

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/250004065>

# Strong VANET security on a budget

Article · January 2006

---

CITATIONS

51

---

READS

223

2 authors, including:



[Kenneth Laberteaux](#)

Toyota Research Institute, North America

35 PUBLICATIONS 2,071 CITATIONS

SEE PROFILE

## STRONG VANET SECURITY ON A BUDGET

Yih-Chun Hu and Kenneth P. Laberteaux

**Abstract:** This article proposes a security authentication process that is well-suited for Vehicular Ad-hoc Networks (VANET). As compared to current Public Key Infrastructure (PKI) proposals for VANET authentication, the scheme is significantly more efficient with regard to bandwidth and computation. The scheme uses time as the creator of asymmetric knowledge. A sender creates a long chain of keys. Each key is used for only a short period of time to sign messages. When a key expires, it is publicly revealed, and then never again used. (The sender subsequently uses the next key in its chain to sign future messages.) Upon receiving a revealed key, recipients authenticate previously received messages. The root of a sender's keychain is given in a certificate signed by an authority. This article describes several possible certificate exchange methods. It also addresses privacy issues in VANET, specifically the tension between anonymity and the ability to revoke certificates.

### 1 INTRODUCTION

On an average day in the United States, vehicular collisions kill 116 and injure 7900. More health care dollars are consumed in the United States treating crash victims than any other cause of illness or injury [3], [4], [5]; the situation in the European Union is similar, with over 100 deaths and 4600 injuries daily, and the annual cost of €160 billion [6]. Governments and automotive companies are responding by making the reduction of vehicular fatalities a top priority [4], [7].

Dedicated Short Range Communications (DSRC) [8], [9], [10] is the leading wireless technology under consideration for vehicular safety applications [1]. Significant progress has been made in standardizing the lower layer protocols for DSRC. Further, as discussed in Section 2, several industry/government consortiums strive to identify which vehicular safety applications (and related technologies) will provide the greatest safety benefits.

This paper describes a method for authenticating message packets for wireless vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communications, which is envisioned to enable a new generation of vehicular safety applications [1], [2]. The efficacy and reliability of a system where information is gathered and shared among autonomous entities raises concerns about the authenticity of the received data. For example, a bad actor could misrepresent its observations in order to gain advantage (e.g. a vehicle  $V$  falsely reports that its desired road  $R$  is stopped with traffic, thereby encouraging others to avoid  $R$  and providing a less-congested trip for  $V$  on  $R$ ). More malicious actors could impersonate other vehicles or road-side infrastructure in order to trigger safety hazards. Vehicles could reduce this threat by creating networks of trust, and ignoring, or at least distrusting, information from un-trusted senders.

A trusted communication generally requires two properties are met:

- The sender is conclusively identified as a trusted source.
- While in transit, the contents of the sender's message are not tampered.

Since the various players in the vehicular application space are converging on DSRC, we formulate our problem to account for the FCC rules for DSRC [9] and the current industry consensus as reflected by the draft IEEE 802.11p standard and its related draft standards, IEEE P1609 [11]. However, this article proposes an alternative approach to the current draft standard IEEE P1609.2 (previously named P1556), which addresses security services for DSRC. Specifically, this article proposes a lightweight authentication protocol to achieve strong security with relatively smaller overheads.

### 2 VANET-SUPPORTED VEHICULAR SAFETY APPLICATIONS

Several industry/government consortiums strive to identify which vehicular safety applications (and related technologies) will provide the greatest safety benefits. These organizations include the Crash Avoidance Metrics Partnership<sup>1</sup> (CAMP), the Cooperative Intersection Collision Avoidance (CICAS) consortium<sup>2</sup>, the Car2Car Communications Consortium<sup>3</sup>, the Advanced Safety Vehicle (ASV) Project<sup>4</sup>, and others. For example, the deliberations by the US National Highway Traffic Safety

---

Yih-Chun Hu (yihchun@crhc.uiuc.edu) is with the University of Illinois at Urbana-Champaign, Dept. of Electrical and Computer Engineering, 1308 West Main St., Urbana, IL 61801, +1-217-333-4220. Kenneth P. Laberteaux (klaberte@acm.org) is with the Toyota Technical Center, USA, Inc., 1555 Woodridge Ave., Ann Arbor, MI 48105, +1-734-995-2600.

<sup>1</sup> CAMP comprises the following vehicle companies: BMW, DaimlerChrysler, Ford, GM, Honda, Nissan, Toyota, and Volkswagen. It works in partnership with NHTSA. Several of CAMP's project reports can be found at [http://www-nrd.nhtsa.dot.gov/departments/nrd-12/pubs\\_rev.html](http://www-nrd.nhtsa.dot.gov/departments/nrd-12/pubs_rev.html).

<sup>2</sup> <http://www.its.dot.gov/cicas/index.htm>

<sup>3</sup> <http://www.car-2-car.org/>

<sup>4</sup> The ASV is a partnership by Japan's 14 automobile, truck, and motorcycle manufactures, sponsored by Japanese Ministry of Land, Infrastructure and Transport (<http://www.mlit.go.jp/>).

Administration (NHTSA), the US Department of Transportation (USDOT), and the Vehicle Safety Communications Consortium (VSCC) of CAMP have identified eight such applications<sup>5</sup> [1],[2]. They are:

**Near-term:**

1. **Traffic Signal Violation Warning**
2. **Curve Speed Warning**
3. **Emergency Electronic Brake Lights**

**Mid-term:**

4. **Pre-Crash Warning**
5. **Cooperative Forward Collision Warning**
6. **Left Turn Assistant**
7. **Lane Change Warning**
8. **Stop Sign Movement Assistance.**

The communication requirements of these eight applications are shown in Table 1. Note that communication frequency ranges from 1-50 Hz, and the maximum communication range spans from 50-300 meters. Further, high-level data element requirements are specified. Several of these data elements (e.g. Position and Heading) are needed by multiple applications. These messages can be efficiently composed using a *Message Dispatcher* [12], which results in messages ranging from 25 bytes to several hundred bytes. However, the small 25 byte messages (often called *heartbeat messages*) are the most common and comprise a significant fraction of all messages sent.

Application	Comm. type	Freq.	Latency	Data Transmitted	Range
Traffic Signal Violation	I2V One-way, P2M	10 Hz	100msec	SignalStatus, Timing, Surface Heading, Light Posn.,Weather,	250m
Curve SpeedWarning	I2V One-way , P2M	1 Hz	1000msec	Curve Location, Curvature, SpeedLimit, Bank, Surface	200m
EmergencyBrake Lights	Vehicle to Vehicle Two-way, P2M	10 Hz	100msec	Position, Deceleration Heading, Velocity,	200m
Pre-CrashSensing	Vehicle to Vehicle Two-way, P2P	50 Hz	20msec	Vehicle Type, Yaw Rate, Position, Heading, Accel.	50m
Collision Warning	Vehicle to Vehicle One-way, P2M	10 Hz	100msec	Vehicle Type, Position, Heading Velocity, Acceleration, Yaw Rate	150m
Left Turn Assist	I2V and V2I One-way, P2M	10 Hz	100msec	Signal Status, Timing, Posn. Direction, RoadGeom., Vel. Heading	300m
Lane Change Warning	Vehicle to Vehicle One-way, P2M.	10 Hz	100msec	Position, Heading, Velocity Accel., Turn Signal Status,	150m
StopSign Assist	I2V and V2I One-way	10 Hz	100msec	Position, Velocity Heading, Warning.	300m

Table 1 Eight high-priority vehicular safety applications as chosen by NHTSA and VSCC [2]. Note that communication frequency ranges from 1-50 Hz and maximum communication range span 50-300 meters. P2M represents 'Point-to-Multipoint', I2V represents 'Infrastructure-to-Vehicle' and V2I represents 'Vehicle-to-Infrastructure'.

The majority of the remainder of this paper is focused on identifying a method of authenticating wireless safety messages, which are nominally frequent (10 Hz per car) and small (a large fraction near 25 bytes)

### 3 BROADCAST AUTHENTICATION

*Authentication* is a cryptographic primitive that allows the recipient of a message to ascertain that the contents of a message were not tampered with, and to determine the source of that message. There are two types of authentication schemes: *unicast* and *broadcast* authentication. In unicast authentication, it is sufficient to prove that the message must have come from either the sender or the receiver. Because the receiver knows that it did not originate the message, it can ascertain that the sender sent it. Unicast authentication is easily achieved when the sender and receiver can share a private value. Broadcast authentication, on the other hand, requires *asymmetric* information: the sender must know more information than the receiver; otherwise, any receiver can send an authenticated message pretending to be the sender.

#### 3.1.1 Conventional Public Key Signatures

Broadcast authentication is typically achieved through the use of public key signatures. This is the scheme adopted by P1609.2 [11]. In a public key signature, the signer possesses a private key  $K^-$ , whereas verifiers possess only a public key  $K^+$ . Knowing the private key allows the signer to generate signatures, and any signature can be verified using the public key.

<sup>5</sup> The authors are aware of similar deliberations in Europe and Asia, with similar results.

However, with the public key alone, it is computationally infeasible to generate a valid signature. A number of public-key signature schemes have been proposed, including RSA [14] and ElGamal [15]. In general, public key approaches require significantly more computation time and storage space. For example, signatures in both DSA [16] and ECDSA [17] are 40 bytes long, and the ECDSA signature specified in IEEE 1609.2 [11] requires signatures that are at least 64 bytes. When messages are small (e.g., 25 bytes), 40 or 64 bytes represents a significant security overhead.

In most public key systems, any computer can generate a public-private key pair, so to ensure that the public key belongs to the node being authenticated, a structure called a Public Key Infrastructure (PKI) is needed. In a PKI, certificate authorities (CAs) sign bindings between public keys and node identifiers; these bindings are called *certificates*. Any entity that trusts this CA will then store its public key. Certificate generation can be hierarchical; for example, a CA may delegate limited certificate-signing authority to another authority. For example, an email-address PKI may start from the domain registry, which would sign a delegation for any `doe.com` email address to the administrator. The administrator could then sign a certificate for the `john@doe.com` email address. In a vehicular network, the same structure that is used to assign VIN numbers in each country could be used as a certificate authority, delegating signature authority whenever it delegates a block of VIN numbers. A federal governmental agency (e.g. United States Department of Transportation) could designate regional (e.g. Michigan Department of Transportation) or municipal agencies to create and manage certificates. The challenging issue of certificate revocation is not addressed in this paper. Readers interested in further exploration of certificates are referred to [21].

### 3.1.2 Lightweight Broadcast Authentication

TESLA [18] is a lightweight broadcast authentication mechanism that turns a unicast authentication mechanism into a broadcast authentication mechanism. A brief summary of TESLA is now given.

The intuition of TESLA is to use time as the creator of asymmetric knowledge. In this section, we will use the term HMAC to refer to a general *message authentication code*.<sup>6</sup> Message authentication codes rely on a shared key  $K$  between the sender and the receiver; the holder of the shared key can compute the authenticator  $a = \text{HMAC}_K(m)$  on any message  $m$ . TESLA also relies on collision-resistant one-way hash functions  $H(x)$ ; such functions have the property that  $H(x)$  is fast to compute, but are *one-way* (meaning that it is computationally infeasible to determine  $x$  from  $H(x)$ ) and *collision-resistant* (meaning that given some  $H(x)$  it is computationally infeasible to find a  $y$  such that  $H(y) = H(x)$ ).<sup>7</sup> TESLA is parameterized with a start time  $T_0$ , an interval time  $\Delta$ , and a number of intervals  $N$ . Given these parameters, TESLA chooses a key  $K_N$  at random, and proceeds to compute keys  $K_i = H(K_{i+1})$  until it computes  $K_0$ . The value  $K_0$ , together with  $T_0$  (and, if not specified by the standard,  $\Delta$  and  $N$ ) form the TESLA anchor (somewhat akin to the public key used in PKI authentication).

TESLA specifies that a key  $K_i$  can be released at time  $T_0 + i\Delta$ . Each message must be authenticated with a key that has not yet been released according to the key schedule (because if the key has been released, then anyone could have generated the authenticator). When a node receives a message, if it is authenticated with a key that has already been scheduled for release, the node drops the message as being inauthentic. Otherwise, the node buffers the message and waits for the key's release. When a value  $v$  purporting to be key  $K_i$  is released, it can be checked by any node having an authentic earlier key  $K_j$ , because  $K_j = H^{i-j}(K_i)$ , where  $H^l(x) = H(H^{l-1}(x))$  and  $H^1(x) = H(x)$ . If the key is authentic, then any message that is authenticated using  $K_i$  can be checked. (Note that this property makes the scheme resilient to individual keys lost in transmission.)

To authenticate a message that will be received at some time  $t \in [T_0 + (i-1)\Delta - 2\epsilon, T_0 + i\Delta - 2\epsilon)$ , where  $\epsilon$  represents the maximum skew between any two clocks, the sender computes the HMAC on the message using key  $K_i$ . When the message is received, the receiver's clock is strictly less than  $T_0 + i\Delta - \epsilon$ , so it knows that the key  $K_i$  has not yet been released. It then buffers this message and waits for the key to be disclosed. This happens at time  $T_0 + i\Delta$  on the sender's clock, which is between  $2\epsilon$  and  $2\epsilon + \Delta$  after the message is sent; we call this the *authentication delay*. Because authentication delay is a function of both the clock skew  $\epsilon$  and the interval time  $\Delta$ , the performance of TESLA depends heavily on these values.

A simplified example of TESLA is shown in Figure 1.

In a vehicular network where connections are generally short-lived, it does not substantially decrease overhead to use a TESLA private key for more than around 10 minutes. If the key interval is selected to be 100 ms, then each hash chain would need to be 6000 elements long. A modern CPU can compute 6000 hashes in 6ms; however, if longer chains or more efficient representations are desirable, alternative data structures for hash chains are overviewed in [20].

Since a TESLA key is valid only for a short period of time (e.g. 100 ms), it can potentially be much smaller (or weaker) than general symmetric keys. As discussed in Section 5.2, for the foreseeable future, a 80 bit key appears to provide adequate security during its short lifetime.

<sup>6</sup> HMAC is the name of a specific type of message authentication code based on hash functions; however, to avoid confusion between message authentication codes and medium access control protocols, MAC will mean medium access control protocols, and HMAC will be a general type of message authentication code

<sup>7</sup> This property is called *second-preimage collision resistance* in cryptography.

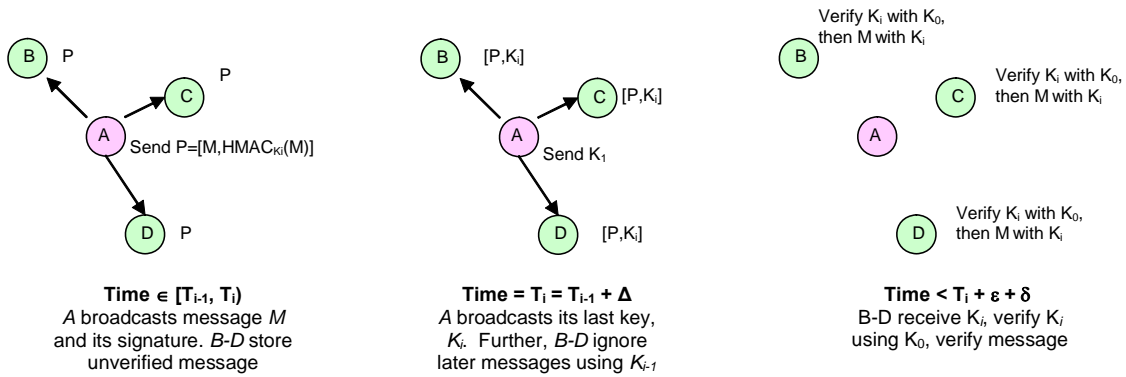


Figure 1 An example of TESLA [18]. First the node  $A$  sends a packet  $P$  containing message  $M$  signed by key  $K_i$ . Then  $A$  sends its key  $K_i$ . The other nodes then verify  $M$  with  $K_i$ .

#### 4 CERTIFICATE DISTRIBUTION

One of the more attractive ways to validate the binding of a vehicle’s ID to its anchor is by having this binding signed by a trusted third-party, or Certificate Authority (CA). Typically this is done using PKI; the Public Key of the CA is widely known. Therefore, if the binding of  $A$ ’s identity with its anchor is signed by a CA (or the CA’s trusted proxy with a *chain of trust* connecting to the CA),  $B$  can trust that messages signed by  $A$  were authentically generated by  $A$ .

To authenticate a message from vehicle  $A$ , a vehicle  $B$  must obtain a valid certificate for  $A$ . In addition to containing  $A$ ’s identity,  $A$ ’s certificate contains the *anchor* data necessary for  $B$  to evaluate  $A$ ’s future signatures. For public key signatures described in Section 3.1.1, this anchor is  $A$ ’s public key  $K^+$ . For TELSAs signatures described in Section 3.1.2, the anchor is the root of the hash chain,  $K_0$ , together with  $T_0$  (and, if not specified by the standard,  $\Delta$  and  $N$ ).

Note that all of the certificate distribution methods described in this section use PKI to connect the chain of trust back to the CA. *This is true whether Vehicle  $A$  subsequently uses PKI signatures (Section 3.1.1) or TESLA signatures (Section 3.1.2) to authenticate future messages.* While both schemes use PKI to distribute certificates, the certificate structure is slightly different. These certificates are shown in Figure 2 and Figure 3. In other words, the certificate exchange of either method uses PKI, and the TESLA certificate is larger than the certificate of Figure 2. However, as shown in Section 6, signatures resulting from the TESLA scheme are much smaller.

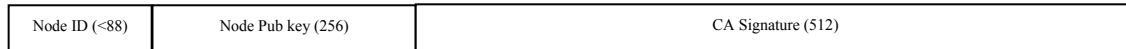


Figure 2 Certificate of PKI strawman. The CA Signature binds the Node ID to the Node’s Public Key. If using the Vehicle Identification Number (VIN), the Node ID will be less than 88 bits. The size and contents of Node’s public key and CA Signature depends on the PKI signature scheme used. P1609.2 dictates 256 bit ECDSA public keys and 512 bit (or larger) CA Signatures.

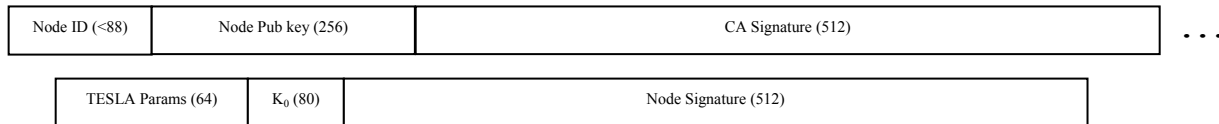


Figure 3 Certificate of TESLA strawman. CA Signature binds the Node ID to the Node’s Public Key. In turn, the Node’s public key is used to authenticate the Node’s anchor, i.e. the TESLA Parameters and the hash chain root  $K_0$ . If using the Vehicle Identification Number (VIN), the Node ID will be less than 88 bits. The size and contents of Node’s public key and CA and Node Signatures depends on the PKI signature scheme used. P1609.2 dictates 256 bit ECDSA public keys and 512 bit (or larger) CA and Node Signatures.

Below we consider several strategies for certificate exchange. However, this section does not attempt to choose among these schemes and their appropriateness for various VANET environments.

##### 4.1 Certificate exchange in P1609.2

When two cars come within radio range of each other, they must exchange certificates in order to ascertain each other’s validity. In the current IEEE P1609.2 proposal, after vehicle  $B$  receives and verifies  $A$ ’s certificate,  $A$ ’s messages are authenticated using the ECDSA signature scheme. However, P1609.2 recommends that a copy of  $A$ ’s certificate be included in every message  $A$  sends<sup>8</sup>. However, we feel that including a signature with every packet is wasteful of bandwidth.

<sup>8</sup> In the P1609.2 case, since each certificate must include an identifier, private key, and a signature, ECDSA certificates can be no smaller than 60 bytes. The

#### 4.2 Generalized Public Key Certificate Exchange

When two cars come within radio range of each other, they must exchange certificates in order to ascertain each other's validity. Periodic messages can be used to detect the presence of vehicles entering this node's wireless range. For example (see Figure 4, when a safety message is received by A from X, A can check to see if it has a valid short-term public key for X. If not, then it is also likely that X does not have a certificate for A. Therefore, A will broadcast its certificate together with the signed short-term public key (or anchor). The message will additionally contain the identifiers of nodes for which A needs a certificate. When X receives this information, X can validate the certificate and cache A's validated certificate so that it can validate future transmissions from A. If X finds its own identifier in A's list of identifiers for which a key is requested, X will also broadcast a certificate.

To reduce the overhead of this certificate exchange mechanism, the certificate broadcast does not take place until the link is determined to be reliable; for example, if a certain number of messages are received from the other vehicle, or if the message was received at a sufficiently high signal-to-noise ratio, the other vehicle will be deemed to be within range. In addition, when two vehicles introduce themselves in this way, any other vehicle within wireless transmission range of these vehicles will have a valid public key for them. Such a vehicle will broadcast its public key, but will not need to request a key from the new neighbor, since it has previously received, verified, and cached a valid public key.

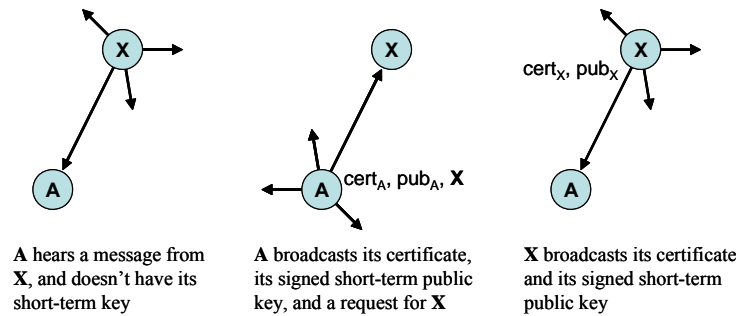


Figure 4 An example of the Generalized Public Key Certificate Exchange. When vehicle A hears a message from a previously unknown vehicle X, it replies with a query for the certificate and short-term public key of X.

#### 4.3 Reduced Bandwidth Certificate Exchange

Compared to the P1609.2 approach in Section 4.1, the approach in Section 4.2 reduces the required bandwidth by only sending a node's certificate once upon each first-encounter. However, this scheme still triggers each vehicle to rebroadcast its certificate each time a new vehicle is encountered. To further reduce certificate-exchange bandwidth, it is possible for each node periodically (1-2 Hz) broadcast its certificate. Therefore, when two vehicles make first contact, each will validate each other's messages only after certificates have been transmitted per a periodic schedule. This periodic scheme can be supplemented by methods to trigger a fast certificate exchange when a high-priority message must be authenticated as soon as possible; such exchanges, however, are unlikely since most messages are urgent only to vehicles in very close proximity, so a periodic certificate transmission should be complete before a node receives an urgent message.

#### 4.4 Infrastructure-Supported Certificate Distribution

Another certificate distribution approach simply loads all of the certificates to be loaded onto the vehicle before it encounters other vehicles. This strategy can minimize delay-until-first-authentication and bandwidth needed for real-time safety messages. For example, vehicles can download certificates of likely-encountered vehicles while parked. Alternatively, when a DSRC Road Side Unit is present, the RSU host can optimize key exchange. Instead of requiring each host to verify certificates for each vehicle that it communicates with, the validation can be performed only at the infrastructure host. The infrastructure host then periodically broadcasts a list of all the valid certificates within range of that host. This list would be signed using the infrastructure host's public key, and such transmission would need to occasionally include a certificate that authorizes this infrastructure, as described in Section 5.1. An infrastructure host may desire to keep all information that it used to generate the key lists, allowing it to address any questions raised about the validity of key lists that it disseminates.

## 5 PROTOCOL DESIGN

### 5.1 Assumptions

Our system relies on a combination of traditional digital signature approaches and lightweight broadcast authentication primitives (such as TESLA [18]). In particular, we use traditional signatures to bootstrap TESLA keys (as described in Section

4); these keys are then used to authenticate the small data messages. This architecture takes advantage of the small authenticators in lightweight broadcast authentication. At any point in time, each vehicle has two keys: its long-term public key (for which it holds a certificate generated by a CA or its proxy), and a short-term signing key. To allow others to verify its short-term signing key, it signs the short-term signing key with the long-term public key (See Figure 3).

We assume that users maintain time synchronization. When each vehicle keeps a clock slaved to GPS, it can ensure that the maximum skew is on the order of a few hundred nanoseconds. However, because of the limitations of GPS, e.g. in tunnels and urban canyons, we also assume that vehicles keep the time on a temperature-compensated crystal oscillator. When two vehicles have not had a GPS lock for one hour, their oscillators should not have skew exceeding 54  $\mu$ s.

We assume that infrastructure may be present in various roadways. For a variety of reasons, such as incremental deployment and low traffic, it seems unlikely that infrastructure will cover all roadways. It is important to assure safety applications can function both with and without this infrastructure. Furthermore, we need to assure that the infrastructure is authorized. This authorization can be based on delegation through the hierarchical government authority. For example, the United States government could have a master key that authorizes infrastructure hosts anywhere in the United States and its protectorates. This key could then be used to delegate the same authorization to the Federal Highway Administration, which would then delegate authorizations to individual infrastructure nodes located on Interstate highways. In addition, the United States master key would delegate authorization to each state's Department of Transportation, limiting the extent of the delegation to the geography covered by that state.

### 5.2 Authentication of Messages and Key Disclosure

In TESLA, each message must include an authenticator (which is the output of a single HMAC). Because the signing keys are inherently short-lived, it is reasonable to allow them to be a shorter length (such as 8–10 bytes). Each message includes an HMAC computed over the message using these short-lived keys, so the authenticator overhead is 8–10 bytes. Whenever a vehicle sends a message authenticated using a key, it will disclose that key at the key disclosure time.

There are two possibilities for choosing a time interval duration  $\Delta$  based on the frequency of data packets and the allowable latency. When several packets are sent within the allowable latency, we set  $\Delta$  roughly equal to the allowable latency. In this case, we can include the key disclosure inside one of the periodic messages. This reduces the MAC overhead, because no additional packets are sent for authentication. Furthermore, the cost of the key disclosure will be amortized over the several packets that are sent using the same key; for example, if each key is used for 10 messages, and the authenticator is 10 bytes long, then each packet will include 11 bytes (amortized) of authenticators. In the other cases, packets are sent at a frequency roughly equal to the allowable latency, or at a rate much lower than the allowable latency. Then we set the time interval  $\Delta$  to the time between messages. Though the authentication latency is between  $2\epsilon$  and  $2\epsilon+\Delta$ , as long as messages are sent close to the end of a time interval, the latency will tend towards the lower end of this range. By keeping the utilization of the wireless medium low, we can keep the MAC latency low. With more-deterministic MAC latencies, we can consistently attempt transmission towards the end of each time interval, which keeps the authentication latency close to  $2\epsilon$ .

### 5.3 Multi-Hop Communications

Some information may be useful across multiple hops. When the originator of the message signs it using a digital signature, the entire message can be passed further along the road, for example by cars moving in the opposite direction. When such relaying is necessary, the system must use asymmetric cryptography, and session keys must be longer lived. Further, when a vehicle A hears another vehicle B first establish a session public key, it must record this message in case A wishes to pass B's information to other vehicles.

## 6 ANALYSIS

### 6.1 The size of signed packets

For ease of presentation, in this section, we set aside the issue of certificate exchange and assume that receiver  $B$  has a validated certificate for sender  $A$ . If so, then  $A$  only needs to send its message as well as a signature to authenticate the message. Consider the example when the message is a 25 byte Heartbeat Message (discussed in Section 2). If using a PKI method to authenticate its messages,  $A$  must send a message similar to that shown in Figure 5. Note that the signature is 256% larger than the message itself.

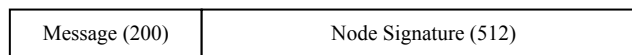


Figure 5 PKI signature case: A 200 bit Heartbeat message along with its 512 bit signature, assuming that signatures are 64 byte ECDSA as proscribed by P1609.2.

However, if using a TESLA signature strategy, each signature is much smaller. Even in the worst case, i.e. when each signature requires the transmission of one TESLA key, the per-packet security overhead is significantly smaller, as shown in

Figure 6.

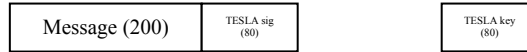


Figure 6 TESLA signature case: A 200 bit Heartbeat message along with its 80 bit signature. In addition, a key is subsequently released to verify the TESLA sig. In some cases, a single TESLA key can be used for multiple TESLA signatures. In this case we assume that signatures and keys are 10 bytes.

As shown in Figure 2 and Figure 3, the TESLA based scheme requires certificates that are 82 bytes longer. However, as shown in Figure 5 and Figure 6, each subsequent TESLA signed message and key require (at least) 44 fewer bytes to transmit. The break-even point occurs when a node sends at least two heartbeat messages for every certificate it sends. If a vehicle broadcasts its certificate once a second using the scheme as described in Section 4.3, and if ten heartbeats are sent per second per vehicle, the TESLA scheme requires  $(44 \cdot 10) - 82 = 358$  fewer bytes of security overhead per second *for each vehicle*. If there are 100 vehicles within wireless interference range of each other, the security overhead savings of the TESLA signature scheme is roughly 286 Kbps.

## 7 RELATED WORK

The spectrum definition in the US for DSRC is specified in [9]. Ongoing related standards and technology evaluations are documented in [10].

Xu [23] and Korkmaz [13] present preliminary ad-hoc protocol designs to support acceptable safety broadcast message reception over a single safety 802.11 DCF channel. Parno and Perrig [24] discuss various security issues in vehicular networks, including some attempts to address privacy. Raya and Hubaux [25] propose the use of public key ECDSA signatures, and show that computation in a vehicular environment is feasible.

## 8 PRIVACY ISSUES

Because each vehicle periodically broadcasts a fixed authenticator and fixed identifier, it can be tracked throughout the road network. This allows substantial privacy compromise, because an adversary can track a user as they drive around, and may even be able to associate a vehicle's identifier with a user identity.

### 8.1 Anonymity with Trusted Third Party

In an anonymous certificate scheme, a participant proves (in zero-knowledge) a user certificate that has been aggregated into some public key. The overhead of such schemes is around one kilobyte per exchange, which must be performed per-neighbor. This anonymous certificate proof can be used to bootstrap a session public key, which is then used to authenticate messages. When using such schemes, because the proof of belonging is zero-knowledge, information cannot be forwarded without loss of integrity.

The trusted third party maintains this public key, and is able to de-aggregate a node if that node is found to be malicious. In vehicular networks, it is unlikely that a node will be provably malicious; rather, a complaint graph can be formed. When a node has high degree in the complaint graph, it can either be evicted from the system entirely, or it can be placed into a special group. Messages coming from a special group will be ignored unless corroborated by a message from outside the special group. These special groups are meant to address situations where a group of bad guys accuses a group of enforcers, and vice versa. When computing the degrees of each node, the third party may find that both the bad group and good group have high degree, but only one of them is necessarily bad. By clumping the bad group together and the good group together and requiring corroboration for either group, the third party can punt on the intractable problem of determining which accusation is actually correct.

Public keys would be distributed through the VANET from a few injection points. A new public key would be issued periodically and be given a sequence number. When two nodes with different public keys first encounter each other, they may choose to perform the exchange at a common (less current) public key, or the node with the newer public key may provide the new (signed) public key to the other node to ensure that the most recent public key is used for the exchange.

Individual user certificates in this scheme can be treated as a pseudonym of a vehicle. In order to help de-correlate pseudonyms from actual identities, previous work has proposed that a vehicle should alternate between multiple pseudonyms. In addition to the reduction of performance inherent in such an approach, we show in the following section that pseudonymous schemes conflict substantially with the necessary function of revocation.

### 8.2 Anonymous revocations

#### 8.2.1 Feasibility of anonymous revocations

In this section, we will examine the mathematical foundations of anonymous revocation techniques. We assume that the certificate authority is subject to government subpoena and that a revocation is issued individually (that is, revocations are not batched). In addition, we assume that each time the certificate authority revokes keys, it disseminates an algorithm that allows each vehicle to detect whether or not a particular key has been revoked. The authority issuing the subpoena is assumed to have



access to this algorithm both before and after the revocation step has been taken, and can therefore compare the effect of that particular (individual) revocation. Keys previously heard from a vehicle can be checked in this way. Many of these assumptions are simplifying assumptions, and our analysis can be generalized to cases where some of these assumptions do not hold.

### 8.2.2 Anonymous Revocation for Single Proven Misbehavior

We compare a “guilty” vehicle to an “innocent” vehicle in a pool of  $N$  vehicles. This pool could consist of all vehicles with VANET capability, or could be more specifically limited to those known to be in the vicinity at the time of the misbehavior. The government will rely on a threshold test to prove its case, checking to see whether or not the number of revoked keys is greater than or equal to a threshold; this threshold can either be a number of keys or a fraction of the known keys. In either case, we denote the probability that a car exceeds this threshold as  $P[\text{exceeds threshold}]$ .

When only the guilty car has any keys revoked, there is no privacy whatsoever, so without loss of generality we examine the case where the probability that an innocent car exceeds the threshold is nonzero; that is, when  $P[\text{exceeds threshold} \mid \text{innocent}] = p > 0$ . We then choose a  $P[\text{exceeds threshold} \mid \text{guilty}] = cp$ , where  $c$  is a constant. To be useful,  $c$  should be greater than 1, so that the probability that the number of revocations exceeds the threshold is greater for the guilty than the innocent.

We earlier said that there is a pool of  $N$  vehicles, so  $P[\text{guilty}] = 1/N$  and  $P[\text{innocent}] = (N-1)/N$ . Then  $P[\text{exceeds threshold}] = (1/N)*cp + ((N-1)/N)*p = p(1+(c-1)/N)$ . We calculate  $P[\text{guilty} \mid \text{exceeds threshold}]$  based on Bayes’ Rule:

$$\begin{aligned} P[\text{guilty} \mid \text{exceeds threshold}] &= P[\text{exceeds threshold} \mid \text{guilty}] * P[\text{guilty}] / P[\text{exceeds threshold}] \\ &= cp * (1/N) / (p(1+(c-1)/N)) = c/(N+c-1). \end{aligned}$$

This equation shows that there is a fundamental tradeoff between fast revocation and anonymity. First, the compromiser of anonymity will choose a definition of threshold that maximizes  $c$ . To provide good anonymity, we need to minimize  $c$  for all possible thresholds. However, to allow a single misbehavior to result in revocation,  $cp$  must approach 1 for reasonably high thresholds. This means that at each such threshold,  $p$  will be approximately  $1/c$ . If we assume that the number of revocations is linear in  $N$ , then  $c$  must be linear in  $N$  (or substantially all users will be revoked in larger networks); however, when  $c$  is linear in  $N$ , then  $P[\text{guilty} \mid \text{exceeds threshold}]$  will be constant, thereby providing little anonymity.

## 8.3 Protocol-Independent Anonymity Issues

### 8.3.1 When latency matters:

The usefulness of information will not be independent of distance; rather, from vehicles that are very close by, information will be less useful because those vehicles are easily perceived by the driver; when information is very far away, it is likewise not useful, because conditions may change before the driver arrives. At 300 meters, information is probably quite useful, so protocol latency matters more at this distance than at longer or shorter distances.

### 8.3.2 Wireless fingerprinting:

Due to imperfections in the RF analog electronics, it is possible to identify nodes based only on the analog properties of the emitted waveform. Experts believe different things about the feasibility of this attack.

### 8.3.3 Movement trends:

Privacy can be compromised simply because of the diurnal movement of each node, and the fact that people do not drive random routes. Previous work has documented the necessity of silent periods in order to disassociate from previous transmissions [19]. As density increases, silent periods can get shorter, but it is unlikely that density will ever increase to an extent that the tracking methods used in this literature will not be applicable. As a result, some risk still remains from wireless eavesdropping.

### 8.3.4 Existing privacy issues:

Driving today is not without its privacy compromises. The existence of constant-valued computer-recognizable license plates allows an adversary with several road cameras to track each car. Electronic toll collection hardware can track users not only at toll collection points, but anywhere there is a reader. Such devices are often used to determine provide real-time traffic status.

## 8.4 Privacy Challenges

Privacy is very desirable in VANET. However, the challenges described in this section are significant, and should be considered when evaluating future VANET privacy proposals.

## 9 CONCLUSIONS

This article explores the challenge of providing strong authentication in VANET. The scheme uses a light-weight scheme using primitives of TESLA and PKI, which are reviewed in Section 3. Time synchronization requirements for VANET nodes are addressed and found to be feasible given current technology. The proposal includes an overview of methods for efficient certification distribution. The proposed scheme significantly reduces the security overhead compared to the current DSRC draft

standard on security (IEEE P1609.2); in a wireless channel shared by 100 vehicles, the security overhead of the proposal saves 286 Kbps over the P1609.2 scheme. Privacy challenges are discussed in Section 8. In particular, a mathematical basis of the tension between anonymity and the ability of certificate revocation is given.

The authors are currently implementing this security scheme in our research test-bed. Results will be forthcoming.

#### REFERENCES

- [1] A Carter, The Status of Vehicle-to-Vehicle Communication as a Means of Improving Crash Prevention Performance, NHTSA Paper No. 05-0264, <http://www-nrd.nhtsa.dot.gov/pdf/nrd-01/esv/esv19/05-0264-W.pdf>, 2005.
- [2] Vehicle Safety Communications Project-Final Report, USDOT HS 810 591, [http://www-nrd.nhtsa.dot.gov/departments/nrd-12/pubs\\_rev.html](http://www-nrd.nhtsa.dot.gov/departments/nrd-12/pubs_rev.html), April 2006.
- [3] NHTSA (2003). Traffic safety facts. Report DOT HS 809 767, <http://www-nrd.nhtsa.dot.gov>.
- [4] Press Release, *U.S. Transp. Sec. Mineta Announces Opening of Crash Preventing "Intelligent Intersection" Test Facility*, <http://www.its.dot.gov/press/fhw2003.htm>, Jun 24, 2003.
- [5] J. Paniati (Dir., ITS, U.S Dept. of Transp.), Intelligent Safety Efforts in America, 10th ITS World Conf. <http://www.its.dot.gov/speeches/madridvii2003.ppt>, Nov. 17, 2003.
- [6] CARE (2004). Community road accident database, <http://europa.eu.int/comm/transport/care/>.
- [7] Toyota (2004). Toyota safety: Toward realizing zero fatalities and accidents, [http://www.toyota.co.jp/en/safety\\_presen/index.html](http://www.toyota.co.jp/en/safety_presen/index.html).
- [8] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle System – 5GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ASTM E2213-03, Sep. 2003
- [9] USFCC, Report and Order, FCC 03-324, Dec. 2003.
- [10] Minutes of IEEE DSRC Standards Group meetings, <http://www.leearmstrong.com/DSRC/DSRCHomeset.htm>.
- [11] ITS Standards Program , <http://www.standards.its.dot.gov/StdSummary.asp>.
- [12] C. Robinson, L. Caminiti, D. Caveney, K. Laberteaux, Efficient Coordination and Transmission of Data for Cooperative Vehicular Safety Applications, *Proceedings of VANET 2006*, Sep. 2006.
- [13] G. Korkmaz, E. Ekici, F. Ozguner, U. Ozguner. Urban Multi-Hop Broadcast Protocol for Inter-Vehicle Communication Systems, In Proc. of the 1st ACM Workshop on Vehicular Ad-hoc Networks, October 2004, Philadelphia, USA.
- [14] Rivest, R., Shamir, A., and Adelman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM* 21(2):120-126, 1978.
- [15] T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Trans. Information Theory*, IT-31(4):469-472, 1985.
- [16] National Institute of Standards and Technology. Digital Signature Standard, FIPS Pub 186-2, Jan 2000.
- [17] Accredited Standards Committee X9. Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA). American National Standard X9.62-2005, Nov 2005.
- [18] Perrig, A., Canetti, R., Tygar, J.D., and Song, D. The TESLA Broadcast Authentication Protocol, *RSA Cryptobytes* 5(2):2-13, 2002.
- [19] Hu, Y. and Wang, H.J. Location Privacy in Wireless Networks. *Proceedings of the ACM SIGCOMM Asia Workshop 2005*, ACM, Beijing, China, April 2005.
- [20] Hu, Y., Jakobsson, M. and Perrig, A.. Efficient Constructions for One-Way Hash Chains. *Proceedings of Applied Cryptography and Network Security 2005*, New York, New York, July 2005.
- [21] Schneier, B. *Applied Cryptography*. John Wiley & Sons, 1996.
- [22] Private correspondence with Scott Andrews (Cogenia Partners), 2005.
- [23] Q. Xu, Vehicle-to-Vehicle Messaging in DSRC, First ACM Workshop on Vehicular Ad Hoc Networks(VANET), 2004.
- [24] Parno, B. and Perrig, A. Challenges in Securing Vehicular Networks. *Workshop on Hot Topics in Networks (HotNets-IV)*, 2005.
- [25] Raya, M. and Hubaux, J.P. The security of vehicular ad hoc networks. *Workshop on Security of ad hoc and Sensor Networks*, 2005.
- [26] J. Paniati (Dir., ITS, U.S Dept. of Transp.), *Intelligent Safety Efforts in America*, 10<sup>th</sup> ITS World Conf. <http://www.its.dot.gov/speeches/madridvii2003.ppt>, Nov. 17, 2003.