TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs

Ahren Studer[†], Elaine Shi[†], Fan Bai[§], & Adrian Perrig[†]
[†] Carnegie Mellon University [§] General Motors
{astuder, rshi, perrig}@cmu.edu fan.bai@gm.com

Abstract*—Vehicular Ad Hoc Networks (VANETs) require a mechanism to help authenticate messages, identify valid vehicles, and remove malevolent vehicles. A Public Key Infrastructure (PKI) can provide this functionality using certificates and fixed public keys. However, fixed keys allow an eavesdropper to associate a key with a vehicle and a location, violating drivers' privacy. In this work we propose a VANET key management scheme based on Temporary Anonymous Certified Keys (TACKs). Our scheme efficiently prevents eavesdroppers from linking a vehicle's different keys and provides timely revocation of misbehaving participants while maintaining the same or less overhead for vehicle-to-vehicle communication as the current IEEE 1609.2 standard for VANET security.

Keywords: Key Management, Vehicular Ad Hoc Networks, Revocation, Privacy

I. Introduction

In Vehicular Ad Hoc Networks (VANETs), vehicles are equipped with sensors and wireless communication devices, allowing vehicles to sense traffic and road conditions, and warn other nearby vehicles about potential emergency situations and traffic jams. VANETs present a promising approach to reduce the 43,000 traffic fatalities and $260 billion spent annually on traffic-related health care in the US [10], [19]. In addition to helping prevent accidents, VANETs also provide convenience and business services that will help improve a driver's experience [1].

In VANETs, a vehicle's *On Board Unit (OBU)* communicates with other vehicles' OBUs and fixed infrastructure called *Road Side Units (RSUs)*. For VANETs to operate securely and reliably, participants needs to validate received messages; otherwise, an attacker can easily inject bogus messages to disrupt the normal operation of VANETs. To allow authentication, we need to build key management mechanisms that allow senders to establish and update keys for security-sensitive operations.

While RSUs can utilize traditional Public Key Infrastructure approaches, designing an OBU key management mechanism for secure VANET operation turns out to be a surprisingly intricate and challenging endeavor, because of multiple seemingly conflicting requirements. Recipients need to authenticate OBUs that they communicate with; and road authorities would like to trace drivers that abuse the system. However, VANETs need to protect a driver's privacy. In particular, drivers may not wish to be tracked wherever they travel.

Ideally, an OBU key management mechanism should provide the following desirable properties:

Authenticity. VANET participants need to authenticate legitimate OBUs and messages from those senders.

Privacy. RSUs and wireless eavesdroppers should not be able to track a driver in the long term. Authorities can already track vehicles through cameras and automatic license-plate readers. However, VANETs should not make such tracking any simpler by repeatedly broadcasting identifying information about the vehicle. The privacy requirement is seemingly contradictory to the authenticity requirement: if each OBU presents a certificate to vouch for its validity, then eavesdroppers can link any use of that certificate back to the OBU and thus the vehicle.

Short-term Linkability. For privacy, an eavesdropper should not be able to link messages from the same OBU in the long-term. However, as we explain in Section II, some VANET applications require that in the short-term, a recipient be able to link two messages sent out by the same OBU.

Traceability and Revocation. An authority should be able to trace an OBU that abuses the VANET. In addition, once a misbehaving OBU has been traced, the authority should be able to revoke it in a timely manner. This prevents the misbehaving OBU from causing any further damage.

Efficiency. OBUs have resource-limited processors to make VANETs economically viable. Therefore, the cryptography used in VANETs should incur limited computational overhead.

We propose Temporary Anonymous Certified Keys (TACKs), an efficient OBU key management system which meets all of these requirements. In the TACKs system, roadways are divided into geographic regions with *Regional Authorities (RAs)* acting as certificate authorities for their region. Within a region, an RA certifies OBU generated temporary keys which are used for authentication. As traffic enters a region, each OBU anonymously requests a certificate from the RA. If the requesting OBU has not been revoked, the RA responds with a certificate. Since all OBUs entering the region change keys simultaneously, the TACK update provides unlinkability between prior and current keys, similar to the privacy provided in MIX networks [7].

Contributions. The contributions of this work include the following: 1) We identify the properties that an OBU key

This research was supported by CyLab at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office, and by General Motors through the GM-CMU Collaborative Research Laboratory. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, GM, or the U.S. Government or any of its agencies.

management scheme should provide. 2) We propose a scheme called TACKs that achieves all of the properties. Although TACKs are based on a combination of standard techniques, combining these techniques to provide an economically viable solution for OBU key management is a challenging task. 3) We analyze and simulate TACKs in realistic settings and show that TACKs represent a practical OBU key management solution.

II. Problem Definition

VANETs require an OBU key management scheme that fulfills a number of properties. Before defining the properties and stating assumptions, we define the following notation for the four sets of VANET participants:

M: A managing authority acting as the root of trust. This is the Certificate Authority/Authorities of the VANET Public Key Infrastructure (VPKI), and could be a Department of Motor Vehicles (DMV) or some commercial entity (e.g., Verisign). To avoid a single point of trust, multiple entities may jointly act as the authority.

R: The set of valid Regional Authorities. These RAs act as intermediary authorities in the VPKI and can grant vehicles temporary region-specific certificates. An authority issues certificates to RAs, and certifies them as valid intermediary authorities.

V: The set of valid OBUs. Any OBU with a valid certificate from M or a region-specific short-lived certificate from R (while in the proper region) is considered part of V.

$\overline{V}$: The set of expired/revoked OBUs. In TACKs, any OBU listed in the authority's current Certificate Revocation List (CRL) that does not have a certificate from some member of R is a member of $\overline{V}$.

A. Requirements for OBU Key Management

Due to the unique characteristics of VANET, we identify the following properties necessary for an OBU key management scheme.

Sender validity and message integrity. In VANETs, a recipient[1] should be able to verify that a message came from a valid OBU, i.e., a member of the set V. In addition, the recipient should be able to verify that the message has not been tampered with in transit.

Sender validity and message integrity are also referred to as *authenticity* in this paper. Authenticity prevents malicious outsiders from injecting bogus messages that might disrupt the normal operation of the VANET.

Short-term linkability. When the same sender sends two or more messages within a small time frame Δt, a recipient should be able to verify that these messages came from the same sender. We would like to enforce short-term linkability in a way such that a malicious OBU cannot launch a Sybil attack [9] where a single OBU impersonates multiple vehicles. Short-term linkability is a desirable property in several VANET applications [13]. For example, one promising VANET safety application is to help drivers decide when it is safe to change lanes. This can be achieved by having OBUs frequently broadcast beacons with their current location, speed, and acceleration. A receiver uses these beacons to build a map of OBUs nearby and predict if changing lanes will cause an accident. In this application, an OBU needs to be able to identify which messages come from the same sender. A malicious OBU might attempt to disrupt this application by impersonating multiple OBUs. Sybil attacks like this should not be possible.

Short-term linkability does not hurt drivers' privacy. Vehicles' mobility patterns are constrained by roads and other vehicles. If a vehicle is detected at some location X at time t, then at $t + \Delta t$ (where Δt represents a small time increment), the vehicle must be in the vicinity of location X. Therefore, being able to track a vehicle in the short-term does not impact drivers' privacy.

Long-term unlinkability. A basic privacy requirement is that an observer cannot link messages sent by a OBU to the driver's name, license plate number, or other personally identifying information.

More specifically, if the same OBU sends two messages m and m' more than Δt time apart, then an adversary should not be able to determine if m and m' originate from the same sender based on message contents and where the messages were received. In particular, this implies that if we use digital signatures to ensure authenticity, certificates should lack identifying information and the keys should change in such a way that an eavesdropper cannot associate an old key with a new key. Tracking based on RF fingerprinting or knowledge of a driver's route are outside the scope of this paper.

Traceability and revocability. If an OBU misbehaves, an authority should be able to trace the identity of the misbehaving OBU from a transcript of the messages it has sent. In addition, the authority should be able to efficiently notify the VANET of the misbehaving OBU and revoke the OBU's identity. Formally, let O denote an OBU found to be misbehaving, revoking O means removing O from the set V and adding it to $\overline{V}$: $V \leftarrow V \backslash \{O\}$, $\overline{V} \leftarrow \overline{V} \cup \{O\}$. After O has been revoked, recipients in the VANET will no longer accept O's messages.

Efficiency. For economic viability, OBUs possess resource-limited processors. To ensure efficient VANET operation, OBUs' necessary cryptographic operations should be lightweight.

B. Assumptions

For TACKs we assume: 1) a trusted authority to manage distribution of privacy preserving keys to OBUs and to certify RAs, 2) OBUs have inexpensive hardware while RAs have greater computational power, and 3) communication coverage exists to allow OBU certificate update and revocation distribution to RAs.

We require an authority to act as the root of trust for the VANET. A trusted entity such as a Department of Motor Vehicles (DMV) or Department of Transportation (DoT) would handle key generation, certification, and distribution in VANETs.

[1] The recipient can either be an OBU or an RSU.

In TACKs, we need trusted authorities to perform mainly two tasks: 1) distributing private long-term privacy preserving keys to OBUs which uniquely identify each OBU; and 2) issuing certificates to RAs and defining regions. The trusted authorities that perform these two tasks are not necessarily the same entity. In practice, to prevent a concentration of trust, we can divide the computation needed to complete a single authority's role. Splitting the role of the group manager into multiple entities can be achieved through standard cryptographic techniques such as secure multi-party computation [8].

We assume RAs are part of a traditional Elliptic Curve Digital Signature Algorithm (ECDSA) based PKI, where an RA's certificate identifies it as a valid RSU RA at a fixed location or ties a given online RA to a region.[2] This type of PKI is commonly assumed in other works on VANET security [21]. In our work, RAs act as authorities for the region near them, so OBUs must be able to link RA-signed certificates back to an RA to determine if that certificate is valid for the current region. The root (e.g., USDoT) would sign state/province certificates, which in turn sign local certificates, and so on. Finally, road authorities sign RAs' certificates which identify the public key of the RA and the position of the RSU RA or the authoritative region of an online RA. Maps (similar to those in current GPS navigation systems) will include metadata about regions' boundaries and how an OBU can contact the appropriate RA for a region (via VANET communication for RSU RAs or a URL for online RAs). OBUs can periodically (e.g., weekly) download authority-signed Certificate Revocation Lists (CRLs) that define which RAs are no longer valid.

We assume that OBUs have relatively slow processors to help reduce vehicle cost. In comparison, RAs have more computational resources. Therefore, if possible, computationally intensive operations (such as the OBU revocation check operation in TACKs) should be offloaded to the RAs.

We assume RSU RA deployment or communication coverage such that OBUs can contact at least one RA when entering a region or requesting a certificate. When away from RSU-based RAs, cellular services integrated into vehicles (e.g., GM's OnStar^{TM}or BMW Assist^{TM}) or WiMax could provide a connection to online RAs. RAs require a means to receive updated revocation information from authorities. Online RAs are reachable via the Internet. RSU-based RAs could connect to the authority through a wired Internet connection or receive data over radio or satellite connections. Given that RSUs act as authorities in a region, we also assume the RSUs are robust to physical tampering. We are not assuming expensive tamper-proof hardware. Instead, a locked box may suffice (similar to traffic light controllers today). Even if attackers manage to compromise an RSU, their actions are limited to that region. Once authorities detect the compromise and OBUs download the relevant revocation information, the stolen keys will be useless. An attacker with RSU keys can issue multiple certificates for the RSU's region and remove any record of previous certificate requests. Even though the attacker gains control of the RSU in that region, such an attacker is unable to track vehicles, generate certificates for other regions, etc.

III. Prior Work

Several prior works have examined OBU key management However, TACKs is the first work to address all of the properties listed in Section II.

Early works proposed installation of numerous authority provided public/private key pairs on an OBU [15], [21]. Since each key is used for a short period of time and the authorities know which OBU possesses which keys, these schemes provide authentication, short-term linkability, traceability, and efficiency. However, an OBU can use multiple keys at the same time, allowing Sybil attacks. The schemes also lack efficient revocation since revoking one OBU requires the use of an expensive secure coprocessor on the revoked OBU, or the distribution of revocation information about hundreds of keys to all VANET participants. Later we discuss how periodically switching keys alone fails to provide long-term unlinkability.

Other works have proposed using group signatures within VANETs. Section IV-A contains background on group signatures. Boneh et al. [2] proposed OBUs generate a group signature for every message broadcast to provide authentication. Group signature's anonymity property makes this the only key management scheme with long-term unlinkability without requiring OBUs to change keys. However, this level of anonymity removes any short-term linkability. In addition, group signatures are computationally expensive, making frequent use of group signatures infeasible on OBU hardware. Calandriello et al. [4] suggested OBUs use group signatures to sign certificates for temporary traditional asymmetric keys. This ensures short-term linkability, but allows for Sybil attacks where an OBU generates multiple concurrent certificates. The technique is also computationally expensive in that OBUs must verify group signatures and check if the group signature is from a revoked OBU. Lu et al. [17] suggested using RSUs as the source of certificates. In such an approach, RSUs (as opposed to OBUs) check the group signature to verify if the sender has been revoked and record values to allow tracing. OBUs then use a RSU provided certificate to achieve authenticity and short-term linkability. However, their scheme is vulnerable to Sybil attacks and requires an unreasonable amount of computation for RSUs (i.e., linear in the size of the revocation information for every certificate request).

Gerlach [12] & Sampigethaya et al. [23] have shown that multiple OBUs need to simultaneously change keys to provide long term linkability. Their solution is to have OBUs communicate to determine when to update keys and ignores other aspects of OBU key management. By using Regional Authorities, TACKs has the advantage that OBUs automatically change keys when entering a new region, providing long-term unlinkability without requiring explicit communication.

[2] RSUs that provide some VANET service, but do not generate certificates, are also part of this PKI.

IV. Temporary Anonymous Certified Keys (TACKs)

At a high level, the TACKs system operates as follows. An OBU signs broadcast messages using a public/private key pair. These signatures ensure message integrity and short-term linkability since only the owner of the private key can generate a signature and that OBU uses a single key pair within a short period of time. An RA provided short-lived certificate identifies the owner of the corresponding key pair as a valid OBU. The OBU anonymously proved to the RA the OBU was a member of V (the set of valid OBUs) to acquire the certificate. Note the short-lived key used to authenticate messages is a Temporary Anonymous Certified Key or *TACK*. To prove validity without revealing identifying information, the OBU uses an authority (M) provided group key to generate a group signature. We discuss group signatures and state their properties in Section IV-A. When the RA provided certificate (also refereed to as a *TACK certificate*) expires or the OBU leaves the region corresponding to the current RA, the OBU must prove to the appropriate RA it is a valid OBU and request a new certificate in what we call a *TACK update*. When a set of OBUs enters a region with a new RA, OBUs in the set will perform a TACK update in an anonymous fashion, such that eavesdroppers and certifying RAs cannot link an old TACK for a given OBU in the set with the OBU's new TACK. If the owner of a TACK is found to have abused the VANET, M can de-anonymize the certificate request corresponding to the TACK and determine the offending OBU. M computes a revocation token corresponding to the offending OBU's private group key and publishes the token to the RAs. This token allows RAs to determine if a revoked OBU is requesting a certificate without learning not yet revoked OBUs' identities. Only when the requesting OBU has not yet been revoked will RAs sign a TACK certificate.

In the remainder of this section, we provide some background on group signatures, define the notation we use, and describe the different aspects of our scheme: long-term key distribution, TACK generation and certification, TACK usage, TACK tracing, and long-term key revocation.

A. Preliminaries and Notation

Group Signatures. Chaum and van Heyst [6] first introduced group signatures. In contrast to normal signatures, group signatures protect the signer's anonymity. A trusted entity (usually referred to as the *group manager*) assigns to each valid member of the group a *group user key*. This group user key allows a group member to sign a message and produce a group signature. Anyone can verify a group signature using the group's public key. A group signature reveals no information about the signer's identity; and only the group manager can trace the identity of the signer from a group signature.

In our system, we need a group signature scheme that provides *tracing* and *revocation*. The group manager can trace the identity of the signer from the group signature, and henceforth revoke that user from the group. We use Verifier-Local Revocation (VLR) [3]. In VLR, the group manager computes and publishes a revocation list RL consisting of a revocation token for each revoked member. When verifying a group signature, the verifier tests the group signature against all revocation tokens in RL, to check if the signer has been revoked. If the signer has been revoked, the verifier rejects the signature. We use Boneh and Shacham's group signature construction [3] because it is one of the most efficient constructions known and it supports revocation and tracing.

Notation.

gSign	group members' algorithm to generate a group signature
gVerify	algorithm for verifying a group signature
guk	an OBU's group user key
gpk	group public key
gmk	group master key, owned by the group manager
RL	revocation list
grt	a token in the revocation list
(K_S^{-1}, K_S^{+})	an OBU's TACK pair: K_S^{-1} is the private key, K_S^{+} is the public key

TABLE I
Notation used in the remainder of the paper.

B. Distribution of Long-term Keys

In the TACKs system, each valid OBU has a group user key that is unique to that OBU. This group user key is issued by a trusted group manager (M). This key is stored in the OBU and remains stable over a long period of time, e.g., between annual vehicle inspections. M first initializes the group signature scheme by calling the group key setup algorithm, to generate a group public key gpk and a group master key gmk. It publishes gpk and retains gmk itself.

To issue a group user key, M generates the key (guk_i) and sends it to V_i. M also maintains a history of all key/OBU pairs it has issued, so that it can later trace misbehaving OBUs.

C. Authenticating Other OBUs

In VANETs, OBUs broadcast messages to communicate with each other. To allow OBUs to authenticate each other in a broadcast environment, a sender can sign each message using the sender's TACK private key K_S^{-1}, and periodically broadcast the RA signed certificate of its TACK public key K_S^{+}. Receivers know the time and the sender's region and the associated RAs, allowing verification that a valid RA certificate was used. A sender could use the TACK to bootstrap a more efficient broadcast authentication mechanism (e.g., TESLA [14], [20]). The remainder of this section discusses how OBUs anonymously acquire certificates from RAs and how an authority can track and revoke misbehaving OBUs.

D. TACK Certificates

In TACKs, RA generated certificates identify valid OBUs. RA generated certificate are only valid for a short period of time while in the region associated with the RA. The short lifetime ensures the timely removal of revoked OBUs from the VANET. Once revoked, an OBU's requests for a new RA generated certificate will fail. To ensure that TACKs expire after a certain period of time (e.g. every few minutes), the RA includes an expiration time when it signs a certificate. A shorter certificate lifetime provides faster removal of revoked OBUs, but more frequent certificate requests and a greater impact on applications. Once VANET applications are better understood a study is needed to determine the optimal lifetime to balance these factors.

Limiting an RA's authority to a geographic region and forcing an OBU to change certificates when entering a new region helps provide long-term unlinkability. A set of vehicles entering a new region has to change certificates simultaneously, preventing an eavesdropper from tracking an OBU. Section II-B discusses our assumptions which ensure an OBU knows its location and can use map metadata to learn the set of valid RAs for the current region and how to contact them when a new certificate is needed.

Updating a TACK. When an OBU enters a region for which it does not have a valid certificate or when the old certificate expires, an OBU must update its short-lived TACK with an RA. Figure 1 contains the steps associated with a TACK update. First, the OBU picks a fresh public/private key pair (K_S^+, K_S^{-1}) at random from the key space. This key pair can be any type of key pair, e.g., an ECDSA key pair as defined by IEEE 1609.2 [16]. Next, the OBU uses its group user key (guk_i) to sign K_S^+ (i.e., K_S^+ is the message being signed), and sends the resulting group signature σ and temporary public key to the appropriate RA. σ proves that the signer is a valid OBU, without revealing the identity of the OBU.

On receiving the certificate request, the RA uses the group signature, the group public key, and revocation list (RL) to verify the signature and check if the requester has been revoked. If the OBU and signature are valid, the RA signs a certificate for the OBU's TACK public key K_S^+, using the RA's secret signing key K_{RA}^{-1}. Next, the RA records the pair (σ, K_S^+) to allow the group manager to track misbehaving OBUs (see Section IV-E). After queueing up all of the certificate requests for a given region within the last δ seconds, the RA broadcasts the resulting certificates to the OBUs. In Section V, we discuss how this delay improves long-term unlinkability.

```
Updating an $(K_S^+, K_S^{-1})$ pair:
1. OBU        : $(K_S^+, K_S^{-1}) \xleftarrow{R}$ key space
2. OBU        : $\sigma \leftarrow \mathsf{gSign}(\mathsf{guk}_i, \mathsf{gpk}, K_S^+)$
3. OBU → RA : $\sigma, K_S^+$
4. RA          : $b \leftarrow \mathsf{gVerify}(\mathsf{gpk}, \mathsf{RL}, \sigma, K_S^+)$
5. RA          : if $b = 0$ then exit
6. RA          : $\mathsf{cert} \leftarrow \mathsf{sign}_{K_{RA}^{-1}}(K_S^+||\text{expiration})$
7. RA          : Add $(\sigma, K_S^+)$ to history table
8.  (at most $\delta$ seconds later)
9. RA → OBU : $\mathsf{cert}$
```

Fig. 1. **Protocol for updating TACKs.** *Refer to Table I for notations.*

Efficient revocation check. In group signature schemes with verifier-local revocation, the verifier (in our case, the RA) keeps a revocation list (RL). RL contains a revocation token $\mathtt{grt}_i$ associated with each revoked OBU ($V_i \in \overline{V}$).

Under Boneh and Shacham's original construction, when the RA verifies a group signature, it needs to check the signature against every token in the revocation list. Hence, the signature verification cost is linear with respect to the size of the revocation list. In TACKs, the long-term keys may be used for up to one year; and during this time period, millions of vehicles may have been revoked. In this case, $O(|\mathsf{RL}|)$ verification cost is too expensive.

To reduce computation, we can use the method proposed by Boneh and Shacham for a more efficient revocation check (see Section 7 of Boneh's work [3]). Restricting the randomness in the signing algorithm maintains the security and anonymity of group signatures and allows verifiers to pre-compute values, such that each revocation check requires a constant number of operations. We divide time into epochs and OBUs are forced to use a function of the current epoch and the RA to generate inputs when generating a group signature. At the beginning of each interval, the RA will perform $O(|\mathsf{RL}|)$ pre-computations, rather than having to perform $O(|\mathsf{RL}|)$ for each request. During periods of low-demand, the RA can utilize idle processor cycles to pre-compute the necessary values rather than waiting until the start of an interval. In this way, verifying a group signature requires only $O(1)$ operations.

Defense against Sybil attacks. A malicious OBU might try to obtain multiple TACK certificates from an RA to impersonate multiple vehicles. Incidentally, the efficient revocation check technique also allows us to defend against the Sybil attack.

By fixing the random numbers used during group signature generation for the same RA during the same time epoch allows us to achieve the following properties:

P1. If an OBU sends two requests for TACK certificates to the same RA within a single epoch, the RA can use the fixed numbers to link the two requests to the same OBU.

P2. If an OBU sends two requests for TACK certificates in different time epochs or to different RAs, these requests are completely unlinkable.

P1 prevents a malicious OBU from requesting multiple TACK certificates at the same RA within the same time epoch. **P2** guarantees legitimate OBUs' anonymity.

E. Tracing and Revocation

When an OBU with TACK public key K_S^+ misbehaves, police (or another trusted entity) can retrieve the group signature σ associated with that $K_S^+$from the RA. The police can then request that the group manager trace and revoke the signer of the group signature σ.

To determine which OBU generated a signature σ, the group manager uses a tracing algorithm, which tests σ against the group user keys of OBUs in the set V. Once M identifies V_i as the misbehaving OBU, M adds a revocation token $\mathtt{grt}_i$ tied to V_i to the current revocation list RL, and distributes the new RL to the RAs.

In Section V, we analyze how TACKs meets the properties set out in Section II. In Sections VI & VII, we investigate if TACKs is efficient enough to operate under real world constraints.

V. TACKs Analysis

In this Section we discuss how TACKs meet the requirements set out in Section II.

Sender Validity. When an OBU requests a certificate from an RA, the RA verifies the group signature and confirms that authorities have not revoked the OBU before returning a TACK certificate. There is a small window of time between when an OBU was revoked and when its TACK certificate expires that allows a revoked OBU to participate in the VANET.

Message Integrity. Provided the underlying cryptography is secure, digital signatures generated using TACK private keys and appended to messages ensure message integrity.

Short-term linkability and Sybil prevention. As an OBU uses the same TACK over a short interval, any messages signed by that TACK can be linked to each other.

A malicious OBU cannot perform a Sybil attack and impersonate arbitrarily many OBUs at the same time. As explained in Section IV-D, during a time epoch T_i, an OBU can only obtain one TACK certificate from an RA for a region.

An attacker who has acquired long-term private keys from multiple OBUs may request multiple TACK certificates from an RA. However, this is equivalent to multiple conspiring vehicles since there still is a one-to-one correspondence between keys and vehicles. In addition, an attacker may request certificates from multiple RAs where each RA controls a different region. However, such an attacker's damage is limited, as the attacker can only use a TACK in its corresponding region.

Long-term unlinkability. To protect drivers' privacy, we require that messages sent by the same vehicle be unlinkable in the long-run. Group signatures and region-based certificates provide long-term unlinkability in TACKs.

Group signatures allow vehicles to anonymously prove their validity to RSUs. However, cryptography alone does not provide a defense against the *correlation attack*. In a correlation attack, an attacker tries to track vehicles by observing the spatial and temporal correlations between different keys. For example, if only a single OBU changes keys at a time, an eavesdropper can associate the new key with the old key. One way to defend against the correlation attack is to have multiple vehicles coordinate their key updates [12], [23]. If numerous vehicles in a physical space update their keys at the same time, an observer can associate the set of old keys that disappeared with the set of new keys that came into use. However, the observer is unable to associate an old key with a specific new key. Prior works have studied coordinated key update techniques, but require explicit communication between vehicles to coordinate key updates [12], [23].

Unique to our work are certificates that are only valid around the RA which signed the certificate. These region-based certificates force OBUs to request a TACK certificate whenever they enter a region, ensuring coordinated key updates without explicit communication, while still providing a MIX function [7]. When a number of vehicles enter a new region, the OBUs send certificate requests and do not sign any new messages until receiving the RA's responses. Even though the requests are not encrypted, the group signatures provide anonymity. Once the RA responds with certificates, OBUs will start signing messages with the corresponding keys. If an eavesdropper is tracking a vehicle, after a key update the eavesdropper will only know that the victim OBU is a member of the set of OBUs which updated keys, but not know exactly which one. Eavesdroppers can correlate vehicle announced location and velocity to help track a specific vehicle in a set of certificate requesters, but if the silent period is on the order of a couple of seconds it is difficult for an attacker to associate the old key with the new key based on radio messages alone [23].

We can measure the level of anonymity TACKs provides a vehicle based on how many OBUs simultaneously change keys a.k.a. the anonymous set size [5]. Traffic models often use a Poisson distribution with a rate of $\lambda = [0.5, 0.8]$ to describe the number of vehicles that drive along a highway [27]. If an RA waits δ seconds between certificate responses (i.e., batching responses for δ seconds), we can describe the number of vehicles that enter a new region and change keys simultaneously as $X \sim \mathsf{Poisson}(\delta \cdot \lambda)$. When an OBU drives across a region boundary and acquires a new certificate, X_1 OBUs update keys and generate an anonymous set of size X_1. If vehicle i from the set exits the region and updates certificates away from other members of the set, $X_{1,i}$ OBUs change keys simultaneously, adding $X_{1,i} - 1$ more entities to the anonymous set size (OBU i was already in the set). Using the rule of iterated expectations, we find the expected anonymous set size after an OBU changes regions n times is $(\delta \cdot \lambda)^n$. As a lower bound, if the OBUs enter and leave the region together, the second key change provides no increase in the anonymous set size and the anonymous set size remains at X_1.

The selection of the maximum RA certificate response delay (δ) presents a need to balance privacy and availability of the VANET. With a large δ, the anonymous set size is larger, but OBUs cannot generate messages until they receive a new certificate. If δ is small OBUs will lack privacy since the anonymous set size will be small. The appropriate value of δ depends on the balance between users' privacy desires and the acceptable time without periodic messages for safety applications.

Traceability and revocability. Authorities require a scheme that allows *Traceability* and *Revocability*. Using the tracing algorithm of the underlying group signature scheme, the group manager and the certifying RA can collaborate to identify the OBU which requested a certificate. The group manager can then revoke the misbehaving OBU by computing and announcing a revocation token for that OBU. When an RA receives a new revocation token, it appends the token to the revocation list RL.

Operation	Comp. Time	Data Size
OBU Group Sig. Creation	$320ms$	228 bytes
RA Group Sig. Verify	$36ms$	228 bytes
RA Creation of Certificate	$3.2ms$	28 bytes

TABLE II
ESTIMATED COMPUTATION TIME AND SIZE OF TACK RELATED CRYPTOGRAPHY FOR A 3.2GHZ RA OR A 400MHZ OBU.

When verifying future group signatures, the RA will check the group signatures against the revocation list RL to make sure they come from valid OBUs that have not been revoked.

Efficiency. In the TACKs system, the most expensive operation is for an OBU to update its short-term key with an RA. This step requires that the requesting OBU sign a group signature, and that the RA verify the group signature. We may assume that the RA has abundant computational resources (e.g., with several GB of RAM and a GHz processor). In contrast, the OBU has limited processing power (e.g., a 400MHz processor [21]). Here we discuss the computation and bandwidth of a TACK certificate request. Section VI contains simulation of TACKs in real traffic scenarios.

Boneh and Shacham's group signature scheme [3] requires the use of bilinear groups, also referred to as pairings. Several types of pairings can be used with trade-offs between size and computation cost. In TACKs, the major concern is the computational overhead of signature generation. We assume the use of type A pairing in TACKs since they are the pairings fastest to compute [18].

Two recent works estimate the performance of running type A pairings on a modern workstation and ECDSA on a memory-constrained 400MHz machine [21], [25]. Table II contains estimated timing based on these works that are relevant to TACKs. We assume that RAs have 3.2GHz Pentium 4 processors with two gigabytes of memory. OBUs have less computational power and memory to help reduce the added cost to vehicles. The results assume that RAs use the efficient revocation check method described in Section IV-D. Moreover, the verification time does not include pre-computation.

VI. TACKS SIMULATION WITH RSU RAS

We use ns-2 [26] to simulate TACKs with RSU RAs in highway and city settings. In Section VII we analyze the use of online RAs. Our goal is to determine if OBUs can successfully update certificates when bandwidth and computation are constrained and how much bandwidth a certificate update consumes. To represent city traffic we use a traffic scenario generator [22] and the 3 kilometer square (9km^2) city topology from Dallas, Texas presented in Figure 2 (a). Our simulated 4 kilometer long 4-lane highway loop is presented in Figure 2 (b). In the simulation, each OBU has a 300 meter broadcast range and broadcasts two signed beacons every second with the OBU's location and speed. These beacons are used for safety applications, and are included to represent realistic VANET channel usage. RSU RAs have the same radio range and wait $\delta = 2$ seconds between responding to certificate requests. This small δ allows OBUs to start using certificates sooner, allowing more OBU beacons and increasing channel contention. First, we describe our simulation environment and the measured quantities.

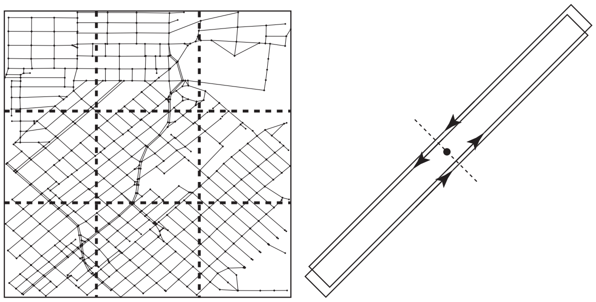

(a) City Topology (b) Highway Topology

Fig. 2. Topologies Used During Simulations

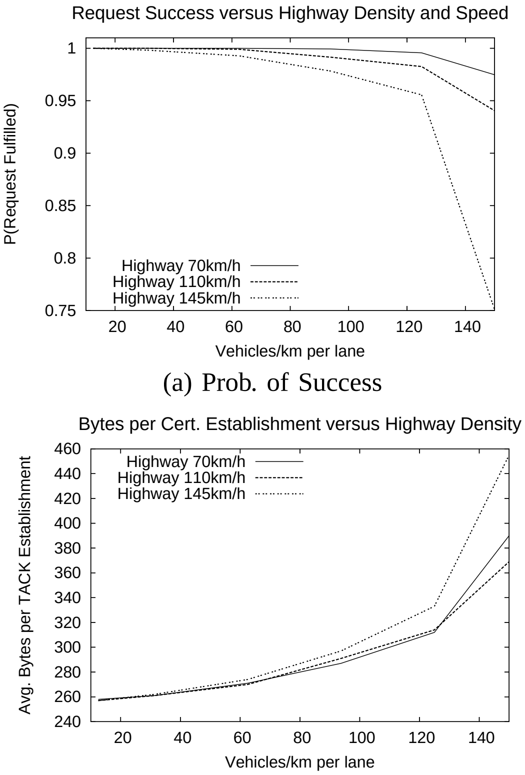
During simulation, we divide each area into regions based on the dotted lines in Figure 2 (1 kilometer square regions in the city and a boundary bisecting the highway loop). In the city, RSUs are placed on the border of regions and spaced such that at least one RSU is within radio range of every entry roadway. In the highway simulation, only a single RSU is present (the dot on the border of the regions). As soon as an OBU enters a new region, it generates and broadcasts a certificate request. If the certificate request is not fulfilled within δ, the OBU rebroadcasts a duplicate certificate request and waits another δ seconds before retrying. In simulation, we measure the probability of an OBU's certificate request being fulfilled within 10 seconds and the average number of bytes an OBU broadcasts when acquiring a new certificate (a good approximation of the additional bandwidth TACKs requires in the region surrounding RSU RAs).

Each scenario was run for 10 minutes of simulated time and repeated several times for each speed and traffic density with the results averaged across all runs for a given speed and density combination.

Probability of Successful TACK Update.

Figure 3 presents the results from our highway simulations with varying vehicle speeds and densities. We also ran several city simulations with varying vehicles densities at posted speed limits from 25km/h to 85km/h (the majority of roads have a speed limit of 55km/h). The results from both scenarios indicate that RSU computation is the limiting factor for OBUs acquiring certificates. As vehicle density and velocity increase, the rate of certificate requests approaches the maximum rate at which an RSU can fulfill requests. As RSU queues fill up and have longer delays, the probability of acquiring a certificate within 10 seconds decreases. However, for realistic traffic scenarios, the probability of acquiring a TACK is over 99%.

In city simulations, over 99% of TACK updates were successful. TACKs performed well in city simulations so we limit the discussion of those results due to space limitations. With 500



(a) Prob. of Success

(b) Request Overhead

Fig. 3. Prob. of TACK Update Success & Overhead versus density of Highway Traffic

nodes/km^2, the probability of a successful certificate request was 99.905%. For reference, sub-compact cars (2.5m × 1.5m) bumper-to-bumper and door-to-door provide a realistic upper limit to traffic density at 267 vehicles/km^2. As such TACKs can successfully handle certificate distribution even under extreme traffic congestion.

At highway speeds, the probability of acquiring a TACK certificate is above 99% until the speed is greater than 110km/h and the density is greater than 100 vehicles/km per lane. Only once the rate of certificate requests approaches the RSU's maximum of 25 requests a second (39.2ms per request with 36ms to verify the group signature and 3.2ms to generate the certificate), OBU requests for certificates start to fail. Given vehicle spacing is inversely proportional to speed (i.e., congestion causes a decrease in speed) and normal vehicle density is around 50 vehicles/km per lane [27], we conclude that even an RSU with modest computational resources can fulfill certificate requests under realistic traffic conditions.

TACKs Bandwidth Overhead.

In TACKs, only certificate requests and responses consume additional bandwidth when compared to fixed OBU keys. Figure 3 (b) indicates the average number of bytes an OBU broadcasts to perform a TACK update versus traffic density on the highway. Each request is 256 bytes plus packet overhead: a 228 byte group signature and a 28 byte ECDSA public key. In our simulation, if an OBU does not receive a beacon after $\delta = 2$ seconds, the OBU rebroadcasts the certificate request. Even while other OBUs are broadcasting safety beacons or requesting certificates for themselves, channel contention is limited such that few requests are lost and thus duplicate requests occur when queuing delays prevent RSUs from servicing requests within δ. In the city with 500 OBUs/km^2, a certificate request takes 281 bytes on average. In highway simulations with 150 OBUs/km^2 at 145km/h, a certificate request takes 454 bytes on average. On the congested highway, vehicles broadcast more requests based on the assumption the RSU did not receive the request, not knowing that the RSU has queued the request and is busy processing earlier requests.

The results in this section show that TACKs is an efficient OBU key management system which can operate with commodity hardware in RSUs under stressed traffic conditions.

VII. Analysis of TACKs with Online RAs

When VANETs are first deployed, RSU coverage will be limited. In the absence of RSUs, online RAs are necessary to allow OBUs to acquire certificates. With online RAs the delay and available bandwidth in the cellular or WiMax connection used to reach the RA are important values. Fortunately, we can ignore other VANET traffic when analyzing online RAs since VANETs use 802.11p [1] and will not interfere with online RA communication. In this section, we focus on the bandwidth and delay of cellular networks. WiMax presents an alternative means with greater bandwidth for communicating with online RAs, but has smaller deployment. Computational load for an online RA is less important since all of operations are easily parallelized.

A 3G network has an expected bandwidth of 348kbps per cell for mobile nodes.[3] Within urban areas where greater customer density exists, each tower covers a region with a radius of 1.5km with 3 cells (120 degree coverage each) [11] or enough bandwidth to support 147 kbps/(km^2) = 64 TACK updates/(s·km^2). During our city simulation with a congested 500 OBUs/km^2, OBUs collectively performed on average 13.25 TACK updates each second within a 1km^2 area. As such, sufficient bandwidth exists in 3G networks to support TACKs and other data, even under times with high demand.

To determine the delay of cellular connections to servers, we ran a network ping application[4] from an N70 smartphone to a number of web servers (i.e., www.google.com, www.yahoo.com, and the local state DMV). With twelve pings to each server, the minimum, maximum, and average round-trip times were 296ms, 467ms, and 371ms. As long as δ is greater than the network and processing delay (roughly half a second total), the cellular network will not interfere with TACKs operation.

Analysis of current mobile connections to the Internet indicates that OBUs could utilize online RAs as an alternative to road side infrastructure to acquire certificates.

[3] http://www.itu.int/osg/spu/imt-2000/technology.html

[4] http://www.aspicore.com/en/products_ping.asp

VIII. Discussion

In this section we discuss some practical issues and concerns when deploying the TACKs system.

A. Impact of TACKs on Applications

For industry and the government to accept a VANET key management scheme, the scheme must not negatively impact VANET applications. Changing temporary keys impacts applications in two major ways: interrupting routing and interrupting ongoing end-to-end communication (e.g., file sharing between OBUs).

Other works have already shown that frequent key changes (10 seconds per key or less) negatively impact routing when OBUs are sparse [24]. However, TACKs require OBUs to change keys on the order of minutes (long enough to keep packet delivery at an acceptable rate).

If two nodes are using VANETs to communicate over several hops, a successful key change will disassociate the old key from the new key. To allow continued routing of data, the receiver can sign the old key using the new key and vice versa to manually link the keys. Such mechanisms would require future work or driver-defined policies to help balance usability (associate keys) and privacy (unlinkability with key changes).

B. Tracking via Online Connections

When OBUs use cellular services to contact online RAs, the cellular provider can identify the source via the SIM card. During a certificate request, the cellular provider (and only the provider) can associate the public key and location with the SIM card. Such associations violate drivers' privacy, but cellular providers can already track users via emergency 911 services or other location specific services. Drivers will have to abandon cell phones in addition to VANETs to prevent tracking by cellular providers.

IX. Conclusion

In this work, we presented Temporary Anonymous Certified Keys (TACKs) as an efficient way to fulfill the security and privacy properties necessary for key management in Vehicular Ad Hoc Networks (VANETs). In TACKs, On-Board Units (OBUs) use short-lived keys to sign messages used for VANET communication. These short-lived keys are certified by Regional Authorities (RAs). During key updates, RAs verify that the requesting OBU is a legitimate OBU that has not been revoked; however, the RAs do not learn the OBU's identity. This allows a valid OBU to acquire a certificate for a temporary key and preserve the OBU's privacy. Since RAs' certificates are only valid in their local region, OBUs must update keys upon entering a new region. When a set of OBUs enters the region, all of the OBUs update keys simultaneously, preventing eavesdroppers from tracking drivers across key changes. If a message is identified as an abuse of the VANET, authorities can trace the certificate request back to the signer. The authorities can revoke the misbehaving OBU so that it is no longer able to participate in the VANET.

References

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

[2] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proceedings of Advances in Cryptology (CRYPTO)*, 2004.

[3] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *Proceedings of the ACM conference on Computer and communications security (CCS)*, pages 168–177, 2004.

[4] G. Calandriello, P. Papadimitratos, A. Lloy, and J.-P. Hubaux. Efficient and robust pseudonymous authentication in VANET. In *Proceedings of the Workshop on Vehicular Ad Hoc Networks (VANET)*, 2007.

[5] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, I(1), 1998.

[6] D. Chaum and E. van Heyst. Group signatures. In *Proceedings of Eurocrypt*, 1991.

[7] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, Feb. 1981.

[8] R. Cramer, I. Damgard, and J. B. Nielsen. Multiparty computation from threshold homomorphic encryption. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 280–299, London, UK, 2001. Springer-Verlag.

[9] J. R. Douceur. The sybil attack. In *First International Workshop on Peer-to-Peer Systems (IPTPS '02)*, Mar. 2002.

[10] J. Duffy. U.S. pitches wireless highway safety plan. *Network World*, Nov. 2005.

[11] T. Fisher. Rural deployments using CDMA. www.e-nc.org/pdf/rural_deployments_using_CDMA.pdf.

[12] M. Gerlach. Assessing and improving privacy in VANETs. In *Proceedings of Workshop on Embedded Security in Cars (ESCAR)*, 2006.

[13] P. Golle, D. Greene, and J. Staddon. Detecting and correcting malicious data in vanets. In *Proceedings of the Workshop on Vehicular Ad Hoc Networks (VANET)*, pages 29–37. ACM, 2004.

[14] Y.-C. Hu and K. P. Laberteaux. Strong VANET security on a budget. In *Proceedings of Workshop on Embedded Security in Cars (ESCAR)*, 2006.

[15] J.-P. Hubaux, S. Capkun, and J. Luo. The security and privacy of smart vehicles. *IEEE Security & Privacy magazine*, 2(3):49–55, 2004.

[16] IEEE. 1609.2: Trial-use standard for wireless access in vehicular environments-security services for applications and management messages. IEEE Standards, 2006.

[17] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen. ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In *to Appear INFOCOM 2008*.

[18] B. Lynn. The Pairing-Based Cryptography (PBC) library. http://crypto.stanford.edu/pbc.

[19] National Highway Traffic Safety Administration. 2005 state traffic data. http://www-nrd.nhtsa.dot.gov/pdf/nrd-30/NCSA/TSF2005/StateTrafficData05.pdf, Sept. 2006.

[20] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. The TESLA broadcast authentication protocol. *RSA CryptoBytes*, 5(Summer), 2002.

[21] M. Raya and J.-P. Hubaux. The security of vehicular ad hoc networks. In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, Nov. 2005.

[22] A. K. Saha and D. B. Johnson. Modeling mobility for vehicular ad hoc networks. In *Proceedings of the Workshop on Vehicular Ad Hoc Networks (VANET)*, 2004.

[23] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki. CARAVAN: Providing location privacy for vanet. In *Proceedings of Embedded Security in Cars (ESCAR)*, Nov. 2005.

[24] E. Schoch, F. Kargl, T. Leinmller, S. Schlott, and P. Papadimitratos. Impact of pseudonym changes on geographic routing in vanets. In *Proceedings of the European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS)*, 2006.

[25] E. Shi, J. Bethencourt, H. Chan, D. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In *IEEE Symposium on Security and Privacy*, May 2007.

[26] VINT Project, University of Berkeley/LBNL. NS-2:network simulator. http://www.isi.edu/nsnam/ns/.

[27] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. K. Tonguz. On the routing problem in disconnected vehicular networks. In *Proceedings of the IEEE INFOCOM Minisymposia*, 2007.