Secure and Efficient Trust Opinion Aggregation for Vehicular Ad-hoc Networks

Chen Chen Jie Zhang[†] Robin Cohen Pin-Han Ho[‡]
David R. Cheriton School of Computer Science, University of Waterloo, Canada
[†]School of Computer Engineering, Nanyang Technological University, Singapore
[‡] Department of Electrical and Computer Engineering, University of Waterloo, Canada
{c32chen}@uwaterloo.ca

***Abstract*—In this paper, we propose a trust opinion aggregation scheme in vehicular ad-hoc networks, to support trust models used to evaluate the quality of information shared among peers in the environment. Our scheme extends an existing identity-based aggregate signature algorithm to correctly combine signatures for multiple messages into one aggregate signature and eliminate signature redundancy. As a result, our proposed scheme is secure and archives both space efficiency and time efficiency, as confirmed by our comparative analysis.**

I. Introduction

Researchers working on road condition systems in the application layer of vehicular ad-hoc networks (VANETs) are faced with the challenge of possibly malicious information shared by peers in the environment. Different trust models [1], [2] have recently been proposed to evaluate the quality of information by modeling the trustworthiness of information senders. One particularly interesting setting for trust modeling is where peers are allowed to share their trust opinions about the information sent by information senders and the aggregation of these trust opinions will be used to evaluate the quality of the information.

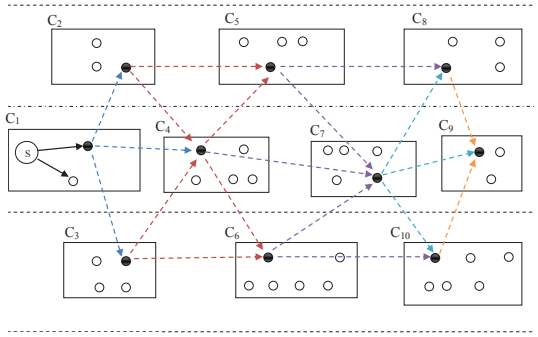

Fig. 1. Trust Opinion Aggregation

Figure 1 illustrates an example of trust opinion aggregation. The topology of aggregation is based on a cluster-based message routing mechanism. For each cluster, a vehicle is randomly chosen from all cluster members (the white nodes) as the cluster leader (the black nodes). Trust opinions[1] are aggregated when message M from sender S (shown as the circle in cluster C_1 in Figure 1) propagates to cluster C_7. Let S_i denote the set of opinions from peers in cluster C_i, i.e. $S_i = \{O_k |$ peer p_k generates trust opinion O_k, and $p_k \in C_i\}$. The cluster leader of C_1 receives the message M and a set of trust opinions S_1, combines them into an aggregate message $A = [M, S_1]$, and then sends A to the next hop clusters. For the cluster leader of C_i, $i \in [2, 7]$, it receives several aggregate messages from previous hops, combines them into a new aggregate, and broadcasts the aggregate to cluster members. It also collects trust opinions, combines them into a newer aggregate, and relays this aggregate to the next hop clusters.

Two important factors, security and efficiency, should be considered for trust opinion aggregation in VANETs. Due to possible attacks (i.e. data repudiation and sybil attacks [3]) during the aggregation of trust opinions, it is required that data is signed before sent so that 1) data repudiation can be easily detected by verifying the data against sender's signature; 2) a message sender or trust opinion provider cannot deny its message because of the existence of its signature; 3) identities cannot be forged or abused because each message is mapped to a valid and unique identity – the only peer that is able to sign the message. Efficiency in trust opinion aggregation consists of two aspects: 1) time efficiency, the time needed to perform an aggregation; 2) space efficiency, the size needed to aggregate all trust opinions and signatures.

Our proposed aggregation scheme is based on an existing identity-based aggregate signature algorithm [4] but introduces two important improvements. It can combine signatures for multiple messages (not just a single message) and it copes with signature redundancy by merging these into the existing signature which remains valid and verifiable. As will be demonstrated, our proposed scheme is secure and improves both space and time efficiency, with the one merged signature remaining of constant size and messages being aggregated without relying on an aggregation chain.

II. Related Work

Current message aggregation schemes in VANETs usually require a digital signature on each message being aggregated so that neither can a message be maliciously repudiated nor can an attacker disseminate bogus information without being traced. Raya et al. [5] illustrate a secure aggregation scheme, which aggregates messages and signatures from different parties signing the message. The model features in three types of

[1]A trust opinion is a message provided by a peer that serves as an evaluation of the sender message, i.e. a pair $(\{trust, \neg trust\}, confidence \in [0, 1])$.

978-1-4244-3574-6/10/$25.00 ©2010 IEEE

aggregate signature schemes, as shown in Figure 2.

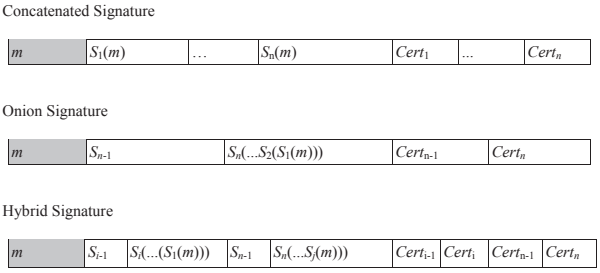

Fig. 2. Concatenated, Onion, and Hybrid Signature

For concatenate signature, a vehicle signs a message m and rebroadcasts the new message to the aggregator that collects all signed messages from all signers. Signatures $S_i(m)$ are concatenated and appended to m so that all signatures are independent from each other. However, the verification of an aggregate takes up n times of signature verification. The size of an aggregate also grows linearly with the number of signatures and certificates, which is space costly. An onion signature borrows the idea of onion routing. A message is over-signed during its propagation, i.e. a signer signs the signed message from its predecessor and forwards the new message to its successor. The n-th node verifies the signature S_{n-1} before over-signing S_{n-1}. This protects data validity and requires at most two signatures, reduces the communication overhead by concatenating signatures, and yet increases the computation overhead because of the inherent property of onion routing. At the same time, it opens the vulnerability window by allowing two colluding attackers, i.e. the n-th and $(n-1)$-th nodes, to modify messages without being detected. A hybrid signature concatenates several onion signatures, each at a given depth, which strengthens the security of signatures by making collusion harder to succeed. However, there still exists a possibility that signatures can be compromised.

Also, onion signature and hybrid signature are based on a strong assumption that signature aggregation is done in a sequential order. An ordered signing sequence would reflect a chained trust relationship among peers, which may not be available or appreciated in VANETs.

Moreover, all these aggregate signatures make use of asymmetric cryptography and rely on public key infrastructure (PKI). In other words, signature verification requires additional information – the signer's public keys and certificates (e.g. $Cert_i$ in Figure 2). Such information, which is usually in a larger size compared to the message itself, needs to be carried along with the signature because it is not preferable to assume that any verifier would have already kept all public keys and certificates in its local repository. As a result, aggregation achieves high security with a trade-off of great space overhead, rendering it inefficient and reversely degrading system performance.

An identity-based aggregate signature scheme [4] has been proposed in recent years and improves the above signature schemes in three aspects. First, multiple signatures are compacted into one single signature while the message can still be verified. Second, instead of storing/carrying public keys and certificates, the verifier only needs to obtain a list of identities of signers, which are much smaller than public keys and certificates. Third, no matter how messages are aggregated, the scheme does not rely a chained trust relationship among peers and thus the final aggregate signature remains unique. With overall information required for verification minimized, high efficiency is achieved without compromising security. However, such a scheme requires n different signatures from n distinct signers. In other words, when it comes to merging two duplicate signed messages from the same signer, which is often true under the trust opinion aggregation strategy, aggregation would fail because the aggregated signature cannot pass verification.

Based on [4], Zhu et al. [6], [7] introduce an aggregated emergency message authentication scheme for vehicular networks, where a vehicle randomly generates a pseudonym as its identity each time a new emergency message is signed. Signed messages can be aggregated and verified even if there are duplicate ones. However, their work cannot trace the signer given the signed message. This leaves ample space for an attacker that signs arbitrary fake messages under different pseudo-identities. A trust modeling method supported by our scheme may be effective to deal with this problem by diminishing the spread of fake messages from untrustworthy peers. A batch message verification method [8] compacts multiple signatures on different messages into one signature. The identity of the signer can be traced by the central authority (CA), but the system relies on a tamper-proof device which might be either hacked or working improperly.

III. Our Trust Opinion Aggregation Scheme

In this section, we illustrate our identity-based aggregation scheme that extends the identity-based aggregate signature algorithm [4]. To start with, we explain the basic concept of identity-based signature and aggregate signature, and introduce the "Bilinear Maps" which serves as the mathematical foundation of the aggregate signature scheme.

A. Preliminaries

An identity-based signature is a signature scheme where the user's identity is used to generate its public key. The identity is a short binary string constructed from what uniquely identifies a user (i.e. an email address, a social insurance number, etc). In a VANET setting, we can assume each car has a unique ID issued by the central authority. Given an ID of a vehicle, the CA computes the unique private key and securely distributes it to the corresponding vehicle. An aggregate signature is a single short binary string which convinces the verifier that for $1 \leq i \leq n$, signer S_i signed the message M_i where the n signers and n messages can be distinct and independent from each other. Instead of keeping n distinct signatures as traditional signature schemes do, the aggregate signature is one single signature that compacts n signatures.

Let G be a cyclic additive group generated by a generator P, and G_T be a cyclic multiplicative group. G and G_T have the

same prime order q, i.e., $|G| = |G_T| = q$. Let $\hat{e} : G \times G \rightarrow G_T$ be a bilinear map, which satisfies the following properties:

- Bilinear: for all $P, Q, R \in G$, and $a, b \in Z_q$, $\hat{e}(Q, P+R) = \hat{e}(P+R, Q) = \hat{e}(P, Q) \cdot \hat{e}(R, Q)$. In particular, $\hat{e}(aP, bP) = \hat{e}(P, bP)^a = \hat{e}(aP, P)^b = \hat{e}(P, P)^{ab}$.
- Non-degenerate: $\exists P, Q \in G$ such that $\hat{e}(P, Q) \neq 1_{G_T}$.
- Computable: there is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in G$.

Such a bilinear map $\hat{e}$, which is often called an *admissible pairing*, can be constructed by the modified Weil pairings on elliptic curves [9]. The groups (G and G_T) on such a map are called bilinear groups, where Decisional Diffie-Hellman problem is easy to resolve but Computational Diffie-Hellman problem is believed to be hard [9].

B. System Setup

The central authority (CA) generates the system parameters:

1) generates groups G and G_T of order q, an admissible pairing $\hat{e} : G \times G \rightarrow G_T$;
2) chooses an arbitrary generator $P \in G$;
3) chooses a random $s \in Z/qZ$, and computes $Q = sP$;
4) chooses three hash functions $H_1, H_2 : \{0,1\}^* \rightarrow G$ and $H_3 : \{0,1\}^* \rightarrow Z/qZ$.

The system parameters $\{G, G_T, \hat{e}, P, Q, H_1, H_2, H_3\}$ are made public. The secret kept by CA is $s \in Z/qZ$.

Next, CA generates the public key K_i and private key k_i for each vehicle V_i. Given a vehicle V_i with ID_i, the CA generates two pairs of keys $(K_{i,0}, k_{i,0})$ and $(K_{i,1}, k_{i,1})$, where $K_{i,j} = H_1(\mathrm{ID}_i, j), k_{i,j} = sK_{i,j}, j \in \{0,1\}$.

Let $K_i = K_{i,0}||K_{i,1}$ and $k_i = k_{i,0}||k_{i,1}$, where $||$ denotes concatenation. Note that both K_i and k_i are unique for each vehicle v_i. Given an ID_i any third party can compute K_i but k_i can only be generated by the CA.

C. Message and Trust Opinion Signing

There are two types of identities that sign messages: 1) the sender, that is the message originator; and 2) the evaluator, that gives a trust opinion. We explain how they sign their messages in the following two cases.

Case 1: a vehicle V_0 is the message originator, and signs its message M as follows:

1) computes $P_M = H_2(M) \in G$;
2) computes $h_0 = H_3(M, \mathrm{ID}_0) \in Z/qZ$;
3) generates a random $r_0 \in Z/qZ$ and initiates $c_0 = 1 \in Z$;
4) computes $S_0 = r_0 P_M + c_0 k_{0,0} + c_0 h_0 k_{0,1}$, $T_0 = r_0 P$;
5) generates signed message $M_0 = [M, c_0, \mathrm{ID}_0, S_0, T_0]$.

Case 2: a vehicle V_i evaluates the message M, generates its trust opinion O_i, and signs M and O_i as follow:

1) computes $P_M = H_2(M) \in G$;
2) computes $h_i = H_3(M, O_i, \mathrm{ID}_i) \in Z/qZ$;
3) generates a random $r_i \in Z/qZ$ and initiates $c_i = 1 \in Z$;
4) computes $S_i = r_i P_M + c_i k_{i,0} + c_i h_i k_{i,1}$, $T_i = r_i P$;
5) generates signed message: $M_i = [M, O_i, c_i, \mathrm{ID}_i, S_i, T_i]$.

The computation of S_i and T_i is implemented over the group G. The combination of both S_i and T_i serves as the signature for the message M (Case 1) or M and O_i (Case 2).

D. Trust Opinion Aggregation

We explain how our scheme aggregates trust opinions under two cases with and without signature redundancy respectively.

Case 1: a third party V' combines the original signed message with n signed trust opinions from n distinct nodes. Specifically, we have $V_0 : M_0 = [M, c_0, \mathrm{ID}_0, S_0, T_0]$ and $V_i : M_i = [M, O_i, c_i, \mathrm{ID}_i, S_i, T_i]$ for $i \in [1, n]$. V' computes

$$S' = \sum_{i=0}^{n} S_i \text{ and } T' = \sum_{i=0}^{n} T_i \tag{1}$$

and generates the aggregate

$$A' = [M, O_1, \ldots, O_n, c_0, \ldots, c_n, \mathrm{ID}_0, \ldots, \mathrm{ID}_n, S', T'] \tag{2}$$

The summation of S_i and T_i is implemented over group G.

Case 2: a third party V' wants to combine two aggregates on the same message M into a larger aggregate, and there may exist duplicate trust opinions. Note that instead of merging aggregates on different messages, our system only combines multiple aggregates for the same message M. Not to lose generality, we assume that V' receives the following aggregates:

$$A_1 = \begin{bmatrix} M, O_1, .., O_p, O_{p+1}, .., O_{p+k}, \\ c_{0,1}, c_{1,1}, .., c_{p,1}, c_{p+1,1}, .., c_{p+k,1}, \\ \mathrm{ID}_0, \mathrm{ID}_1, .., \mathrm{ID}_p, \mathrm{ID}_{p+1}, .., \mathrm{ID}_{p+k}, \\ S_1, T_1 \end{bmatrix} \tag{3}$$

$$A_2 = \begin{bmatrix} M, O_{p+1}, .., O_{p+k}, O_{p+k+1}, .., O_{p+k+q}, \\ c_{0,2}, c_{p+1,2}, .., c_{p+k,2}, c_{p+k+1,2}, .., c_{p+k+q,2}, \\ \mathrm{ID}_0, \mathrm{ID}_{p+1}, .., \mathrm{ID}_{p+k}, \mathrm{ID}_{p+k+1}, .., \mathrm{ID}_{p+k+q}, \\ S_2, T_2 \end{bmatrix} \tag{4}$$

where A_1 and A_2 share k duplicate trust opinions, i.e. O_{p+i} for $i \in [1, k]$. Note that for $i \in \{0\} \cup [p+1, p+k]$, ID_i may have different c_i values in A_1 and A_2 due to various paths of aggregation, so we denote them as $c_{i,1}$ and $c_{i,2}$.

V' computes $S' = S_1 + S_2$, $T' = T_1 + T_2$,

$$c'_i = \begin{cases} c_{i,1} & \text{for } i \in [1, p] \\ c_{i,1} + c_{i,2} & \text{for } i \in \{0\} \cup [p+1, p+k] \\ c_{i,2} & \text{for } i \in [p+k+1, p+k+q] \end{cases} \tag{5}$$

and generates the new aggregate

$$A' = \begin{bmatrix} M, O_1, .., O_p, .., O_{p+k}, .., O_{p+k+q}, \\ c'_0, c'_1, .., c'_p, .., c'_{p+k}, .., c'_{p+k+q}, \\ ID_0, \mathrm{ID}_1, .., \mathrm{ID}_p, .., \mathrm{ID}_{p+k}, .., \mathrm{ID}_{p+k+q}, \\ S', T' \end{bmatrix} \tag{6}$$

We give a simple example showing how aggregation on two aggregates works. Suppose we have four vehicles V_0, V_1, V_2, and V_3 where V_0 is the original message sender, and two existing aggregates $A_1 = [M, O_1, O_2, \mathrm{ID}_0, \mathrm{ID}_1, \mathrm{ID}_2, c_{0,1}, c_{1,1}, c_{2,1}, S_1, T_1]$ that combines messages from V_0, V_1, V_2, and $A_2 = [M, O_2, O_3, \mathrm{ID}_0, \mathrm{ID}_2, \mathrm{ID}_3, c_{0,2}, c_{2,2}, c_{3,2}, S_2, T_2]$ that combines messages from V_0, V_2, V_3. Both aggregates share a common trust opinion O_2 that is duplicate. An aggregator generates the new aggregates A' as

$$A' = [M, O_1, O_2, O_3, \mathrm{ID}_0, \mathrm{ID}_1, \mathrm{ID}_2, \mathrm{ID}_3, c'_0, c'_1, c'_2, c'_3, S', T']$$

where $c'_0 = c_{0,1} + c_{0,2}$, $c'_1 = c_{1,1}$, $c'_2 = c_{2,1} + c_{2,2}$, $c'_3 = c_{3,2}$ and $S' = S_1 + S_2$, $T' = T_1 + T_2$.

E. Signature Verification

We verify our signature scheme under three cases, a signed message with and without trust opinions, and a merged aggregate from two aggregates respectively.

Case 1: verify a signed message without trust opinions yet. Given a signed message $M_0 = [M, c_0, \mathrm{ID}_0, S_0, T_0]$, the verifier checks if

$$\hat{e}(S_0, P) = \hat{e}(T_0, P_M)\hat{e}(Q, c_0 K_{0,0} + c_0 h_0 K_{0,1}) \quad (7)$$

holds, where P, Q are system parameters, $P_M = H_2(M)$, $K_{0,j} = H_1(\mathrm{ID}_0, j)$ for $j \in \{0, 1\}$ and $h_0 = H_3(M, \mathrm{ID}_0)$. The verification is as follows:

$$\begin{aligned}
&\hat{e}(S_0, P) = \hat{e}(r_0 P_M + c_0 k_{0,0} + c_0 h_0 k_{0,1}, P) \\
&= \hat{e}(r_0 P_M, P)\hat{e}(c_0 k_{0,0} + c_0 h_0 k_{0,1}, P) \\
&= \hat{e}(P_M, r_0 P)\hat{e}(c_0 s K_{0,0} + c_0 h_0 s K_{0,1}, P) \\
&= \hat{e}(P_M, r_0 P)\hat{e}(c_0 K_{0,0} + c_0 h_0 K_{0,1}, sP) \\
&= \hat{e}(P_M, T_0)\hat{e}(c_0 K_{0,0} + c_0 h_0 K_{0,1}, Q) \\
&= \hat{e}(T_0, P_M)\hat{e}(Q, c_0 K_{0,0} + c_0 h_0 K_{0,1})
\end{aligned}$$

Case 2: verify an aggregate which contains the message M and trust opinions from n distinct nodes. Given an aggregate

$$A = [M, O_1, .., O_n, c_0, .., c_n, \mathrm{ID}_0, .., \mathrm{ID}_n, S_{n+1}, T_{n+1}] \quad (8)$$

where M is signed by vehicle ID_0 and (M, O_i) is signed by ID_i, for $1 \le i \le n$, the verifier checks if the following holds:

$$\hat{e}(S_{n+1}, P) = \hat{e}(T_{n+1}, P_M)\hat{e}(Q, \sum_{i=0}^{n} c_i K_{i,0} + \sum_{i=0}^{n} c_i h_i K_{i,1}) \quad (9)$$

where $P_M = H_2(M)$, $K_{i,j} = H_1(\mathrm{ID}_i, j)$, for $i \in [0, n], j \in \{0, 1\}$, and $h_0 = H_3(M, \mathrm{ID}_0), h_i = H_3(M, O_i, \mathrm{ID}_i)$, for $i \in [1, n]$. The verification is as follows:

$$\begin{aligned}
&\hat{e}(S_{n+1}, P) = \hat{e}(\sum_{i=0}^{n} S_i, P) \\
&= \hat{e}(\sum_{i=0}^{n} r_i P_M + \sum_{i=0}^{n} c_i k_{i,0} + \sum_{i=0}^{n} c_i h_i k_{i,1}, P) \\
&= \hat{e}(\sum_{i=0}^{n} r_i P_M, P)\hat{e}(\sum_{i=0}^{n} c_i k_{i,0} + \sum_{i=0}^{n} c_i h_i k_{i,1}, P) \\
&= \hat{e}(P_M, \sum_{i=0}^{n} r_i P)\hat{e}(\sum_{i=0}^{n} c_i s K_{i,0} + \sum_{i=0}^{n} c_i h_i s K_{i,1}, P) \\
&= \hat{e}(P_M, \sum_{i=0}^{n} r_i P)\hat{e}(\sum_{i=0}^{n} c_i K_{i,0} + \sum_{i=0}^{n} c_i h_i K_{i,1}, sP) \\
&= \hat{e}(P_M, T_{n+1})\hat{e}(\sum_{i=0}^{n} c_i K_{i,0} + \sum_{i=0}^{n} c_i h_i K_{i,1}, Q) \\
&= \hat{e}(T_{n+1}, P_M)\hat{e}(Q, \sum_{i=0}^{n} c_i K_{i,0} + \sum_{i=0}^{n} c_i h_i K_{i,1})
\end{aligned}$$

Case 3: the verification of a merged aggregate A' from aggregate A_1 and A_2, as shown in Section III-D.

$$\begin{aligned}
&\hat{e}(S', P) = \hat{e}(S_1 + S_2, P) = \hat{e}(S_1, P)\hat{e}(S_2, P) \\
&= \hat{e}(T_1, P_M) \\
&\hat{e}(Q, (c_{0,1} K_{0,0} + c_{0,1} h_0 K_{0,1}) + \sum_{i=1}^{p+k} (c_{i,1} K_{i,0} + c_{i,1} h_i K_{i,1})) \\
&\hat{e}(T_2, P_M) \\
&\hat{e}(Q, (c_{0,2} K_{0,0} + c_{0,2} h_0 K_{0,1}) + \sum_{i=p+1}^{p+k+q} (c_{i,2} K_{i,0} + c_{i,2} h_i K_{i,1})) \\
&= \hat{e}(T_1 + T_2, P_M)\hat{e}(Q, \sum_{i=0}^{p+k+q} (c'_i K_{i,0} + c'_i h_i K_{i,1})) \\
&= \hat{e}(T', P_M)\hat{e}(Q, \sum_{i=0}^{p+k+q} (c'_i K_{i,0} + c'_i h_i K_{i,1}))
\end{aligned}$$

IV. Analysis of Trust Opinion Aggregation

We compare the security level, space efficiency, and time efficiency of our identity-based aggregation (ID-based) with a multiple of existing secure aggregation methods: concatenate-signature-based (Conc.), onion-signature-based (Onion), and hybrid-signature-based (Hybrid) discussed in Section II.

A. Security Level

The security level of concatenate signature and identity-based signature is high because in both cases signatures cannot be forged nor can trust opinions be repudiated without being detected. Onion signature has a low level of security because only two signatures are kept for verification – in case of two colluding peers, the data can be arbitrarily generated or modified without being detected. The hybrid signature strikes a balance between the concatenation signature and onion signature by placing multiple pairs of signatures in the message. So we assign a medium security level to hybrid signatures.

B. Space Efficiency

The sender message "event, confidence, location, time" cost 21 bytes, if we assign 8 bytes to "event", 1 byte to "confidence", 8 bytes to "location", and 4 bytes to "time". Each trust opinion is assigned one byte (1 bit for reaction and 7 bit for confidence), and each vehicle identity in our region uses 3 bytes ($2^{24} \approx 1.6$ million vehicles). Given a message with $n-1$ trust opinions, the space cost is $21 + (n-1) + 3n = 4n + 20$ bytes, plus the cost of signatures, as shown below.

For non identity-based aggregation methods, if we apply the ECDSA signature scheme [9], which is adopted by IEEE 1609.2 standard [10], the signature size is 42 bytes. At the same time, one certificate must be transmitted along with one signature. If we use the certificate presented in IEEE 1609.2 standard, the certificate costs 125 bytes. For concatenate-signature-based aggregation, each vehicle identity would cost $42 + 125 = 167$ bytes for its signature plus certificate, so the size of aggregate message will be $4n + 20 + 167n = 171n + 20$

bytes, since each identity occupies 167 bytes. Since onion-signature-based aggregation only keeps two signatures, the size of aggregate message is $4n + 20 + 167 \times 2 = 4n + 354$ bytes. As for hybrid-signature-based aggregation, we define a degree $k \in [1, \lfloor n/2 \rfloor]$ to denote how many pairs of signatures are kept in the message. The aggregate size falls into $4n + 20 + 167 \times 2k = 4n + 334k + 20$ bytes.

For identity-based aggregation, if we want to achieve the same security level of ECDSA, the signature will be 42 bytes (21 bytes for S and another 21 bytes for T). In order to maintain the list of c_i, $i \in [1, n]$ for n identities, $2n$ bytes are needed. However, since the value of c_i is usually small, less than $2n$ bytes might be consumed in practice if a compaction algorithm such as variable-length encoding is deployed. The ultimate size of aggregate message falls in an interval between $4n + 62$ and $6n + 62$ bytes.

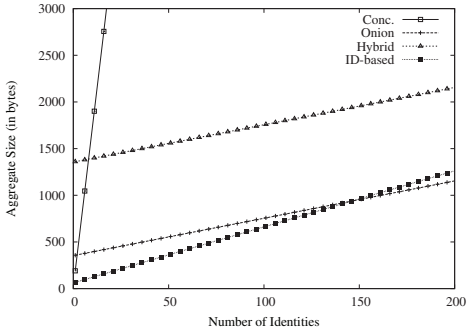

Fig. 3. Space Efficiency

We compare the aggregate size among these aggregation methods in Figure 3. The degree k for hybrid signature is set to 4. Our identity-based aggregation method achieves the best space efficiency when the number of identities is smaller than 146, although it is later surpassed by the onion-signature-based aggregation method, which, in practice, should not be employed due to its inherent disadvantage of low security and high aggregation delay.

C. Time Efficiency

Let T denote the average time cost for a peer to verify one single signature and generate a trust opinion.[2] The concatenate-signature-based and identity-based aggregation schemes take $2T$ to generate an aggregate message with n trust opinions from n peers – for the first T the message is broadcast to each peer that verifies the message, signs and attaches its own opinion; for the second T these opinions are sent back to the aggregator that verifies and combines these opinions with its own trust opinion into the new aggregate message. For the onion-signature and hybrid-signature-based aggregation, it requires an aggregation sequence of n peers where messages are verified, attached an opinion and signed from the head peer to the tail peer, and thus the total cost is nT.

[2]In practice, the verification time T_v is less than 500ms with 200 identities, according to Zhang et. al [8].

A summary of the comparison between the four aggregation methods is presented in Table I.

TABLE I
COMPARISON OF SECURE AGGREGATION METHODS

	Security Level	Size (in bytes)	Delay
Conc.	high	$171n + 20$	$2T$
Onion	low	$4n + 354$	nT
Hybrid	medium	$4n + 334k + 20$	nT
ID-based	high	$4n + 62 \sim 6n + 62$	$2T$

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a secure and efficient trust opinion aggregation scheme. First and foremost, high security is reinforced throughout the aggregation process. Second, our aggregation method achieves high time efficiency since multiple signatures can be aggregated in one pass by summing them up mathematically. High space efficiency is made possible in that signatures are compacted into one aggregate signature, with additional information for verification minimized and redundancy eliminated. Third, our aggregation is flexible because there is no negative effect and little difference on whomever and whenever to perform the aggregation. These desirable properties of our extended scheme are verified through detailed analysis, and become the strong support for building trust modeling methods to effectively evaluate quality of information shared among peers in VANET environment.

For future work, we will look into a variable-length encoding algorithm to further compact the size of an aggregated message, and evaluate its improvement in space efficiency.

REFERENCES

[1] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen, "Towards expanded trust management for agents in vehicular ad-hoc networks," *International Journal of Computational Intelligence (IJCI), accepted*, 2009.

[2] C. Chen, J. Zhang, R. Cohen, and P.-H. Ho, "A trust-based message propagation and evaluation framework in vanets," in *Proc. of IFIP Trust Management*, 2010.

[3] J. R. Douceur, "The sybil attack," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. London, UK: Springer-Verlag, 2002, pp. 251–260.

[4] C. Gentry and Z. Ramzan, "Identity-based aggregate signatures," in *Public Key Cryptography*, 2006, pp. 257 – 273.

[5] M. Raya, A. Aziz, and J.-P. Hubaux, "Efficient secure aggregation in VANETs," in *Proc. of VANET*, 2006.

[6] H. Zhu, X. Lin, R. Lu, P.-H. Ho, and X. Shen, "AEMA: An aggregated emergency message authentication scheme for enhancing the security of vehicular ad hoc networks," in *Proc. of ICC*, 2008.

[7] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. IEEE INFOCOM*, 2008.

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

[9] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. of ASIACRYPT*, 2001.

[10] I. S. 1609.2, "IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages," *IEEE Std 1609.2*, pp. 0–1–105, 2006.