Secure and Privacy-Preserving, Timed Vehicular Communication

Mike Burmester, Emmanouil Magkos, and Vassilis Chrissikopoulos

***Abstract*—In this paper we consider the problem of privacy and security in vehicular (V2V) communication, in particular securing *routine safety messages*. Traditional public key mechanisms are not appropriate for such applications because of the large number of safety messages that have to be transmitted by each vehicle, typically one message every $100-300\,ms$. We first show that a recently proposed V2V communication scheme, TSVC, based on the Time Efficient Stream Loss-tolerant Authentication (TESLA) scheme is subject to an impersonation attack in which the adversary can distribute misleading safety information to vehicles, and propose a modification that secures it against such attacks. We then address general concerns regarding the inappropriateness of TESLA for vehicular applications (caused by the delayed authentication, and buffer overflow issues), and propose a V2V communication scheme based on a variant of TESLA, TESLA^0, in which packets are self-authenticating. This scheme is appropriate for applications in which vehicles are in close proximity. Finally we consider a hybrid protocol that combines both schemes and addresses in a more flexible way the mobility requirements of V2V communications.**

***Index Terms*—Vehicle to vehicle communication, security, privacy, TESLA, timed hash chains.**

I. Introduction

VEHICULAR ad-hoc networks (VANETs) are emerging as one of the most interesting instantiations of mobile ad-hoc networks, aiming at enhancing road safety and transportation efficiency. In a VANET, vehicles equipped with short-range wireless capabilities are able to communicate with each other in an ad-hoc fashion (Vehicle-to-Vehicle, V2V) and with the road infrastructure (Vehicle-to-Infrastructure, V2I), forming a *mesh* network of nodes [1]. A number of automotive safety and convenience-related VANET applications are expected to be deployed in the near future [2], while several proof-of-concept implementations are already in place (*e.g.*, [3], [4]), and the technology is being standardized [5], [6].

The security of vehicular communications has received much attention in the literature (*e.g.*, [7], [8], [9], [10], [11]). On one hand, messages in a VANET should be properly authenticated to prevent internal or external adversaries that replay, modify or fabricate messages. In addition, proper identification may be necessary in order to authorize access to services ($e.g.$ for access control, billing purposes etc), provide personalized, context-aware content, or trace back an identity for accountability / liability purposes ($e.g.$, credential revocation, when investigating an accident). On the other hand, VANET communication is often required to be anonymous (*i.e.*, unlinkable and untraceable), to preserve user privacy.

The *privacy vs. authentication* tradeoff has been an important research area for VANETs [9], [12], [13], [8], [11], [14], [15], [16], [17], [18], [19], [20]. Recently, there has also been a discussion concerning the benefits of using public-key cryptography in VANETs and whether the requirement for infrastructure-based authentication can be relaxed [12], [21], [17], [22], [19], [20]. For example, non-emergency communication such as routine safety messages that are sent by vehicles every $100-300\,ms$, can be based on strict time constraints ([17], [19], [20]). To this end, a number of hybrid solutions that combine asymmetric with lightweight (symmetric) cryptographic primitives for message authentication (*e.g.*, [12], [21], [11], [19], [20]) or confidentiality (*e.g.*, [17], [16], [18]), have been proposed.

In a recent scheme, the *TSVC scheme* [20], privacy is preserved by using a list of uncorrelated, short-lived pseudonyms, where each pseudonym has the form of a public key for verifying digital signatures, certified by a trusted entity. Security in [20] is based on the cryptographic hash chain primitive [23] and the TESLA broadcast authentication protocol [24]. Specifically, each public key authenticates a cryptographic hash chain, where elements of the chain are released after a predefined delay and are used by neighbouring receivers as the MAC keys to authenticate a series of subsequent safety routine messages. The anonymity is conditional, in the sense that the pseudonyms bear information that allows tracing a real-world identity, if needed. Compared with currently available public-key based schemes, the hash chain primitive is very efficient for non-emergency communication, since it only requires computing hash values. Furthermore, communication complexity is reduced as a single message authentication code (MAC) is attached to each data package.

Our contribution: We show that the TSVC message authentication scheme is subject to an impersonation attack, in which the adversary distributes misleading safety information to neighboring vehicles. We then show how to fix this scheme and address general concerns regarding delayed authentication and buffer overflow. Finally we propose a variant of TESLA in which messages are self-authenticating and show how to combine the two schemes so as to address in a flexible way the mobility requirements of V2V communications.

II. Timed Secure Vehicular Communication

A. The TESLA authentication protocol

TESLA (Time Efficient Stream Loss-tolerant Authentication) [24] is a symmetric key broadcast authentication pro-

Florida State University, Department of Computer Science, 268 Love Building, Tallahassee, FL 32306-4530, U.S.A., email: burmester@cs.fsu.edu.
Ionian University, Department of Informatics, Plateia Tsirigoti 7, 49100, Corfu, Greece, email: {emagos, vchris}@ionio.gr.

tocol that requires receivers to be loosely time synchronized. It uses hash chains generated by a cryptographic one-way function H. To generate a hash chain of length n, the last element, say s, is chosen randomly. Then each term of the chain is generated recursively using the relation $h_{i-1} = H(h_i)$, $i = n, \ldots, 2$, with $h_n = s$. The chain is $h_1, h_2, \ldots, h_n$. Its keys h_i are used to authenticate messages with a MAC, and are revealed one-at-a-time within a time interval bounded by a constant $\delta\, ms$.

B. The TSVC protocol

TSVC (Timed efficient and Secure Vehicular Communication) [20] is a strict-schedule beacon broadcasting (application-layer) protocol that uses a hash key chain to authenticate safety messages. The hash keys are trust-linked via public keys and certificates to a certifying authority. Each vehicle has a list of public/private key pairs (PK_i, SK_i), and corresponding certificates $Cert_i$ that link them to pseudo-identities $PVID_i$. For the purpose of traceability, a Registration Authority (RA) keeps records of the certificates and the corresponding identities of vehicles. Each key pair has a relatively short lifespan. Hash keys are linked to a particular public key PK_i and used to authenticate vehicles. TSVC uses a TESLA hash chain $h_1, h_2, \ldots, h_n$ generated by a cryptographic hash function H. Two types of packets are broadcast by a vehicle O: *data packets* P_j and *key release packets* kr_P_j. Data packets have the form:

$$P_j = \langle PVID_0, M_j, MAC_{h_j}(M_j||T_j), T_j, index\rangle,$$

where $PVID_0$ is a pseudo-identity for vehicle O, M_j is a safety message, T_j is the time when the message is broadcast, and $index = j$ is the index of the hash key h_j. The key release packets have the form:

$$kr\_P_j = \langle PVID_0, h_j, index, T'_j\rangle, \quad j > 1,$$

where h_j is the hash key and T'_j is the time when the key release packet is broadcast. The first key release packet is authenticated using the public key of vehicle O:

$$kr\_P_1 = \langle PVID_0, sig_{SK_O}(h_1, 1, T'_1), h_1, 1, T'_1, Cert_O\rangle.$$

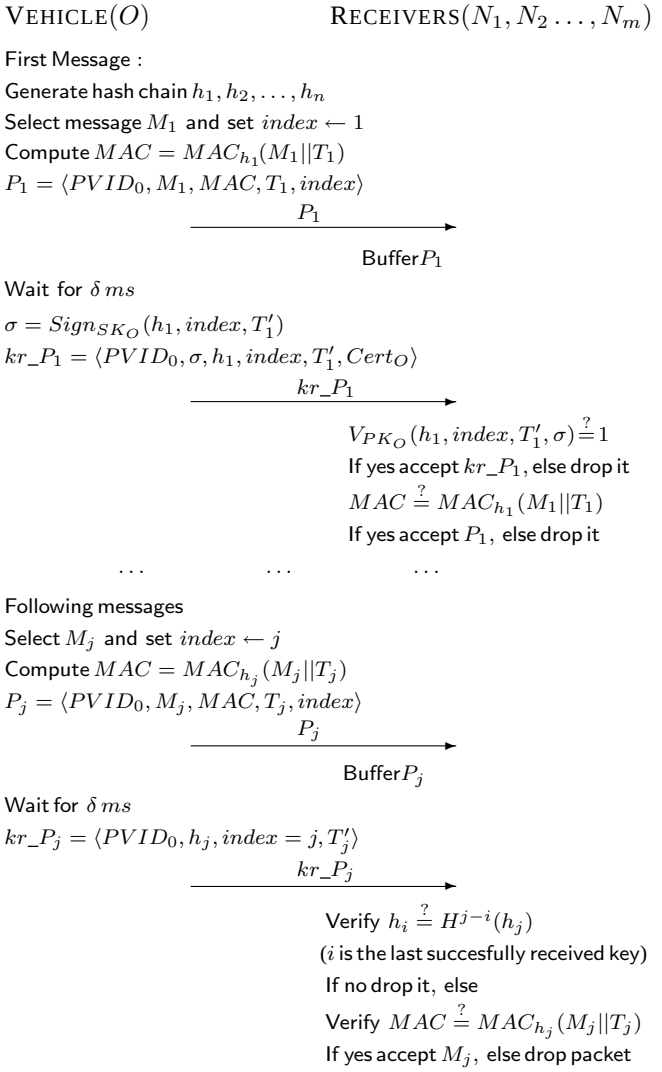
In the TSVC protocol (Figure 1) a vehicle O first broadcasts the data packet P_j and then, after $\delta\, ms$ (typically $\delta = 100\, ms$), the corresponding key release packet kr_P_j. The vehicles in a group formation that receive data packets store these in a buffer, and check their validity when the corresponding key is released.

Each vehicle stores in a database DB, for each source vehicle O, an entry with the following information: $(source, index, key, lifetime)$, with values $PVID_0$, i, h_i, and a timer controlling how long the entry is active. This information is updated after each successful key release packet verification.

C. Threat model

We assume a traditional *Byzantine* adversary [25], *i.e.*, the adversary is able to eavesdrop or modify the contents of the communication channels, provide inputs to honest parties, observe their outputs, and coordinate the actions of corrupted parties. The adversary is an outsider or an insider that attempts to modify messages in transit, or replay messages to disrupt the network. Additionally, the adversary is capable of interacting with other sessions of the protocol that may be executing concurrently. All components of the VANET (the vehicles, the RSUs, the RA) including the adversary are modeled by probabilistic, polynomial-time Turing machines.

Fig. 1. The TSVC Protocol in [20]



D. An impersonation (substitution) attack

We describe an impersonation attack on TSVC in which the adversary sends misleading safety messages on behalf of authorized users. Let the vehicles O, N_1, N_2 form a group with leader O and let N_1 be an adversarial vehicle (Figure 2). Suppose that O has broadcast the messages:

$$P_1, kr\_P_1, \ldots, P_{j-1}, kr\_P_{j-1},$$

and that just after kr_P_{j-1} is broadcast, vehicle N_2 leaves the group formation, but is still in the range of N_1 (Figure

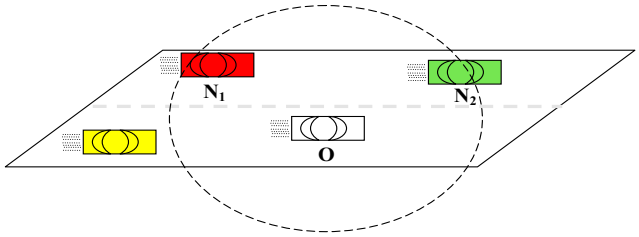


Fig. 2. Groupwise communication in the TSVC scheme

3). Vehicle N_1, after receiving P_j and the key release packet

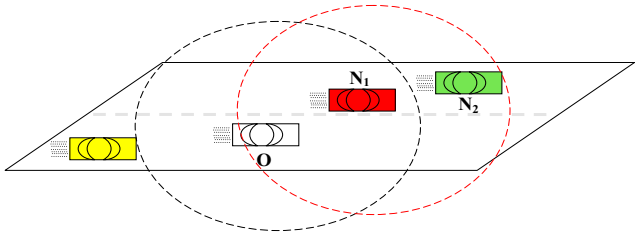


Fig. 3. An impersonation attack on TSVC

$kr\_P_j = \langle PVID_0, h_j, index = j, T'_j \rangle$, prepares a forged message M_j^* and a data packet

$$P_j^* = \langle PVID_0, M_j^*, MAC^*, T_j^*, index = j \rangle,$$

to be sent to Vehicle N_2 at time T_j^* close to T_{j+i}, for some $i \geq 1$ —which allows for i missed packets, where $MAC^* = MAC_{h_j}(M_j^* || T_j^*)$. The packet P_j^* is followed after $\delta\, ms$ by the corresponding key release packet

$$kr\_P_j^* = \langle PVID_0, h_j, index = j, T_j'^* \rangle.$$

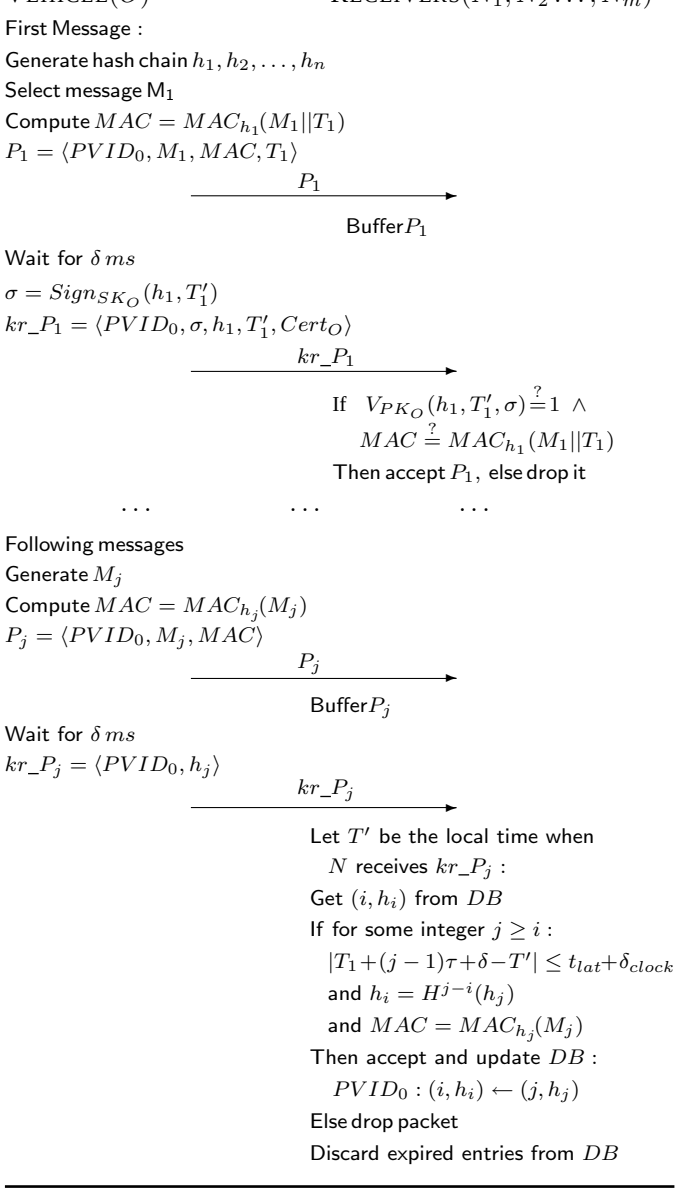
Vehicle N_2 uses the stored key value $(j-1, h_{j-1})$ to verify that $h_{j-1} = H(h_j)$ (vehicle N_2 does *not* check the time interval for the stored key h_j in the TSVC protocol), and then verifies the MAC for M_j^*. Consequently N_2 will accept the (forged) message M_j^* as an authentic message sent by O.

This attack is a timing attack: vehicle N_2 does not check that the key release packet $kr\_P_j^*$ contains a key h_j which is for the much earlier timeslot $[T_j, T_j + \delta]$. In the attack only vehicle O can be linked to the forged packet P_j^*. It follows that the owner of PK_O (with pseudo-identity $PVID_0$) will be traced by the RA as the sender of the (forged) message M_j^*, and not the adversarial vehicle(s).

E. Synchronization

Let $t_{latency}$ be the time it takes a message to reach a vehicle (communication latency) and δ the time taken to release a key. If the difference in time between the clock of the sender O and receiver N_2 is greater than $\frac{1}{2}\delta\, ms$, then the adversary can forge the data packets of O. For example, suppose that the clock of O is $\frac{2}{3}\delta\, ms$ slower than the clock of a vehicle N_2 that is not in the range of O, but in the range of an adversarial vehicle. Then the adversary can forge the message and MAC of the first data packet P_1 of O and forward the forged data packet so the N_2 gets it at (local) time $T_1 + \frac{1}{3}\delta + t_{latency}$,

Fig. 4. A fix for the TSVC Protocol.



where T_1 is the time P_1 was sent by O, with the key release packet following after $\frac{2}{3}\delta\, ms$. Vehicle N_2 will get the key release packet $\delta\, ms$ after time $T_1 + t_{latency}$, where $T_1 + \delta$ is the certified time in the key release packet —vehicle N_2 does not use its own clock to check the actual time that P_1 was sent. To avoid such attacks we require that $\delta >> t_{latency} + 2\delta_{clock}$, where δ_{clock} is an upper bound on the time differences of the clocks of all vehicles. A discussion of approaches to time-synchronization in VANETs is given in [20], [24]. Typically, vehicles are synchronized via an external source, such as GPS signals. Or, the road infrastructure (*i.e.*, an RSU) could regularly broadcast the certified time.

F. A fix for the TSVC protocol

The problem with the TSVC protocol (Figure 1) is that the receivers do not check the validity of the hash key h_j *for the*

transmission interval of the data packet. Although the packet P_j sent by O is timed (with timestamp T_j) and the key release packet kr_P_j is timed (with timestamp T'_j), and the receiver checks that the listed times are within acceptable bounds, the receiver does not check that the value of the key h_j listed in the key release packet is correct for the transmission time interval (the value of $index$ can be forged). The adversary can exploit this weakness and undermine the security of TSVC.

To fix the TSVC protocol we have to make certain that the receiver vehicle uses its own clock to determine that the appropriate key for the transmission interval is used, and does not rely on the value of *index* in the key release packet. We shall assume that clocks are highly accurate, but not necessarily synchronized. However we assume that the difference in time between the clocks of all the vehicles is bounded by a constant δ_{clock} that is significantly less than the key release time: $\delta_{clock} << \delta$. Let $T_j = T_1 + (j-1)\tau$, $j = 2, 3, \ldots$, be the times when vehicle O broadcasts its data packets (typically $\tau = 300\,ms$), and $T'_j = T_j + \delta$ be the times it broadcasts the key release packets (typically $\delta = 100\,ms$). To check the transmission time, the receiver vehicle, say N, uses the first data packet P_1 sent by O. If this is received at time T, and if T_1 is the time listed in P_1, then the difference in time should be bounded by:

$$T - T_1 \leq t_{latency} + \delta_{clock}, \tag{1}$$

where $t_{latency}$ is the communication latency; for a $1000\,m$ range this is bounded by $10\,ms$ [20]. Furthermore, if the clocks are accurate then Equation (1) must apply to all subsequent times T_j, $j = 2, \ldots, n$. It follows that when, later on, vehicle N receives a data packet P from O, if the local time (determined by the clock of N) is T, then $|T_j - T| \leq t_{latency} + \delta_{clock}$, for some integer j. If the packet P is followed shortly by the key release packet kr_P when the local time is T', then we must have $T' - (T_j + \delta) \leq t_{latency} + \delta_{clock}$. Consequently,

$$T' - (T_1 + (j-1)\tau + \delta) \leq t_{latency} + \delta_{clock}.$$

Observe that vehicle N relies totally on the time of its own clock to determine the validity of packets: it does not need a timestamp from O nor the value of $index$—which may be forged. By synchronizing its clock to the clock of O using the (digitally signed) timestamp T_1 of the first key release packet kr_P_1, it can compute on its own the relevant time-periods. N only needs the key h_j: if this arrives during the correct local time-period, then the data packet is authentic. In Figure 4 we illustrate the necessary modifications to secure TSVC.

G. Unsuitability of TESLA for vehicular applications

There are four major concerns regarding the use of TESLA for securing V2V communications.

1) *TESLA is not appropriate for highly dynamic group configurations*, with vehicles leaving or joining groups very frequently [11].
2) *TESLA is not appropriate for delay intolerant networks* [11]. In TSVC, the verification of a data packet is only possible after its key is released, and there is a delay in validating safety messages. Apart from delay-tolerant applications designed for VANETs [26], V2V routine messages are considered as delay-intolerant data [11].
3) *TESLA is subject to buffer overflows* [27]. This may cause a denial-of-service (DOS) attack, in which the attacker floods receivers with invalid messages.
4) *TESLA does not support non-repudiation*: after the hash key is released it is easy to forge messages.

Concern 1 is partially addressed by having vehicles regularly re-broadcast their first message, in particular whenever a new vehicle (with a new $PVID$) sends a data packet (not necessarily the first packet). Concern 2 is partially addressed by having a short key release time δ. In the following section we shall consider a protocol that uses a variant of TESLA, $TESLA^0$, for which there is no delay and packets are self-authenticating. This mechanism also addresses Concern 3. As for Concern 4, TESLA should not be used to protect event safety information, where the source must be identifiable.

H. Security vs reliability

The value $\delta = 100\,ms$ of the key disclosure delay is chosen so that routine safety messages can reach all vehicles in the full transmission range of the source O (typically up to $1000\,m$ [20]). For a vehicle $10m$ away from O, having to wait $100\,ms$ before a safety message can be validated, may be too long for some safety applications, *e.g.*, for close proximity manoeuvering. One may therefore want to adopt a more flexible approach that distinguishes neighbor vehicles, *e.g.* those less than $50m$ away, from vehicles further away. We shall describe such an approach below.

III. Synchronized vehicular communication

A. The $TESLA^0$ authentication protocol

$TESLA^0$ is a variant of TESLA in which a hash chain is used for *origin integrity* (authentication): each key is released *together* with its data packet ($\delta = 0$) and used as a token to identify the sender. The tokens are "*self destructing*" authenticators: they are valid only if "seen" during the period $(T_j, T_j + \varepsilon)$, where T_j is the time the key was sent and $\varepsilon > 0$ a time-bound. This period must be *very* short, with ε less than the time a man-in-the-middle attack takes.

Consequently any message attached to the token is *implicitly* authenticated, provided it is "seen" during the period $(T_j, T_j + \varepsilon)$. There is an affinity between interactive zero-knowledge proofs [28] and $TESLA^0$ authenticators. For both: (i) only the receiver (verifier) gets convinced of a certain truth (in $TESLA^0$: "that the sender is authentic"), and (ii) the evidence of the proof can easily be generated after the protocol is executed (in $TESLA^0$: "the packet can be forged"). $TESLA^0$ authenticators are non-interactive and inherently *one-to-many*, so appropriate for broadcast applications. However their shelf life is short and restricted to settings with synchronized clocks.

The protocol uses strict-schedule broadcasting, with the j-th packet P_j, $j = 1, 2, \ldots$, sent at time:

$$T_j = T_1 + (j-1)\tau.$$

The first packet

$$P_1 = \langle PVID_0, M_1, h_1, T_1, \sigma, Cert_0 \rangle, \quad (2)$$

includes the timestamp T_1 for the start time (chosen arbitrarily by each vehicle), the first key h_1, a message M_1, a digital signature:

$$\sigma = Sign_{SK_O}(h_1, T_1),$$

and the certificate $Cert_O$. The following packets are of the form:

$$P_j = \langle PVID_0, M_j, h_j \rangle, \ j > i,$$

and do not include a MAC, a timestamp or an index.

Let ε be a lower bound for $t_{latency} + t_{forge} - \delta_{clock}$, where $t_{latency}$ is the communication latency, t_{forge} the time it takes to forge a data packet (essentially, to read a hash key, and deliver the forged packet), and δ_{clock} the time discrepancy between clocks. We shall assume that the clocks of all parties are accurate, and that δ_{clock} is significantly less than ε: $\delta_{clock} << \varepsilon$.

The shelf life ε should be sufficiently small to make it impossible for the adversary to forge packets. Whenever P_j is received, for some integer $j > i$, where i is the index of the last validated packet of $PVID_0$, the receiver checks that: (i) $|T - (T_1 + (j-1)\tau)| < \varepsilon$, where T is the time P_j was received—receivers use their own clocks, and (ii) $h_i = H^{j-i}(h_j)$ —this allows for $(j-i)$ missed packets. Packets P_j that satisfy both constraints are valid. All other packets are discarded. Note that the time it takes to validate a packet may be more than ε: it is therefore important that T is calculated using the recorded time when P_j is received, not after it is checked.

$TESLA^0$ does not provide *explicit* data integrity, since the packets do not contain a MAC. However it does provide *implicit* data integrity assuming that: (1) the message is "seen" within the period $(T_j, T_j + \varepsilon)$, where ε is sufficiently small to prevent the adversary from substituting the original message, and (2) we have origin integrity.

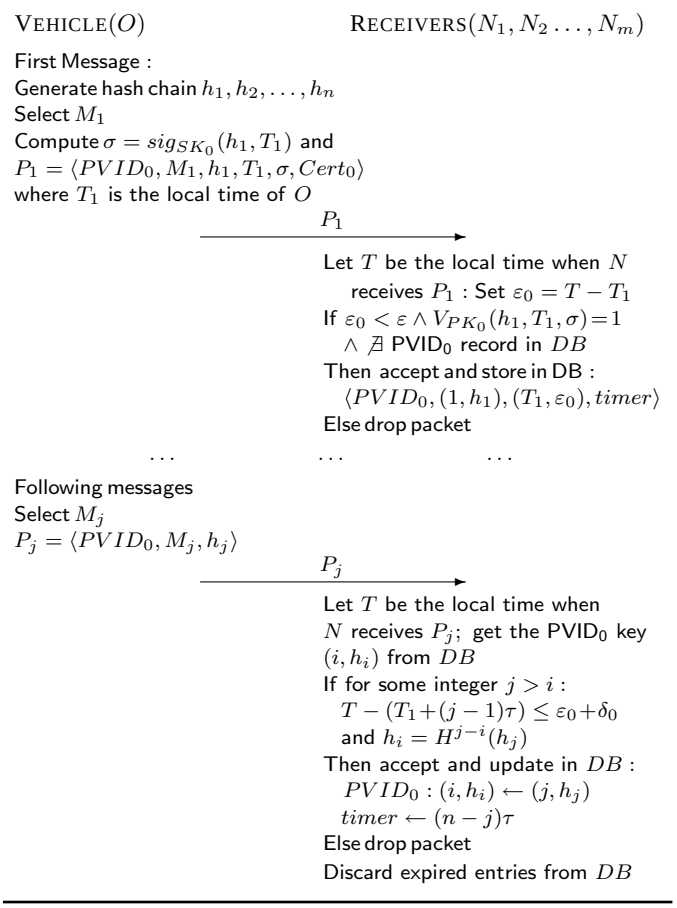
B. Vehicular communication based on $TESLA^0$

We now present a variant of TSVC (as modified in Section II-F) that uses a $TESLA^0$ hash chain for close proximity communication to address impersonation attacks, packet delays and buffer overflows. The protocol is illustrated in Figure 5.

As in TSVC, each vehicle has a list of public/private key pairs, pseudonyms, and certificates that link the vehicle identifier to the pseudonyms for conditional traceability. Note that this does not provide assurance against non-repudiation: an adversarial vehicle can transmit malicious packets P_j^* and later, after the key is released, repudiate them. This applies to all TESLA-based schemes.

Packets are broadcast at regular intervals (strict-schedule broadcasting) and authenticated using the $TESLA^0$ protocol, with each vehicle O broadcasting a data packet P_j at time $T_j = T_1 + (j-1)\tau$, $j = 1, 2, \ldots, n$. The first data packet P_1 (Equation (2)) includes the pseudonym $PVID_0$ for vehicle

Fig. 5. A $TESLA^0$ Vehicular Communication Protocol



O and the transmission time T_1, authenticated by the digital signature σ. If it is received at time T bounded by:

$$T - T_1 < \varepsilon,$$

where ε is a short time interval (Section III-A), then it is accepted as authentic. The receiver also keeps an entry in DB: $(PVID_0; index \leftarrow 1, key \leftarrow h_1; T_1; \varepsilon_0 = T - T_1;\ timer \leftarrow (n-1)\tau)$, where $timer$ controls the lifetime of the hash chain session.

Let $\varepsilon_0 = T - T_1$ and δ_0 be a short time interval (typically $\delta_0 \sim 0.5\,ms$) to allow for vehicle mobility (in a range of $30-50\,m$). The value $\varepsilon_0 + \delta_0$ is used to time all future readings of the receiving vehicle. For the following time intervals, if a data packet P_j sent by O is received at local time T (of the receiver) then the receiver checks that:

$$T - (T_1 + (j-i)\tau) \leq \varepsilon_0 + \delta_0, \text{ for some integer } j > i,$$

where i is the index of the last validated packet from O, and that: $h_i = H^{j-i}(h_j)$. If these hold then P_j is accepted as authentic and the receiver updates the entry of $PVID_0$ in DB with new values: $index \leftarrow j$, $key \leftarrow h_j$ and $timer \leftarrow (n-j)\tau$. When the timer reaches 0, the $PVID_0$ entry is discarded.

Our $TESLA^0$-based communication protocol is suitable for settings where the communication latency is sufficiently small (typically $3-5\,ms$—see also Section V-B) to make it

difficult for the adversary to forge packets. For VANETs this setting covers either unsaturated conditions with medium-to-long communication range (typically, up to $1000\,m$ [5]) or saturated, city traffic conditions with very short range transmissions (typically, below $100\,m$) to reduce communication latency [29]. For all other cases, as for example in saturated conditions where we would also like to warn cars at the maximum range, the TSVC scheme should be used.

IV. A HYBRID SCHEME

We can combine TSVC and $TESLA^0$ to get a hybrid authentication scheme that aggregates their strenghts with only marginally more overhead than TSVC. The hybrid scheme uses two hash chains: $\{h_j\}$ for TSVC and $\{h_j^0\}$ for $TESLA^0$. These are linked to the sender with the digital signatures σ, σ^0 and the certificate $Cert_0$.

The first data packet P_1 of the hybrid system is obtained by appending to the corresponding packet of TSVC (the modified version in Section II-F) the $TESLA^0$ key h_1^0, a digital signature $\sigma^0 = sig_{SK_O}(h_1^0, T_1)$ of the source O authenticating h_1^0 and T_1, and a certificate:

$$P_1 = \langle PVID_0, M_1, h_1^0, MAC, T_1, \sigma^0, Cert_0 \rangle.$$

When a vehicle N receives P_1 it records the time T it was received and stores in a database DB the record: $(PVID_0; T_1; T; (1, h_1^0); timer)$. Then it checks that:

1) $|T - T_1| < \varepsilon_0 + \delta_0$ (the vehicles O, N are in close proximity), and
2) the signature σ^0 on (h_1^0, T_1) is valid, and $Cert_0$ is a valid certificate for the source O.

If these hold then it accepts M_1 as *implicitly* authen- ticated. Otherwise vehicle N waits $\delta\,ms$ for the key release packet:

$$kr\_P_1 = \langle PVID_0, h_1, \sigma \rangle,$$

which contains the TSVC key h_1 and the signature $\sigma = sig_{SK_O}(h_1)$ that link it to the sender, to verify that $MAC = MAC_{h_1}(M_1)$ directly. If P_j is authentic then the record of $PVID_0$ in DB is updated.

The j-th packet, $j > 1$, of the hybrid scheme is:

$$P_j = \langle PVID_0, M_j, h_j^0, MAC \rangle,$$

where $MAC = MAC_{h_j}(M_j)$. The time T it is received and the key h_j^0 are used for close proximity (implicit) authentication. If i is the index of the last received valid packet, then we require that: (1) $|T - T_j| < \varepsilon_0 + \delta_0$ for some $j > i$, and (2) $h_i^0 = H^{j-i}(h_j^0)$. If these are satisfied then M_j is accepted and the record of $PVID_0$ updated. Otherwise vehicle N waits $\delta\,ms$ for the key release packet:

$$kr\_P_j = \langle PVID_0, h_j \rangle,$$

that contains the TSVC key h_j used to verify $MAC = MAC_{h_1}(M_1)$ directly, and authenticate M_j explicitly. If P_j is authenticated then the record of $PVID_0$ in DB is updated.

The threshold $\varepsilon_0 + \delta_0$, the waiting time δ, and the frequency τ of transmission are system parameters. To deal with buffer overflow issues packets that are broadcast outside the expected times: $T_j = T_1 + (j-1)\tau$ and $T_j' = T_j + \delta$, are discarded (we allow for a small deviation, that is at least as large as the upper bound δ_{clock} for the time discrepancy of clocks).

In the following sections we shall see that the hybrid scheme addresses a major weakness of TSVC (the disclosure delay δ dominates the communication latency—Section V-B) and that on average it only requires 8 bytes more than TSVC (taken over $1,000$ packets—Section V-A).

A. Security analysis

Protection involves privacy (anonymity) and integrity. The privacy adversary tries to identify the source O of the transmitted packets, whereas the integrity adversary tries to forge the packets of O. Privacy is assured because O uses the pseudonym $PVID_0$. We have (conditional) unlinkability because the pseudonym of O for each session is linked to independent public keys PK_O.

The $TESLA^0$ integrity adversary may try to forge packets of O within the range of vehicle O, or beyond its range. Since it is hard to forge the key h_j^0 (this follows from the fact that a cryptographic one-way function is used to generate hash keys and a digital signature scheme is used to link it to the sender) and its lifespan is short (less than the time it takes to deliver a forged packet), the adversary cannot send forged packets P_j^* to a vehicle N in the range of O *before* N gets the authorized packet P_j from O (the adversary needs to get the key h_j^0 contained in P_j to forge it). This proves integrity for the close proximity authentication scheme based on $TESLA^0$. Forging packets beyond the range of O takes even longer, and therefore is thwarted. The security of the TSVC component of the hybrid scheme is based on the security of TESLA [20].

V. EFFICIENCY

The hybrid scheme distinguishes between close proximity V2V coommunication (low communication latency) and communication with vehicles further away (high latency). For close proximity communication there is no key disclosure delay in the $TESLA^0$ component ($\delta = 0$). As a result, there is no delay in validating safety messages. This can important for safety applications, *e.g.*, manoeuvering vehicles in close proximity to a sender O do not have to wait $100\,ms$ before validating safety messages. When communication latency is high (*e.g.*, in saturated traffic with long range communication), the TSVC component is invoked.

A. Bandwidth efficiency

Assume that $n = 1000$ routine safety messages are sent at $300\,ms$ intervals, and that the ECDSA [30] signature scheme is used, combined with the SHA-1 algorithm [31] for hashing. The length of the first data packet of the hybrid scheme is,

$$\begin{aligned} \ell(P_1) &= \ell(M_1) + \ell(PVID_0) + \ell(h_1^0) + \ell(MAC) + \ell(T) \\ &\quad + \ell(\sigma^0) + \ell(Cert_0) \\ &= 100 + 4 + 20 + 20 + 4 + 56 + 125 = 329 \text{ bytes}, \end{aligned}$$

allowing for a 100 byte payload, 4 bytes for the $PVID_0$, 20

bytes for the authenticator, 20 bytes for the MAC, 4 bytes for the time, 56 bytes for a signature, and 125 bytes for the certificate. This is 197 bytes more than for TSVC (Section III D.2, [20]—for the hybrid version P_1 contains an extra hash key, signature and certificate, but not the index). For the first key release packet we have,

$$\begin{aligned}\ell(kr\_P_1) &= \ell(PVID_0) + \ell(h_1) + \ell(\sigma) \\ &= 4 + 20 + 56 = 80 \text{ bytes},\end{aligned}$$

which is 133 bytes less than TSVC (for the hybrid version, kr_P_1 does not contain the certificate, index, or time). The other data packets have length

$$\ell(P_i) = \ell(PVID_0) + \ell(M_i) + \ell(h_i^0) + \ell(MAC) = 144 \text{ bytes},$$

as opposed to 132 bytes for TSVC (they contain one extra authenticator, but not an index or the time). The other key release packets have length

$$\ell(kr\_P_i) = \ell(PVID_0) + \ell(h_i) = 20 + 4 = 24 \text{ bytes},$$

which is 4 bytes less than TSVC (they do not contain an index or the time). The average packet length for 1,000 safety messages is:

$$(409 + 999 \times 168)/1000 \approx 168 \text{ bytes},$$

which is 8 bytes more than TSVC.

B. Communication latency

Packet delivery delay is the delay between the time a packet was generated and the time the packet is successfully received. It includes the transmission time, the propagation time, and the medium access time (*e.g.*, due to backoff, busy channel, inter-frame spaces [32], [33]):

$$t_{delivery} = t_{transmission} + t_{propagation} + t_{mac}.$$

In the TSVC protocol received packets are buffered, and only validated when the key release packets are received, which is after $\delta = 100\,ms$. Validation is done at the upper (application) layers. It follows that the actual communication latency of TSVC is:

$$t_{latency}(TSVC) = t_{delivery} + t_{application} + \delta, \qquad (3)$$

where $t_{application}$ includes all delays at the upper layers (*e.g.*, queuing, processing, etc). For the *TESLA*^{0} communication protocol the key release packets are sent together with the safety packets. So,

$$t_{latency}(TESLA^0) = t_{delivery} + t_{application}. \qquad (4)$$

As an illustration, suppose that the transmission rate is $6\,Mbps$ (the base rate of 802.11a) and the range is $1\,Km$. Then the transmission delay for a 500-byte routine safety message is roughly:

$$t_{transmission} = \frac{4\,Kb}{6\,Mbps} \sim 0.7\,ms,$$

and the propagation time is:

$$t_{propagation} = \frac{1\,Km}{3 * 10^5\,Km/s} = \frac{1}{3 * 10^5}\ s = 3.3\,\mu s,$$

assuming an electromagnetic wave velocity of $3 * 10^5\,Km/s$. The delays from upper-layer processing, in particular computing (verifying) a MAC are also small. For example, SHA-1 of 500-byte data can be computed on a 2.2 GHz AMD Opteron 8354 in less than $0.5\,\mu s$ [34], so the upper layer latency is:

$$t_{application} < 1\mu s.$$

The medium access control layer delays are harder to estimate as the collision probability in a VANET varies with the vehicle density, the velocity of vehicles and other factors [35]. Typical estimations [36], [37] are based on simulations that distinguish between *unsaturated* traffic (no more that 10 vehicles per Km) and *saturated* traffic (greater than 100 vehicles per Km). The medium access delays for the TSVC protocol are estimated for both simulations in [20]. The simplest case is with unsaturated traffic for which we get the upper bound $t_{mac} = 1\,ms$ for a transmission range of $1\,Km$ [36]. For saturated traffic the estimated delays are higher—*e.g.*, an upper bound of $14\,ms$ for a transmission range of $1\,Km$ is given in [36]. To keep delays below $10\,ms$ the authors in [29], [37] propose to reduce the broadcast range to less than $200\,m$. For this range, $t_{mac} \sim 9\,ms$. Using Equation (3) this gives us:

$$t_{latency\_unsat}(TSVC) \sim 2\,ms + \delta,$$

and

$$t_{latency\_sat}(TSVC) \sim 10\,ms + \delta.$$

In both cases the delay $\delta = 100\,ms$ in releasing the authentication keys dominates the latency, which highlights a basic weakness of delayed authentication. Of course we can reduce the delay to say, $\delta = 10\,ms$. However one has to be careful when reducing the key release time in case that for some vehicles (in the extremes of the broadcast range) the keys arrive before the safety messages are processed, which may result in attacks of the type described in Section II-D.

Our hybrid approach is designed to address such issues, in particular to exploit the "quadratic" reduction effect on saturated traffic with close proximity communication. More specifically, 100 vehicles in a $1\,Km$ range are reduced to $100 \times (\frac{100}{1000})^2 = 1$ vehicle in the $100\,m$ range. Consequently even when the traffic is saturated in the $1\,Km$ range, in the $30 - 50\,m$ range where the TESLA^0 communication protocol is used the number of vehicles cannot be more than 10, so the latency for unsaturated traffic applies. For this range using the simulations in [36] we get: $t_{mac} < 1\,ms$, so that from equation (4) we have:

$$t_{latency}(TESLA^0) \sim 2\,ms.$$

It is clear that a hybrid approach that distinguishes short range communication from long range communication to address traffic density has to be adopted, for the safety packages to be secured.

C. Collisions with strict-schedule broadcasts

The TSVC protocol as well as our modification in Section II-F and the $TESLA^0$ vehicular communication protocol rely on *strict-schedule beacon broadcasting* (typically every $\tau = 300\,ms$). This means that a collision of packet P_j will affect the whole broadcast stream of data packets $P_j, P_{j+1}, P_{j+2}, \ldots$—assuming the parties involved adhere strictly to their schedule.

We distinguish three cases: (i) the lead data packets P_1^A of vehicle A and P_1^B of vehicle B collide, (ii) the lead packet P_1^A collides with the j-th packet P_j^B of vehicle B (vehicle B joins an established group), (iii) P_i^A collides with P_j^B (vehicles A, B join an established group).

In the first case the consequences of the collision are minimized if both vehicles select a different time schedule: T_1^j, $T_2^j = T_1^j + \tau$, $T_3^j = T_1^j + 2\tau$, $\ldots$, $j = A, B$. In the second case only vehicle B selects a different time schedule: T_1^B, $T_2^B = T_1^B + \tau$, $T_3^B = T_1^B + 2\tau$, $\ldots$, while vehicle A adheres to its schedule $T_{i+1}^A = T_1^A + (i+1)\tau$, $T_{i+2}^A = T_1^A + (i+2)\tau$, $\ldots$ (the visiting vehicle B must start a new session). The last case is treated as the first one: both vehicles must select new time schedules. The same procedure is used if the key release packets collide.

D. Performance comparison

The main difference between the hybrid communication scheme and TSVC is that the key disclosure delay in the $TESLA^0$ component is $\delta = 0$. As a result, the average packet delay is negligible, as with traditional public-key (signature) protocols.

- *Impact of Vehicle Density.* There are no packet delays (PD) with $TESLA^0$. Consequently for low density (typically highway) traffic there is little variance in PDs and in the packet ratio loss (PRL) between TSVC and the hybrid scheme. However with high density (typically city) traffic, as observed in Section V-B, there is a significant improvement since the latency for short range communications in the hybrid scheme (when $TESLA^0$ is used) approximates that for low density traffic.
- *Impact of Vehicle Moving Speed.* A range of $10 - 40\,m/s$ is considered with the traffic simulations in [20] for initial inter-vehicle distance 30 meters. It is shown that for TSVC the PD is within the maximum allowable $100\,ms$ latency and the variation of speed does not significantly impact the PD and PLR.

  For the hybrid scheme with communication in the $1000\,m$ range there is no difference (TSVC is invoked). However for short range communication ($< 100m$) there are no PDs and PLR is reduced to the low density traffic case.

VI. Conclusion

We have shown that the TSVC scheme is subject to an impersonation attack and proposed a modification that addresses such attacks. We have also proposed a vehicular communication scheme for close proximity formations based on a variant of TESLA, in which messages are self-authenticated. Finally we have combined this scheme with the modified TSVC scheme to address dynamic vehicular group formations.

References

[1] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," *Communications Magazine, IEEE*, vol. 43, no. 3, pp. 123–131, 2005.

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

[3] CVIS Project, "Cooperative Vehicle-Infrastructure Systems." http://www.cvisproject.org/.

[4] IntelliDrive, "IntelliDrive Project." http://www.intelli-driveusa.org/.

[5] DSRC, "Dedicated Short Range Communications." http://www.secg.org/download/aid-385/sec1_final.pdf, 2007.

[6] Task Group p, "IEEE 802.11p wireless access for vehi- cular environments, draft standard," 2009.

[7] J. Blum and A. Eskandarian, "The threat of intelligent collisions," *IT Professional*, vol. 6, no. 1, pp. 24–29, Jan.-Feb. 2004.

[8] B. Parno and A. Perrig, "Challenges in securing vehicular networks," *Workshop on Hot Topics in Networks (HotNets-IV)*, 2005.

[9] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, (New York, NY, USA), pp. 11–21, ACM, 2005.

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

[11] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[12] J. Y. Choi, M. Jakobsson, and S. Wetzel, "Balancing auditability and privacy in vehicular networks," in *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pp. 79–87, ACM, 2005.

[13] K. Sampigethaya, L. Huang, K. Matsuura, R. Poovendran, and K. Sezaki, "Caravan: Providing location privacy for vanet," in *Escar 2005: 3rd Embedded Security in Cars Workshop*, 2005.

[14] S. Rahman and U. Hengartner, "Secure crash reporting in vehicular ad hoc networks," in *Third International Conference on Security and Privacy in Communication Networks (SecureComm 2007)*, (New York, NY, USA), To appear, 2007.

[15] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 8, pp. 1557–1568, Oct. 2007.

[16] J. Sun, C. Zhang, and Y. Fang, "An id-based framework achieving privacy and non-repudiation in vehicular ad hoc networks," *Military Communications Conference, 2007. IEEE*, pp. 1–7, 29-31 Oct. 2007.

[17] N.-W. Wang, Y.-M. Huang, and W.-M. Chen, "A novel secure communication scheme in vehicular ad hoc networks," *Computer Communications, Elsevier*, 2008.

[18] M. Burmester, E. Magkos, and V. Chrissikopoulos, "Strengthening Privacy Protection in VANETs," in *WIMOB '08: IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*, pp. 508–513, IEEE Computer Society, 2008.

[19] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "RAISE: An Efficient RSU-Aided Message Authentication Scheme in Vehicular Communication Networks," in *Proceedings of IEEE International Conference on Communications, ICC 2008*, pp. 1451–1457, IEEE, 2008.

[20] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, and X. Shen, "TSVC: timed efficient and secure vehicular communications with privacy preserving," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 12, pp. 4987–4998, 2008.

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

[22] K. Plőíl and H. Federrath, "A privacy aware and efficient security infrastructure for vehicular ad hoc networks," *Comput. Stand. Interfaces*, vol. 30, no. 6, pp. 390–397, 2008.

[23] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[24] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA Broadcast Authentication Protocol," *RSA CryptoBytes*, vol. 5, 2002.

[25] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, pp. 198–207, 1983.

[26] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "SMART: A Secure Multi-Layer Credit based Incentive Scheme for Delay-Tolerant Networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4628–4639, 2009.

[27] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient vanet authentication," in *Proceedings of the 6th Embedded Security in Cars Workshop (ESCAR)*, Nov. 2008.

[28] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems," *J. ACM*, vol. 38, no. 3, pp. 690–728, 1991.

[29] M. van Eenennaam, G. Karagiannis, and G. Heijenk, "Towards Scalable Beaconing in VANETs," in *Fourth ERCIM workshop on eMobility*, pp. 103–108, 2010.

[30] SECG, "Standards for efficient cryptography group. SEC 1: Elliptic curve cryptography." Available at: http://www.secg.org/download/aid-385/sec1_final.pdf, 2005.

[31] National Institute of Standards and Technology, *FIPS PUB 180-3: Secure Hash Standard*. pub-NIST:adr: pub-NIST, October 2008.

[32] X. Ma, X. Chen, and H. Refai, "Performance and reliability of DSRC vehicular safety communication: a formal analysis," *EURASIP J. Wirel. Commun. Netw.*, vol. 2009, pp. 1–13, 2009.

[33] X. Ma and X. Chen, "Delay and broadcast reception rates of highway safety applications in vehicular ad hoc networks," in *Proceedings of IEEE Workshop on Mobile Networking for Vehicular Environments (MOVE07)*, pp. 85–90, 2008.

[34] W. Dai, "Crypto++ 5.6.0 Benchmarks." http://www. cryptopp.com/benchmarks.html, 2009.

[35] J. An, X. Guo, and Y. Yang, "Analysis of collision probability in vehicular ad hoc networks," in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pp. 791–794, ACM, 2009.

[36] N. Wisitpongphan, O. Tonguz, J. Parikh, P. Mudalige, F. Bai, V. Sadekar, *et al.*, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 14, no. 6, p. 84, 2007.

[37] R. Chen, D. Ma, and A. Regan, "TARI: Meeting Delay Requirements in VANETs with Efficient Authentication and Revocation," in *2nd International Conference on Wireless Access in Vehicular Environments (WAVE)*, IEEE, 2009.