AN EFFICIENT AUTHENTICATION SCHEME WITH PRIVACY PRESERVING FOR VEHICULAR AD-HOC NETWORKS

Ye Li
Tongji University
The Key Laboratory of Road and Traffic Engineering, Ministry of Education
4800 Cao’an Road, Shanghai, P.R.China 201804
Email: jamesli@tongji.edu.cn

Hui Duan, Corresponding Author
Tongji University
The Key Laboratory of Road and Traffic Engineering, Ministry of Education
4800 Cao’an Road, Shanghai, P.R.China 201804
Tel: +86 13671863463 Email: dellon5961@163.com

Haopeng Deng
Tongji University
The Key Laboratory of Road and Traffic Engineering, Ministry of Education
4800 Cao’an Road, Shanghai, P.R.China 201804
Email: tiankong8949@gmail.com

Word count: 5589 words text + 6 tables/figures x 250 words (each) = 7089 words

August 1, 2015
ABSTRACT

Vehicular ad hoc networks (VANETs) provide opportunities to exchange traffic information among vehicles and infrastructures enhancing the transportation safety and driving experiences. Due to the criticality of exchanged information, message authentication which will not expose the privacy of vehicles is required. The majority of current authentication schemes for VANETs depend primarily on public-key cryptography which brings extra overhead in terms of delay and requires infrastructure support for certificate verification. Symmetric-key based techniques can be more efficient, but they introduce significant key maintenance overheads. In this paper, we propose an efficient and lightweight symmetric-key based authentication scheme for VANETs based on group communication. We analyze the security properties of our proposed schemes to show the applicability when there is little or no infrastructure support. In addition, the proposed scheme was implemented and tested with real-world vehicle data. Simulation results confirmed the efficiency in terms of delay with respect to other proposed techniques.

Keywords: Vehicular ad-hoc networks, Authentication, Privacy Preservation
INTRODUCTION

Vehicular ad-hoc networks (VANETs) have started to receive increasing interest recently due to their potential to be used in traffic and safety applications in the upcoming years (1, 2). In such an ad hoc network, vehicles equipped with a wireless transmission device can send and receive messages at significantly higher speeds compared to traditional mobile ad-hoc networks (3). Vehicles exchange traffic information as they move through the network, which allows drivers to adjust their routes to avoid congestion, obtain road-condition warnings, and be warned in advance for potential traffic accidents.

While the majority of recent research focused on medium access control and routing protocols with the goal of handling the dynamic behavior of VANETs (2, 4), an important aspect that needs to be considered is the security in transmitting messages. Security of VANETs is critical in preventing collisions and thus minimizing the risk for major accidents. Yet the privacy of the driver sending those safety-related messages against unauthorized observers must be guaranteed. However, this anonymity service should be made conditional, meaning it can be revoked for law enforcement purposes whenever necessary. Besides those aforementioned requirements, a secure VANET system should also support availability against various common attacks such as denial-of-service (DoS) attacks and replay attacks.

Most of the prior works on VANET security make exclusive use of public-key cryptography (PKC), requiring that every message be digitally signed and attached with public-key certificates. This incurs significant overhead in terms of both computational cost and bandwidth. In addition, signature verification is a much slower process than signature generation in proposed digital signature schemes, such as ECDSA (5), which makes it even more vulnerable to DoS attacks. Furthermore, the use of PKC may require infrastructure support (e.g. certificate revocation list distribution) which may not always be available everywhere even though VANET deployments are planned in the near future. The concerns mentioned above suggest that symmetric-key cryptography, which is overall much more efficient than public-key techniques, should be used so that VANET's real-time requirements can be met. However, it is infeasible to have every two vehicles share a secret session key due to the huge scale of VANETs. To address this problem, one solution is to use group communication which is motivated by the fact that in a VANET vehicles typically move in groups. The use of groups, together with symmetric-key cryptography, can solve the key distribution problem and improve the efficiency of secure VANETs.

There are many VANET applications that require the formation of groups, particularly for vehicles in geographical proximity. The most prominent example is platooning (6), which groups vehicles in a way that allows them to accelerate or brake simultaneously, thereby avoiding collisions as well as increasing road capacity. VANET groups may also find numerous applications in infotainment in the future, such as multi-player games and chat rooms for vehicle passengers. There are a number of ways to construct groups in VANET applications. The most useful category of groups in terms of functionality is a geodynamic group, where a group leader is elected dynamically, group membership is changed dynamically, and the group boundary also moves dynamically along the road with the vehicles in the group. Geodynamic groups are naturally the best choice for platooning-like applications.

Nonetheless, the design of secure VANETs is further complicated if groups are allowed to form, because security measures must be implemented to ensure only legitimate vehicles can join a group. In particular, the overhead associated with the formation and management of geodynamic groups poses a significant challenge in designing efficient security schemes (i.e. secure group communication). Herein, we propose a lightweight geodynamic group-based authentication scheme for VANETs which can efficiently create geodynamic groups and provide secure
communication among the members of the groups via symmetric-key cryptography. We strive to design a geodynamic group scheme that makes a reasonably good balance among all the different security requirements as well as efficiency and that is readily deployable, yet it can be easily extended to take full advantage of PKC while retaining its efficiency when widespread infrastructure support is available for VANETs. Our scheme is lightweight in the sense that it utilizes symmetric-key cryptography by maintaining a group key for each VANET group. Group keys are created and distributed by group leaders providing efficiency in key distribution and maintenance. Groups are dynamically formed within the transmission range of the leader and managed via a small set of control messages.

LITERATURE REVIEW

Despite the IEEE 1609.2 standard, there has been a significant number of newly proposed authentication schemes over the past few years. The most common objectives of recent authentication schemes were to lower communication overhead, preserve node anonymity, isolation of misbehaving nodes, and non-repudiation. These, however, were not the only objectives stated in the following works. One key note, is that most of the works aimed at providing various security properties which would be appropriate for one particular application or privacy level, but is more than required for another. This is where the main differences lie between the proposed schemes for authentication.

In general, symmetric key algorithms are less computationally intensive than asymmetric algorithms. Though the OBU (On Board Units) are considered powerful enough to make this a moot point in some cases, any protocol which would be intended for utilization in time sensitive safety applications should aim to keep the speed of their protocol as fast as possible as even a few milliseconds can make a big difference in cases such as braking to avoid an accident. Symmetric key algorithms also provide the same level of security with smaller key sizes, thus less storage requirement. However, key exchange becomes an issue when dealing with symmetric key algorithms as a node has to have/establish a common key between any other nodes with whom they wish to communicate. Whereas asymmetric key algorithms do not. Additionally, in asymmetric key algorithms, once a private key is compromised, only the owner of the compromised key is adversely affected. With symmetric algorithms, anyone using the symmetric key is compromised.

Node anonymity, message authentication, non-repudiation, and computational and communication efficiency are some of the very common core concepts identified. Other concerns such as those dealing with RSUs (Road Side Units) in (7), are introduced due to the specifics of the proposed authentication scheme. Although a few schemes (7-9) directly identified attack vectors within their security concerns in VANETs, all works identified and addressed a possible attack vector towards each security issue (e.g. message authentication with an attack vector of node impersonation).

Most of the schemes attempt to address the respectively identified shortcomings of IEEE 1609.2 and/or various other similar works. While a few, such as (10, 11) would still rely on the same proposed cryptographic system (i.e. elliptic curve cryptography) and proposed architecture as in IEEE 1609.2, other schemes take completely different approaches to achieve their goals, such as (9, 12, 13). Some other schemes, such as (14, 15), tried to make improvements by creating a hybrid scheme which only periodically utilizes public-key cryptography.

Nearly every work included an analysis section for their new authentication scheme. Most works, however, did not provide comparisons against other possible solutions or even against IEEE 1609.2 which is a significant drawback. While some did utilize similar test scenarios (e.g. node
densities, communication ranges, map layout) there was no relative consensus on the testbed setup. Additionally, the metrics utilized in assessing the proposed schemes were, in general, not similar across the works. As a result, any direct comparison of efficiency, even though addressed as a concern in several papers, would only allow for very rough estimates.

Several of the reviewed authentication schemes require infrastructure in the form of RSUs and as such will not be complete solutions in areas with little to no infrastructure coverage. A few in fact, such as (10, 16, 17), are so heavily dependent on RSU type infrastructure that they will not work without vast coverage. However, a few schemes, such as (9, 12-14), need very little infrastructure to function and thus provide more flexibility in deployment options.

THE PROPOSED SCHEME
In this section, we propose a Group-based Lightweight Authentication Scheme (GLAS). It makes exclusive use of symmetric-key cryptographic techniques to authenticate messages sent among nodes of a VANET group. It takes into account the lack of infrastructure support for vehicles. Besides message authentication, GLAS guarantees user privacy in face of passive adversaries, yet provides non-repudiation for law enforcement purposes.

System Model and Assumptions
We assume that the on-board unit (OBU) of each vehicle is a tamper-proof device (TPD) (18) which securely stores all cryptographic credentials and performs all cryptographic computations. Given that the underlying cryptographic algorithms are secure, no active adversaries are able to forge a message with a valid MAC by cryptanalytic means. Adversaries can obviously inject falsified and malicious data by other means, for example, by cheating the sensors. This threat can be mitigated through the use of data crosschecking and scoring techniques (19).

Each vehicle has a unique identifier V, such as an electronic license plate (ELP), that relates to other information about the vehicle and its registered owner. Furthermore, each vehicle has a unique pseudonym P\textsubscript{V} whose mapping with the vehicle identifier V is kept solely by the transportation authority. Both V and P\textsubscript{V} are preloaded into the OBU by the transportation authority at the time of vehicle registration.

Furthermore, all OBUs manufactured share two symmetric keys: a join key K\textsubscript{J} used by vehicles joining a VANET group, as well as a law enforcement key K\textsubscript{L}, which is also possessed by the law enforcement agency (not the transportation authority). Since these two keys are universal, their lengths should be sufficiently long, for example, at least 128 bits.

Finally, time synchronization among vehicles is assumed. This can be easily achieved if each OBU is equipped with a GPS receiver.

Group Construction
Non-Member Nodes
Nodes which are not currently associated with any groups will construct a search message S which consists of nothing more than a randomly generated number x and they will periodically broadcast this message as seen in line 2 of Algorithm 1. The actual range of the number is irrelevant so long as there is at least one higher number than another. Although in essence this number could be represented by a single bit, we allocated a few bytes in our experiments to allow greater flexibility should it be needed.

Upon receipt of an S message from a non-member node, a contention will be performed at the receiving node, say V, as detailed in Algorithm 2. That is, the random number y will be extracted from the S message and compared versus V’s random number x. If V’s random number is greater...
than the sender’s, V will promote itself to group leader status and begin broadcasting GL messages, constructed using the join key $K_J$ as seen in lines 8-11 of Algorithm 2. If V’s number is smaller, it will simply continue the procedure of broadcasting S messages and doing contentions. Any node which is already a member of a group or a group leader will simply discard any S message received.

**TABLE 1 Algorithm 1 GLAS Group Construction & Maintenance**

<table>
<thead>
<tr>
<th>Algorithm 1 GLAS Group Construction &amp; Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: if group leader == false &amp;&amp; member == false then</td>
</tr>
<tr>
<td>2: if S received then</td>
</tr>
<tr>
<td>3: node contention</td>
</tr>
<tr>
<td>4: else if GL received then</td>
</tr>
<tr>
<td>5: join group associated with GL message</td>
</tr>
<tr>
<td>6: else</td>
</tr>
<tr>
<td>7: create and broadcast S</td>
</tr>
<tr>
<td>8: end if</td>
</tr>
<tr>
<td>9: else if group leader == true &amp;&amp; member == false then</td>
</tr>
<tr>
<td>10: if GL received then</td>
</tr>
<tr>
<td>11: node contention</td>
</tr>
<tr>
<td>12: else</td>
</tr>
<tr>
<td>13: create and broadcast GL</td>
</tr>
<tr>
<td>14: end if</td>
</tr>
<tr>
<td>15: else if group leader == false &amp;&amp; member == true then</td>
</tr>
<tr>
<td>16: if GL received then</td>
</tr>
<tr>
<td>17: join group associated with GL message</td>
</tr>
<tr>
<td>18: end if</td>
</tr>
<tr>
<td>19: recentTS ← query L for first T</td>
</tr>
<tr>
<td>20: if recentTS &gt; threshold then</td>
</tr>
<tr>
<td>21: revoke all group membership</td>
</tr>
<tr>
<td>22: member ← false</td>
</tr>
<tr>
<td>23: else</td>
</tr>
<tr>
<td>24: unicast a message-MAC pair to all group leaders</td>
</tr>
<tr>
<td>25: end if</td>
</tr>
<tr>
<td>26: end if</td>
</tr>
</tbody>
</table>

Alternatively, we may instead initialize all nodes as group leaders, and this would eliminate the need for an additional, unsecured, message S. As a result, nodes would only alternate between group members and group leaders. This, however, increases the usage of the join key $K_J$ significantly. This was undesirable and thus the overhead incurred for sending our new additional message is seen as a fair trade-off.

**TABLE 2 Algorithm 2 Node Contention**

<table>
<thead>
<tr>
<th>Algorithm 2 Node Contention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Require: (group leader == true &amp;&amp; GL received)</td>
</tr>
<tr>
<td>1: x ← own number to compare against</td>
</tr>
<tr>
<td>2: y ← extracted number from message received</td>
</tr>
<tr>
<td>3: if x &lt; y then</td>
</tr>
</tbody>
</table>
if GL received then
  group leader ← false
  join group associated with GL received
end if

else
  if group leader == false then
    group leader ← true
    generate group key $K_G$ randomly
    begin broadcasting GL messages
  end if
end if

Group Leaders and Group Members

Once a node assumes the role of a group leader, $G$, it will begin periodically broadcasting a GL message (lines 13 in Algorithm 1) which is constructed by encrypting a triple, consisting of the timestamp $T$, the group key $K_G$ and the metadata field $m$, using the join key $K_J$:

$$G \rightarrow *: \{T, K_G, m\}_{K_J}$$

Any vehicle receiving such a GL message is aware of the existence of a group leader, and can thereby join the group by decrypting that message with the join key $K_J$ and verifying the timestamp $T$, finally obtaining the group key $K_G$.

The group key should be chosen randomly by a node when it first becomes a group leader. The metadata field $m$ can be used to store any control data for group management (e.g. message count, neighbor count, or random number), which will be detailed in the next subsection.

Group Maintenance

Group Members

To support multiple group memberships, each member node maintains a group-info list $L$ that stores the information of each group of which it maintains membership. Whenever a node receiving a GL message joins a group, a list element will be constructed consisting of the timestamp $T$, the group key $K_G$, and the group leader's address extracted from the GL message just received. This element will then be inserted at the top of $L$. As such, $L$ becomes sorted by freshness of the timestamps. As shown in Algorithm 1 beginning on line 19, a member node will periodically perform a check on the most recent group information on the top of $L$.

Group information from received GL messages are not blindly added to the list $L$ however. Whenever a GL message is received and verified, an iterative search should first be performed on $L$ to check if the node is already a member of the group associated with that GL message. If so, the information for the group is simply updated by removing the old entry from $L$ and inserting the new information at the top of $L$. Since the maximum allowable number of group memberships (i.e. the capacity of $L$) is typically small and fixed, any iterative search on $L$ can be performed in constant time.

Group Joining

It was identified early on that should a node encounter a group, no matter for how brief of a time, the node would join the group and start sending data to it. This, in essence, was not immediately identified as undesirable as it would allow data to flow more freely throughout the network. However, given our restriction the capacity of the group list $L$ be kept small, say two for example, then data flow is negatively impacted. This occurs as a node will constantly cycle through
membership of the groups which are located around the node. As such, more time is spent
performing the join process than on actually sending data to those groups. To remedy this issue the
most basic solution would be to only allow a node to join another group should there be room
within the node's group listing L. As expired group information is already periodically checked for
and removed from L, it is simply a matter of ensuring L is not full when attempting to insert the
new information into L. For sake of simplicity we utilized this approach in modifying GLAS.

Group Leaders

Once a node becomes a group leader, it will only lower its status when another group leader is
encountered, or the system becomes inactive (e.g. the vehicle is turned off). Upon receiving a rival
group leader's GL message, the receiving node will decrypt and verify the message as all other
nodes do. Once validity is assured, the node will perform a node contention as outlined in
Algorithm 2 lines 1-6, in which the node's control value x (such as message count) stored in the
metadata field m of its own GL messages is compared against the control value y extracted from
the GL message received. Should the receiver have a larger control value it will continue to serve
as the group leader normally would and thus returns to its normal duties. Otherwise it will demote
itself to a member of the sending group leader's group. It does not have to wait for another GL
message as it has already verified the validity of the current GL message and may join the group
with it.

Data Communication and Authentication

Every node should regularly send updated safety messages to other nodes within the same group,
for each group recorded in the group-info list L. The data D of a safety message M would typically
contain information such as the current speed, location and trajectory of the node. A node V can
send a safety message to a particular group as follows:

1. Construct the law enforcement access field (LEAF) by concatenating V's pseudonym
PV with the group key KG obtained from L, padding the result with random bitstrings, and
encrypting it with the law enforcement key KL:

\[
\text{LEAF} = \{r_1 \| PV \| KG \| r_2\} \quad \kappa_L (2)
\]

where \( r_1, r_2 \in_R \{0, 1\}^{32} \).

2. Construct the safety message M, consisting of the data D, the LEAF, and the timestamp
T:

\[
M = <D, \text{LEAF}, T> (3)
\]

3. Compute the HMAC (2I) of the message M using the group key KG, and send the
message-MAC pair via unicast to the group leader, G (unless V is the group leader itself). For this
unicast, V should use a fake source address (i.e. at the link layer) generated randomly using a
pseudo-random number generator:

\[
V \rightarrow G : M, \text{HMAC}_{KG} (M) (4)
\]

4. G would broadcast that message-MAC pair received from V to the entire group. Note
that the group leader's message can reach to every node within the group:

\[
G \rightarrow * : M, \text{HMAC}_{KG} (M) (5)
\]

Any node within the same group receiving such a message-MAC pair would authenticate
the data D by verifying both the timestamp T and the MAC (recomputing it using the group key
KG). The LEAF would be discarded.

SECURITY ANALYSIS

The proposed GLAS scheme meets all the security requirements.
Privacy preservation: GLAS preserves privacy, as no vehicle identifiers are included in any GLAS messages. Furthermore, since the law enforcement key KL used for encrypting the LEAF is possessed by tamper-proof OBUs and the law enforcement agency only, no eavesdroppers are able to decrypt the LEAF to obtain the vehicle pseudonym, which could be used as an alternative label for identifying a particular vehicle.

Efficiency: In GLAS, encryptions are performed in the construction of GL messages and the LEAF. Since public-key cryptography is slower than symmetric-key cryptography by orders of magnitude, GLAS’s exclusive use of symmetric-key cryptography is guaranteed to be much more efficient than the IEEE 1609.2 standard, which proposes the use of ECDSA to digitally sign every safety message. In addition, GLAS consumes less bandwidth by eliminating the attachment of public-key certificates to safety messages.

Non-repudiation: GLAS provide a mechanism that allows law enforcement to track the sender of a safety message. To uncover the identity of the vehicle sending a message-MAC pair that was previously captured, the law enforcement agency can decrypt the LEAF within that message using the law enforcement key KL whose availability will be discussed shortly under separate subsections. As GLAS utilizes a universal law key KL, it is available at the law enforcement agency and thus any LEAF can be easily decrypted.

Denial-of-Service Resistance: In GLAS, a malicious node will be unable to create a group for a legitimate node to join because it is incapable of constructing a valid GL message without the knowledge of the join key KJ. GLAS could be made slightly more efficient, in terms of grouping, by having group leaders respond to any S message received by re-broadcasting the GL message right away. However, this is disallowed in GLAS to prevent Smurf-like DoS attacks.

Replay Resistance: For GLAS, the inclusion of timestamps in GL, GLa, GLb, safety messages and the LEAF prevents replay attacks. Since a safety message goes through a group leader before being forwarded to the recipients, the timestamp of a safety message should sustain a slightly longer lifetime than that of any GL message.

PERFORMANCE EVALUATION
Simulation Setup
In evaluating our algorithm we utilized the ns-2 simulator for simulating a VANET environment. The topography was generated utilizing the mobility modeling software from (22). It was a 2.4 kilometer square area which resembled an urban environment. The vehicles would move, roughly, anywhere from 25 mph to 45 mph depending on the road they were traveling on. The vehicles would also periodically make stops, simulating the need for stopping at appropriate traffic control devices (e.g. stop signs or stop lights). Each run lasted for approximately 5 minutes of simulation time. The results reported are averaged over 30 different runs. For tests where the node count varied, the transmission range was set to 150 meters. For tests where the transmission range varied, the node count was set to 130 nodes.

For GLAS, while the search messages S were broadcast every 900ms, GL messages were sent approximately every 1/3 of a second once a group leader had been established. Safety messages were sent every 300ms once a node joined a group. The threshold for checking the validity of GL message timestamps was set to 100ms. This threshold was chosen to be as it achieved over 99% validation rate even in the most delay imposing scenarios. The other threshold for group information freshness (i.e. scanning the list L) was picked as 700ms. In our experiments, we allowed the member nodes to join up to two different groups which helped to ensure inter-group connectivity.
The join key $K_J$ and the law enforcement key $K_L$ were generated, from fixed values, upon node creation as they are to be loaded into the OBU upon vehicle registration. To establish a baseline for our own cryptographic solutions, we implement a version of GLAP without any encryption or authentication being performed. We denote this implementation as insecure GLAP (iGLAP). We additionally implement a relatively simplistic and base version of an ECDSA scheme, which mimics IEEE 1609.2, for comparison. In addition to ECDSA, we also ran experiments with VAST (23), which was also implemented in ns-2. In the implementation, the actual cryptographic operations were simulated by adding in approximate benchmark delay times on both the sender and receiver. The packet's size was generated to simulate a packet containing a signature and/or MAC. As described in (23), packets are broadcast every 100ms.

**Simulation Results**

**End-to-end Delay**

Figure 1a shows the average end-to-end delay of a typical safety message within our simulation, in seconds, given various node densities. For a better comparison of iGLAS, GLAS and ECDSA, we also include Figure 1b. The transmission range of the radio device on the vehicle is fixed at 150 meters in these experiments.

![Graphs](image_url)

(a) Average Delay vs Node Count  
(b) Average Delay vs Node Count, VAST is not included

**FIGURE 1** Average delay values when varying node count.

We see from Figure 1a that VAST suffers the highest average delay. This is due primarily because, for message $M_i$, a node must wait for the next heartbeat message $M_{i+1}$, which is broadcast approximately every 100ms, before the node can verify $M_i$. From Figure 1b we can see that in all tested scenarios, GLAS has less average delay than the generic implementation of an ECDSA scheme. We also note the efficiency of symmetric-key cryptography in terms of delay as both GLAS and iGLAS produced similar delay results.

After further investigation it was noted a few nodes had extremely high delays, one node in one simulation had an average delay of 343.3ms per message, as compared to the other nodes. As such, a standard deviation was performed on the results which proved a nominal variance in the lower node counts, but a significantly high variance in the upper two node counts. Those two counts had a standard deviation of 4.12ms for the 190 node count, which had an average delay of 7.93ms, and a standard deviation of 8.97ms for the 210 node count plot, which had an average delay of 9.90ms.
Figure 2a and Figure 2b are similar in concept to the previous two figures, Figure 1a and Figure 1b respectively, except instead of varying node densities the transmission range of a node was altered. In these experiments, the number of nodes was fixed to 130 nodes. VAST again has the highest average delay, as seen in Figure 2a, due to the aforementioned need to wait for an additional heartbeat message. From Figure 2b we see the optimum range for minimum average delay with GLAS and iGLAS, would be 50 meters which was measured to be about 3.9ms. The three intersect with an average delay of 4.0ms at around 65-70 meters. Again, all versions of GLAS have a relatively linear growth when the transmission range is varied. GLAS and iGLAS max out average delay around 10.0ms at 300 meters in range. As before the average delay growth depicted in Figure 2 is due to group leaders causing a bottleneck in data flow. We see from Figure 2b the ECDSA algorithm has a relatively constant delay of about 14.0ms. Although the increased number of nodes should increase queuing delay, since more packets are dropped with the increased range they effectively cancel each other and thus the delay stays relatively constant.

![Figure 2a](image1.png) ![Figure 2b](image2.png)

(a) Average Delay vs Transmission Range  (b) Average Delay vs Transmission Range, VAST is not included

**FIGURE 2** Average delay values when varying transmission range.

**Packet Drop Ratio**

Figure 3a shows the drop ratio of packets within the simulation given the various node densities while Figure 3b shows the drop ratio given various transmission ranges. As previously mentioned, the packets checked for dropping were all packets besides routing related packets.

![Figure 3a](image3.png) ![Figure 3b](image4.png)

(a) Average Drop Ratio vs Node Count  (b) Average Drop Ratio vs Transmission Range
FIGURE 3 Packet drop percentages.

Figure 3a shows that all algorithms dropped relatively few packets, that is less than 10%. All versions of GLAS had relatively similar packet drop ratios. This is attributed to the utilization of asymmetric-key cryptography as it takes slightly longer to process the packets than that of symmetric-key cryptography. Additionally, there will be more control packets as an extra message has to be utilized to transfer grouping information between two nodes. Packet drop reasons were primarily collisions with several retry count expirations as well.

From Figure 1 we can see merely increasing node density does not have as dire of an impact on this primarily due to the sparsity of nodes when taking the entire simulation setup into account. However, Figure 2 shows quite a different story. Increasing the transmission range has a drastic impact over the area even with only 130 nodes in the simulation. When thinking of this in the sense of a graph, an increase in transmission range will increase the connectivity of all nodes throughout the entire simulation, while an increase in node count will only increase the connectivity of a few nodes intermittently during the simulation.

Message Throughput

Figure 4a shows the average throughput of message packets within the simulation given the various node densities while Figure 4b shows the average throughput of message packets given various transmission ranges. Throughput does not count any control messages.

FIGURE 4 Average message throughput in kilobits per second (kbps).

From both Figure 4a and Figure 4b we see that VAST and ECDSA have the highest overall throughput. This is due in part as all messages in VAST and ECDSA are considered data messages in regards to our simulation test, as there is no grouping in either. Additionally, all packets in ECDSA contain not only the data, but also the signature on the data as well as the signer's certificate for usage in verifying the signature. VAST is similar and both protocols data packet sizes are over 30 bytes larger. This coupled with the fact that VAST sends messages every 100ms and ECDSA every approximate 1/3 of a second help in showing the somewhat drastic difference in throughput between the protocols.

CONCLUSION
In this paper, an efficient and group-based lightweight authentication scheme, GLAS, was proposed by exploiting the group-based behavior of vehicles in traffic. The scheme maintains the privacy of an individual while allowing for non-repudiation of a message by law enforcement if so needed. Additionally, it allows for a high degree of DoS and replay resistance. As the use of symmetric-key cryptography is proposed in GLAS, we have also identified several ways to maintain key freshness, a desirable property. The performance of GLAS have been tested under ns-2 using real-life traffic data.

Extensive simulation was conducted which showed that it prove to be a fast and efficient scheme with all message delays being less than 14ms within our simulation environment and a low drop ratio for the majority of scenarios tested. The results also have shown that for the optimal performance in terms of both delay and drop ratio, smaller groups (i.e. smaller transmission ranges such as 50m) should be preferred. Compared to ECDSA and VAST, GLAS can significantly reduce the end-to-end delay for packets while providing the desired authentication with privacy and non-repudiation.

REFERENCES

11. A. Studer, M. Luk and A. Perrig. Efficient Mechanisms to Provide Convoy Member and Vehicle Sequence Authentication in VANETs. Proceedings of the


