This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3004459, IEEE Transactions on Vehicular Technology

ESAC: An Efficient and Secure Access Control Scheme in Vehicular Named Data Networking

Shunrong Jiang*, Jianqing Liu⋆, Liangmin Wang†, Yong Zhou*, and Yuguang Fang§
*The Department of Information Science, China University of Mining and Technology, Xuzhou 221116, China
⋆Department of Electrical and Computer Engineering, The University of Alabama in Huntsville, AL, USA
†The Department of Cyberspace Security, Jiangsu University, Zhenjiang 212013, China
§Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, 32611, USA
Email:jsywow@gmail.com, jianqing.liu@uah.edu, wanglm@ujs.edu.cn, yzhou@cumt.edu.cn, fang@ece.ufl.edu

***Abstract*—In Vehicular Ad Hoc Networks (VANETs), the mobility of vehicles and urban obstacles may make the traditional IP-based content delivery mechanism work sluggishly. To address this problem, we adopt the Vehicular Named Data Networking (V-NDN) architecture to enable efficient content delivery. However, secure access control and incentive design are rarely taken into account in existing solutions. In this paper, we propose an efficient and secure access control (ESAC) scheme for content delivery in V-NDN. Specifically, we construct the proxy re-encryption method to achieve access control and data confidentiality, while using pseudonyms and the identify-based signature to ensure anonymous authentication and content integrity. Furthermore, the revocation of illegal vehicles and updating operations are performed by the constructed proxy re-encryption method, which significantly reduces communication overhead. Finally, an incentive scheme is designed to improve the utility of NDN in VANETs. The security analysis shows that ESAC can meet the security requirements of the content delivery in VANETs while significantly lowering the overhead.**

I. Introduction

To improve the traffic safety, road environment, and infotainment dissemination, VANETs [2] are proposed for drivers and passengers. The nature of VANET is that the mobility of vehicles and urban obstacles may cause frequent network changes and connectivity disruptions [3]. Consequently, traditional IP-based communications may not work well for content delivery due to its inability to deal with continuous disruptions and topology changes.

We consider a specific scenario where the Service Provider ($\mathcal{SP}$) sells digital contents in VANETs like e-books, music, and movies. Since the communications among vehicles are dynamic and intermittently available, a favorable choice is that $\mathcal{SP}$ pre-fetches the traffic demands using in-network caching. To achieve this, we can use Named Data Networking (NDN) [4, 5] as an alternative to realize data delivery/forwarding in VANETs [6–10]. In the architecture of NDN, we utilize names of applications directly for content delivery or forwarding. As applications and their namespaces have priorities [11], NDN enables vehicles within vicinity of each other to exchange packets as soon as their signals can reach each other. By employing NDN in VANETs, coined as V-NDN [12], the shortness of IP-based scheme can be avoided and efficient content delivery can be achieved.

The cached content is preferred in the content delivery process of V-NDN, so several important security challenges should be addressed during the content delivery [13] even when $\mathcal{SP}$ is offline. The most important one is access control. Attributed to the presence of in-network cache, any request can be fulfilled by the vehicles in the forwarding path whereas $\mathcal{SP}$ cannot control the vehicles' behaviors. Thus, unauthorized vehicles can easily acquire their desired contents from the network without the permission of $\mathcal{SP}$, which damage the interests of $\mathcal{SP}$.

In order to address the access control issue in NDN, many schemes have been proposed, which can be categorized into two kinds: authentication-based [14–16] schemes and encryption-based [13, 17–20] schemes. On the one hand, in authentication-based schemes, an authentication process can be launched by the cache hit nodes to determine whether to return the requested content or not. Obviously, this process requires an access control server or interactions with $\mathcal{SP}$, which may lead to heavy overhead and may degrade the delivery performance [21]. Moreover, the integrity and authenticity of contents cannot be guaranteed. On the other hand, in the encryption-based schemes, only authorized users can decrypt the encrypted contents, and hence the access control can be guaranteed. However, to achieve authorization, $\mathcal{SP}$ usually needs a special architecture or relies on static end-to-end communications, which cannot ensure flexible content sharing[1]. Moreover, the network resources are easily exhausted by flooding requests [16]. These issues make existing solutions ineffective for efficient content delivery in V-NDN with high mobility and high flexibility.

Apart from the above challenges in designing the access control scheme, privacy-protection is another design concern in VANETs, which should be guaranteed during the content delivery. However, only a few works [21] take the privacy protection into account. Moreover, the revocation operation is considered in few of these schemes, which will hinder the large-scale deployment of V-NDN in practice. Furthermore, to

The preliminary result was presented at the IEEE GLOBECOM 2018 [1].

[1]Flexible sharing means that $\mathcal{SP}$ can design access control policies without knowing the subscribed identities of vehicles in advance.

guarantee the utility of NDN in VANETs and provide better content delivery, the incentive scheme is necessary to make sure that vehicles are able to cache contents. Due to the above problems, the security of V-NDN faces more challenges than that of the traditional NDN.

Motivated by the challenges discussed above, we demonstrate an efficient and secure access control (ESAC) scheme for V-NDN. In ESAC, we focus on content delivery and access control without interactions with $\mathcal{SP}$ after subscription. Specifically, to enable flexible access control and revocation operation, we construct a proxy re-encryption to ensure that data is kept confidential and only authenticated vehicles can decrypt and retrieve the content. For privacy protection, we employ the encryption-based name obfuscation to encrypt the interest packets, while adopting the identify-based signature and pseudonyms to achieve anonymous authentication and content integrity during the sharing process. To sum up, the main contributions of our ESAC are listed as follows:

- We present a secure scheme to deal with privacy-preserving access control and the integrity verification for content delivery in V-NDN.
- We utilize the proxy re-encryption method to enable periodic update and revocation operation, which transfers the main computation overhead to vehicles while reducing the inter-vehicle communication cost.
- We design an incentive scheme using hashed certificates to encourage vehicles to cache contents, which boosts the utility of NDN in VANETs.

The rest parts are presented as follows: Section II summarizes the related works. The system model and design goals are illustrated in Section III. Section IV describes our ESAC scheme. Section V and Section VI demonstrate the security analysis and the performance evaluation of ESAC, respectively. Finally, we conclude our paper in Section VII.

II. Related Works

Confidentiality, Integrity and Authenticity: Security design for NDN can be traced back to the emergence of Content-Centric Networking (CCN) [11]. DiBenedetto et al. [22] introduce onion routing in NDN, utilizing several cryptographic operations to offer end-to-end content privacy protection. In another work by Nabeel et al. [23], paillier homomorphic cryptography is used to achieve secure message delivery in publish/subscribe networks. Both schemes [22, 23] attempt to use the end-to-end content encryption to achieve content integrity verification as well as content access control, but unfortunately both will result in significant overheads. Different from [22] and [23], Li et al. [18] consider the content caching and access control of NDN in the network layer, in which a lightweight integrity verification architecture is designed by using the Merkle Hash Tree [24]. To realize physical layer security, Kiskani et al. [25] design a random vector-based coded caching scheme to resist eavesdropping attacks and ensure asymptotic perfect secrecy during content delivery.

Access control: Fotiou et al. [14] demonstrate a delegation-based access control framework for Information-Centric Networking (ICN), a similar concept of NDN, which depends on the synchronization and state maintenance between access control policy servers and content routers. However, scalability will be an issue for content networks in large scale. Hamdane et al. [15] introduce an identity-based cryptographic access control system using hierarchical tree-assisted content naming in NDN. To achieve the access control of a sub-tree's content, the root of the sub-tree is assigned with an encryption/decryption key pair and a symmetric content encryption key. Wood et al. [17] present a secure content distribution framework for CCN according to the proxy re-encryption, providing strong end-to-end content security and at the same time reducing the number of messages needed by key retrieval and user authentication. Ghali et al. [26] propose an Interest-Based Access Control (IBAC) scheme, which only uses the information contained in interest messages to enforce access control. Misra et al. [13] illustrate a new access control architecture, named AccConF, to deliver content to legal users in a secure way in ICNs based on broadcast encryption. Fan et al. [19] introduce a proxy re-encryption based access control scheme in NDN, which is inefficient since each router has to execute re-encryption operations for every forwarding. Tseng et al. [20] develop a fine-grained access control scheme based on DBDH assumption for NDN, which can support mobile receivers, and preserve data confidentiality. To sum up, we list the comparison results of the aforementioned access control schemes in TABLE I.

TABLE I
Comparison of access control schemes.

Scheme	Flexible sharing	Integrity	Privacy	Revocation	Incentive
Fotiou et al.[14]	×	×	✓	×	×
Hamdane et al.[15]	×	✓	×	×	×
Wood et al.[17]	✓	✓	×	×	×
Li et al. [18]	×	✓	×	×	×
Misra et al.[13]	✓	✓	✓	✓	×
Fan et al. [19]	✓	✓	×	×	×
Tseng et al.[20]	✓	✓	×	×	×
ESAC	✓	✓	✓	✓	✓

III. System Model and Design Goals

A. System Model of V-NDN

The system model contains three entities: a service provider ($\mathcal{SP}$), road side units (RSUs), and vehicles with on-board units (OBUs), as demonstrated in Fig. 1.

- $\mathcal{SP}$ is a service provider, which serves as a certificate and registration center for vehicles. $\mathcal{SP}$ connects with RSUs through a secure channel. $\mathcal{SP}$ offers different services, such as multimedia streaming, navigation services, and instant messaging. To improve the quality of service (QoS), $\mathcal{SP}$ encourages vehicles to cache data contents.
- RSUs connect with vehicles by wireless links and use wired links to connect one another. As gateways, RSUs could directly deliver contents to the requesting vehicles. Or RSUs could assist vehicles in content delivery through multi-hop transmissions.
- OBUs in vehicles broadcast the information about routine traffic-related status periodically to enhance the road conditions and traffic safety. Vehicles need infotainment

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3004459, IEEE Transactions on Vehicular Technology

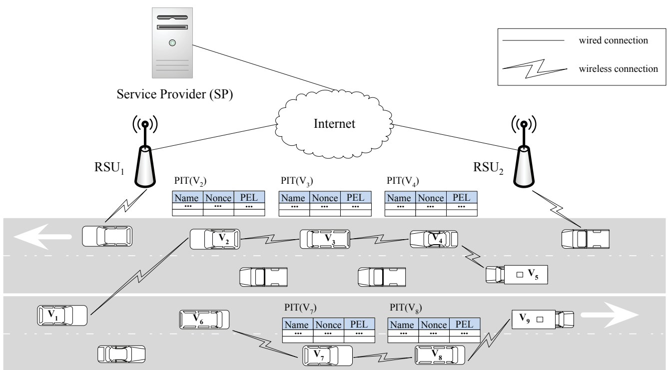


Fig. 1. The system architecture.

data as well for drivers and passengers by NDN. In V-NDN, a vehicle can serve as any one of the four roles: data producer, data consumer, "data mule" (when it carries data over distance but does not connect with anyone else [27]), and forwarder as shown in [7].

Moreover, we assume that secrets are pre-distributed to entities using a secure channel.

B. Design Goals

The design goals are listed as follows [18, 22]:

- **Integrity and authentication**: During the content delivery, $\mathcal{SP}$ should digitally sign each content. As a result, each vehicle could verify the authenticity and integrity of the content. Besides, the interest requester and the corresponding data responder should achieve mutual authentication.
- **Access control**: For the purpose of business, $\mathcal{SP}$ generally has policies on which contents can be accessed by which domain. Hence, V-NDN should allow controllable content access to be implemented even when $\mathcal{SP}$ is offline during the content delivery.
- **Privacy preservation**: During the content delivery, no entity except the vehicle itself could link the real identity with the location for each content.
- **Incentive**: To guarantee the utility of NDN in VANETs, an incentive scheme is desired to encourage vehicles to cache contents. Besides, a payment method should be designed for these vehicles.

C. Attack Model

In our trust model, $\mathcal{SP}$ is considered fully trusted. Vehicles and RSUs are semi-trusted. That is to say, they will follow the predefined protocol, but may be able to infer some sensitive information from content delivery. Attackers can be categorized into *internal attackers* and *external attackers* in VANETs. *Internal attackers* are compromised vehicles and they have access to shared secrets. *External attackers* can eavesdrop on the communication channels between any two entities and launch attacks as follows: (a) inject fake packets, (b) replay or modify packets, (c) pretend to be a legal vehicle, (d) compromise a vehicle. As external attackers are not included in the system, they do not possess secret materials and thus are not able to sign or decrypt messages. Notice that we do not consider the denial-of-service (DoS) attack.

TABLE II
NOTATIONS

Notations	Descriptions
T_i/t_i	The expiration time/the timestamp
$\mathcal{SP}/\mathcal{V}_i$	The service provider/ith mobile vehicles
PID_A	The pseudonym of A
IK_{T_i}	The interest encryption key of T_i
PK_A/SK_A	The public/secret key of A
$e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$	The bilinear map for proxy re-encryption
$\widehat{e} : \widehat{\mathbb{G}}_1 \times \widehat{\mathbb{G}}_1 \rightarrow \widehat{\mathbb{G}}_T$	The bilinear map for signature operation
$H_1(\cdot); H_2(\cdot)$	$H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$; $H_2 : \widehat{\mathbb{G}}_1 \rightarrow \mathbb{Z}_q^*$
$H_3(\cdot); H_4(\cdot)$	$H_3 : \mathbb{G}_T \rightarrow \mathbb{Z}_q^*$; $H_4 : \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$
$E_A(M)$	Encrypt the message M by the key A

IV. ESAC SCHEME

A. Overview

In our ESAC, only $\mathcal{SP}$ works as the content producer and we focus on achieving content delivery by the cache contents among vehicles. Specifically, $\mathcal{SP}$ first generates the encrypted contents with the proxy re-encryption scheme and signs them using identify-based signature [28]. Then, it caches these ciphertexts in vehicles and issues the corresponding pseudonyms and keys to the subscribed vehicles. After that, each subscribed vehicle can generate interest request packets and recover the corresponding content with the authorized keys. Thus, access control can be guaranteed even when SP is offline. Besides, during the content delivery process, each interest request vehicle also pays the cache-hit vehicles incentive certificates to guarantee the utility of NDN in VANETs. Furthermore, ESAC also supports anonymous authentication and revocation operations by using different cryptographic primitives.

B. System Initialization

$\mathcal{SP}$ initializes the system with the following steps:

1) System initialization:

- With the security parameter 1^λ, $\mathcal{SP}$ selects a bilinear map: $\widehat{e} : \widehat{\mathbb{G}}_1 \times \widehat{\mathbb{G}}_1 \rightarrow \widehat{\mathbb{G}}_T$, in which $\widehat{\mathbb{G}}_1$ and $\widehat{\mathbb{G}}_T$ are bilinear groups of a prime order $\widehat{q}$. P is a random generator of $\widehat{\mathbb{G}}_1$. $H_1(\cdot)$ and $H_2(\cdot)$ are the cryptographic hash functions where $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ and $H_2 : \widehat{\mathbb{G}}_1 \rightarrow \mathbb{Z}_q^*$.
- $\mathcal{SP}$ randomly selects $s \in \mathbb{Z}_q^*$ as its secret key $SK_{\mathcal{SP}}$ and calculates $PK_{\mathcal{SP}} = sP$ as the public key;
- $\mathcal{V}_i$ produces the secret key $SK_{\mathcal{V}_i}$ and the public key $PK_{\mathcal{V}_i}$ by RSA.

2) Cache source generation: Prior to using NDN to offer services, $\mathcal{SP}$ needs to cache these contents in vehicles. The original content usually has a very large size. Therefore, $\mathcal{SP}$ divides the original content name (CN) into n content blocks B_i. Each B_i has a fixed size, e.g., 1 Mbytes. For every B_i, a source name as: $/\mathcal{SP}/CN/B_i$ will be given by $\mathcal{SP}$. Besides, to achieve the name obfuscation, $\mathcal{SP}$ generates an interest encryption key IK_{T_1} before the expiration time T_1 such as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3004459, IEEE Transactions on Vehicular Technology

the end of the month. Then, $\mathcal{SP}$ updates the source name as $/\mathcal{SP}/CN/E_{IK_{T_1}}(B_i)$.

We encrypt content blocks in order to guarantee confidentiality and access control. Here, a Proxy Re-Encryption scheme (including six algorithms: PRE.Setup, PRE.KeyGen, PRE.Enc, PRE.ReKey, PRE.ReEnc, PRE.Dec) based on [29] is constructed to encrypt blocks as follows:

- PRE.Setup (1^λ): $\mathcal{SP}$ selects a bilinear map: $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ where $\mathbb{G}_1$ and $\mathbb{G}_T$ are bilinear groups of the prime order q. g is a random generator in $\mathbb{G}_1$, $H_3 : \mathbb{G}_T \to \mathbb{Z}_q^*$ and $H_4 : \mathbb{G}_1 \to \mathbb{Z}_q^*$, which are cryptographically secure hash functions. Then, $\mathcal{SP}$ randomly selects $\epsilon \in \mathbb{Z}_q^*$ as its re-encryption secret key $RSK_{\mathcal{SP}}$ and calculates the re-encryption public key $RPK_{\mathcal{SP}} = g^\epsilon$. The public parameters are: $pm = (q, g, \mathbb{G}_1, \mathbb{G}_T, RPK_{\mathcal{SP}}, H_3, H_4)$;
- PRE.KeyGen $(pm,\ RSK_{\mathcal{SP}})$: For the expiration time T_1, $\mathcal{SP}$ randomly chooses $r_1 \in \mathbb{Z}_q^* \backslash \{\epsilon\}$ as the decryption key SK_{T_1} and generates the corresponding encryption key $PK_{T_1} = g^{r_1}$ to encrypt content blocks;
- PRE.Enc $(PK_{T_1},\ RPK_{\mathcal{SP}},\ B_i)$: $\mathcal{SP}$ computes $t = H_1(T_1||B_i)$ and encrypts each B_i as $EB_i = (c_1, c_2)$, where $c_1 = g^t$ and $c_2 = B_i \oplus H_3(e(PK_{T_1}, RPK_{\mathcal{SP}})^t)$;
- $\mathcal{SP}$ computes $h_{i,T_1} = H_1(H_4(c_1)||c_2||T_1)P$ and $Sig_{i,T_1} = sh_{i,T_1}$;
- Finally, $\mathcal{SP}$ broadcasts $EB_i||T_1||Sig_{i,T_1}$ to vehicles.

When these contents are received by vehicles, they will be cached randomly. Before that, vehicles can verify the signatures by checking if $\widehat{e}(Sig_{i,T_1}, P) \stackrel{?}{=} \widehat{e}(h_{i,T_1}, PK_{\mathcal{SP}})$. Note that the validity of the cached contents should be guaranteed by each vehicle before returning data packets.

C. Vehicles Subscription

When a vehicle $\mathcal{V}_i$ subscribes the service with $\mathcal{SP}$ before the expiration time T_1, it completes the registration process as below:

- $\mathcal{SP}$ issues a set of pseudonyms for $\mathcal{V}_i$ as $PID_{\mathcal{V}_i,j}$;
- For $PID_{\mathcal{V}_i,j}$, $\mathcal{SP}$ calculates $\tau_{\mathcal{V}_i,j} = H_1(PID_{\mathcal{V}_i,j})$ and $SK_{\mathcal{V}_i,j} = \frac{1}{s+\tau_{\mathcal{V}_i,j}}P$ as the corresponding private key;
- $\mathcal{SP}$ issues the certificate of $PID_{\mathcal{V}_i,j}$ by computing $h_{\mathcal{V}_i,j} = H_1(PID_{\mathcal{V}_i,j}||T_1)P$ and $\sigma_{\mathcal{V}_i,j} = sh_{\mathcal{V}_i,j}$;
- Finally, $\mathcal{SP}$ sends the registration information $Cert_{\mathcal{V}_i,j} = (PID_{\mathcal{V}_i,j}, T_1, \sigma_{\mathcal{V}_i,j})$, $SK_{\mathcal{V}_i,j}$, the interest encryption key IK_{T_1}, and the secret key $SK_{T_1} = r_1$ via a secure channel.

On receiving $Cert_{\mathcal{V}_i,j}$, $\mathcal{V}_i$ first verifies the expiration time T_1 and subsequently confirm the validity of $Cert_{\mathcal{V}_i,j}$ through checking $\widehat{e}(\sigma_{\mathcal{V}_i,j}, P) \stackrel{?}{=} \widehat{e}(h_{\mathcal{V}_i,j}, PK_{\mathcal{SP}})$. Notice that for vehicles which only join the system for caching data and getting rewards, they can only get $Cert_{\mathcal{V}_j,j}$.

D. Content Delivery

The content delivery process could be divided into three phases as follows:

Algorithm 1 Interest Request Verification

Input: $\mathsf{EIn}_i, t_1, Sig_1, Cert_{\mathcal{V}_i,j}$

```
1: V_j first checks the validity of t_1;
2: if t_2 ≤ ΔT + t_1 then
3:    V_j calculates h'_{V_i,j} = H_1(PID_{V_i,j}||T_1)P according to Cert_{V_i,j};
4:    if ê(σ_{V_i,j}, P) = ê(h'_{V_i,j}, PK_SP) then
5:       V_j verifies the validity of signature Sig_1 by calculating τ'_{V_i,j} = H_1(PID_{V_i,j}), θ'_1 = PK_SP + τ'_{V_i,j}P, and h'_1 = H_1(PID_{V_i,j}||EIn_i||t_1||H_2(X_1));
6:       if ê(Y_1, θ'_1) = ê(X_1, P) ê(P, P)^{h'_1} then
7:          The message (1) is a valid interest request.
8:       end if
9:    end if
10: end if
```

1) Interest request: During the content delivery process, the encryption-based name obfuscation [26] is used to avoid creation of interests by unauthorized users and hide the target of interests from eavesdroppers. Moreover, to reduce the privacy threat during the forwarding process, the name components (e.g., suffix of the name) are encrypted partially. When $\mathcal{V}_i$ requests the interest of name In_i (e.g., /youtube.com/movie/Godfather), it broadcasts an interest packet as:

$$\mathcal{V}_i \to * : \mathsf{EIn}_i, t_1, Sig_1, Cert_{\mathcal{V}_i,j}, \tag{1}$$

where EIn is the encrypted suffix of the interest name (e.g., /youtube.com/movie/$E_{IK_{T_1}}(Godfather)$). t_1 is the current timestamp and Sig_1 is the signature which is computed as follows:

- $\mathcal{V}_i$ randomly picks $\delta_1 \in \mathbb{Z}_q^*$ and computes $X_1 = \delta_1 P$;
- $\mathcal{V}_i$ computes $h_1 = H_1(PID_{\mathcal{V}_i,j}||\mathsf{EIn}_i||t_1||H_2(X_1))$ and $Y_1 = (\delta_1 + h_1)SK_{\mathcal{V}_i,j}$;
- $\mathcal{V}_i$ then returns a signature $Sig_1 = (X_1, Y_1)$.

Once receiving the interest request message at t_2, each vehicle will forward this message based on the proposed geolocation-based data forwarding schemes [6, 8]. If $\mathcal{V}_j$ has the data satisfying the interest, it will perform the interest request verification as **Algorithm 1**.

Here, ΔT is the valid time interval. The verification of the signature is described in formula (2).

$$\begin{aligned}
&\widehat{e}(Y_1, \theta_1) = \widehat{e}((\delta_1 + h_1)SK_{\mathcal{V}_i,j}, PK_{\mathcal{SP}} + \tau_{\mathcal{V}_i,j}P)\\
=&\widehat{e}((\delta_1 + h_1)\frac{1}{s + H_1(PID_{\mathcal{V}_i,j})}P, sP + H_1(PID_{\mathcal{V}_i,j})P)\\
=&\widehat{e}((\delta_1 + h_1)\frac{1}{s + H_1(PID_{\mathcal{V}_i,j})}P, (s + H_1(PID_{\mathcal{V}_i,j}))P)\\
=&\widehat{e}(P, P)^{\delta_1}\widehat{e}(P, P)^{h_1}\\
=&\widehat{e}(X_1, P)\widehat{e}(P, P)^{h_1}.
\end{aligned} \tag{2}$$

2) Data response: After confirming the validity of the interest packet, $\mathcal{V}_j$ will return the corresponding data packet as:

$$\mathcal{V}_j \to \mathcal{V}_i : t_3, EB_i, Sig_{i,T_1}, \beta_i, Sig_2, Cert_{\mathcal{V}_j,i}, \tag{3}$$

where t_3 is the current timestamp, EB_i is the ciphertext of B_i and Sig_{i,T_1} is the signature of B_i at T_1 signed by $\mathcal{SP}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3004459, IEEE Transactions on Vehicular Technology

Algorithm 2 Data Response Verification

Input: $t_3, EB_i, Sig_{i,T_1}, \beta_i, Sig_2, Cert_{\mathcal{V}_j,i}$
Output: B_i
$\mathcal{V}_i$ first checks the validity of t_3;
if $t_4 \leq \Delta T + t_3$ **then**
  $\mathcal{V}_i$ checking the validity of $Cert_{\mathcal{V}_j,i}$ by calculating $h'_{\mathcal{V}_j,i} = H_1(PID_{\mathcal{V}_i,j}||T_1)P$;
  if $\widehat{e}(\sigma_{\mathcal{V}_j,i}, P) = \widehat{e}(h'_{\mathcal{V}_j,i}, PK_{\mathcal{SP}})$ **then**
    $\mathcal{V}_i$ calculates $\tau'_{\mathcal{V}_j,i} = H_1(PID_{\mathcal{V}_j,i})$, $\theta'_2 = PK_{\mathcal{SP}} + \tau'_{\mathcal{V}_j,i}P$, and $h'_2 = H_1(PID_{\mathcal{V}_j,i}||t_3||H_4(c_1)||c_2||H_2(Sig_{i,T_1})||\beta_i||H_2(X_2))$;
    if $\widehat{e}(Y_2, \theta'_2) = \widehat{e}(X_2, P)\widehat{e}(P, P)^{h'_2}$ **then**
      $\mathcal{V}_i$ calculates $B_i = H_3(e(RPK_{\mathcal{SP}}^{SK_{T_1}}, c_1)) \oplus c_2$.
    end if
  end if
end if

$\beta_i = (\mathsf{EIn}_i \oplus H_1(\gamma_i))$ where $\gamma_i \in \mathbb{Z}_q^*$ and $Cert_{\mathcal{V}_j,i}$ is the ith certificate of $\mathcal{V}_j$. Sig_2 is computed as follows:

- $\mathcal{V}_j$ chooses a random number $\delta_2 \in \mathbb{Z}_q^*$ and calculates $X_2 = \delta_2 P$;
- $\mathcal{V}_j$ computes $h_2 = H_1(PID_{\mathcal{V}_j,i}||t_3||H_4(c_1)||c_2||H_2(Sig_{i,T_1})||\beta_i||H_2(X_2))$ and $Y_2 = (\delta_2 + h_2)SK_{\mathcal{V}_j,i}$;
- $\mathcal{V}_j$ then returns a signature $Sig_2 = (X_2, Y_2)$.

When $\mathcal{V}_i$ receives the data packet as response at t_4, it will execute the process as **Algorithm 2**.

There could be a number of blocks satisfying In_i, so $\mathcal{V}_i$ may receive a few response data. For different signatures $Sig_i = (X_i, Y_i)$, $\mathcal{V}_i$ could perform batch verification as follows:

- $\mathcal{V}_i$ calculates $\tau_{\mathcal{V}_j,i} = H_1(PID_{\mathcal{V}_j,i})$;
- $\mathcal{V}_i$ computes $\theta_i = PK_{\mathcal{SP}} + \tau_{\mathcal{V}_j,i}P$ and h_i;
- $\mathcal{V}_i$ verifies $\widehat{e}(\sum Y_i, \sum \theta_i) \stackrel{?}{=} \widehat{e}(\sum X_i, P) \cdot \widehat{e}(P, P)^{\sum h_i}$.

The correctness of B_i is described as follows:

$$\begin{aligned} &H_3(e(RPK_{\mathcal{SP}}^{SK_{T_1}}, c_1)) \oplus c_2 \\ =&H_3(e(g^{\epsilon r_1}, g^t)) \oplus B_i \oplus H_3(e(PK_{T_1}, RPK_{\mathcal{SP}})^t) \\ =&H_3(e(g^{\epsilon r_1}, g^t)) \oplus B_i \oplus H_3(e(g^{r_1}, g^{\epsilon})^t) \\ =&B_i. \end{aligned} \tag{4}$$

Notice that if $\mathcal{V}_j$ returns a wrong data packet[2] which is not generated by $\mathcal{SP}$ at T_1 (as described in the cache source generation phase, $\mathcal{V}_j$ should ensure the correctness of the cached data sources before content delivery), $\mathcal{SP}$ will revoke $\mathcal{V}_j$ by broadcasting its certificates to other vehicles. Thus, when vehicles receive packets, they should first check the certificate revocation list (CRL) to ensure the validity of users. When T_1 expires, $\mathcal{SP}$ and legal vehicles will execute the update process to exclude the revoked vehicles. Thus, legal vehicles will discard the CRL for T_1 and generate a new CRL at T_2.

3) Incentive certificate: After getting B_i, $\mathcal{V}_i$ should return an incentive certificate for $\mathcal{V}_j$'s contribution as follows:

- $\mathcal{V}_i$ computes $H_1(\gamma_i) = \beta_i \oplus \mathsf{EIn}_i$;
- $\mathcal{V}_i$ randomly chooses $\delta_3 \in \mathbb{Z}_q^*$ and computes $X_3 = \delta_3 P$;
- $\mathcal{V}_i$ calculates $h_3 = H_1(PID_{\mathcal{V}_i,j}||\mathsf{EIn}_i||t_5||H_1(\gamma_i)||H_2(X_3))$ and $Y_3 = (\delta_3 + h_3)SK_{\mathcal{V}_i,j}$;
- $\mathcal{V}_i$ sets $Cer_{\mathcal{V}_i,\mathsf{EIn}_i} = (X_3, Y_3)$ and returns $Cer_{\mathcal{V}_i,\mathsf{EIn}_i}$ to $\mathcal{V}_j$.

[2]This implies that $\mathcal{V}_j$ has passed the identity verification (by $Cert_{\mathcal{V}_j,i}$) and message integrity verification (by Sig_2).

TABLE III
THE INCENTIVE TABLE.

Request	Interest	Time	Value	Hash value	Certificate
$PID_{\mathcal{V}_1}$	EIn_1	t_1	γ_1	$H_1(\gamma_1)$	$Cer_{\mathcal{V}_1,\mathsf{EIn}_1}$
$PID_{\mathcal{V}_2}$	EIn_2	t_2	γ_2	$H_1(\gamma_2)$	$Cer_{\mathcal{V}_2,\mathsf{EIn}_2}$
$PID_{\mathcal{V}_3}$	EIn_3	t_3	γ_3	$H_1(\gamma_3)$	$Cer_{\mathcal{V}_3,\mathsf{EIn}_3}$
$PID_{\mathcal{V}_4}$	EIn_4	t_4	γ_4	$H_1(\gamma_4)$	$Cer_{\mathcal{V}_4,\mathsf{EIn}_4}$
...	...	...	...	...	...

When $\mathcal{V}_j$ receives the certificate, it verifies $Cer_{\mathcal{V}_i,\mathsf{EIn}_i}$. If the verification is passed, it will store $\{PID_{\mathcal{V}_i,j}, \mathsf{EIn}_i, t_5, \gamma_i, H_1(\gamma_i), Cer_{\mathcal{V}_i,\mathsf{EIn}_i}\}$, as shown in Table III. When $\mathcal{V}_j$ intends to get reward for its contribution, it only needs to send $\{PID_{\mathcal{V}_i,j}, \mathsf{EIn}_i, t_5, \gamma_i, H_1(\gamma_i), Cer_{\mathcal{V}_i,\mathsf{EIn}_i}\}$ to $\mathcal{SP}$. When $\mathcal{SP}$ receives this rewarding request, the following operations will be processed:

- $\mathcal{SP}$ first checks the validity of $PID_{\mathcal{V}_i,j}$;
- If $PID_{\mathcal{V}_i,j}$ is valid and does not belong to $\mathcal{V}_j$, then $\mathcal{SP}$ checks the validity of γ_i. $\mathcal{SP}$ calculates $t' = H_1(\gamma_i)$;
- If $t'_i = H_1(\gamma_i)$, then$\mathcal{SP}$ checks the validity of $Cer_{\mathcal{V}_i,\mathsf{EIn}_i}$ by calculating $\tau'_{\mathcal{V}_i,j} = H_1(PID_{\mathcal{V}_i,j})$, $\theta'_3 = PK_{\mathcal{SP}} + \tau'_{\mathcal{V}_i,j}P$, $h_3 = H_1(PID_{\mathcal{V}_i,j}||\mathsf{EIn}_i||t_5||H_1(\gamma_i)||H_2(X_3))$, and verifies $\widehat{e}(Y_3, \theta'_3) \stackrel{?}{=} \widehat{e}(X_3, P)\widehat{e}(P, P)^{h'_3}$. If the verification is passed, $\mathcal{V}_j$ is the real contributor and can get corresponding rewards. Notice that $\mathcal{SP}$ can also perform batch verification.

E. Periodic Update

Upon expiration of T_1, $\mathcal{SP}$ needs to update the encrypted content EB_i, the corresponding decryption key SK_{T_1}, and interest encryption key IK_{T_i} to achieve access control. The processes are illustrated as follows:

1) $\mathcal{SP}$ update: According to the expiration time T_1, $\mathcal{SP}$ should generate new interest encryption key, a new encrypted content, and the corresponding decryption key for the next expiration time T_2. The details are described as follows:

- $\mathsf{PRE.ReKey}$ $(SK_{T_1}, SK_{T_2}, RPK_{\mathcal{SP}})$: For the expiration time T_2, $\mathcal{SP}$ computes the new decryption key $SK_{T_2} = r_2$ by $\mathsf{PRE.KeyGen}$ $(pm, RSK_{\mathcal{SP}})$ and computes the re-encryption key $rk_{T_1 \to T_2} = RPK_{\mathcal{SP}}^{SK_{T_1}} \cdot g^{H_1(SK_{T_2}||T_2)}$;
- $\mathsf{PRE.ReEnc}$ $(EB_i, rk_{T_1 \to T_2})$: $\mathcal{SP}$ updates the content blocks by calculating $c'_1 = e(g, c_1)$ and $c'_3 = e(c_1, rk_{T_1 \to T_2})$. Then, $\mathcal{SP}$ outputs the re-encryption ciphertext $EB'_i = (c'_1, c_2, c'_3)$;
- Finally, $\mathcal{SP}$ calculates $h_{i,T_2} = H_1(H_4(c'_1)||c_2||H_3(c'_3)||T_2)P$ and $Sig_{i,T_2} = sh_{i,T_2}$ as the corresponding certificates.

2) Vehicles update: Vehicle $\mathcal{V}_i$ which subscribes the service of $\mathcal{SP}$, sends an interest request packet to $\mathcal{SP}$ upon expiration of T_1:

$$\mathcal{V}_i \to \mathcal{SP} : /\mathcal{SP}/Key/T_2, t_7, Sig_3, Cert_{\mathcal{V}_i,\zeta}, \quad (5)$$

where t_7 is the current timestamp and Sig_3 is the signature which is computed as Sig_1. $PID_{\mathcal{V}_i,\zeta}$ is a special pseudonym of $\mathcal{V}_i$ for implementing the update process.

Upon receiving the interest packet, $\mathcal{SP}$ needs to perform the verification process before returning the data packet as shown in **Algorithm 1**. Only when all the verifications are passed, $\mathcal{SP}$ will return the data packet as:

$$SP \to \mathcal{V}_i : t_8, E_{PK_{\mathcal{V}_i}}(rk_{T_1\to T_2}, SK_{T_2}, IK_{T_2}), Sig_4, \quad (6)$$

where t_8 is the current timestamp, $E_{PK_{\mathcal{V}_i}}(rk_{T_1\to T_2}, SK_{T_2}, IK_{T_2})$ means encrypting $rk_{T_1\to T_2}$, SK_{T_2}, and IK_{T_2} by the public key $PK_{\mathcal{V}_i}$ of $\mathcal{V}_i$. Sig_4 is the signature calculated similarly as Sig_1.

Once receiving the updated data packet, $\mathcal{V}_i$ should first verify Sig_4 to guarantee its validity. After that, $\mathcal{V}_i$ extracts the new data decryption key SK_{T_2} and updates the encrypted content blocks according to PRE.ReEnc $(EB_i, rk_{T_1\to T_2})$ to get $EB_i' = (c_1', c_2, c_3')$. Notice that when $\mathcal{V}_j$ gets EB_i', it computes $H_3(\frac{c_3'}{c_1'^{H_1(sk_{T_2}||T_2)}}) \oplus c_2$ according to equation (7) to get B_i. Then it verifies $e(g,g)^{H_1(T_1||B_i)} = c_1'$. If the equation holds, it accepts B_i.

$$\begin{aligned}
&H_3(\frac{c_3'}{c_1'^{H_1(SK_{T_2}||T_2)}}) \oplus c_2 \\
=&H_3(\frac{e(g^{H_1(T_1||B_i)}, g^{\epsilon r_1}) \cdot e(g,g)^{H_1(T_1||B_i)\cdot H_1(SK_{T_2}||T_2)}}{e(g,g)^{H_1(T_1||B_i)\cdot H_1(SK_{T_2}||T_2)}}) \quad (7)\\
&\oplus B_i \oplus H_3(e(g^{r_1}, RPK_{\mathcal{SP}})^t) \\
=&B_i.
\end{aligned}$$

Notice that for vehicles which only join the system for caching data to get rewards, they can only get $rk_{T_1\to T_2}$ to update EB_i.

3) Certificate request: After re-encrypting the content blocks EB_i', $\mathcal{V}_i$ needs to get the corresponding certificates from $\mathcal{SP}$. To be illustrative, $\mathcal{SP}$ calculates $\{H_1(H_4(c_1')||c_2||H_3(c_3')), Sig_{i,T_2}\}$ and sends it to vehicles through RSUs according to the resource allocation schemes in [30] or with the assistance of Unmanned Aerial Vehicles (UAVs) [31]. Each vehicle can get the certificate Sig_{i,T_2} according to $H_1(H_4(c_1')||c_2||H_3(c_3'))$ and verify their validity before accepting it. After finishing the update process, vehicles can continue with content delivery.

V. Security Analysis

We first give the security proof about our constructed proxy re-encryption scheme. Then, we present the detailed security analysis according to different security requirements.

Theorem V.1. *Our proposed re-encryption scheme is CCA secure in the random oracle model under the DBDH assumption.*

Proof: Assuming that there exists an adversary $\mathcal{A}$ breaking the CCA security of our scheme, we build an algorithm $\mathcal{B}$ solving the DBDH problem $S = e(g,g)^{abc}$. $\mathcal{B}$ maintains three tables T_{pk}, T_{sk}, and T_{rk}, which are initially empty. Besides, we assume that the signatures of the issued re-encryption/decryption queries are valid [32]. Finally, $\mathcal{B}$ sets $RPK_{\mathcal{SP}} = g^b$ and runs the follows steps.

Phase 1: $\mathcal{O}_{pk}$: Upon input of an index i, $\mathcal{B}$ first chooses x_i and sets a random $\alpha_i \in \{0,1\}$ so that $Pr[\alpha_i = 1] = \delta$. If α_i=1, $\mathcal{B}$ calculates $pk_i = g^{x_i}$. Otherwise, $\mathcal{B}$ calculates $pk_i = (g^a)^{x_i}$. Finally, $\mathcal{B}$ records the tuple (pk_i, x_i, α_i) in table T_{pk} and returns pk_i to $\mathcal{A}$.

We assume that $\mathcal{A}$ has made the appropriate $\mathcal{O}_{pk}$ queries before executing one of the following queries:

- $\mathcal{O}_{sk}$: Upon input pk_i, $\mathcal{B}$ queries T_{pk} by pk_i, and obtains the corresponding value:
  - If $\alpha_i = 1$, $\mathcal{B}$ returns x_i to $\mathcal{A}$ and records pk_i in T_{sk}.
  - If $\alpha_i = 0$, $\mathcal{B}$ returns $failure$ and aborts the simulation.
- $\mathcal{O}_{rk}$: On input (pk_i, pk_j), $\mathcal{B}$ first searches table T_{rk} to check if there is a record of (pk_i, pk_j, $rk_{i\to j}$). If it exists, $\mathcal{B}$ returns " having accessed the re-encryption key before". Otherwise, $\mathcal{B}$ obtains (pk_i, x_i, α_i) and (pk_j, x_j, α_j) according to T_{pk}:
  - If $\alpha_i = \alpha_j = 1$, $\mathcal{B}$ queries $\mathcal{O}_{sk}$ with pk_i and pk_j, and then uses the obtained private keys to compute the corresponding re-encryption key $rk_{i\to j}$ with PRE.ReKey.
  - Otherwise, $\mathcal{B}$ chooses a random number $R_{i,j}$ from $\mathbb{G}_1$ as the re-encryption key $rk_{i\to j}$.
  - Finally, $\mathcal{B}$ records (pk_i, pk_j, $rk_{i\to j}$) in table T_{rk}.
- $\mathcal{O}_{re}$: Upon input (pk_i, pk_j, C_i), $\mathcal{B}$ first verifies the validity of the signature of the issued query. If it does not hold, output $\perp$ and abort; otherwise, it queries $\mathcal{O}_{pk}$ with pk_i, pk_j to obtain (pk_i, x_i, α_i) and (pk_j, x_j, α_j), respectively.
  - If $\alpha_i = 0$ and $\alpha_j = 1$, $\mathcal{B}$ chooses a random number r from $\mathbb{Z}_q^*$, and calculates $c_1' = e(g, c_1)$ and $c_3' = e(g, g^b)^{ax_i} \cdot c_1'^{H_1(x_j||r)}$.
  - Otherwise, $\mathcal{B}$ queries $\mathcal{O}_{rk}$ with (pk_i, pk_j) to obtain $rk_{i\to j}$. At last, $\mathcal{B}$ returns PRE.ReEnc $(C_i, rk_{i\to j})$ to $\mathcal{A}$.
- $\mathcal{O}_{dec}$: On input (pk_i, C_i), where C_i is an initial ciphertext or a re-encrypted ciphertext according to different forms as (c_1, c_2) or (c_1, c_2, c_3), $\mathcal{B}$ verifies the validity of the signature of the issued query. If the signature is invalid, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ obtains pk_i and performs the following operations:
  - For the initial ciphertext (c_1, c_2), $\mathcal{B}$ continues the following operation:
    1) If $\alpha_i = 0$, $\mathcal{B}$ returns $failure$ and aborts the simulation.
    2) Otherwise, if $\alpha_i = 1$, $\mathcal{B}$ calculates $B_i = H_3(e(g^{bx_i}, c_1)) \oplus c_2$.
    3) Finally, $\mathcal{B}$ returns B_i to $\mathcal{A}$.
  - For the re-encrypted ciphertext (c_1, c_2, c_3):
    1) If $\alpha_i = 0$, $\mathcal{B}$ returns $failure$ and aborts the simulation.
    2) Otherwise, $\mathcal{B}$ calculates $B_i = H_3(\frac{c_3'}{c_1'^{H_1(x_i||i)}}) \oplus c_2$ and checks if $e(g,g)^{H_1(i||B_i)} = c_1'$. If not, $\mathcal{B}$

returns $\perp$.

3) Otherwise, $\mathcal{B}$ returns B_i to $\mathcal{A}$.

Challenge: On input pk^*, m_0, and m_1, where m_0 and m_1 are of the same length.

- If α^*=1, $\mathcal{B}$ outputs *failure* and aborts the simulation;
- Otherwise, for the initial ciphertext, $\mathcal{B}$ chooses a random bit **b**, and calculates $c_1^* = g^c$, $c_2^* = m_{\mathbf{b}} \oplus H_3(S^{x^*})$, x^* is the corresponding value in tuple (pk^*, x^*, α^*).
- Then, for the re-encryption ciphertext, from pk_i to pk^* (α_i=0), $\mathcal{B}$ chooses a random bit **b**, and computes ${c_1}^{*\prime} = e(g, g^c)$, $c_2^* = m_{\mathbf{b}} \oplus H_3(S^{x^*})$, ${c_3}^{*\prime} = e(g^c, R^*)$, where R^* is a random element from $\mathbb{G}_1$.
- $\mathcal{B}$ returns (c_1^*, c_2^*) and $({c_1}^{*\prime}, c_2^*, {c_3}^{*\prime})$ to $\mathcal{A}$ as the challenge ciphertext.

Phase 2: Almost the same as that in **Phase 1**, except with the following restrictions:

- $\mathcal{A}$ is not permitted to launch an $\mathcal{O}_{sk}$ query on pk^*.
- $\mathcal{A}$ is not permitted to launch any query of the form $\mathcal{O}_{dec}(pk_i, C_i)$ where $(pk_i, C_i) = (pk^*, C^*)$ or $(pk_i, C_i) = (pk^*, C^{*\prime})$
- If $\mathcal{A}$ issues an $\mathcal{O}_{sk}$ query on pk_i, $\mathcal{A}$ is not permitted to make an $\mathcal{O}_{rk}(pk^*, pk_i)$
- If $\mathcal{A}$ issues an $\mathcal{O}_{rk}(pk^*, pk_i)$, $\mathcal{B}$ checks whether $C_i = \mathsf{PRE.ReEnc}(C_i, rk_{*\to i}, C^*)$. If it holds, $\mathcal{A}$ is not permitted to launch an $\mathcal{O}_{dec}(pk_i, C_i)$ query.

Guess: $\mathcal{A}$ outputs $\mathbf{b}'$. If $\mathbf{b}'$=**b**, $S = e(g,g)^{abc}$; otherwise, $S \neq e(g,g)^{abc}$.

With the similar methods used in [29], we have that the above simulator succeeds with a non-negligible probability. ■

A. Integrity and Authentication

During the content delivery, we use the signature and certificate to achieve integrity and mutual authentication. The interest packet broadcasted by $\mathcal{V}_i$ contains Sig_1 and $Cert_{\mathcal{V}_i,j}$. Both of them contain the secret key $SK_{\mathcal{SP}} = s$. This means that the probability to compromise s from $PK_{\mathcal{SP}} = sP$ is approximately $2^{b/2}$, where the number of elements in group $\widehat{\mathbb{G}}_1$ is $|\widehat{\mathbb{G}}_1| = 2^b$. The identity authentication of $\mathcal{V}_i$ based on $Cert_{\mathcal{V}_i,j}$ is as follows:

$$\widehat{e}(\sigma_{\mathcal{V}_i,j}, P) = \widehat{e}(s \cdot h_{\mathcal{V}_i,j}, P) = \widehat{e}(h_{\mathcal{V}_i,j}, PK_{\mathcal{SP}}).$$

The probability of $\widehat{e}(s' h_{\mathcal{V}_i,j}, P) = \widehat{e}(\sigma'_{\mathcal{V}_i,j}, P)$, which results in $\widehat{e}(s' h_{\mathcal{V}_i,j}, P) = \widehat{e}(h_{\mathcal{V}_i,j}, PK_{\mathcal{SP}})$, is approximately $2/(2^b - 1)$.

To verify the signature Sig_1, we should calculate formula (2). To construct the correct signature, the attacker should solve the k-CAA (the collusion attack algorithm with k traitors) problem [33] which is defined as follows:

Definition V.1 (k-CAA problem)**.** *For an integer k, and $x \in \mathbb{Z}_q$, $P \in \widehat{\mathbb{G}}_1$, given*

$$\{P, Q = xP, h_1, \cdots, h_k \in \mathbb{Z}_q, \frac{1}{h_1 + x}P, \cdots, \frac{1}{h_k + x}P\}$$

to compute $\frac{1}{h+x}P$ for some $h \notin \{h_1, \cdots, h_k\}$.

Thus, for all $t-$time adversaries $\mathcal{A}$, we have $Adv_{\mathcal{A}}^{k-\mathrm{CCA}}$ as a negligible function as in formula (8).

To reply with the data packet according to the interest packet, user should return the message (3). It also contains the signature Sig_2 and certificate $Cert_{\mathcal{V}_j,i}$ that has the same function as the interest packet sent by $\mathcal{V}_i$. Only valid users can generate these contents. By using these contents, we can ensure that the content delivery process can achieve integrity and mutual authentication.

B. Access Control

In a distributed situation without central control, content is encrypted as EB_i =(c_1, c_2), where $t = H_1(T_1 || B_i)$, $c_1 = g^t$, $c_2 = B_i \oplus H_3(e(PK_{T_1}, RPK_{\mathcal{SP}})^t)$. To decrypt B_i, user should use $SK_{T_1} = r_1$ to get B_i as in formula (4). When the time expires, EB_i is updated to $EB'_i = (c'_1, c_2, c'_3)$, where $c'_1 = e(g, c_1)$ and $c'_3 = e(c_1, rk_{T_1 \to T_2})$. To get B_i from EB'_i, user should compute B_i as in formula (7). Without decryption keys, the attacker is required to solve DBDH problem as proved in $\mathbf{Theorem V.1}$. to get B_i. Thus, for all $t-$time adversaries $\mathcal{A}$, we have $Adv_{\mathcal{A}}^{\mathrm{DBDH}}$ as a negligible function as in formula (9) to get the plaintext. By using these encrypted contents, we can ensure content confidentiality.

Apart from that, only the legal users have the secret key SK_{T_i} according to the lifecycle. When the time expires, users' SK_{T_i} should be updated. Moreover, for inside attackers, we adopt the CRL to record them between the expiration time T_{i-1} and T_i and use the updated operation to revoke them at the beginning of the next expiration time T_{i+1}. Thus, by these manners, we can guarantee access control.

C. Anonymous

The proposed scheme adopts the pseudonym technique to avoid the leakage of the real identities of users. In terms of location privacy, attackers cannot reveal the location of a specific user because: 1) the pseudonym $PID_{\mathcal{V}_i,j}$ of each user is periodically changed and 2) we use the geolocation-based forwarding to deliver data instead of using IP address. As a result, attackers cannot figure out the real identity of a user or track trajectories by the pseudonym.

D. Incentive

To achieve the incentive purpose, each data responder $\mathcal{V}_j$ sends $\beta_i = (\mathsf{EIn}_i \oplus H_1(\gamma_i))$ to the interest requester $\mathcal{V}_i$ and gets the corresponding incentive certificate $Cer_{\mathcal{V}_i, \mathsf{EIn}_i}$. With these stored incentive certificates in Table III, $\mathcal{SP}$ can verify the validity of these certificates during the rewarding phase. For attackers without knowing the secret key $SK_{\mathcal{V}_j,j}$, the probability of correctly guessing $Cer_{\mathcal{V}_i,\mathsf{EIn}}$ is negligible. Thus, we can guarantee the non-repudiation about incentive.

To resist the imitating attack about incentive, $\mathcal{V}_j$ lets $\mathcal{V}_i$ sign on $H_1(\gamma_i)$ which is the hash value of γ_i. During the rewarding phase, $\mathcal{V}_j$ will send γ_i and $H_1(\gamma_i)$ to $\mathcal{SP}$. Thus, $\mathcal{SP}$ can ensure that this incentive belongs to $\mathcal{V}_j$. For attackers not knowing γ_i, the probability of correctly guessing γ_i from $H_1(\gamma_i)$ is about $1/2^{l-1}$, where l is the output length of $H_1(\cdot)$. Therefore, we can prevent the incentive imitating attack.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3004459, IEEE Transactions on Vehicular Technology

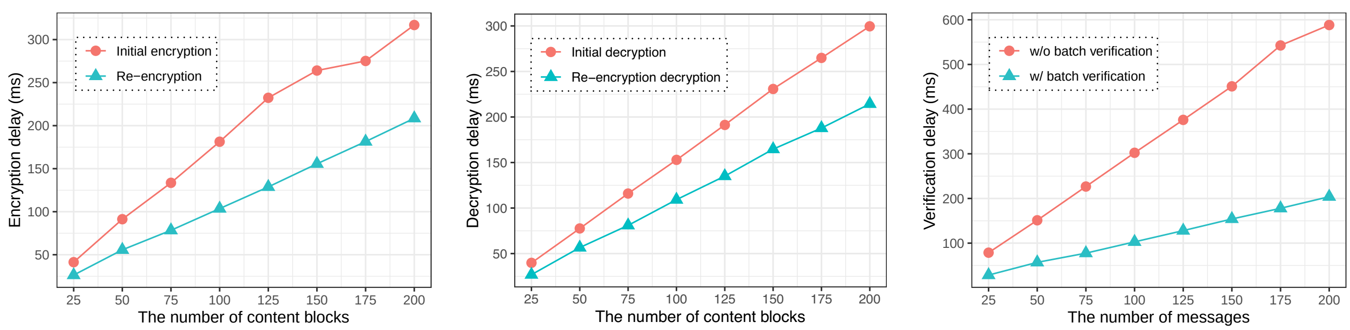
$$Adv_{\mathcal{A}}^{\text{k-CCA}} = Pr\left[\begin{array}{c} \mathcal{A}(P, sP, \frac{1}{H_1(PID_1)+s}P, \cdots, \frac{1}{H_1(PID_k)+s}P) = \frac{1}{H_1(PID)+s}P \\ s \in \mathbb{Z}_q,\ P \in \widehat{\mathbb{G}}_1, H_1(PID_1), \cdots, H_1(PID_k) \in \mathbb{Z}_q, PID \notin \{PID_1, \cdots, PID_k\} \end{array}\right] < \varepsilon. \quad (8)$$

$$Adv_{\mathcal{A}}^{\text{DBDH}} = \left|\begin{array}{c} Pr[a,b,c \leftarrow \mathbb{Z}_q^*; 1 \leftarrow \mathcal{A}(g, g^a, g^b, g^c, e(g,g)^{abc})] - \\ Pr[a,b,c \leftarrow \mathbb{Z}_q^*; z \in \mathbb{G}_T; 1 \leftarrow \mathcal{A}(g, g^a, g^b, g^c, e(g,g)^z)] \end{array}\right| < \varepsilon. \quad (9)$$

TABLE IV
THE COMPUTATION OVERHEAD.

Phase	Entity	Computation Overhead[1]	Communication Overhead (bits)[2]
Cache source generation	$\mathcal{SP}$	$(T_{exp} + T_{par} + T'_{exp} + 2\widehat{T}_{mul})n_B$	N/A
Vehicles Subscription	$\mathcal{SP}$	$3\widehat{T}_{mul} n_{PID_{\mathcal{V}_i}}$	$(29 * 8 + 2\|\widehat{\mathbb{G}}\|)n_{PID_{\mathcal{V}_i}}$
Content Delivery	$\mathcal{V}_i(\mathcal{V}_i \to *)$	$2\widehat{T}_{mul}$	$(37 * 8 + 3\|\widehat{\mathbb{G}}\|)$
	$\mathcal{V}_j(\mathcal{V}_j \to \mathcal{V}_i)$	$3\widehat{T}_{mul} + 5\widehat{T}_{par} + \widehat{T}'_{exp}$	$(37 * 8 + 4\|\widehat{\mathbb{G}}\| + S_{H_1} + S_B)$
	$\mathcal{V}_i$	$n_D\widehat{T}_{mul} + (2n_D + 3)\widehat{T}_{par} + \widehat{T}'_{exp} + n_D T_{par}$	N/A
Periodic Update	$\mathcal{V}_i$	$2n_{\mathcal{V}_i,B} T_{par}$	$(58 * 8 + 3\|\widehat{\mathbb{G}}\|)$
	$\mathcal{SP}$	$T_{exp} + T_{mul} + 2n_B T_{par} + 2n_B \widehat{T}_{mul}$	$(29 * 8 + 2\|\widehat{\mathbb{G}}\|)$

[1] n_B and $n_{\mathcal{V}_i,B}$ are the number of content blocks of $\mathcal{SP}$ and $\mathcal{V}_i$, respectively. $n_{PID_{\mathcal{V}_i}}$ and n_D represent the number of pseudonymous of $\mathcal{V}_i$ and the number of data packets received by $\mathcal{V}_i$, respectively.
[2] S_{H_1} and S_B are the size of $H_1(\cdot)$ and B_i, respectively.



(a) Encryption; (b) Decryption; (c) Signature verification.

Fig. 2. The computation overhead.

VI. Performance Evaluation

A. Theoretical Analysis

We quantify the performance of our ESAC in term of computation overhead and communication overhead. Here, $\widehat{T}_{mul}$, $\widehat{T}_{par}$, and $\widehat{T}'_{exp}$ refer to the time required to perform one point multiplication over $\widehat{\mathbb{G}}_1$, one pairing operation over $\widehat{\mathbb{G}}_T$, and one exponentiation operation over $\widehat{\mathbb{G}}_T$, respectively. Moreover, we use T_{mul}, T_{exp}, T_{par}, and T'_{exp} to refer to the time of performing one point multiplication over $\mathbb{G}_1$, one exponentiation operation over $\mathbb{G}_1$, one pairing operation over $\mathbb{G}_T$, and one exponentiation operation over $\mathbb{G}_T$, respectively. We omit the computation overhead of hash operations and the detailed performance analysis is listed as follows:

Cache source generation: In this phase, $\mathcal{SP}$ should generate the encrypted content blocks EB_i of T_1. To achieve this, $\mathcal{SP}$ computes c_1 and c_2 for each EB_i, which consumes $T_{exp} + T_{par} + T'_{exp}$. After that, $\mathcal{SP}$ generates a signature for each EB_i, which consumes $2\widehat{T}_{mul}$. Thus, the total computation overhead of this phase is $(T_{exp} + T_{par} + T'_{exp} + 2\widehat{T}_{mul})n_B$ where n_B is the number of content blocks. Notice that this phase can be performed offline.

Vehicles Subscription: During this phase, $\mathcal{SP}$ issues $Cert_{\mathcal{V}_i,j}$ for the registered vehicles, which consumes $\widehat{T}_{mul}$ to compute $SK_{\mathcal{V}_i,j}$ and $2\widehat{T}_{mul}$ to compute the signature $\sigma_{\mathcal{V}_i,j}$. Therefore, the total computation overhead of this phase is $3\widehat{T}_{mul} n_{PID_{\mathcal{V}_i}}$, where $n_{PID_{\mathcal{V}_i}}$ is the number of pseudonyms of $\mathcal{V}_i$. The corresponding communication overhead is $(29 * 8 + 2|\widehat{\mathbb{G}}|)n_{PID_{\mathcal{V}_i}}$ bits. Notice that this phase can be performed offline, too.

Content Delivery: When $\mathcal{V}_i$ requests an interest packet, it should broadcast message (1), which requires $2\widehat{T}_{mul}$ to compute the signature Sig_1.

On receiving the interest request message (1), $\mathcal{V}_j$ should execute $2\widehat{T}_{par}$ to verify $Cert_{\mathcal{V}_i,j}$. Then, it requires $\widehat{T}_{mul} + 3\widehat{T}_{par} + \widehat{T}'_{exp}$ to verify Sig_1. When $\mathcal{V}_j$ returns the corresponding data packet (3), it consumes $2\widehat{T}_{mul}$ to generate the signature Sig_2. The total computation overhead is $(3\widehat{T}_{mul} + 5\widehat{T}_{par} + \widehat{T}'_{exp})$. The corresponding message size of message (1) and data packet (3) is about $(37 * 8 + 3|\widehat{\mathbb{G}}|)$ bits and $(37 * 8 + 4|\widehat{\mathbb{G}}| + S_{H_1} + S_B)$ bits, respectively. Here $|\widehat{\mathbb{G}}|$ is the length of $\mathbb{G}_1$. S_{H_1} is the size of $H_1(\cdot)$ and S_B is the size of B_i.

After receiving n_D data packets from different vehicles, $\mathcal{V}_i$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3004459, IEEE Transactions on Vehicular Technology

should first verify $Cert_{\mathcal{V}_j,i}$, which consumes $2n_D\widehat{T}_{par}$. Then it verifies the validity of signatures by performing a batch verification, which consumes $n_D\widehat{T}_{mul}+3\widehat{T}_{par}+\widehat{T}'_{exp}$. $\mathcal{V}_i$ gets B_i by computing formula (4) which requires T_{par}. Therefore, for n_D data packets, $\mathcal{V}_i$ consumes $(n_D\widehat{T}_{mul}+2n_D\widehat{T}_{par}+3\widehat{T}_{par}+\widehat{T}'_{exp}+n_DT_{par})$.

Periodic Update: During the update phase, $\mathcal{SP}$ should compute $rk_{T_1\to T_2}$, which consumes $T_{exp}+T_{mul}$. For n_B content blocks, it needs $2n_BT_{par}$ to update EB_i to EB'_i and $2\widehat{T}_{mul}$ to generate new signatures. Thus, $\mathcal{SP}$ consumes $(T_{exp}+T_{mul}+2n_BT_{par}+2n_B\widehat{T}_{mul})$. For $\mathcal{V}_i$ which has $n_{\mathcal{V}_i,B}$ content blocks, it needs $2n_{\mathcal{V}_i,B}T_{par}$ to update EB_i to EB'_i. The message sizes of message (5) and (6) are $(58*8+3|\widehat{\mathbb{G}}|)$ bits and $(29*8+2|\widehat{\mathbb{G}}|)$ bits, respectively. Notice that we omit the size of the encryption contents such as $E_{PK_{\mathcal{V}_i}}(rk_{T_1\to T_2}, SK_{T_2}, IK_{T_2})$.

Finally, we summarize the computation overhead in Table IV.

B. Numerical Simulations

1) Implementation: We implement our ESAC using PBC library [34] with Type A pairing parameters which is equivalent to 1024 bits Discrete Logarithm security. We perform our design on a computer with an Intel(R) Core(TM) i9-8950HK CPU of 2.90 GHz and 8 GB memory with Ubuntu 18.04. We evaluate ESAC in terms of three performance metrics: (1) Initial encryption (before re-encryption) time and re-encryption time; (2) Initial decryption time and the re-encryption decryption time; (3) Signature verification time with and without batch verification. The detailed simulation results are presented as follows:

Fig. 2(a) shows the initial encryption time and re-encryption time under various number of blocks ranging from 25 to 200 for our ESAC. It costs about 316.9 ms and 208.5 ms to realize initial encryption and re-encryption of 200 messages for ESAC, respectively. Notice that we omit the XOR operation with B_i. As shown in Fig. 2(a), the encryption time of these two phases almost increases linearly with the number of blocks while the initial encryption causes more computation overhead than the update phase. The reason is that to initially encrypt a content block, vehicle should conduct two exponentiation operations while the time for executing one exponentiation operation is larger than one pairing operation in our implementation. Here, we make vehicles implement the re-encryption or update process to save bandwidth while the computation overhead is acceptable.

Fig. 2(b) shows the corresponding decryption time under various number of blocks (from 25 to 200). We observe that it costs about 299.63 ms and 214.4 ms to decrypt 200 initial encryption and re-encryption messages in ESAC, respectively. As shown in Fig. 2(b), the decryption time of these two phases almost grows linearly with the increasing number of blocks while the decryption time in re-encryption phase consumes about two times as the initial phase. However, the computation overhead of these two phases are both acceptable.

In order to display the efficiency of our batch signature verification, we implement the signature verification process with and without batch verification, respectively. We repeat each implementation for 10 times. Fig. 2(c) shows that verification delay rises linearly when the number of messages increases. In contrast, the batch verification is much more efficient. It costs about 588.2 ms and 203.93 ms to verify 200 messages without batch verification and with batch verification, respectively. We use the batch verification, which makes the number of pairing operations consistent with respect to the number of verification messages.

TABLE V
SIMULATION PARAMETERS FOR NDNSIM

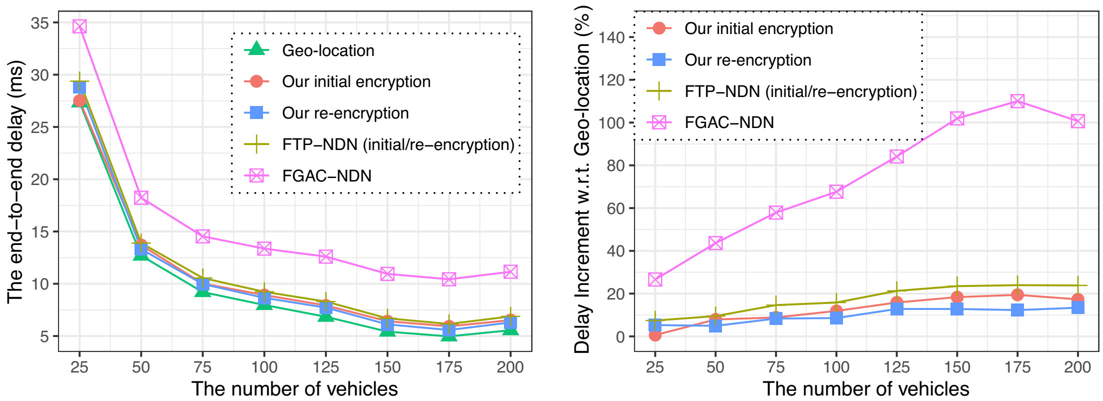
Parameter	Value
Simulation area	10 km× 100 m
Simulation time	1000 s
Speed of vehicle	20 m/s
Data packet size	1024 bytes
Wireless protocol	802.11a
Wireless data rate	OfdmRate24Mbps
Radio propagation model	Nakagami

2) Networking Implementation: To assess the network performance, our ESAC is performed in a NS3-based open-source NDN simulator named ndnSIM [35] and the geolocation-based forwarding scheme [6, 8] is used as the data forwarding strategy. As studied in [6, 8], geolocation-based NDN forwarding strategy can ensure efficient and reliable packet delivery in urban VANET scenarios. The simulation parameters are listed in TABLE V. To better study our network performance, we conduct the simulation in two scenarios: scenario 1 where vehicles are distributed with different densities, and scenario 2 where vehicles have different distances between each other.

Scenario 1: Different distribution densities of vehicles. In this scenario, a different number of vehicles are set along a straight line which are uniformly distributed within 1000 m. A data publisher is arranged at the beginning of the vehicle cohort in the moving direction while an interest requester is arranged at the end of the vehicle cohort. We compare ESAC with AccConF [13], FTP-NDN [19], FGAC-NDN [20], and geo-location forwarding scheme [6, 8]. Here, we omit the results of AccConF in figures because of its bad performance.

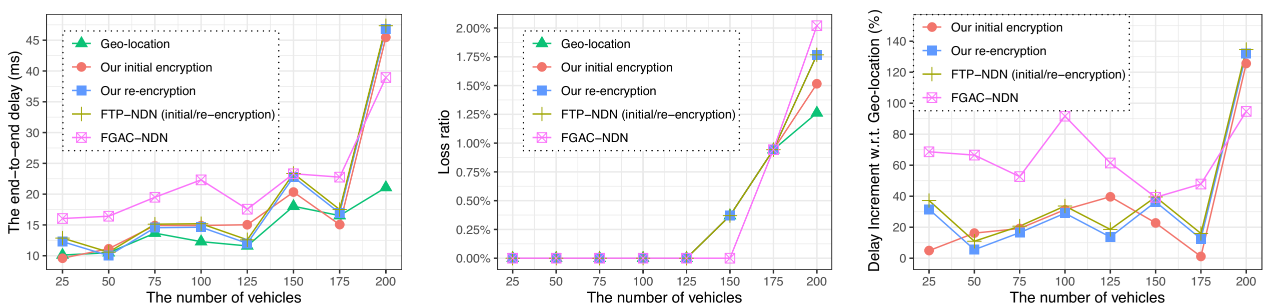
First, we focus on the incurred communication overhead caused by the security operations. Here, we set the size of B_i to 512 bits. Fig. 3 shows the average end-to-end delay to retrieve data packets from 1000 m away for an interest packet under different densities of vehicles. For the initial ciphertext (before re-encryption) of ESAC, it needs about 27.5 ms and 6.5 ms to retrieve a data packet when the number of vehicles is 25 and 200, respectively. While for the re-encryption ciphertext, the corresponding delay is about 28.8 ms and 6.3 ms, respectively. The average delay is about 10.9 ms and 10.8 ms for the initial ciphertext and re-encryption ciphertext, respectively. In contrast, it takes 159.9 ms for AccConF, 11.4 ms for the initial/re-encryption ciphertext of FTP-NDN, 15.7 ms for FGAC-NDN, and 10.0 ms for geo-location scheme. Therefore, the efficiency of our scheme is obvious. Compared to geo-location scheme, the additional delay is mainly due to the computation operations of our

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3004459, IEEE Transactions on Vehicular Technology



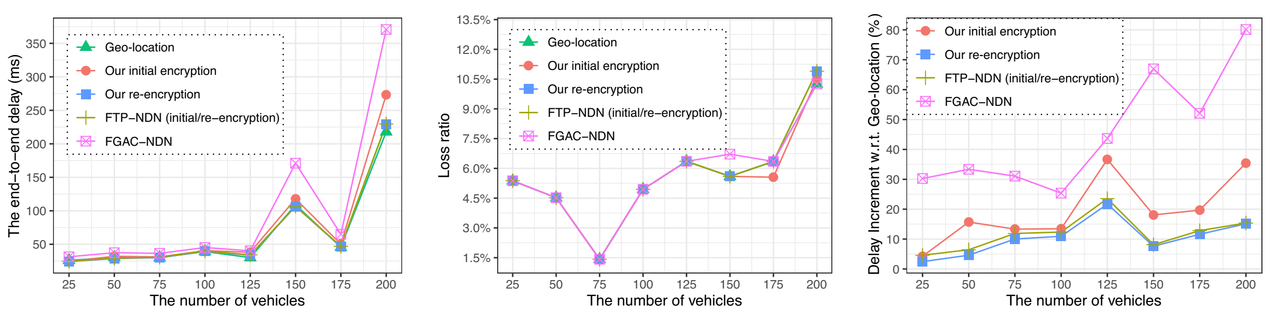
(a) The average end-to-end delay; (b) Retrieval delay increment ratio.

Fig. 3. The average end-to-end delay under different densities of vehicles ($B_i = 64B$).



(a) The average end-to-end delay; (b) The packet loss ratio; (c) Retrieval delay increment ratio.

Fig. 4. The network performance for 40% vehicles sending requests simultaneously ($B_i = 10KB$).



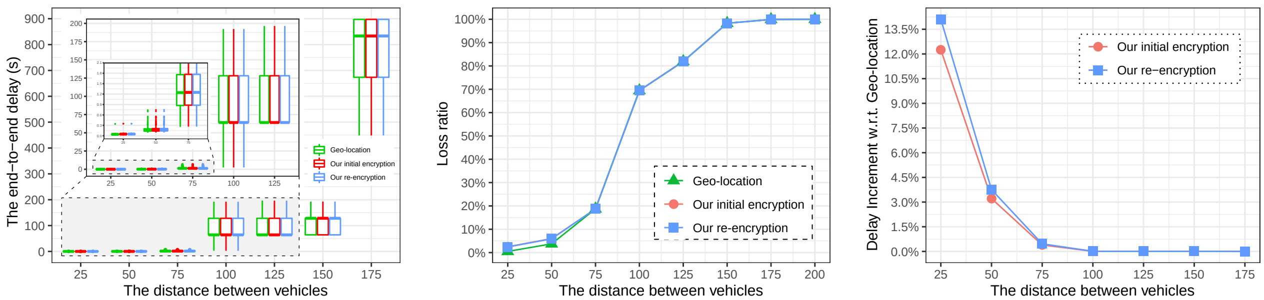
(a) The average end-to-end delay; (b) The packet loss ratio; (c) Retrieval delay increment ratio.

Fig. 5. The network performance for 40% vehicles sending requests simultaneously ($B_i = 20KB$).

secure scheme. Besides, the end-to-end delay decreases with increasing density of vehicles as the number of hops decreases. Finally, we present the corresponding delay increment ratio comparing with geo-location scheme [6, 8]. Fig. 3(b) shows that the delay increment ratio increases with more dense vehicles. This is because with the end-to-end delay decreasing, the computation overhead has more influence on the delay.

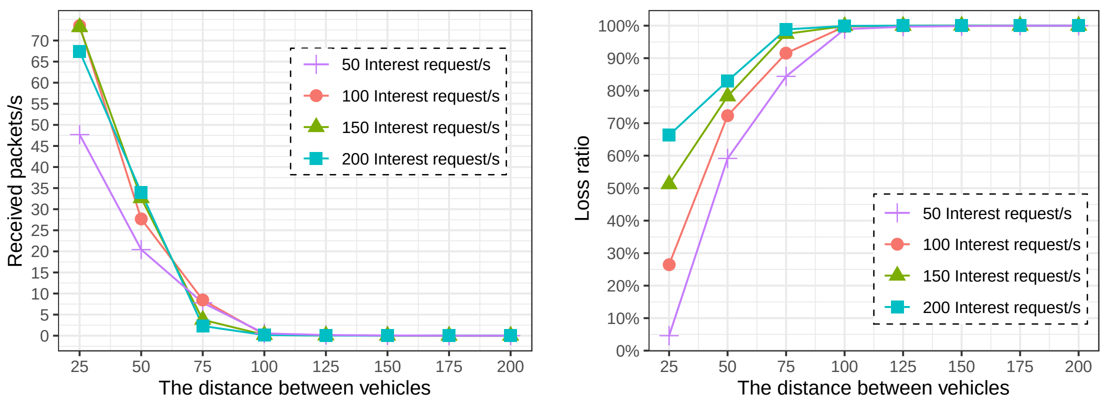
Moreover, we study the reliability of our scheme by sending more interest requests under different densities of vehicles. We assume that 40% vehicles send requests simultaneously and examine the corresponding average end-to-end delay. As shown in Fig. 4, when $B_i = 10$ KB, the average end-to-end delay decreases to 18.3 ms, 18.8 ms, 161.7 ms, 19.3 ms, 22.1 ms, and 14.2 ms for initial ciphertext of ESAC, re-encryption ciphertext of ESAC, AccConF, the initial/re-encryption ciphertext of FTP-NDN, FGAC-NDN, and geo-location scheme, respectively. While for $B_i = 20$ KB, it increases to 75.4 ms, 67.2 ms, 265.2 ms, 67.8 ms, 99.6 ms, and 66.0 ms, respectively, since the end-to-end delay increases sharply as shown in 5(a). We also present the corresponding message loss ratio as show in 5(b) and 5(c). Note that the end-to-end delay of 75 vehicles when $B_i = 20$ KB decreases significantly. This is because the distance between vehicles increases to a certain level while the increment of vehicles still have little influence on the network overhead, which leads to the lowest end-to-end delay.

Scenario 2: Different distances between vehicles. In this scenario, 50 vehicles are arranged along a straight line and



(a) The end-to-end delay; (b) The packet loss ratio; (c) Retrieval delay increment ratio.

Fig. 6. Network performance with different distances (m) between vehicles.



(a) The received data packets; (b) The loss ratio.

Fig. 7. The received packets with different interest request ratio under different distance (m).

travel at the same constant speed of 20 m/s. The distance between two neighbouring vehicles is randomly chosen from 25 m to 200 m. A data publisher is arranged in the front of the vehicle cohort in the moving direction while an interest requester is arranged at the tail of the vehicle cohort.

In the execution, the consumer/requester sends 10 interest packets every second. Fig. 6(a) shows the average end-to-end delay to retrieve data packets from 1 km to 10 km away for an interest packet. In the initial encryption phase (before re-encryption), it needs about 49.5 ms and 760.24 s to retrieve a data packet when the distance is 25 m and 175 m, respectively. In contrast, in the re-encryption phase, the corresponding delay is about 50.6 ms and 760.25 s. It is obvious that the delay increases dramatically when two vehicles are far away from each other. Moreover, we present the corresponding packet loss ratio in Fig. 6(b). Obviously, the loss ratio increases significantly when two vehicles are far away from each other, especially when the distance is more than 75 m. Apart from this, we observe that the extra cost incurred by the computation operation has little impact on network performance (i.e., loss ratio) when there is a larger inter-vehicle distance. This is because with the end-to-end delay increasing, the computation delay has less influence on the delay.

We also present the number of received packets under different frequencies of Internet requests (e.g., 50 interests/s, 100 interests/s, 150 interests/s, and 200 interests/s) under different distances between two adjacent vehicles as shown in Fig. 7. The average number of received packets is 9.6 for 50 packets/s, 13.8 for 100 packets/s, 13.7 for 150 packets/s, and 13.0 for 200 packets/s. Specifically, when the inter-vehicle distance is more than 100 m, vehicles cannot receive packets anymore. Thus, to guarantee the content delivery performance, we should decrease the number of interest request frequencies and the distance between vehicles.

VII. Conclusion

In this study, we reveal the security challenges when applying NDN to VANETs, and propose corresponding solutions (ESAC) to support efficient and privacy-preserving access control for content delivery in V-NDN. First, we design a proxy re-encryption algorithm to support access control, revocation operations and updated operations, even when trusted authority is offline. To deal with the anonymous authentication and integrity verification, we adopt pseudonyms and identify-based signature. Finally, we design an incentive method with hashed certificate to ensure the utility of NDN in VANETs. The security analysis shows that ESAC can meet the security requirements in V-NDN. Moreover, simulation results show that the proposed secure scheme incurs insignificant overhead on network performance.

VIII. Acknowledgement

This work was supported by the National Natural Science Foundation of China (No.61702231, No.U1764263) and US

National Science Foundation under grant IIS-1722791 and CNS-1717736. The Natural Science Foundation of Jiangsu Province (BK20170556), Jiangsu Provincial Research Scheme of Natural Science for Higher Education institutions (No. 17KJB520005).

References

[1] S. Jiang, J. Liu, L. Wang, and Y. Fang, "A secure data forwarding scheme in vehicular named data networking," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 206–212.

[2] J. Sun, X. Zhu, C. Zhang, and Y. Fang, "Rescueme: Location-based secure and dependable vanets for disaster rescue," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 659–669, 2011.

[3] A. Rowstron and G. Pau, "Characteristics of a vehicular network," *University of California Los Angeles, Computer Science Department, Tech. Rep*, pp. 9–17, 2009.

[4] "Named data networking," http://named-data.net/.

[5] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[6] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Rapid traffic information dissemination using named data," in *Proceedings of the 1st ACM workshop on emerging name-oriented mobile networking design-architecture, algorithms, and applications*. ACM, 2012, pp. 7–12.

[7] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Vanet via named data networking," in *Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2014*. IEEE, 2014, pp. 410–415.

[8] C. Bian, T. Zhao, X. Li, and W. Yan, "Boosting named data networking for efficient packet forwarding in urban vanet scenarios," in *Proceedings of IEEE International Workshop on Local and Metropolitan Area Networks (LANMAN), 2015*. IEEE, 2015, pp. 1–6.

[9] G. Grassi, D. Pesavento, G. Pau, L. Zhang, and S. Fdida, "Navigo: Interest forwarding by geolocations in vehicular named data networking," *arXiv preprint arXiv:1503.01713*, 2015.

[10] R. A. Rehman and B. S. Kim, "Lomcf: Forwarding and caching in named data networking based manets," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9350–9364, 2017.

[11] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[12] S. H. Ahmed, S. H. Bouk, M. A. Yaqub, D. Kim, H. Song, and J. Lloret, "Codie: Controlled data and interest evaluation in vehicular named data networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3954–3963, 2016.

[13] S. Misra, R. Tourani, F. Natividad, T. Mick, N. E. Majd, and H. Huang, "Accconf: An access control framework for leveraging in-network cached data in the icn-enabled wireless edge," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[14] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012, pp. 85–90.

[15] B. Hamdane, A. Serhrouchni, and S. G. E. Fatmi, "Access control enforcement in named data networking," in *IEEE Internet Technology & Secured Transactions (ICITST)*. IEEE, 2013, pp. 576–581.

[16] Q. Li, P. P. C. Lee, P. Zhang, P. Su, L. He, and K. Ren, "Capability-based security enforcement in named data networking," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 5, pp. 2719–2730, 2017.

[17] C. Wood and E. Uzun, "Flexible end-to-end content security in ccn," in *Proceedings of the IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE, 2014, pp. 858–865.

[18] Q. Li, X. Zhang, Q. Zheng, R. Sandhu, and X. Fu, "Live: Lightweight integrity verification and content access control for named data networking," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 308–320, 2015.

[19] C. I. Fan, I. T. Chen, C. K. Cheng, J. J. Huang, and W. T. Chen, "Ftp-ndn: File transfer protocol based on re-encryption for named data network supporting non-designated receivers," *IEEE Systems Journal*, vol. 12, no. 1, pp. 473–484, 2018.

[20] Y. Tseng, C. Fan, and C. Wu, "Fgac-ndn: Fine-grained access control for named data networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 143–152, 2019.

[21] K. Xue, P. He, X. Zhang, Q. Xia, D. S. L. Wei, H. Yue, and F. Wu, "A secure, efficient, and accountable edge-based access control framework for information centric networks," *IEEE/ACM Transactions on Networking (TON)*, 2019.

[22] S. DiBenedetto, P. Gasti, G. Tsudik, and E. Uzun, "Andana: Anonymous named data networking application," *arXiv preprint arXiv:1112.2205*, 2011.

[23] M. Nabeel, N. Shang, and E. Bertino, "Efficient privacy preserving content based publish subscribe systems," in *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*. ACM, 2012, pp. 133–144.

[24] C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology-CRYPTO'87*. Springer, 1988, pp. 369–378.

[25] M. K. Kiskani and H. R. Sadjadpour, "A secure approach for caching contents in wireless ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 249–10 258, 2017.

[26] C. Ghali, M. A. Schlosberg, G. Tsudik, and C. A. Wood, "Interest-based access control for content centric networks," in *Proceedings of the 2nd International Conference on Information-Centric Networking*. ACM, 2015, pp. 147–156.

[27] H. Ding, C. Zhang, Y. Cai, and Y. Fang, "Smart cities on wheels: A newly emerging vehicular cognitive capability harvesting network for data transportation," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 160–169, 2018.

[28] K. A. Shim, "A round-optimal three-party id-based authenticated key agreement protocol," *Information Sciences*, vol. 186, no. 1, pp. 239–248, 2012.

[29] H. Lin, J. Shao, C. Zhang, and Y. Fang, "Cam: cloud-assisted privacy preserving mobile health monitoring," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 985–997, 2013.

[30] L. Zhao, J. Wang, J. Liu, and N. Kato, "Optimal edge resource allocation in iot-based smart cities," *IEEE Network*, vol. 33, no. 2, pp. 30–35, 2019.

[31] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "Ac-poca: Anticoordination game based partially overlapping channels assignment in combined uav and d2d-based networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1672–1683, 2018.

[32] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1786–1802, 2011.

[33] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *Public Key Cryptography-PKC 2004*. Springer, 2004, pp. 277–290.

[34] https://crypto.stanford.edu/pbc/.

[35] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnsim: Ndn simulator for ns-3," *Technical Report NDN-0005*, 2012.