TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs

Ahren Studer, Elaine Shi, Fan Bai, Adrian Perrig

March 14, 2008
CMU-CyLab-08-011

CyLab
Carnegie Mellon University
Pittsburgh, PA 15213

TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs

Ahren Studer
astuder@ece.cmu.edu
Carnegie Mellon University

Elaine Shi
rshi@cmu.edu
Carnegie Mellon University

Fan Bai
fan.bai@gm.com
General Motors Research

Adrian Perrig
adrian@ece.cmu.edu
Carnegie Mellon University

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) require some 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 examine 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.

1. 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 [15, 27]. In addition to helping prevent accidents, VANETs also provide convenience and business services that will help improve a driver's experience [2].

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

Providing key management mechanisms for secure VANET operation turns out to be a surprisingly intricate and challenging endeavor, because of multiple seemingly conflicting requirements. On one hand, vehicles need to authenticate vehicles that they communicate with; and road authorities would like to trace drivers that abuse the system. On the other hand, VANETs need to protect a driver's privacy. In particular, drivers may not wish to be tracked down wherever they travel.

Ideally, a VANET key management mechanism should provide the following desirable properties:

Authenticity. A vehicle needs to authenticate other legitimate vehicles, and messages sent out by other legitimate vehicles. A vehicle should filter out bogus messages injected by a malicious outsider, and accept only messages from legitimate participants.

Privacy. RSUs and casual observers should not be able to track down a driver's trajectory in the long term. Authorities can already trace vehicles through cameras and automatic license-plate readers, however, VANETs should not make such tracing any simpler. The privacy requirement is seemingly contradictory to the authenticity requirement: suppose each vehicle presents a certificate to vouch for its validity, then different uses of the same certificate can be linked to each other. In particular, suppose a vehicle presents the certificate to an RSU in one location; and later presents the same certificate to another RSU in a different location. Then if these two RSUs compare the information that they have collected, they can easily learn that the owner of the certificate has traveled from one location to another.

Short-term Linkability. For privacy, an eavesdropper should not be able to link messages in the long-term. However, as we explain in Section 2, some VANET applications require that in the short-term, a recipient be able to link two messages sent out by the same vehicle. We observe that short-term linkability does not violate drivers' privacy, because vehicles mobility pattern is constrained (i.e., vehicles cannot teleport). 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 users' privacy.

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

Efficiency. To make VANETs economically viable, the OBUs have resource-limited processors. Therefore, the cryptography used in VANET should not incur heavy computational overhead.

In this work, we propose Temporary Anonymous Certified Keys (TACKs), an efficient VANET 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, a RA certifies vehicle generated temporary keys which are used to authenticate vehicles. As traffic enters a region, each vehicle anonymously requests a certificate from the RA. If the requesting vehicle has not been revoked, the RA responds with a certificate. Since in our system all vehicles entering the region change keys simultaneously, the TACK update provides unlinkability between prior and current keys, similar to the privacy provided in MIX networks [11].

Contribution. The contribution of this work includes the following: 1) We identify the desirable properties that a VANET key management scheme should provide. 2) We propose a scheme called TACKs that achieves all of the desirable properties. Although TACKs are based on a combination of standard techniques, combining these techniques to an economically viable solution for VANETs is a challenging task. To accomplish a viable solution, we need a deep understanding of the characteristics of VANETs as well as the cryptographic techniques used. 3) We provide a solution to the vehicle revocation problem: previous works distributed revocation information for vehicles to OBUs which is inherently not scalable. 4) We analyze and simulate TACKs in a realistic setting and show that TACKs represent a practical solution to providing security and privacy in VANETs.

Previously, researchers have also studied the problem of how to provide security and privacy in VANETs. However, as far as we know, TACKs is the first system for VANET key management that supports all the desirable properties that we identify. We refer readers to Section 8 for discussion of prior work.

2. REQUIREMENTS AND ASSUMPTIONS

We now describe the unique challenges of key management in VANETs, and state our assumptions.

2.1 Requirements for VANET Key Management

In this section, we present the notation that we use in the remainder of the paper and the properties needed for a viable key management system.

We consider four sets of VANET participants:

M: An 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 for a region and can grant vehicles temporary region-specific certificates. An authority issues certificates to RAs, and certifies them as valid intermediary authorities. RSUs or online entities could act as RAs.

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 $\overline{V}$.

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

Sender validity and message integrity. In VANET, 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 be referred to as *authenticity* in this paper. This property is to prevent 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 [14] where a single vehicle impersonates multiple vehicles. Short-term linkability is a desirable property in several VANET applications [20]. 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 each vehicle broadcast a beacon every 100ms with its current location, speed, and acceleration (i.e., braking status). A receiver uses these beacons to build a map of vehicles nearby and predict if changing lanes will cause an accident. In this application, a vehicle needs to be able to identify which messages come from the same sender. A malicious vehicle might attempt to disrupt this application by impersonating multiple vehicles. However, Sybil attacks like this should not be possible.

As mentioned in Section 1, short-term linkability does not hurt drivers' privacy. Short-term linkability allows an observer to correlate two or more messages sent by the same vehicle over a short duration of time. Based on where these messages are overheard, the observer is now able to track the vehicle in that small time period. However, as vehicles do not teleport, they must be in similar locations over a short duration of time.

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

More specifically, if the same vehicle sends two messages M and M' more than Δt time apart, then an adversary should not be able to distinguish whether or not M and M' originate from the same sender (excluding other external information such as RF fingerprinting or knowledge of a vehicle's trajectory which are outside the scope of this paper). In particular, this implies that if we use message authentication codes (MACs) or digital signatures to ensure the authenticity of messages, then the MACs and signatures should not carry identifying information.

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 its identity. Formally, let O denote an OBU found to be misbehaving, revoking O means removing O from the set V and adding

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

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 often possess resource-limited processors. To ensure efficient VANET operation, we require that the required cryptographic operations be light-weight. On the other hand, we assume that RAs possess greater computational resources. Therefore, if possible, we should try to offload computationally intensive cryptographic operations (e.g., revocation checks in our system) to the RAs.

2.2 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. 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. The trusted authorities that perform these two tasks are not necessarily the same entity. In practice, to prevent a concentration of trust, we can potentially have multiple entities jointly perform each task. Section 7.3 proposes a technique that allows the splitting of a single authority into multiple ones, thereby reducing the trust placed into any single entity.

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. This type of PKI is commonly assumed in other works on VANET security [29]. 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 federal transportation authority (e.g., USDoT) could act as the root of this key hierarchy. The root signs 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 the 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). In the case of compromised RAs, periodically (e.g., daily or weekly) OBUs could automatically 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 revocation check operation in TACKs) should be offloaded to the RAs.

We assume RSU deployment or communication coverage such that OBUs can contact at least one RA when entering a region or requesting a certificate. In regions with limited infrastructure or while in the center of a region, cellular services integrated into vehicles (e.g., GM's OnStar$^{\text{TM}}$, Mercedes Benz Tele Aid$^{\text{TM}}$, or BMW Assist$^{\text{TM}}$) or WiMax could provide a connection to online RAs. Our scheme requires that RAs be accessible to verify the validity of vehicles and to act as an authority to issue short-term certificates for a region. RAs require connectivity 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, their damage is contained within that region, and once authorities detect the compromise and OBUs download the relevant revocation information the attacker's 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. If RSU key theft is considered too serious of a threat, online RAs could replace RSU-based RAs.

3. TEMPORARY ANONYMOUS CERTIFIED KEYS (TACKS)

We now give a high level overview of how the TACKs system operates.

Using TACKs for authentication and short-term linkability. In TACKs, an OBU uses a short-lived public/private key pair (also referred to as the *TACK*) to sign messages it send.This ensures the integrity of messages sent by legitimate OBUs. Meanwhile, as a vehicle uses the same TACK over a short time duration, different messages sent by the same vehicle can be linked to each other within that time frame.

TACK update, long-term unlinkability. A vehicle frequently updates its TACKs, e.g., whenever it enters a new geographic region, or whenever an old TACK certificate expires. In addition, to protect drivers' privacy, we need to guarantee long-term unlinkability, that is, two different TACKs used by the same vehicle cannot be linked to each other.

In our system, TACK keys are certified by a RA. Whenever a vehicle updates its TACK key, it needs to request a new certificate (also referred to as a *TACK certificate*) from a RA. We refer to this process as a *TACK update*.

During a TACK update, the OBU needs to prove to the RA that it is a member of the set V of valid OBUs. In our system, this proof is done in an anonymous fashion without revealing the requesting OBU's identity information. This is achieved through the use of group signatures. In a group signature scheme, a trusted entity (e.g., the Department of Motor Vehicles) issues a long-term private key to each vehicle. In Section 3.1, we introduce group signatures and state their properties. This long-term private key uniquely identifies each vehicle; and it allows a vehicle to compute a group signature and prove its validity to an RA without revealing its identity. In this way, a set of vehicles entering a region performs TACK updates in an anonymous fashion, such that eavesdroppers and certifying RAs cannot link an old TACK for a given vehicle in the set with the vehicle's new TACK.

Tracing and revocation. If an OBU is found to have misbehaved, the group manager can remove it from the valid set V. To revoke an OBU, the group manager computes a revocation token corresponding to that OBU and publishes the revocation token to RAs. In a TACK update, the certifying RA can use this revocation information to verify that the requesting OBU has not been revoked. This revocation check is also done in an anonymous way: the RA learns whether or not the requesting OBU has been revoked without learning the OBU's identity. The RA only signs a TACK certificate if the requesting OBU is a valid member of the set V.

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.

3.1 Preliminaries and Notation

Group Signatures. Group signatures were first introduced by Chaum and van Heyst [10]. 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 member of the group to sign a message and produce a group signature. Group signatures can be verified by anyone 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*. When a group member misbehaves, the group manager can trace the identity of the signer from the group signature, and henceforth revoke that user from the group. In TACKs, we use a revocation method called Verifier-Local Revocation (VLR) [5]. 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 make sure that the signer has not been revoked. The verifier only accepts the signature if it comes from a valid signer that has not been revoked. We use Boneh and Shacham's group signature construction [5]. We choose this scheme because it is one of the most efficient constructions known and it supports revocation and tracing.

The TACKs system utilizes a group signature scheme in the following way. The group manager is a trusted entity such as the Department of Motor Vehicles. Each OBU is a group member and obtains a group user key (a.k.a. a long-term private key) from the group manager. To obtain a certificate for a short-lived Temporary Anonymous Certified Key (TACK), an OBU needs to present a group signature to the appropriate RA. The RA is then able to verify that the requesting OBU is a valid member of the set V, without learning any identifying information about the OBU.

Notation. We use the following notation:

gSign	group members' algorithm to generate a group signature
gVerify	algorithm for verifying a group signature
$\mathsf{sign}_{K^{-1}}(M)$	a traditional signature for a message M signed with private key K^{-1}
guk	an OBU's group user key
gpk	group public key
gmk	group master key, owned by group manager
RL	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

Having introduced background on group signatures as well as notations, we now proceed to describe the TACKs system.

3.2 Distribution of Long-term Keys

In the TACKs system, each valid OBU has an embedded long-term private key that uniquely identifies the OBU. This long-term private key is issued by a trusted group manager such as the Department of Motor Vehicles (DMV). This key is stored in the OBU and remains stable over a long period of time, e.g., between annual vehicle inspections. We now describe how the group manager generates long-term private keys for OBUs.

The trusted entity 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 guk to an OBU, the trusted entity generates a group user key guk_i. This group user key will serve as the OBU's long-term private key. The group manager then sends guk_i to V_i; meanwhile, it maintains a history of all the group user keys it has issued, so that it can later trace misbehaving OBUs.

3.3 Anonymous Update of TACKs

We now explain how an OBU updates its short-lived TACK with an RA when the OBU enters a region for which it does not have a valid certificate or when its old certificate expires. 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 [23]. Then the OBU uses its guk_i to sign K_S^{+} (i.e., K_S^{+} is the message being signed), and sends the resulting group signature σ to the appropriate RA for the region. The group signature σ vouches for the fact that the signer is a valid OBU, without revealing the identity of the OBU.

On receiving the group signature σ, the RA calls the gVerify algorithm of the group signature scheme, and verifies whether the requesting OBU is a valid OBU. If so, 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 adds the pair (σ, K_S^{+}) to a history table to be stored locally for some extended period of time (the length of storage depends on how quickly authorities investigate VANET abuses). In this way, if an OBU with K_S^{+} misbehaves, the RA can retrieve the group signature σ associated with K_S^{+}; and the group manager can use σ to trace the identity of the misbehaving OBU and revoke it from the set V. After queueing up all of the certificate requests for a given region within the last δ seconds, the RA sends the resulting certificate to the OBU. The delay δ helps improve unlinkability by removing timing relations between when a given vehicle independently enters a region and requests a certificate and receives a certificate as part of a set. We discuss the selection of δ during the analysis of TACKs.

The protocol for updating an OBU's temporary key when it enters a region or the old certificate expires is described in Figure 1.

```
Updating an $(K_S^+, K_S^{-1})$ pair :
    OBU : $(K_S^+, K_S^{-1}) \overset{R}{\leftarrow}$ key space
    OBU : $\sigma \leftarrow \mathsf{gSign}(\mathsf{guk}_i, \mathsf{gpk}, K_S^+)$
    OBU → RA : $K_S^+, \sigma$
    RA : $b \leftarrow \mathsf{gVerify}(\mathsf{gpk}, \mathsf{RL}, \sigma, K_S^+)$
    RA : if $b = 0$ then exit
    RA : $\mathsf{cert} \leftarrow \mathsf{sign}_{K_{RA}^{-1}}(K_S^+ || \text{expiration})$
    RA : Add $(\sigma, K_S^+)$ to history table
    (less than $\delta$ seconds later)
    RA → OBU : $\mathsf{cert}$
```

Figure 1: Protocol for updating TACKs.

TACK expiration mechanism. As we mentioned, whenever a vehicle updates its TACK, the new TACK cannot be linked to the old TACKs that have been used by the same vehicle in the past. To ensure that TACKs expire after a certain period of time (e.g. every few minutes), the RA includes expiration information when it signs a certificate for a TACK public key K_S^+. Our system uses the following two mechanisms to enforce TACK key expiration: *time-based* and *region-based*. The RA included expiration time ensures a TACK is only valid for a certain period enforcing time-based expiration. Given RAs are only authorities in their own region, a TACK from one region is not valid in another region enforcing region-based expiration. With this combination of mechanisms, our system enforces that a TACK certificate is valid only for a short period of time within the region associated with a certifying RA.

In practice, to support region-based expiration, a vehicle needs to know its current location and the set of RAs for a region to verify a TACK certificate. GPS can provide location information and map metadata can provide RA contact information (see Section 2.2). Recall that each RA is part of a PKI, and has been certified by a trusted authority. A certificate for an RA includes the RA's region information. Through such location-aware certificates, we can guarantee that malicious RAs do not lie about the region they control.

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 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 revoked member on the revocation list, to make sure that the signer has not been revoked. Each check requires that the verifier perform some mathematical operations in certain algebraic structures. 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 may be too expensive.

Boneh and Shacham propose a method for a more efficient revocation check (See Section 7 of the Boneh and Shacham work [5]). Specifically, by restricting the randomness in the gSign algorithm, the verifier can perform some pre-computations, such that each revocation check boils down to a constant number of operations plus a table look-up.

In TACKs, we can adopt the same strategy. In particular, using the same notations as Boneh and Shacham [5], the signer uses a hash function to compute two random numbers (u and v) in the process of generating a group signature. We can fix the numbers u and v for the same RA over a short duration T (e.g., every 10 minutes):

$$(u, v) \overset{R}{\leftarrow} H(\mathsf{gpk},\quad \text{time\_epoch},\quad \text{RA\_id})$$

At the beginning of each time epoch, the RA performs $O(|\mathsf{RL}|)$ operations and saves the result of the pre-computation in a table. Or, rather than performing these pre-computations at the start of each time epoch, the RA can utilize idle processor cycles to pre-compute them. In this way, verifying a group signature requires only $O(1)$ operations.

If an OBU issues two group signatures to the same RA in a single time epoch, the RA can test whether these two signatures were generated by the same group user key (i.e., the OBU's master key). RAs will only respond to the first request from the OBU, and will prevent an OBU from receiving more than one certificate per time epoch. However, signatures issued at different RA's or in different time epochs still remain unlinkable.

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

Recall that in order to achieve more efficient revocation check, we fix the random numbers u and v needed for the group signature generation for the same RA and same time epoch. This allows us to achieve the following properties:

P1. If an OBU sends two requests for TACK certificates to the same RA within a single time epoch, the RA is able to link these two requests to the same OBU.

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

Property **P1** prevents a malicious OBU from requesting multiple TACK certificates at the same RA within the same time epoch. On the other hand, property **P2** guarantees legitimate senders' anonymity in the long run.

3.4 Tracing and Revocation

When an OBU with TACK public key K_S^+ misbehaves, police (or another trusted entity) can retrieve from the RA the group signature σ associated with that K_S^+. 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 long-term secret keys (guk_i's) of OBUs in the set V, and identifies the signer. The tracing algorithm takes time linear in the size of V. Since in VANETs the size of V can be large (i.e., if the state DMV is the manager $|V|$ = number of registered vehicles in the state), $O(|V|)$ computation may be expensive in practice. In Section 4.2, we discuss an alternative group signature construction that provides $O(1)$ tracing at the cost of slower signature generation and verification.

Suppose that the group manager identifies that V_i is the misbehaving OBU. To revoke V_i, the group manager adds a revocation token grt_i tied to V_i to the current revocation list RL, and publishes the updated RL to the RAs.

3.5 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^+.

We can also use more efficient methods to achieve broadcast authentication, such as TESLA [21,28]. In this case, we will use an OBU's TACK private/public key pair (K_S^{-1}, K_S^+) and the certificate for K_S^+ to bootstrap the symmetric authentication keys required by TESLA.

4. TACKS ANALYSIS

In this Section we discuss how TACKs meet the requirements set out in Section 2. We also describe several practical concerns, such as how to defend against eavesdroppers correlating TACKs in an attempt to track vehicles, and how to pick an appropriate life-time for TACK keys.

4.1 Security Analysis

In this section, we explain how TACKs satisfy the various security requirements posed in Section 2.

Authenticity. In TACKs, *message integrity* is guaranteed through means of digital signatures created using the TACK private key. If signatures are not used for message authentication, vehicles can use TACKs to bootstrap other broadcast authentication mechanisms (e.g., TESLA).

We now explain how TACKs guarantees *sender validity*. When an OBU requests a certificate from an RA, the RA verifies the request and confirms that authorities have not revoked the OBU. Provided that the OBU has a valid TACK certificate, a recipient can infer that the OBU was not revoked when it received the certificate, proving *sender validity*. However, there still is a window of time between when an OBU was revoked and when it must request a new certificate. During that small time window, a revoked OBU is still able to participate in the VANET. In Section 7.1, we discuss practical issues and concerns when we pick the lifetime for TACK keys.

Short-term linkability and Sybil prevention. As a vehicle uses the same TACK over a short duration of time, during that time frame, two or more messages sent by the same vehicle can be linked to each other.

In addition, a malicious OBU cannot impersonate arbitrarily many OBUs at the same time. As explained in Section 3.3, during a time epoch T_i, an OBU can only obtain a single TACK certificate from an RA for a region. This provides a defense against the Sybil attack.

Nevertheless, an attacker who has acquired long-term private keys from multiple OBUs may request multiple TACK certificates from an RA. However, this is the same situation as when multiple vehicles conspire in the VANET 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, defense against the correlation attack. To protect drivers' privacy, we require that messages sent by the same vehicle be unlinkable in the long-run. TACKs leverages group signatures to allow vehicles to prove their validity to RSUs without leaking their identifying information.

However, cryptography alone does not provide 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 there is only a single OBU changing 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 [19, 33]. 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 in the set of new keys. Prior works have studied coordinated key update techniques, but these works require explicit communication between vehicles to coordinate key updates [19, 33].

TACKs implicitly forces OBUs to request new keys whenever they enter a region, ensuring coordinated key updates without explicit communication while still providing a MIX function when multiple vehicles enter a region [11]. When a number of vehicles enter a new region, each vehicle sends a certificate request and does not sign any new messages until receiving the RA's response. Even though the request is not encrypted, the group signature hides the requesting OBU's identity in the TACK request, and there is no relation between the old temporary key and the new randomly selected temporary key. Once the RA responds with certificates for OBUs' new temporary keys, OBUs will start signing messages with those keys. If an eavesdropper is tracing a vehicle, after a key update the eavesdropper will only know that the victim car is a member of the set of vehicles which updated keys, but not know which one exactly. Eavesdroppers can correlate vehicle announced location and velocity to help track a specific vehicle in a cluster of certificate requesters, but if the silent period is on the order of seconds and regions change at intersections, or other places where vehicles can turn in one of several directions, it will be hard for an attacker to associate the old key with the new key based on radio messages alone (i.e., the eavesdropper may not know if the vehicle turned or went straight).

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 [8]. Prior works on traffic models use a Poisson distribution with a rate of $\lambda = [0.5, 0.8]$ to describe the number of vehicles that drive along a highway for the majority of the day [37]. If an RA waits δ seconds between certificate responses (i.e., batching responses to reduce correlation between request and response time), we can describe the number of vehicles that change keys simultaneously (i.e., the anonymous set size) using a Poisson distribution with a parameter of $\delta \cdot \lambda$. If an eavesdropper associates an OBU with a given key in region A, the attacker will associate the OBU with X_{AB} vehicles after the OBU enters region B (where $X_{AB} \sim Poisson(\delta \cdot \lambda)$). If the OBUs that entered B together exit the region at different times or locations, the number of vehicles changing keys with vehicle i from that set is an independent Poisson random variable X_{BC_i} with the same parameter $\delta \cdot \lambda$. After all X_{AB} OBUs leave the region, each of the X_{AB} original OBUs will contribute X_{BC_i} additional OBUs to the anonymous set. Using the rule of iterated expectations, we find that the expected number of vehicles in the anonymous set size after n region changes is $(\delta \cdot \lambda)^n$. As a lower bound, if the OBUs that enter the region together leave the region together, the second key change provides no increase in the anonymous set size and the anonymous set size remains at X_{AB}. When an OBU's certificate expires in a region, that vehicle must perform a TACK update with an online or nearby RSU RA. However, unless the OBU is near the edge of a region there is a small probability of other nearby vehicles simultaneously updating their TACKs, without additional OBUs changing keys the anonymous set size remains the same.

The selection of δ presents a need to balance privacy and availability of the VANET. With a long RA certificate response delay, the larger the anonymous set size, but OBUs could leave radio range if an RSU RA is used. In addition, OBUs cannot send authenticated messages for the new region until receiving the certificate from the RA. OBUs without certificates still allow routing since the nodes can still forward messages other nodes sign. If δ is small OBUs will lack privacy since the anonymous set size will be small. As such, the upper bound on δ is $\frac{r}{v}$ where r is the reliable radio range of an RSU (300 meters [23]) and v is the speed limit (or some fraction larger to permit some speeding). We use r (rather than $2r$) to reflect that RSU RAs will be on the border of regions and a vehicle will request a certificate as soon as it enters the new region. For example, system managers should impose a δ of 10 seconds or less on highways with a speed limit of 70mph or 30m/s. The lower bound on δ depends on the processing time of a request. 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 which specific group member requested a certificate for a public key. 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 it to the revocation list RL. When verifying future group signatures, the RSUs will check them against the revocation list RL to make sure they come from valid OBUs that have not been revoked.

Ideally, a misbehaving OBU should be revoked as soon as possible to prevent it from causing further damage. In TACKs a revocation operation takes effect as soon as the revoked OBU's current TACK key expires. To ensure timely revocation, it is desirable to have TACK keys expire rapidly. Section 7.1 discuss the various implications and trade-offs when we pick different TACK key life-times.

4.2 Cryptographic Overhead

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 1: Estimated Computation Time and Size of TACK Related Cryptography for a 3.2GHz RA or a 400MHz OBU.

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 [29]).

Performance of Group Signature Schemes. Boneh and Shacham's group signature scheme [5] requires the use of bilinear groups, also referred to as pairings [16]. Several types of pairings can be used in the construction with tradeoffs between storage cost and computation cost. In TACKs, the major concern is the computational overhead of signature generation, since OBUs have to be economically viable, and thus have limited computational power. Among known pairings, type A pairings are the fastest to compute [26]. Therefore, we will assume the use of type A pairings when analyzing the performance of the intended group signature construction.

Two recent works estimate the performance of running type A pairings on a modern workstation and ECDSA on a memory-constrained 400MHz machine [29, 35]. Table 1 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 3.3. Moreover, the verification time does not include pre-computation.

Boneh and Shacham also point out that using type D pairings, the signature length can be reduced to 1192 bits, or 149 bytes. However, the type D pairing operation is roughly 5 times slower than a type A pairing, resulting in increased signature generation time.

Improving tracing efficiency. Having studied the signature generation and verification cost for OBUs and RAs respectively, we now investigate the time required for an authority to trace a signer of a group signature. Using the original Boneh and Shacham construction, it takes a group manager time linear in the number of valid OBUs, to trace a signature to an OBU. If five million vehicles are in the same group, it will take on average half a day to trace an OBU.

It is possible to bring the tracing cost down to $O(1)$. The technique has been used in another recent group signature scheme by Boyen and Waters [6]. This new variant uses bilinear groups of composite order. Although it preserves the signature generation and verification cost asymptotically, in practice, pairing operations in composite order bilinear groups are $20 \sim 30$ times slower than the fastest pairing operation in prime order groups (type A pairing). Therefore, this indicates a tradeoff between tracing efficiency and signature generation/verification efficiency. If we consider signature generation and verification to be far more frequent than tracing, the original Boneh and Shacham scheme is more attractive, as it provides more efficient signature generation and verification. However, the variant using bilinear groups of composite order may still be a valuable alternative to keep in mind, especially because slower tracing also implies that a malicious vehicle is allowed to participate in the VANET longer and cause more damage.

To summarize, the results in this section show that group signatures and other cryptographic primitives adopted by TACKs are both computationally and space efficient enough for use in VANETs.

5. TACKS SIMULATION WITH RSU RAS

We use ns-2 [36] to simulate TACKs with RSU RAs in highway and city settings. In Section 6 we analyze the use of online RAs. To represent city traffic we use a traffic scenario generator [32] and the 3 kilometer square (9km^2) city topology (a section of 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. In the following subsections, we analyze the probability of an OBU receiving a certificate when entering a new region and the additional communication overhead associated with TACKs.

During simulation, we divide each area into regions based on the dotted lines in Figure 2 (1 kilometer square regions in the city and half way across the highway). 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 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 (a crucial operation for TACKs to work) and the average number of bytes an OBU broadcasts when requesting a certificate (a good approximation of the additional bandwidth TACKs requires in the region surrounding RSU RAs).

Each scenario was allowed to run for 10 minutes of simulated time and repeated several times (3 times for each vehicle speed and traffic density for highway simulations and 6 times for each traffic density in the city simulations) with the results averaged across all runs for a given speed and density combination to reduce variance.

5.1 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 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 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%.

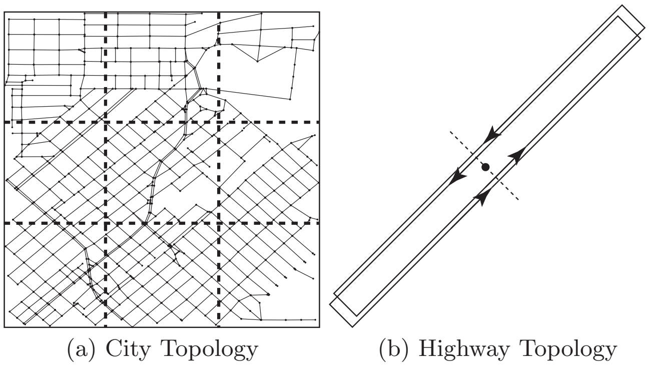

(a) City Topology (b) Highway Topology

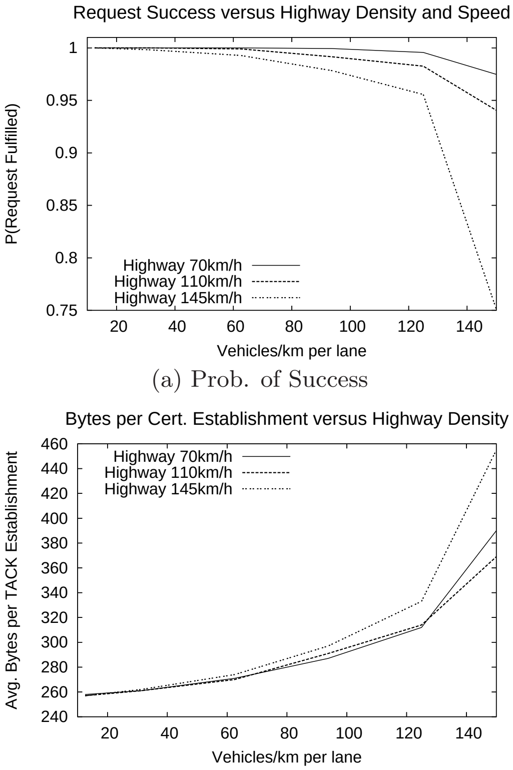
Figure 2: Topologies Used to Simulate Traffic

In city simulations, over 99% of TACK updates were successful. Due to space limitations, we only discuss city simulation results.With 222 nodes/km^2, the probability of success is 99.935%. With 500 nodes/km^2 traveling an average of 55km/h, the probability of success is still 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.

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 25 requests a second (the maximum an RSU can handle based on the numbers from Sec. 4.2), OBU requests for certificates start to fail. Simulations with 1 OBU every 5.33 meters or 187 vehicles/km per lane at 110km/h (≈ 22.5 requests a second) had a success rate of 21%. However, it is unrealistic to have such congested traffic at such high speeds (even if everyone drove cars that are only 2.5m long that leaves 1 car length between each vehicle traveling at 110km/h). Thus we conclude, even an RSU with modest computational resources can fulfill certificate requests under realistic traffic scenarios.

5.2 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 updateversus traffic density on the highway. Note that 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 two seconds, the OBU rebroadcasts the certificate request. Our results show that even while other OBUs are broadcasting safety beacons or



(a) Prob. of Success

(b) Request Overhead

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

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. The issue is that as requests are queued longer, vehicles broadcast more requests based on the assumption the RSU did not receive the request, not knowing that the RSU is busy processing earlier requests.

The results in this section show that TACKs is efficient enough to operate with commodity RSU hardware under the most stressed traffic conditions and still meet the requirements necessary of a VANET key management system.

6. ANALYSIS OF ONLINE RAS

When online RAs are used, the bandwidth and delay associated with 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 [2] and will not interfere with cellular or WiMax certificate traffic. Given the limited deployment of WiMax and its greater capabilities we focus on analysis of cellular networks in this section. Computation load for an online RA is less important since all of the key operations are easily parallelized.

A 3G network has an expected bandwidth of 348kbps per cell for mobile nodes[2]. 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) [17] or enough bandwidth to support 147 kbps/(s·km^2) = 64 TACK updates/(s·km^2). During our simulation of a city with a crowded 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 traffic.

To determine the delay of cellular connections to servers, we ran a network ping application[3] 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 time was 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 currently available mobile connections to the Internet indicates that OBUs could use online RAs as an alternative to road side infrastructure to acquire certificates.

7. DISCUSSION

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

7.1 Selecting TACK Certificate Life-time

The life-time of a TACK certificate (and equivalently the key) can have several implications on our system. On one hand, rapid expiration of certificates has the following advantages: 1) A shorter TACK life-time means better privacy, as messages sent by the same vehicle more than a TACK life-time apart cannot be linked to each other. 2) As mentioned earlier, a shorter TACK life-time means more timely revocation. As a revocation only takes effect after the revoked OBU's current certificate expires. This ensures that we can prevent the misbehaving OBU from causing further damage in a timely manner.

On the other hand, rapid expiration of TACKs can be undesirable in the following sense: 1) Rapid key updates incur more computational and communication cost to the VANET. 2) As some VANET applications require short-term linkability over a small duration Δt, the TACK life-time should not be smaller than Δt.

Taking all of the above factors into account, we suggest 5 to 20 minutes as a reasonable TACK key life-time. Also note that in practice, the TACK lifetime is enforced through a combination of two expiration mechanisms: time-based expiration and region-based expiration (See Section 3.3). This means that we need to pick an appropriate value for both the time till expiration, and the area within which a TACK key is valid.

7.2 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).

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

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

Other works have already shown that frequent key changes (10 seconds per key or less) negatively impact routing when OBUs are sparse [34]. However, TACKs require OBUs to change keys when they enter new regions or the old certificate expires. A realistic TACKs deployment will have regions a few kilometers in each direction, forcing OBUs to change keys once every few 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. When this happens nodes may no longer know where to route packets. This requires a "handover" mechanism similar to mobile IP. The simplest solution would sacrifice unlinkability for connectivity by associating the previous key with the current key. An OBU with previous key pair (K_N^+, K_N^{-1}) and new pair (K_{N+1}^+, K_N^{-1}) would transmit $(\{K_{N+1}^+\}_{K_N^{-1}}, \{K_N^+\}_{K_{N+1}^{-1}})$ so nodes would know where to direct traffic. Such mechanisms would require future work or driver-defined policies to help balance usability (associate keys) and privacy (unlinkability with key changes).

7.3 Avoiding a Single Point of Trust

In TACKs, the group manager is able to open every group signature produced by OBUs and trace the signer. This means that if RAs collaborate with the group manager, then they are able to track any vehicle. The solution to this problem is to avoid a single point of trust. In particular, we wish to split the role of the group manager into multiple entities, such that only when a threshold number of them collaborate, can they trace the signer of a group signature.

Splitting the group manager into multiple entities can be achieved through a combination of standard techniques known in the cryptography literature, including secure multi-party computation [12], discrete-log-based zero-knowledge proofs [9, 13, 18, 24], and public key threshold encryption systems [3].

7.4 Tracking via Online Connections

When OBUs connect to cellular services, the cellular provider can identify the source via the SIM card. When an OBU makes a certificate request, the cellular provider can associate the public key and associated region 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. To achieve perfect location privacy drivers will have to abandon cell phones in addition to VANETs.

Fortunately, only the cellular provider can track OBUs using online RAs. Traffic between the OBU and the cellular provider is encrypted so parties who eavesdrop on cellular traffic cannot access data in certificate traffic associated with a given OBU to tower connection[4]. To prevent Internet eavesdroppers or online RAs from relating IP addresses with cellular customers, cellular providers should use a small number of IP addresses (similar to a NAT) to translate from cellular device to IP address.

8. RELATED WORK

We give an overview of prior work in this space, and point out why previous schemes fail to meet the requirements of VANETs.

Several research papers have examined VANET key management [1, 4, 7, 19, 22, 25, 29–31, 33]. In addition to verifying valid parties through signatures, these works focus on efficient mechanisms to provide privacy through long-term unlinkability. In these works, privacy is provided through one of three mechanisms: each vehicle has multiple pre-installed public/private key pairs, group signatures are used to sign every message, or vehicles use group signatures to sign their own certificates for temporary public/private key pairs. In this subsection, we present an overview of previous work in this space, and discuss their advantages and drawbacks.

8.1 Multiple Pre-installed Keys

To protect the privacy of drivers, researchers have proposed to install numerous public/private key pairs on vehicles [22, 29–31]. In this way, vehicles can update their keys periodically and a new key should be unlinkable to an old key used by the same vehicle. The drawbacks with this approach include:

Key renewal problem. Once the pre-installed keys have all been used, some mechanism is required for the OBU to renew its pool of pre-installed keys. This can be achieved in two ways: 1) the driver goes to some authority (e.g., Department of Motor Vehicles) to request new keys periodically; or 2) intermediary authorities such as RSUs issue new keys to OBUs. In the first case, we need that the renewal operation is relatively infrequent (e.g. on a yearly basis); as drivers will find it cumbersome and inconvenient to have to go to the DMV for key renewals. Thus the OBU must have ample storage to hold a sufficient number of keys needed per annum. This may incur unreasonable cost for building the OBUs. On the other hand, if we allow RSUs to issue new keys to vehicles, unless some cryptographic mechanism (e.g. group signature) is used, a set of colluding RSUs now have the ability to link a vehicle's old keys with new keys. However, in practice, we would like to avoid exposing such an amount of trust to an intermediary authority like the RSU.

Expensive revocation. Using multiple pre-installed keys on OBUs makes revocation hard. Here revocation can potentially be done in two ways:

- The authority distributes the list of revoked keys to all OBUs and RSUs, so they can check whether a sender is using a key that has been revoked. This operation is expensive because OBUs now have to store a list linear in the length of the revoked OBUs' keys. TACKs also requires distribution and storage of revocation information. However, in TACKs OBUs only maintain RA revocation information which is much smaller (i.e., number of regions versus number of OBUs).

- Assuming tamper-resistant hardware on the OBUs, we can also have the authority broadcast a revoke message to a revoked OBU, such that the tamper-resistant hardware on the revoked OBU can erase the pre-installed keys on the OBU. This approach incurs additional cost on the OBUs due to the use of tamper-resistant hardware, and therefore may not be economically viable.

[4] In RSU-based certificate requests, OBUs wirelessly broadcast certificate requests without source addresses to disassociate the source from the message.

Storage cost. As mentioned earlier, one way for a vehicle to renew its pre-installed keys is to have drivers go to a trusted entity (e.g., Department of Motor Vehicles) to obtain a new pool of keys periodically. As drivers may find this cumbersome and inconvenient, we would like to keep such key renewals at an infrequent basis, for example, every year. As a result, OBUs are required to have sufficient storage to hold enough keys for one year. Assume we use 1024-bit keys (2048 bits for a public-private pair), and that OBUs rotate its keys every 10 minutes, this means that each OBU needs roughly $13MB$ storage to store keys for one year. While normal storage is cheap, in practice, it may be more desirable to store sensitive keying material using tamper-proof storage. However, $13MB$ of tamper-proof storage may be too costly for an OBU.

8.2 Using Group Signatures

Groups signatures allow vehicles to prove that they are valid members of the set V without revealing their identifying information. Through the use of group signatures, we can potentially achieve both authenticity and anonymity in VANETs.

In prior work, Boneh et al. [4] propose that OBUs should use group signatures to sign VANET messages. Armknecht et al. [1] and Calandriello et al. [7] propose for OBUs to use group signatures to anonymously sign their own short-lived certificates and bootstrap keys required for authentication. These approaches help reduce the security overhead in VANETs. With these scheme, a single computationally expensive operation is done to generate the certificate. After that, OBUs use relatively short signatures that can be generated and verified quickly to authenticate messages.

The main difference between TACKs and the works by Armknecht et al. and Calandriello et al. is that rather than having OBUs directly verify group signatures, in TACKs, the RAs issue certificates for TACK keys, thus offloading group signature verification to the RAs. In particular, if we wish to support revocation, having the RAs perform group signature verification and revocation check has the following advantages:

Avoid distribution of revocation information to OBUs. In group signatures with verifier-local revocation, the verifier needs to have the revocation list. In TACKs, by having the RAs verify the group signatures, authorities only have to distribute OBU revocation information to RAs. OBUs still need RA revocation information. However, since the RA population is smaller and more stable (e.g., vehicles are likely to change owners, but RAs will rarely change), distribution and management of RA revocation information is a simpler less time critical task.

Avoid costly operations on OBUs. In group signatures with verifier-local revocation, the verifier needs to check if the signer of a group signature has been revoked. As we explain in Section 3.3, this is a relatively expensive operation. As OBUs usually have limited computational resource, we do not wish to have the OBUs perform such costly operations.

In TACKs, an OBU only needs to perform the verification operation of a standard digital signature algorithm (e.g., RSA signatures), and these signatures are much faster to verify than a group signature (especially, one with verifier-local revocation). In practice, we can further reduce the cryptographic cost on the OBUs by using one digital signature to establish TESLA keys which allows efficient broadcast authentication. In this way, the cost of one digital signature can be amortized across multiple messages.

Lu et al. [25] propose a key management scheme that is similar to ours in that OBUs use group signatures to acquire certificates from RSUs. However, there scheme fails to address two major issues: tracking prevention and a lack of infrastructure. In their scheme, an OBU anonymously requests a new certificate from a RSU whenever the old certificate expires. However, as we discuss in Section 4.1, a vehicle anonymously changing keys does not prevent tracking, an obvious violation of privacy. In their scheme they assume OBUs are always within radio range of RSUs, this is a very strong assumption since at this time no RSUs exist and complete coverage will not exist when VANETs are first deployed. In TACKs, the option of online RAs allows operation under the more realistic assumption of widespread cellular connectivity, something that is true today.

8.3 Context-aware Key Changes

Whether one uses multiple pre-installed key pairs on OBUs; or uses group signatures to bootstrap short-lived keys, as pointed out in Section 4, these cryptographic mechanisms alone do not prevent correlation attacks using information derived through other channels (see Section 4.1).

Gerlach [19] and CARAVAN [33] build upon Hubaux and Raya's works and focus on when OBUs should change keys. Gerlach uses "Mix Contexts" to help OBUs determine when to change keys. Under Mix-Contexts, OBUs evaluate the number of nearby vehicles and other useful information to determine the entropy of the context, and change keys when the entropy is above a certain threshold. In CARAVAN, an OBU changes its key when it joins a network (e.g., merges onto a roadway), but failed to prevent tracking unless all OBUs on the roadway simultaneously changed keys. As mentioned in Section 4.1, the drawback with such approaches is the need for communication between OBUs to decide when to change keys. In TACKs, when OBUs enter a new region, the OBUs change keys together without any additional communication overhead.

9. 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 to abuse the VANET, authorities can trace the certificate request back to the signer. The authorities can further revoke the misbehaving OBU so that it is no longer able to participate in the VANET.

10. REFERENCES

[1] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng. Cross-layer privacy enhancement and non-repudiation in vehicular communication. In *Workshop on Mobile Ad-hoc networks (WMAN)*, Mar. 2007.

[2] 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.

[3] D. Boneh, X. Boyen, and S. Halevi. Chosen ciphertext secure public key threshold encryption without random oracles. In D. Pointcheval, editor, *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 226–243. Springer, 2006.

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

[5] 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.

[6] X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *Public Key Cryptography (PKC2007)*, volume 4450 of *Lecture Notes in Computer Science*, pages 1–15, 2007.

[7] 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.

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

[9] D. Chaum and T. P. Pedersen. Wallet databases with observers. In *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 89–105, London, UK, 1993. Springer-Verlag.

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

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

[12] 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.

[13] R. Cramer, I. Damgard, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO '94: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, pages 174–187, London, UK, 1994. Springer-Verlag.

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

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

[16] R. Dutta, R. Barua, and P. Sarkar. Pairing-based cryptographic protocols: A survey. Cryptology ePrint Archive, Report 2004/064, 2004. http://eprint.iacr.org/.

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

[18] E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, pages 16–30, London, UK, 1997. Springer-Verlag.

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

[20] 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.

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

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

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

[24] E. F. Ivan Damgard. An integer commitment scheme based on groups with hidden order. In *Advances in Cryptology, ASIACRYPT 02,*, 2002.

[25] 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*.

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

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

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

[29] 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.

[30] M. Raya and J.-P. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks*, 2007.

[31] M. Raya, P. Papadimitratos, and J.-P. Hubaux. Securing vehicular communications. *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications*, 2006.

[32] 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.

[33] 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.

[34] E. Schoch, F. Kargl, T. Leinmüller, 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.

[35] 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.

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

[37] 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.