet.info/?q=node/11`.

[29] L. Wischoff, A. Ebner, H. Rohling, M. Lott, and R. Halfmann. SOTIS - a self-organizing traffic information system. In *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, volume 4, pages 2442 – 2446 vol.4, april 2003.

APPENDIX

A. CRYPTOGRAPHY IN SEDYA

In this appendix, we briefly discuss the cryptographic primitives required to implement our scheme with additional bandwidth efficiency. First we review different types of signatures, followed by a discussion of multisignatures and aggregate signatures.

Signature A digital signature is a cryptographic primitive that guarantees a it was generated using a message M and a private key sk, when verified with the corresponding public key pk. The private key is held only by the signer, while the public key is available to anyone. The signature can then be verified by anyone that has possession of a legitimate copy of the corresponding public key.

Multisignature Given a group of n participants, each with their own key-pair (pk_i, sk_i) a multisignature on a message M can be generated by any subset of participants L by computing this multisignature from the individual signature

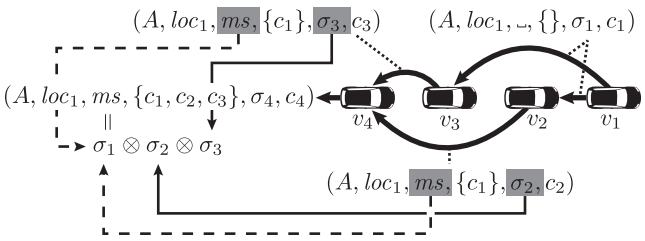


Figure 8: Vehicle v_4 receives multisignatures with overlapping content from v_2 and v_3. Because signatures are only embedded with 1 vehicle delay, v_4 can use the separately received σ_2 and σ_3 together with $ms\,(=\sigma_1)$ to construct a duplicate-free multisignature representing $\sigma_1, \sigma_2, \sigma_3$.

that each participant in L generates on M. Then, the signature can be verified by using all the public keys of the participants in L and computing from them the public key of L.

Aggregate Signature Given a group of n distinct participants and n messages, an aggregate signature is a signature that can be computed from the signatures that are generated by each participant on its own message. The necessary inputs for verification are the n messages, the n public keys and the aggregate signature.

In [14], an identity-based multisignature scheme is developed, is used as a first step to construct an identity-based aggregate signature scheme, which we will also briefly discuss. Although signing is relatively cheap, verification requires three pairing operations and n point additions (for n participants) and it is not possible to work with multiple CAs in the same multisignature. The identity-based multisignature scheme is defined as follows (we refer interested readers to the original work for details):

Setup Generate two groups G_1, G_2 of prime order q and a pairing $e : G_1 \times G_1 \to G_2$, a generator P of G_1, a master secret key $s \in \mathbb{Z}/q\mathbb{Z}$ and master public key sP, and two hash functions H_1, H_2 that both map text to G_1.

Key generation Given identity I, compute the key-pair with public key $H_1(I)$ and secret key $sH_1(I)$.

Sign Choose random $r_i \in \mathbb{Z}/q\mathbb{Z}$ and compute the signature as $(r_i H_2(m) + sH_1(I), r_i P)$.

Aggregation To aggregate signatures, simply perform point addition for all the individual signatures; $(\sum_i S_i, \sum_i T_i)$ for signatures written as (S_i, T_i), given that all signatures are on the same message m.

Verify Given a multisignature (S_n, T_n), verify that $e(S_n, P) = e(T_n, H_2(m))e(sP, \sum_i P_i)$.

Note that a strong requirement for the usage of multisignatures is that the message is the same. The goal of aggregate signatures is to side-step this requirement. However, aggregate signatures are not a viable alternative to multisignatures, at least for SeDyA, because they require all messages to be included. Nevertheless, they can be used to provide compression of signatures in SAS [16], as also noted by the original work. Thus, we briefly show the identity-based aggregate signature scheme that [14] presents:

Setup Generate two groups G_1, G_2 of prime order q and a pairing $e : G_1 \times G_1 \to G_2$, a generator P of G_1, a master secret key $s \in \mathbb{Z}/q\mathbb{Z}$ and master public key sP, two hash functions H_1, H_2 that both map text to G_1, and a hash function H_3 that maps text to $\mathbb{Z}/q\mathbb{Z}$.

Key generation Given identity I, the CA will compute two key-pairs with public keys $P_{i,0} = H_1(I_i, 0)$, $P_{i,1} = H_1(I_i, 1)$ and private keys $sH_1(I, 0)$, $sH_1(I, 1)$ respectively.

Sign Determine a w that has not been used before, which will be the same for each message. It need not be random, only unique to this signing procedure. Compute $H_2(w)$, $c_i = H_3(m_i, I_i, w)$ and generate random $r_i \in \mathbb{Z}/q\mathbb{Z}$; then the signature is $(w, r_i H_2(w) + sP_{i,0} + c_i sP_{i,1}, r_i P)$.

Aggregation To aggregate signatures, simply perform point addition for all the individual signatures; $(w, \sum_i S_i, \sum_i T_i)$ for signatures written as (w, S_i, T_i). It is not allowed to aggregate signatures with different w.

Verify Given a multisignature (w, S_n, T_n), verify that $e(S_n, P) = e(T_n, H_2(w))e(sP, \sum_i P_{i,0} + \sum_i P_{i,1})$.

B. MULTISIGNATURE COMPUTATION

Instead of using just the multisignature as signature for a message, SeDyA includes a multisignature and a regular signature while the message is in the finalization phase. The reason for this is shown in Figure 8, which shows four vehicles with comparable speeds. In this figure, v_1 is the finalizing vehicle: it generates the finalized message A and adds its location, loc_1, to the message, in addition to his signature and certificate. When both v_2 and v_3 receive this message, both of them set the multisignature to the signature of v_1, σ_1 and put the certificate c_1 into the certificate list. Then, both vehicles create their own signature and forward the message in the same format. Now, v_4 receives both messages and computes a new multisignature as $\sigma_1 \otimes \sigma_2 \otimes \sigma_3$. However, if v_2 and v_3 had directly computed their multisignatures as $\sigma_1 \otimes \sigma_2$ and $\sigma_1 \otimes \sigma_3$, then v_4 would have computed $(\sigma_1 \otimes \sigma_2) \oplus (\sigma_1 \otimes \sigma_3)$. To see why this is problematic, note that the certificate set needs to explicitly specify the set of certificates that were included. If a counter is used for this purpose, then an attacker has the opportunity to perform a denial of service attack on the verification process, because this process involves the multiplication of all certificates. Thus, we include the necessary certificates explicitly; repeated certificates therefore lead to additional computational cost.