EDR: Efficient Decentralized Revocation Protocol for Vehicular Ad Hoc Networks

Albert Wasef, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—It is well recognized that security is vital for the reliable operation of vehicular ad hoc networks (VANETs). One of the critical security issues is the revocation of misbehaving vehicles, which is essential for the prevention of malicious vehicles from jeopardizing the safety of other vehicles. In this paper, we propose an efficient decentralized revocation (EDR) protocol based on a novel pairing-based threshold scheme and a probabilistic key distribution technique. Because of the decentralized nature of the EDR protocol, it enables a group of legitimate vehicles to perform fast revocation of a nearby misbehaving vehicle. Consequently, the EDR protocol improves the safety levels in VANETs as it diminishes the revocation vulnerability window existing in conventional certificate revocation lists (CRLs). By conducting detailed performance evaluation, the EDR protocol is demonstrated to be reliable, efficient, and scalable.**

***Index Terms*—Ad hoc, decentralized, revocation protocol, vehicular networks.**

I. Introduction

VEHICULAR ad hoc networks (VANETs) have become a promising technology for increasing the efficiency and the safety levels of transportation systems. VANETs consist of two main network entities: 1) vehicles and 2) infrastructure roadside units (RSUs). Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications are two basic vehicular communication modes that respectively allow vehicles to communicate with each other or with the infrastructure RSUs. Due to the open-medium nature of wireless communications and the high-speed mobility of a large number of vehicles in spontaneous vehicular communications, message authentication, integrity, nonrepudiation, and privacy preservation are identified as the primary security requirements for VANETs [1], [2]. According to [3], vehicular networks will rely on the public key infrastructure (PKI) as a comprehensive method to achieve these security requirements. In PKI, a central certification authority (CA) issues an authentic digital certificate for each node in the network. An efficient certificate management is essential for the reliable and robust operation of any PKI. A critical part of any certificate-management scheme is the revocation of misbehaving nodes. Certificate revocation can be centralized or decentralized. For centralized revocation, a central entity, such as the CA, is the only entity in the network that can take the revocation decision for a certain node. For decentralized revocation, the node revocation is done by the neighboring nodes of the misbehaving node.

According to the IEEE 1609.2 standard [3], vehicular networks depend on certificate revocation lists (CRLs) and short-lifetime certificates to achieve revocation. In such a case, to revoke a vehicle, a CRL has to be issued by the CA and broadcast by the infrastructure RSUs. The network scale of VANETs is expected to be very large. Hence, the distribution of CRLs is prone to long delays [4], [5]. Moreover, centralizing the revocation decision to the CA renders the CA to be a bottleneck as well as a single point of failure. In addition, during the early deployment of VANETs, it is expected that RSUs will not uniformly be distributed in the network. Hence, CRL is not proper for applications requiring fast revocation of misbehaving vehicles. Revocation can also be achieved by relying on certificates with short lifetimes, where a certificate is automatically revoked after its lifetime expires. In VANETs, each vehicle takes life-critical actions based on the received messages from its neighboring vehicles. Hence, VANETs cannot solely depend on the short-lifetime certificates, as a misbehaving vehicle can harm other vehicles until its certificate lifetime expires.

For a practical revocation method, it is required that the revocation of misbehaving vehicles should take place as fast as possible to prevent these vehicles from jeopardizing the safety of other vehicles. In addition, the revocation should be done in a decentralized way to alleviate the load on the CA. In addition, the revocation method should be independent of RSUs, which may not uniformly be distributed in the network. Finally, the revocation method should not contradict other security requirements so that it can efficiently be integrated with other security mechanisms. To address the aforesaid challenges, we propose an efficient decentralized revocation (EDR) protocol for VANETs, which enables a group of neighboring vehicles to revoke a nearby misbehaving vehicle. The EDR protocol is independent of the RSUs and the CA, which makes it suitable for the early deployment phase of VANETs, where a nonuniform RSU distribution is expected. In addition, the EDR distributes the revocation load to all the vehicles, thus avoiding overwhelming the CA. Moreover, it achieves fast revocation of misbehaving vehicles, thus decreasing the time window during which a misbehaving vehicle can broadcast malicious messages. Consequently, the EDR protocol increases the security level provided by VANETs. In addition, the revocation messages, which are broadcast by the vehicles, have a security strength equivalent to that of the revocation messages issued by the CA. The EDR protocol has a modular nature that makes it

Manuscript received March 3, 2009; revised May 7, 2009. First published May 26, 2009; current version published November 11, 2009. The review of this paper was coordinated by Prof. Y. Zhang.

The authors are with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: awasef@bbcr.uwaterloo.ca; xshen@bbcr.uwaterloo.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2009.2023662

integrable with any PKI system. It can also be used as a standalone revocation protocol or integrated with the CRL technique to compensate the absence of RSUs in some areas.

The remainder of this paper is organized as follows. Section II discusses the related work. In Section III, preliminaries are presented. The proposed EDR protocol is presented in Section IV. The performance analysis of the EDR protocol is discussed in Section V. Section VI evaluates the EDR protocol from the security point of view. Section VII concludes this paper.

II. Related Work

Security of VANETs has gained extensive interest in the last few years. In [1] and [2], PKI systems were proposed to secure VANETs. Systems based on symmetric key cryptosystems were developed in [6]–[8] to provide security to VANETs while minimizing the overhead and increasing the encryption/decryption speeds compared with those of the PKI. In [9]–[13], the group signature technique was used to provide security and privacy to VANETs. All of the foregoing systems rely on the CRL as a revocation method without considering the special characteristics of VANETs. Short-lifetime certificates were proposed in [14] and [15]. Each vehicle needs to acquire a new certificate from any RSU before its current certificate expires. Compromised or faulty nodes can still endanger other vehicles until the end of their certificate lifetimes.

The probabilistic approach is a promising technique for key management in ad hoc networks [16], [17]. Zhu *et al.* [18] used the probabilistic approach to establish a pairwise key between the network nodes. Later, they introduced the GKMPAN, which is an efficient group rekeying scheme for secure multicast in ad-hoc networks protocol [19], which is considered the most complete work in the context of key management for ad hoc networks. The GKMPAN adopts a probabilistic key distribution technique, which is based on predeployed symmetric keys. The GKMPAN is efficient and scalable for wireless mobile networks, because it takes node mobility into consideration. In [20], a probabilistic random key distribution technique was proposed to achieve an efficient privacy-preserving group communication protocol for VANETs.

Although the security in VANETs was the focus of many works, only a few of them addressed the revocation problem. Golle *et al.* [21] proposed a technique for detecting and correcting malicious data. The main idea is that each vehicle, based on its sensor capabilities, could maintain a model for the network status that specifies the possible events in the network. In that model, the physics and safety control some rules, e.g., two vehicles cannot exist at the same location. Each vehicle could modify its network model based on the directly observed data. The received data from the other vehicles are accepted if it is consistent with the network model developed by the vehicle. If the data are inconsistent, then a heuristic-termed adversial parsimony is developed, where the vehicle looks for the simplest explanation for the inconsistency in the data assuming an attack that contains a small number of vehicles. Then, the vehicle ranks the possible explanations and accepts the data that agree with the highest rank explanation. It is shown that this technique can efficiently detect many attacks when accurate information about the vehicle positions can be obtained by methods other than communication, e.g., using sensors, cameras, etc. Although this technique is efficient for the detection and correction of malicious data, it cannot completely revoke the malicious vehicle that sent the data.

Raya *et al.* [22], [23] proposed an eviction technique consisting of the following components: centralized revocation of a node by the CA, localized misbehavior detection system (MDS), and local eviction of attackers by voting evaluators (LEAVE). In the centralized revocation of a node by the CA, two techniques were proposed: 1) revocation using compressed CRLs, where the traditional CRLs issued by the CA are adopted; however, a CRL is compressed using Bloom filters prior to broadcasting it; and 2) revocation of the tamper-proof device, which is used in the case in which all the certificates of a vehicle are to be revoked. In such a case, the CA sends a message to the tamper-proof device in the designated vehicle and informs it to stop all the security functions. MDS and LEAVE can be used to isolate misbehaving nodes before the revocation data from CA are available to all the vehicles. In MDS, the misbehavior that can be identified by monitoring specific parameters of a node and the data anomalies that do not follow any known pattern are distinguished. In LEAVE, a group of neighboring vehicles perform a voting on the misbehavior of a specific vehicle. If the accumulation of votes exceeds a predefined threshold, then a warning message is broadcast to the neighboring vehicles which informs them to ignore all the messages transmitted by the misbehaving vehicle. This way, the neighbors of a misbehaving vehicle can quarantine the misbehaving vehicle until a centralized revocation is issued by the CA. Although this method is effective for isolating malicious vehicles, it makes the revocation decision centralized by the CA, which renders the CA as a bottleneck.

Different from the aforementioned research works, we propose an EDR protocol, where a permanent revocation of a misbehaving vehicle is completely performed by its neighbors, hence alleviating the burden on the CA and simplifying the revocation process in VANETs.

III. Preliminaries

In this section, we give a brief overview of bilinear pairing [24], which is one of the foundations of the proposed protocol. Then, we present the system and security models adopted by the EDR protocol.

A. Bilinear Pairing

Let $\mathbb{G}_1$ denote an additive group of prime order q, and let $\mathbb{G}_2$ be a multiplicative group of the same order. Let P be a generator of $\mathbb{G}_1$, and let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear mapping with the following properties:

1) Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q$.
2) Nondegeneracy: $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$.
3) Symmetric: $\hat{e}(P, Q) = \hat{e}(Q, P)$ for all $P, Q \in \mathbb{G}_1$.
4) Admissible: The map $\hat{e}$ is efficiently computable.

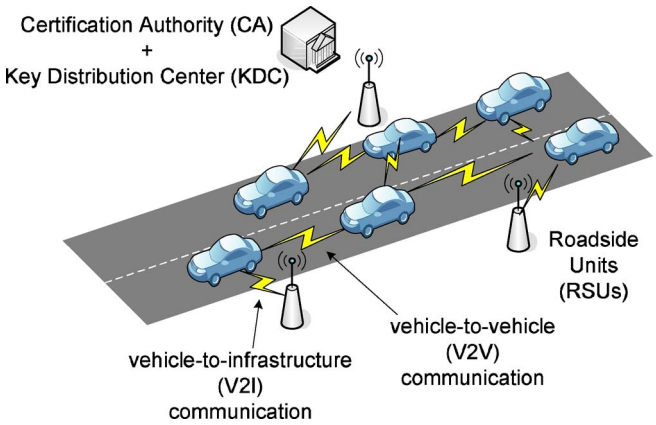

Fig. 1. System model.

The bilinear map $\hat{e}$ can be implemented using the Weil [25] and Tate [26] pairings on elliptic curves.

The security of the proposed protocol depends on solving the following hard computational problem:

- Elliptic Curve Discrete Logarithm Problem (ECDLP): Given a point P of order q on an elliptic curve and a point Q on the same curve, the ECDLP problem [27] is to determine the integer l, $0 \leq l \leq q-1$ such that $Q = lP$.

B. System Model

As shown in Fig. 1, the system model under consideration consists of the following:

1) a CA, which is responsible for generating initial certificates for all the vehicles in the network, which also acts as a key distribution center (KDC). Therefore, the CA is also responsible for distributing keys to all the vehicles in the network;
2) RSUs, which are fixed units distributed in the network. RSUs can securely communicate with the CA;
3) vehicles, which can communicate either with other vehicles through V2V communications or with the infrastructure RSUs through V2I communications.

It should be noted that the system model under consideration is mainly a PKI system, where each vehicle has a short-lifetime certificate used to secure its communication with other entities in the network.

C. Security Model

In this section, we outline the security model adopted by the EDR protocol as follows.

1) The CA is fully trusted by all the network entities. In addition, it has sufficient physical securing mechanisms such that it cannot be compromised by any attacker regardless of his capabilities.
2) The RSUs are fixed in place, and they are fully controlled by the CA. Moreover, the CA can instantly quarantine any compromised RSU.
3) Vehicles have abundant resources in computation and storage. In addition, vehicles can freely move in the network, and they can easily be compromised by an attacker.
4) Revoked vehicles can collude, trying to revoke a legitimate innocent vehicle.
5) Legitimate vehicles do not have sufficient incentives to disclose security materials to the revoked vehicles, i.e., legitimate vehicles cannot collaborate with the revoked vehicles.

D. Security Objectives

On the design of the EDR protocol, we aim at achieving the following security objectives.

1) *Resistance to forging attacks:* The generated revocation messages in the EDR protocol should be unforgeable such that any entity in the network must not be able to generate a fake revocation message, even if it has previously generated revocation messages.
2) *Resistance to collusion attacks:* The revoked vehicles must not be able to collude to revoke an innocent vehicle.
3) *Resistance to internal revocation-denial attacks:* A legitimate vehicle should not be able to deliberately fail the revocation process of a misbehaving vehicle.
4) *Resistance to external revocation-denial attacks:* An external attacker is defined as the attacker who has neither a valid certificate nor valid keys. An external attacker must be able to neither illegitimately share in any revocation process nor fail the revocation process of a misbehaving vehicle.

IV. Efficient Decentralized Revocation Protocol

The proposed protocol is based on the probabilistic random key distribution technique and a novel pairing-based threshold scheme.

A. System Initialization

The system is initialized as follows.

1) The CA issues a short-lifetime certificate for each vehicle in the network. Each vehicle can update its certificate from either the RSUs or the CA.
2) Initially, the CA selects a generator $P \in \mathbb{G}_1$ of order q and a key pool consisting of l keys, where each key $k_j \in \mathbb{Z}_q$ has a fixed identity $j \in \{1, 2, \ldots, l\}$. Each vehicle in the network randomly picks from the key pool a key set (R) consisting of m distinct keys.
3) The CA selects x random revocation secret keys $\mathrm{SK}_{\mathrm{SHARE}} = \{s_1, s_2, s_3, \ldots, s_x\} = \{s_i | 1 \leq i \leq x\}$ from the key pool such that $\sum_{i=1}^{x} s_i \bmod q = S$, where S is the secret key of the CA to sign a message and for all $i \in [1, x]$ $\exists\ j \in [1, l]$ such that $s_i = k_j$. It should be noted that S and the revocation secret keys are equivalent to a threshold scheme [28], where the key S is equivalent to the secret to be shared between multiple entities, and the revocation secret keys are equivalent to the shadows. In addition, the CA calculates the revocation public keys $\mathrm{PK}_{\mathrm{SHARE}} = \{\mathrm{PK}_{\mathrm{share}_1}, \mathrm{PK}_{\mathrm{share}_2}, \ldots, \mathrm{PK}_{\mathrm{share}_x}\} = \{\mathrm{PK}_{\mathrm{SHARE}_i} | 1 \leq i \leq x\} = \{(1/s_1)P, (1/s_2)P, \ldots, (1/s_x)P\} = \{(1/s_i)P | 1 \leq i \leq x\}$ that correspond to

the revocation secret keys $\mathrm{SK}_{\mathrm{SHARE}} = \{s_i | 1 \leq i \leq x\}$. In addition, the CA calculates its public key $P_\circ = \mathrm{SP}$ that corresponds to the private key S and chooses a hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$.

4) The CA announces H, $P_\circ$, $\mathrm{PK}_{\mathrm{SHARE}}$ and the key's identities (j's) corresponding to the revocation secret keys $\mathrm{SK}_{\mathrm{SHARE}}$ to all the vehicles.

After the system is initialized, each vehicle should have the following information:

1) a short-lifetime certificate;
2) a set of m keys;
3) the keys' identities (j's) corresponding to the revocation secret keys $\mathrm{SK}_{\mathrm{SHARE}} = \{s_i | 1 \leq i \leq x\}$;
4) the revocation public keys $\mathrm{PK}_{\mathrm{SHARE}} = \{\mathrm{PK}_{\mathrm{SHARE}_i} | 1 \leq i \leq x\}$;
5) the hash function H, P, and the public key $P_\circ$.

The main idea of the proposed protocol is to use the revocation secret keys $\mathrm{SK}_{\mathrm{SHARE}} = \{s_i | 1 \leq i \leq x\}$ to revoke the PKI certificate of any misbehaving vehicle.

B. Revocation Process

A misbehaving vehicle can be revoked as follows.

1) When a vehicle exhibits a misbehavior, its neighbors vote to revoke the misbehaving vehicle. The details of the voting scheme are not in the scope of this paper. However, the proposed protocol has a modular nature that makes it integrable with any voting scheme, e.g., the voting scheme proposed in [22]. When the voting exceeds a predefined threshold, the misbehaving vehicle should be revoked.
2) The vehicle, which accumulates votes exceeding the defined threshold to revoke a vehicle, takes the role of the revocation coordinator or one of the neighbors of the misbehaving vehicle volunteers to take the role of the revocation coordinator.
3) The revocation coordinator broadcasts to its one-hop neighboring vehicles a request to share in the revocation process and a message msg containing the certificate of the misbehaving vehicle, the reason for revocation, the current time stamp, the revocation coordinator signature on the entire message msg, and the revocation coordinator certificate.
4) Any vehicle receiving the request and the message msg verifies the signature of the revocation coordinator on msg using the revocation coordinator's public key contained in its certificate and checks the time stamp to ensure the freshness of the message msg. In addition, it searches its key set (R) for revocation secret keys belonging to $\mathrm{SK}_{\mathrm{SHARE}} = \{s_i | 1 \leq i \leq x\}$. For each possessed revocation secret key, it calculates its revocation share as $\mathrm{Rev}_i = s_i H(\mathrm{msg}) \in \mathbb{G}_1$, where $i \in \{1, 2, 3, \ldots, x\}$, and sends $(i \| \mathrm{Rev}_i)$ to the revocation coordinator.
5) When the revocation coordinator receives any revocation share $(i \| \mathrm{Rev}_i)$ calculated by a revocation secret key i, it uses the corresponding revocation public key ($\mathrm{PK}_{\mathrm{share}_i} = (1/s_i)P$) to verify the received revocation share by checking that $\hat{e}(\mathrm{Rev}_i, \mathrm{PK}_{\mathrm{share}_i}) = \hat{e}(H(\mathrm{msg}), P)$. This verification holds since

$$\begin{aligned}\hat{e}(\mathrm{Rev}_i, \mathrm{PK}_{\mathrm{share}_i}) &= \hat{e}\left(s_i H(\mathrm{msg}), \frac{1}{s_i} P\right) \\ &= \hat{e}\left(H(\mathrm{msg}), P\right)^{s_i \cdot \frac{1}{s_i}} \\ &= \hat{e}\left(H(\mathrm{msg}), P\right). \end{aligned} \tag{1}$$

If the revocation share Rev_i does not pass verification, it is immediately rejected and dropped. Instead of verifying the revocation shares one by one, the revocation coordinator can wait until the revocation shares corresponding to all the x revocation secret keys are received; then, it can simultaneously verify all the x revocation shares by checking that

$$\hat{e}\left(\sum_{i=1}^{x} \mathrm{Rev}_i, P\right) = \hat{e}\left(H(\mathrm{msg}), P_\circ\right). \tag{2}$$

This verification holds since

$$\begin{aligned}\hat{e}\left(\sum_{i=1}^{x} \mathrm{Rev}_i, P\right) &= \hat{e}(\mathrm{Rev}_1 + \mathrm{Rev}_2 + \cdots + \mathrm{Rev}_x, P) \\ &= \hat{e}(\mathrm{Rev}_1, P)\hat{e}(\mathrm{Rev}_2, P) \cdots \hat{e}(\mathrm{Rev}_x, P) \\ &= \hat{e}\left(s_1 H(\mathrm{msg}), P\right) \hat{e}\left(s_2 H(\mathrm{msg}), P\right) \\ &\quad \cdots \hat{e}\left(s_x H(\mathrm{msg}), P\right) \\ &= \hat{e}\left(H(\mathrm{msg}), P\right)^{s_1} \hat{e}\left(H(\mathrm{msg})P, P\right)^{s_2} \\ &\quad \cdots \hat{e}\left(H(\mathrm{msg}), P\right)^{s_x} \\ &= \hat{e}\left(H(\mathrm{msg}), P\right)^{s_1 + s_2 + \cdots + s_x} \\ &= \hat{e}\left(H(\mathrm{msg}), P\right)^{S} \\ &= \hat{e}\left(H(\mathrm{msg}), \mathrm{SP}\right) \\ &= \hat{e}\left(H(\mathrm{msg}), P_\circ\right). \end{aligned}$$

6) When the revocation coordinator receives and correctly verifies all the required revocation shares, i.e., $\mathrm{Rev}_1, \mathrm{Rev}_2, \ldots, \mathrm{Rev}_x$, the revocation coordinator computes the total revocation message signature as

$$\mathrm{Rev} = \sum_{i=1}^{x} \mathrm{Rev}_i.$$

The total revocation message signature Rev can be verified as follows:

$$\hat{e}(\mathrm{Rev}, P) = \hat{e}\left(H(\mathrm{msg}), P_\circ\right). \tag{3}$$

The proof of (3) directly follows from the proof of (2). It should be noted that the CA is also able to revoke any vehicle using its secret revocation key (S) by directly calculating the total revocation message signature $\mathrm{Rev} = \mathrm{SH}(\mathrm{msg})$. The total revocation message signature issued by the revocation coordinator is identical to that issued by the CA. Hence, the revocation message signature Rev, which was generated by either the CA or the revocation coordinator, can be verified by any vehicle using the CA public key $P_\circ$, as indicated in (3). As a result, a vehicle verifying Rev notices no difference between the

verification of the revocation messages transmitted by the revocation coordinators and those transmitted by the CA.

7) The revocation coordinator broadcasts a certificate revocation message $\text{Cert}_{\text{rev}} = \{\text{msg}\|\text{Rev}\|T_{\text{stamp}}\|\text{sgn}_{\text{coord}}\}$ to the neighboring vehicles, where T_{stamp} is the current time stamp, and $\text{sgn}_{\text{coord}}$ is the signature of the revocation coordinator on $(\text{msg}\|\text{Rev}\|T_{\text{stamp}})$. Note that the certificate of the revocation coordinator is included in the message msg.
8) Any vehicle receiving Cert_{rev} checks the freshness of the time stamp T_{stamp} compared with that in msg to ensure that the revocation process is done in a timely manner, verifies the signature of the coordinator $\text{sgn}_{\text{coord}}$ using the coordinator's public key included in its certificate, and validates Rev, as shown in (3). Any vehicle verifying Cert_{rev} correctly forwards it to other vehicles. The dissemination of Cert_{rev} continues until the lifetime of the revoked certificate ends.
9) When any RSU captures the message Cert_{rev}, it checks the message validity and then forwards the message to the CA, which keeps a list of the revoked vehicles.

Since the message Cert_{rev} is broadcast to all the vehicles in the neighborhood of the revoked vehicle, all the neighboring vehicles ignore the messages from the revoked vehicle.

C. Vehicle Rekeying

All the keys of the revoked vehicles are considered compromised. The rekeying process is triggered by the CA when the number of compromised keys in the key pool or when the number of the compromised revocation secret keys exceeds a predefined threshold. All the legitimate vehicles must securely update their compromised keys [19]. The rekeying process is as follows.

1) The CA searches its database to determine the identity (M) of the noncompromised key k_M that is shared by the majority of the unrevoked vehicles. The CA then generates an intermediate key $k_{\text{im}} = f(k_M) \in \mathbb{Z}_q^*$, where f is a family of pseudorandom functions, which is unique and publicly known to all the network entities. This intermediate key is used by all the vehicles to update their compromised keys. In addition, the CA calculates the updated revocation public key(s) corresponding to the compromised revocation secret key(s) $s_i = k_j$ as $\text{PK}'_{\text{share}_i} = (1/f_{k_{\text{im}}}(s_i))P$ and its new secret key $S' = \sum_{i=1}^{x} s'_i \bmod q$, where

$$s'_i = \begin{cases} f_{k_{\text{im}}}(s_i), & \text{if } s_i \text{ is compromised} \\ s_i, & \text{otherwise.} \end{cases}$$

In addition, the CA calculates its new public key $P'_\circ = S'P$. After that, the CA broadcasts a key update message

$$\text{Kmsg} = \left(M\|\text{IDrev}_{\text{vehicle}}\|\text{IDrev}_{\text{key}}\left\|\{\text{PK}'_{\text{share}_i}\}\right\|P'_\circ\right)$$

where $\text{IDrev}_{\text{vehicle}}$ is a list of the identities of the revoked vehicles, $\text{IDrev}_{\text{key}}$ is a list of the identities of the revoked keys, $\{\text{PK}'_{\text{share}_i}\}$ is the set of updated revocation public keys, and $P'_\circ$ is the CA new public key corresponding to the new secret key S'. The CA also sends with the previous message its signature $\text{sgn}_{\text{Kmsg}} = \text{SH}(\text{Kmsg})$ on the message Kmsg.

2) After receiving the message Kmsg and the signature sgn_{Kmsg}, each vehicle verifies the received message as $\hat{e}(\text{sgn}_{\text{Kmsg}}, P) = \hat{e}(H(\text{Kmsg}), P_\circ)$. This verification holds since

$$\begin{aligned} \hat{e}(\text{sgn}_{\text{Kmsg}}, P) &= \hat{e}\left(\text{SH}(\text{Kmsg}), P\right) \\ &= \hat{e}\left(H(\text{Kmsg}), \text{SP}\right) \\ &= \hat{e}\left(H(\text{Kmsg}), P_\circ\right). \end{aligned}$$

If the message is correctly verified, the vehicle checks if it has k_M or not. If yes, then the vehicle independently computes the intermediate key k_{im}.

3) When a vehicle v does not have the key k_M, it will not be able to update its compromised keys and must get k_{im} from its neighboring vehicles. The vehicle v broadcasts its certificate and a request to get k_{im}, and starts its own timer.
4) Any neighboring vehicle of vehicle v having k_{im} uses the public key of the vehicle v, which is included in its certificate, to encrypt the intermediate key k_{im} and sends the encrypted k_{im} to vehicle v.
5) When vehicle v receives the encrypted k_{im}, it uses its secret key to decrypt k_{im}. Otherwise, if the timer of the vehicle v is timed out without receiving the required data, then go to step 3.
6) The revoked vehicles cannot compute k_{im} since they do not have k_M. In addition, they cannot receive k_{im} from other vehicles since the key update message contains the identities of the revoked vehicles, which prevents others from forwarding k_{im} to them.
7) When a vehicle possesses a key k_j that is contained in the revoked vehicle key sets, i.e., compromised key, it updates the compromised key as follows:

$$k'_j = f_{k_{\text{im}}}(k_j).$$

8) After performing the key set update, each vehicle erases k_{im}, the original compromised revocation public keys $\text{PK}_{\text{share}_i}$'s, and the original compromised keys k_j's.

Remarks:

1) Note that if a vehicle missed a rekeying process, it is still able to share in the upcoming revocation processes since only the compromised keys are updated; hence, it can use its noncompromised revocation secret key(s) in the future. However, if the number of missed rekeying processes increases, then it may be necessary for the vehicle to contact the CA through RSUs to get the required security materials to update its key set.
2) It is clear that only one key update message is broadcasted after several revocations took place. Consequently, the number of messages broadcasted by the CA is substantially reduced compared with the centralized revocation scheme, where the CA has to broadcast a message for each revocation process. It should be noted that the

TABLE I
NOTATIONS

Symbol	Notation
l	the key pool size of the key server
m	the key set size stored in each vehicle
N	the number of the collaborating vehicles to revoke a vehicle
x	the number of the revocation secret keys
P_x	the probability of having at least one key of x
P_{rev}	the revocation success probability
w	the number of revoked keys
P_{half}	the probability that at least half of the x keys are safe when there are w keys revoked
$P_{rev}(w)$	the probability that the revocation is successful and it is performed with at least $(x/2)$ non-compromised revocation secret keys

rekeying process can be done after every revocation process to increase the security level of the proposed protocol. However, this results in increasing the communication overhead of the rekeying process.

3) It should be noted that the EDR protocol has a modular nature, which makes it integrable with any PKI system. In other words, the proposed protocol does not require any modification to the core of the PKI architecture, but all that is needed to implement the proposed protocol is to add a KDC to the CA.
4) The EDR protocol can be used as a stand-alone revocation method, or it can coexist with the conventional CRL revocation method, where the proposed protocol helps to revoke the misbehaving vehicles in geographic areas where RSUs are not available.
5) The EDR protocol is suitable not only for VANETs but also for any type of network employing PKI as well.

V. Performance Analysis

In this section, we evaluate the performance of the EDR protocol in terms of its feasibility and reliability. The notations used throughout the rest of this section are given in Table I.

A. Probability of Having at Least One Revocation Secret Key

The probability P_x of having at least one key of the revocation secret keys (x) in the key set of a vehicle can be calculated as

$$P_x = 1 - \frac{\binom{l-x}{m}}{\binom{l}{m}}. \tag{4}$$

Fig. 2 shows P_x as a function in x. It can be seen that P_x increases as x and m increase and l decreases. This can be explained as follows: For a fixed x, the probability that a vehicle has at least one revocation secret key increases with the number of keys (m) a vehicle gets from the key pool. A similar analogy applies to the number of the revocation secret keys x and the key pool size l. Therefore, a desired value for P_x can be achieved by appropriately selecting the values of l, m, and x.

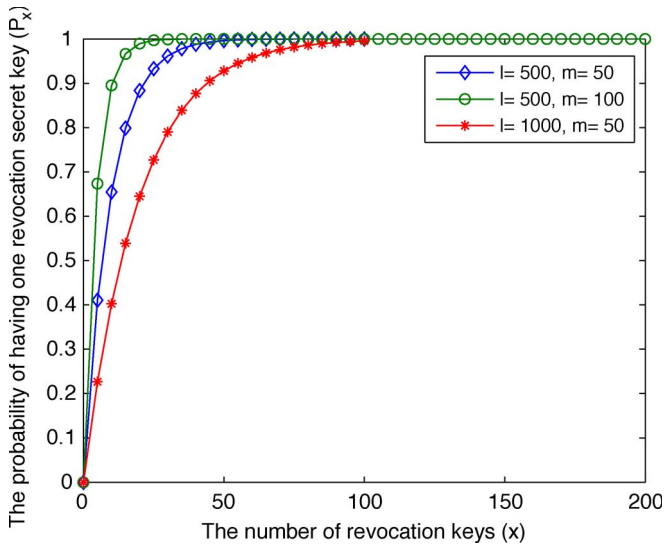

Fig. 2. Probability (P_x) of having at least one key out of x in the key set of a vehicle.

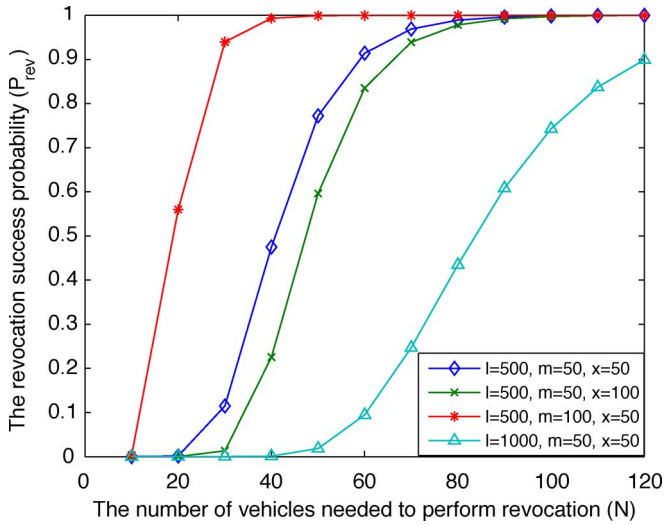

Fig. 3. Revocation success probability $P_{\rm rev}$.

B. Revocation Success Probability

In this section, we are interested in calculating the revocation success probability $P_{\rm rev}$, which is defined as the probability that any N collaborating vehicles have all the required revocation secret keys (x) to revoke a vehicle. The revocation success probability $P_{\rm rev}$ can be calculated as

$$P_{\rm rev} = \left(1 - \frac{\binom{l-1}{m}^N}{\binom{l}{m}^N}\right)^x. \tag{5}$$

Fig. 3 shows the relation between the revocation success probability $(P_{\rm rev})$ and the number of the collaborating vehicles (N) for different values of l, m, and x. It can be seen that for a constant l, m, and x, $P_{\rm rev}$ increases as N increases. Generally speaking, the value of N should be set according to the real-life measurements of the average number of vehicles within the communication range of a vehicle. In addition, it can be seen that $P_{\rm rev}$ increases as the vehicle key set size (m) increases

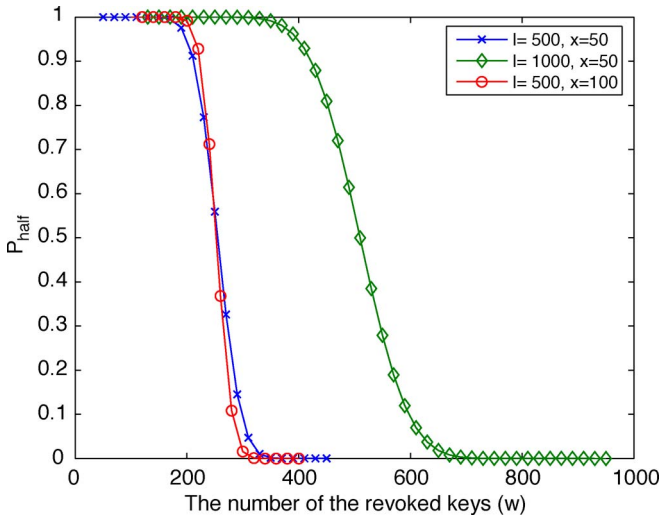

Fig. 4. Probability that at least half of the revocation secret keys are safe.

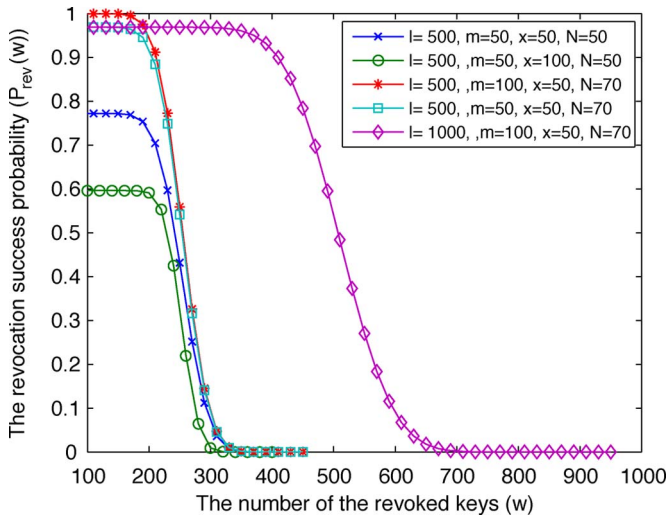

Fig. 5. Revocation success probability with at least half of the revocation secret keys being safe.

and the size of the key pool (l) decreases. This is due to the fact that increasing m or decreasing l increases the probability P_x of having at least one revocation secret key in the key set of a vehicle, as seen in Fig. 2, hence increasing the probability for each vehicle of the neighbors of a misbehaving vehicle to share in the revocation process, which increases the revocation success probability $P_{\rm rev}$. However, increasing the value of m results in increasing the vulnerability of the system, because the more keys a single vehicle has, the more information an attacker can get by compromising a single vehicle. In addition, decreasing the value of l results in lowering the security, because an attacker gets more information about the key pool if a few number of vehicles are compromised.

From the above discussion, the values of l, m, N, and x and the desired security level should carefully be selected to get the desired value of $P_{\rm rev}$.

C. Impact of the Number of Revoked Keys

In this section, we study the effect of revoking w keys on the safety of the revocation secret keys and the revocation success probability.

To ensure the correctness of the revocation process, we set the following requirement: At least half of the revocation secret keys sharing in the revocation of a vehicle must be noncompromised. It should be noted that the keys of any revoked vehicle are considered compromised. The probability $P_{\rm half}$ that at least half of the x revocation secret keys are safe can be calculated as a function of the number of the revoked keys w as follows:

$$P_{\rm half} = \sum_{i=x/2}^{x} \frac{\binom{l-w}{i} \cdot \binom{w}{x-i}}{\binom{l}{x}}. \quad (6)$$

Fig. 4 shows the relation between $P_{\rm half}$ and the number of revoked keys w. It can be seen that changing x has a slight effect on $P_{\rm half}$ because the number of revocation secret keys (x) is relatively small compared with the number of keys (l) in the key pool, which alleviates the effect of revoking keys from the key pool on the safety of the revocation secret keys. In addition, $P_{\rm half}$ decreases as w increases, and $P_{\rm half}$ increases as l increases.

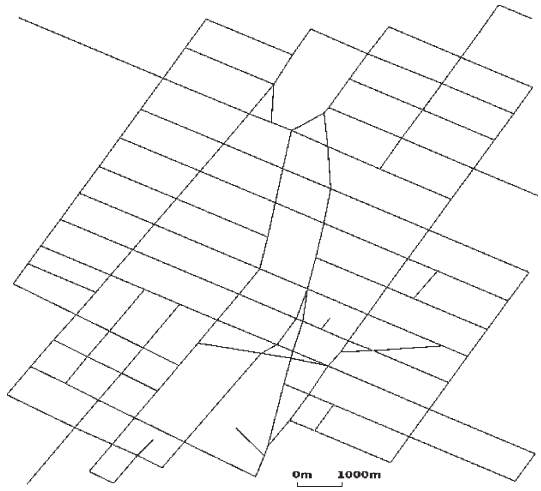

Fig. 6. City street simulation scenario.

The probability $P_{\rm rev}(w)$ that the revocation is successful and it is performed by at least $(x/2)$ noncompromised revocation secret keys is as follows:

$$P_{\rm rev}(w) = P_{\rm rev} \cdot P_{\rm half} = \left(1 - \frac{\binom{l-1}{m}^N}{\binom{l}{m}^N}\right)^x \cdot \sum_{i=x/2}^{x} \frac{\binom{l-w}{i} \cdot \binom{w}{x-i}}{\binom{l}{x}}. \quad (7)$$

Fig. 5 shows the relation between $P_{\rm rev}(w)$ and w. It can be seen that $P_{\rm rev}(w)$ decreases as w increases. In addition, $P_{\rm rev}(w)$ increases as m and N increase, and $P_{\rm rev}(w)$ decreases as l and x increase.

D. Revocation Delay

In this section, we evaluate the revocation delay of the EDR protocol and the conventional CRL by conducting ns-2 [29] simulation for the city street scenario shown in Fig. 6.

TABLE II
NS-2 SIMULATION PARAMETERS

simulation area	$13.4\ Km \times 12.3\ Km$
simulation time	$100\ sec$
max. vehicle speed	$60\ Km/h$
vehicle transmission range	$300\ m$
vehicle information dissemination interval	$300\ msec$
number of vehicles	4486
wired channel capacity	100 Mb/s
wireless channel capacity	6 Mb/s
number of RSUs	576
distribution of RSUs	uniform
key pool size l	500
vehicle key set size m	100
number of revocation secret keys x	20

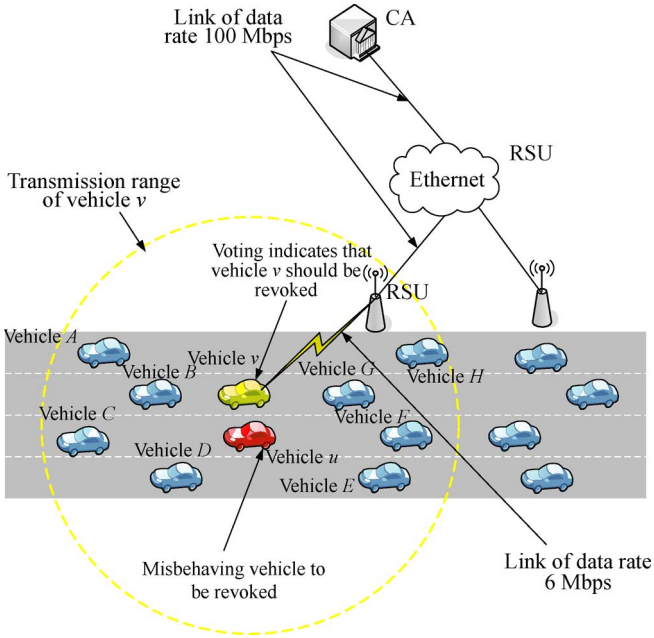

Fig. 7. Different revocation scenarios.

The adopted simulation parameters are given in Table II. The mobility traces adopted in this simulation are generated using TraNS [30].

VANETs have two types of links: 1) wireless links connecting vehicles to each other and to the RSUs and 2) wired links connecting the RSUs and the CA, as shown in Fig. 7. According to the dedicated short-range communication (DSRC) specifications, each wireless data channel in VANET has a bandwidth of 10 MHz that corresponds to a channel data rate in the range of 3–27 Mb/s [31]. We select a data rate of 6 Mb/s for the wireless channels in a VANET. The RSUs are connected via Ethernet to the CA [3]. We consider the links of the Ethernet connecting the RSUs and CA to have a data rate of 100 Mb/s. The RSU connection pattern employed in our simulation is shown in Fig. 8. The adopted RSU connection considers a well-deployed VANET, where the RSUs are uniformly distributed with the distance between any pair of adjacent RSUs is 500 m. The CA is located at the top-left corner of the city scenario shown in Fig. 6. To simulate real-life revocation scenarios,

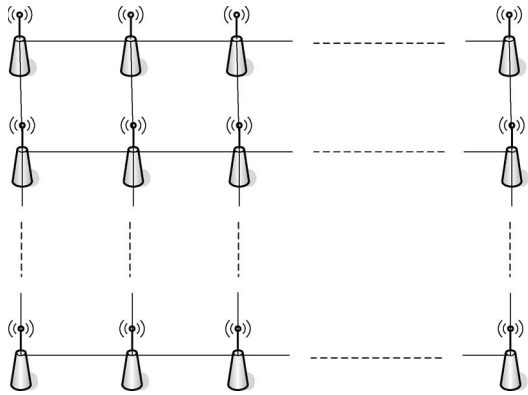

Fig. 8. RSU connection pattern.

we conduct revocation scenarios imposed on VANET safety-related applications, where each vehicle has to disseminate information about the road condition every 300 ms according to DSRC.

In the conducted simulation, we consider the cryptography delay only due to pairing and point multiplication operations on an elliptic curve, as they are the most time-consuming operations in the proposed protocol and the conventional CRL. Let T_{pair} and T_{mul} denote the time required to perform a pairing operation and a point multiplication, respectively. In [32], T_{pair} and T_{mul} are found for an MNT [33] curve with embedding degree $k = 6$ that is equal to 4.5 and 0.6 ms, respectively. Elliptic curve digital signature algorithm [34] is the digital signature method chosen by the VANET standard IEEE1609.2, where a certificate and signature verification takes $4T_{\text{mul}}$, and a signature generation takes T_{mul}.

We consider two revocation scenarios, as shown in Fig. 7. The first scenario is the conventional CRL revocation method combined with a generic voting scheme. In Fig. 7, vehicle u is misbehaving, and the accumulation of votes in vehicle v reaches the threshold where a revocation of vehicle u should be performed. Hence, vehicle v should send a revocation request to the CA via the nearest RSU. After the request reaches the nearest RSU, the request will be forwarded through the RSUs' Ethernet to the CA, where the request message experiences a delay of $4T_{\text{mul}}$ at each intermediate RSU, as each RSU has to verify the certificate and the signature of the sender before forwarding the request. When the revocation request reaches the CA, it has to verify the request, which takes $4T_{\text{mul}}$, and generate a new signed CRL, which takes T_{mul}. In VANETs, the most important issue in any revocation method is the delay of delivering the revocation message to the neighboring vehicles of a misbehaving vehicle to prevent that misbehaving vehicle from jeopardizing the safety of its neighbors. Consequently, the CRL total revocation delay T_{CRL} is the delay from the moment a vehicle issues a revocation request until the moment the new CRL is broadcast in the geographic area containing vehicle u.

The second scenario is the EDR protocol. In Fig. 7, when the accumulation of votes in vehicle v exceeds the threshold where a revocation of vehicle u should be performed, vehicle v acts as the revocation coordinator and sends a revocation request to the neighboring vehicles located within one hop connectivity (vehicles $A, B, C, \ldots$, and H in Fig. 7). Any

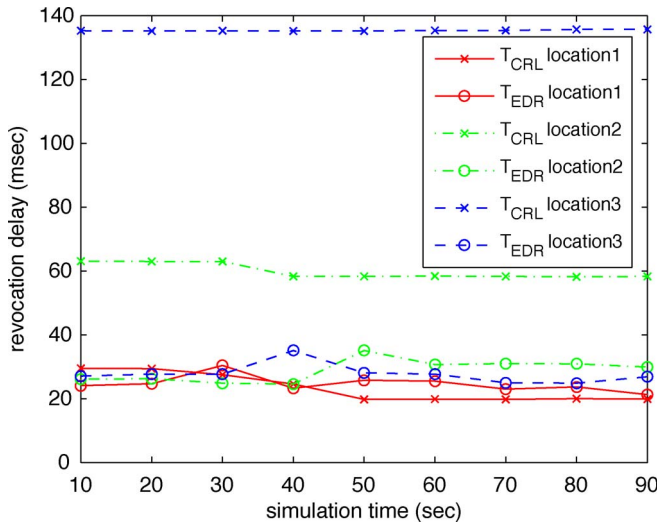

Fig. 9. Revocation delay for different revocation scenarios.

vehicle receiving the revocation request and having a revocation secret key verifies the request, which takes $4T_{\text{mul}}$, calculates its revocation share Rev_i, which takes T_{mul}, and broadcasts its revocation share. When the revocation coordinator receives the required revocation shares to calculate the final revocation message of vehicle u, it verifies all the revocation shares using (2), which takes $2T_{\text{pair}} + T_{\text{mul}}$, and then, it calculates the final revocation message. Finally, vehicle v broadcasts the final revocation messages to its neighboring vehicles. Consequently, the EDR revocation delay T_{EDR} is the delay from the moment the revocation coordinator issues a revocation request until the moment the revocation of vehicle v is broadcast in the geographic area containing vehicle u.

Fig. 9 shows the CRL revocation delay T_{CRL} and the EDR revocation delay T_{EDR} in milliseconds versus the simulation time. We conducted simulation for the two revocation scenarios triggered by the vehicle v at three different locations, i.e., location1, location2, and location3, which correspond to initial distances of 2.7, 4.7, and 10.3 km, respectively, from the CA at the beginning of the simulation. The revocation process is triggered every 10 s during the simulation, and the corresponding revocation delay is measured. The variations in T_{CRL} is due to the number of intermediate RSUs existing in the connection between the CA and the vehicle sending the revocation request. In addition, the variations in T_{EDR} are due to the variation in the number of neighboring vehicles of the revocation coordinator. It can be seen that T_{EDR} is almost the same for the three locations and is confined within the range of 21–35 ms. This is due to the fact that the proposed protocol is independent on the CA. On the other hand, it can be seen that T_{CRL} increases with the distance from the CA. Consequently, the delay saving of the proposed EDR protocol compared with the conventional CRL revocation increases with distance from the CA. For example, the average CRL revocation delay is 59.87 ms for location2, whereas the average CRL revocation delay for the EDR protocol is 28.83 ms. Consequently, the EDR protocol decreases the revocation delay by 51.85% compared with the conventional CRL in that case. It should be noted that T_{EDR} and T_{CRL} correspond to the vulnerability window that a misbehaving vehicle has until it is revoked for the EDR protocol and CRL, respectively. During the vulnerability window, the misbehaving vehicle can still jeopardize the safety of the neighboring vehicles. It can be seen that the EDR protocol has a small vulnerability window compared with the CRL technique, which increases the safety level in VANET.

VI. Security Analysis

In this section, we analyze the proposed protocol against the achieved security objectives in Section III-D. It should be noted that these security objectives combat most of the common revocation attacks.

1) Resistance to Forging Attacks: To forge the revocation share $\text{Rev}_i = s_i H(\text{msg})$ of any vehicle, an attacker has to solve the following ECDLP problem: Given $H(\text{msg})$ and Rev_i, find s_i such that $\text{Rev}_i = s_i H(\text{msg})$. A similar analogy applies to finding the CA secret key S from the total revocation message signature $\text{Rev} = \text{SH}(\text{msg})$. Since ECDLP is a hard computational problem, i.e., it cannot be solved in a subexponential time, the revocation shares and the total revocation message signature Rev are unforgeable. Similarly, finding the CA secret value S from $P_\circ = \text{SP}$ is an ECDLP problem, which makes it unforgeable. Furthermore, the revocation request sent by the revocation coordinator to his neighboring vehicles is unforgeable since this request is signed by the revocation coordinator. From the aforementioned discussion, the EDR protocol is resistant to forging attacks.

2) Resistance to Collusion Attacks: According to the EDR protocol, the rekeying process is performed before the number of compromised revocation secret keys exceed half the total number of revocation secret keys. Therefore, it is guaranteed that the revoked vehicles can never have all the revocation secret keys; hence, they cannot collude to revoke any vehicle. Consequently, the EDR protocol is resistant to collusion attacks. Moreover, the key update in each vehicle mainly depends on the intermediate key k_{im}, which cannot be generated by any revoked vehicle. In addition, any compromised vehicle cannot lead to the old k_{im}'s since after each rekeying process, each vehicle erases the current k_{im}. As a result, the revoked vehicles are able to neither update their keys nor share in future revocation processes.

3) Resistance to Internal Revocation–Denial Attacks: When a legitimate vehicle deliberately sends an erroneous revocation share to fail the revocation process, the revocation coordinator immediately detects and discards the erroneous revocation share, as it will fail to pass the revocation share verification in (1). In addition, since the EDR protocol adopts a probabilistic key distribution technique, the same revocation secret key may be found with more than one vehicle. Consequently, the revocation coordinator may receive multiple copies of the same revocation share Rev_i, alleviating the effect of a vehicle intentionally choosing not to send its revocation share. Hence, the EDR protocol exhibits robust performance against internal revocation-denial attacks.

4) Resistance to External Revocation–Denial Attacks: When an external attacker tries to send a fake revocation share during the revocation process, the revocation coordinator will

immediately detect and exclude the fake revocation share as it will fail to pass the revocation share verification in (1). Consequently, the EDR protocol is resistant to external revocation-denial attacks.

VII. Conclusion

In this paper, we have proposed a robust EDR protocol for VANETs that substantially reduces the complexity of the certificate revocation problem while achieving fast revocation of the misbehaving vehicles. The EDR protocol decreases the vulnerability window that a misbehaving vehicle has, resulting in a higher safety level for VANET. The EDR protocol is resistant to the most known revocation attacks. In addition, it can efficiently be integrated with any PKI and/or any misbehavior-detection scheme for VANETs. Our future work will focus on classifying the received messages according to their correctness degree and triggering the revocation process for a misbehaving vehicle transmitting malicious messages.

Acknowledgment

The authors would like to thank R. Lu for his valuable comments on this paper.

References

[1] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proc. 3rd ACM Workshop Security Ad Hoc Sens. Netw.*, 2005, pp. 11–21.
[2] F. Dötzer, "Privacy issues in vehicular ad hoc networks," in *Proc. 2nd ACM Workshop Veh. Ad Hoc Netw.*, Sep. 2006, pp. 197–209.
[3] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages*, IEEE Std. 1609.2-2006, 2006.
[4] K. P. Laberteaux, J. J. Haas, and Y. Hu, "Security certificate revocation list distribution for VANET," in *Proc. 5th ACM Int. Workshop Veh. Inter-NETw.*, 2008, pp. 88–89.
[5] P. P. Papadimitratos, G. Mezzour, and J. Hubaux, "Certificate revocation list distribution in vehicular communication systems," in *Proc. 5th ACM Int. Workshop Veh. Inter-NETw.*, 2008, pp. 86–87.
[6] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks," in *Proc. ICC*, 2008, pp. 1451–1457.
[7] J. Y. Choi, M. Jakobsson, and S. Wetzel, "Balancing auditability and privacy in vehicular networks," in *Proc. 1st ACM Int. Workshop Quality Service Security Wireless Mob. Netw.*, 2005, pp. 79–87.
[8] Y. Xi, K. Sha, W. Shi, L. Schwiebert, and T. Zhang, "Enforcing privacy using symmetric random key-set in vehicular networks," in *Proc. ISADS*, 2007, pp. 344–351.
[9] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," in *Proc. 4th ACM Int. Workshop Veh. Ad Hoc Netw.*, 2007, pp. 19–28.
[10] J. Guo, J. Baugh, and S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," in *Proc. Mobile Netw. Veh. Environ.*, 2007, pp. 103–108.
[11] K. Sha, Y. Xi, W. Shi, L. Schwiebert, and T. Zhang, "Adaptive privacy-preserving authentication in vehicular networks," in *Proc. ChinaCom*, 2006, pp. 1–8.
[12] X. Sun, X. Lin, and P.-H. Ho, "Secure vehicular communications based on group signature and id-based signature scheme," in *Proc. ICC*, 2007, pp. 1539–1545.
[13] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.
[14] A. Wasef, Y. Jiang, and X. Shen, "ECMV: Efficient certificate management scheme for vehicular networks," in *Proc. IEEE GLOBECOM*, 2008, pp. 1–5.
[15] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. INFOCOM*, 2008, pp. 1229–1237.
[16] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. IEEE Symp. Security Privacy*, 2003, pp. 197–213.
[17] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. ACM Conf. Comput. Commun. Security*, 2002, pp. 41–47.
[18] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing pairwise keys for secure communication in ad hoc networks: A probabilistic approach," in *Proc. 11th IEEE Int. Conf. Netw. Protocols*, 2003, pp. 326–335.
[19] S. Zhu, S. Setia, S. Xu, and S. Jajodia, "GKMPAN: An efficient group rekeying scheme for secure multicast in ad-hoc networks," *J. Comput. Secur.*, vol. 14, no. 4, pp. 301–325, 2006.
[20] A. Wasef and X. Shen, "PPGCV: Privacy preserving group communications protocol for vehicular ad hoc networks," in *Proc. ICC*, 2008, pp. 1458–1463.
[21] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proc. 1st ACM Int. Workshop Veh. Ad Hoc Netw.*, 2004, pp. 29–37.
[22] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1557–1568, Oct. 2007.
[23] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, and J. P. Hubaux, "Certificate revocation in vehicular networks," Swiss Fed. Inst. Technol., Lausanne, Switzerland, Tech. Rep. LCA-Rep.-2006-006, 2006.
[24] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. 21st Annu. Int. Cryptology Conf. Adv. Cryptology*, 2001, pp. 213–229.
[25] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, Sep. 2004.
[26] M. Scott, "Computing the Tate pairing," in *Topics in Cryptology*. Berlin, Germany: Springer-Verlag, 2005, pp. 293–304.
[27] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs Codes Cryptogr.*, vol. 19, no. 2/3, pp. 173–193, Mar. 2000.
[28] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.
[29] *The Network Simulator—ns-2*. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/User_Information
[30] *Traffic and Network Simulation Environment—TraNS*. [Online]. Available: http://trans.epfl.ch/
[31] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *Proc. 1st ACM Int. Workshop Veh. Ad Hoc Netw.*, 2004, pp. 1–9.
[32] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, 2008, pp. 246–250.
[33] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reductions," *IEIC Tech. Rep.*, vol. 100, no. 323, pp. 99–108, 2000.
[34] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.

Albert Wasef (M'09) received the B.Sc. and M.Sc. degrees in electrical communication engineering from El Menoufia University, El Menoufia, Egypt, in 1998 and 2003, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

He is currently with the Broadband Communications Research (BBCR) Group, Department of Electrical and Computer Engineering, University of Waterloo. His research interest includes wireless network security, privacy preservation in vehicular networks, and group communications.

Xuemin (Sherman) Shen (M'97–SM'02–F'09) received the B.Sc. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees in electrical engineering from Rutgers University, Camden, NJ, in 1987 and 1990, respectively.

He is currently a Professor and a University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on mobility and resource management in interconnected wireless/wired networks, ultrawideband wireless communication networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks. He is the coauthor of three books and has published more than 400 papers and book chapters on wireless communications and networks, control, and filtering.

Dr. Shen is a Registered Professional Engineer in Ontario and a Distinguished Lecturer of the IEEE Communications Society. He served as the Tutorial Chair for the 2008 IEEE International Conference on Communications, the Technical Program Committee Chair for the 2007 IEEE Global Communication Conference, the General Co-Chair for the 2007 International Conference on communications and networking in China and the 2006 International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, and the Founding Chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He serves as a Founding Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, Editor-in-Chief of *Peer-to-Peer networking and Applications*, Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the *KICS/IEEE Journal of Communications and Networks*, *Computer Networks, ACM/Wireless Networks, Wireless Communications and Mobile Computing* (Wiley), etc. He has also served as Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, *IEEE Communications Magazine*, *ACM Mobile Networks and Applications*, etc. He received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award in 2003 from the Province of Ontario, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo.