IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 17, 2020, accepted December 27, 2020, date of publication December 30, 2020,
date of current version January 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048273

Secure, Robust and Flexible Cooperative Downloading Scheme for Highway VANETs

YAN ZHANG[1,2], LEI ZHANG[1,2], (Member, IEEE), DINGKAI NI[1,2], KIM-KWANG RAYMOND CHOO[3,4], (Senior Member, IEEE), AND BURONG KANG[1,2]

[1]Engineering Research Center of Software/Hardware Co-design Technology and Application, Ministry of Education, East China Normal University, Shanghai 200062, China
[2]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China
[3]Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, USA
[4]Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, USA

Corresponding author: Lei Zhang (leizhang@sei.ecnu.edu.cn)

This work was supported in part by the National Key R&D Program of China (No. 2017YFB0802000), in part by the NSF of China under Grant 61972159, Grant 61572198, Grant 61321064, and Grant 61702259; in part by the Open Research Fund of Engineering Research Center of Software/Hardware Co-design Technology and Application, Ministry of Education (East China Normal University); and in part by the Fundamental Research Funds for the Central Universities. The work of Kim-Kwang Raymond Choo was supported by the Cloud Technology Endowed Professorship.

ABSTRACT The increasing popularity of smart vehicles and vehicular ad hoc networks (VANETs) has reinforced the importance of Internet connectivity. However, existing solutions (e.g., those based on cooperative downloading via drive-thru Internet) have a number of limitations, partly due to the challenges in selecting reliable proxy vehicles in a fast moving driving situation, and balancing between flexible data downloading and achieving strong security. Therefore, in this paper we propose a secure, robust and flexible cooperative downloading scheme based on our reputation based selection mechanism and ordered signatures. Using the reputation based selection mechanism, only vehicles with the highest expected downloading capacities will be selected as proxy vehicles. This helps us to avoid selecting less reliable proxy vehicles. The flexible data downloading is achieved by dividing a file into small blocks so that a proxy vehicle can flexibly vary the file / data download, for example based on existing condition. Our scheme also achieves strong security, in the sense that it realizes traditional authentication, privacy preservation and message confidentiality, as well as a newly introduced security requirement (i.e., process authentication: authenticates the order of file blocks and the order of the assisting vehicles using ordered signatures).

INDEX TERMS Vehicular ad hoc network, cooperative downloading, security, privacy, authentication.

I. INTRODUCTION

Vehicular ad-hoc network (VANET) can facilitate vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, for example in an intelligent transportation system (ITS). The primary goal of VANET is to improve traffic efficiency and safety [1]–[7]. However, drivers/passengers might prefer to access the Internet on-the-move, and enjoy online services, such as downloading of contents (e.g., video and audio) [8]. Although cellular network can support such Internet services on-the-move, the signal may be poor with low coverage when the vehicle is moving at high speed in most countries and the cost may be relatively high [9]. Also, cellular networks may face severe traffic overload problems caused by excessive mobile data demands [23]. Roadside infrastructures (also referred to as wireless Internet access points), such as those along expressways or highways, can also be leveraged to improve Internet access, with lower costs. Such Internet access that is provided by the roadside infrastructure is also referred to as drive-thru Internet in the literature. Drive-thru Internet, generally characterized as high-capacity, low-cost and low-coverage, is becoming more popular [10].

Downloading of files (broadly defined to include video, audio, and other content) is one of the most widely used applications in drive-thru Internet scenarios. According to [11], for example, only about 10 MB data can be downloaded by a high mobility vehicle on highway when it passes through

The associate editor coordinating the review of this manuscript and approving it for publication was Oussama Habachi.

a roadside infrastructure due to the low-coverage limitation of drive-thru Internet. Further, there are generally limited roadside infrastructures deployed along a long stretch of highway especially at rural areas, partly due to costs (e.g., installation and maintenance). Therefore, it is challenging to provide vehicles on highways with ubiquitous access to the Internet 24/7, and this affects the quality of service (QOS) and quality of experience (QoE) [24]. A promising method to download large files is cooperative downloading, which enables a file requested by a client vehicle to be downloaded with the help of other vehicles in the vicinity (also referred to as proxy and assisting vehicles; See Section II-A) [12]–[15]. The proxy vehicles can help download files and the the assisting vehicles can help forward this file. However, there are various issues have to be addressed before cooperative downloading is practical.

Security is one of the major concerns. In VANET, an attacker may inject fake messages to mislead one or more vehicles and vehicles may also behave maliciously [16]. For instance, a malicious proxy vehicle may first accept a cooperative downloading request of a client vehicle, but subsequently refuse to help the client to download the file. Hence, ensuring the authentication of messages in VANET is crucial. If fake message or malicious behavior of an entity is detected, then we need to have some form of investigation capability. Vehicle privacy is another challenge, in the sense that vehicles' identities and location history should not be disclosed to other entities. Finally, file downloading is usually a type of paid service. Messages sent and received by a vehicle may contain sensitive data (e.g., the content of the requested file such as an electronic transaction). Hence, we have to protect the confidentiality of a message if it contains sensitive data as well.

In addition to security and privacy, both robustness and flexibility are equally important features. Robustness means that the system is reliable and not easy to be breakdown even though there are malicious behaviours. In cooperative downloading, malicious (e.g., accept a cooperative downloading request, but does not not honestly execute the scheme) or inactive (e.g., does not want to forward the file blocks) vehicles in the VANET can result in communication breakdown. Hence, this reinforces the importance of robustness. The latter guarantees the reliability of a cooperative downloading scheme even if one or several proxy/assisting vehicles are malicious and/or inactive. Robustness of a cooperative downloading scheme also implies an adequate initiative mechanism, where the vehicles in the system are willing to participant in cooperative downloading tasks and honestly execute the tasks. Flexibility implies that the downloading capacity of each vehicle is not fixed but determined by its own condition, such as position, speed, and network condition. We remark that flexibility is an important property, since many factors (e.g., vehicle failure, difference between the actual driving status from the estimated one, and network congestion) can influence the download capacity of a proxy vehicle and lead to large download failure rate if we assume the downloading capacity of a proxy vehicle is predetermined.

A. RELATED WORK

Cooperative downloading in VANETs was first studied in [17]. However, this scheme as well as several later constructions [18]–[20] did not consider the security issues. We note that message confidentiality in cooperative downloading is easy to achieve, since we just need to guarantee the end-to-end security which can be easily realized by using encryption schemes [21]–[23]. While authentication is more challenging, particulary, if flexibility and the fact that the file blocks downloaded by proxy vehicles have to be forwarded to the client vehicle with the help of assistance vehicles, have to be addressed. In fact, existing cooperative downloading schemes assume a file is separated into large file blocks whose sizes are pre-determined by the downloading capacities of proxy vehicles [19], [23]. However, to realize flexibility, unlike the existing ones, the size of a file block should not be large, so that a proxy vehicle may determine the number of file blocks that it may download based on its own condition. This leaves the problem that a receiver has to check whether the received file blocks are in order. Further, for the later fact, it would be useful if we may learn the order of the assisting vehicles, so that they can be rewarded according to this order. Existing schemes [21], [22] only deal with traditional authentication property which cannot be used to verify the order of the file blocks and the order of the assisting vehicles.

Several mechanisms have been introduced to enable robust cooperative downloading which is mostly determined by selecting reliable proxy vehicles and enabling reliable forwarding. For the choosing of proxy vehicles, the wildly used mechanisms are those based on reputation, credit and/or contract. The reputation based one relies on keeping track of each vehicle's reputation value so that vehicles with low reputation values can be excluded from the candidate lists [15], [23], [25]–[27]. The credit based one mainly uses virtual currency (or even e-cash) to motivate the initiativeness of vehicles and punish the malicious behaviors of vehicles [23], [28]–[30]. The contract based one generally uses contract to restrict the behaviors of vehicles [26], [27], [31], [32]. We note that, to ensure efficient cooperative downloading, it is prefer to select most reliable vehicles. That is, the vehicles with highest downloading capacities have to be selected as proxy vehicles. The existing mechanisms use more or less random selection mechanism and cannot address this issue well. As to reliable forwarding, mechanisms based on credit and/or multi-path are generally applied [15], [23]. Our solution also employs these two mechanisms.

B. OUR CONTRIBUTION

Existing cooperative downloading schemes face the challenges of selecting most reliable proxy vehicles, allowing flexible data downloading and authenticating the order of the file blocks/assistance vehicles. To deal with these challenges,

we propose a scheme for secure, robust and flexible cooperative downloading.

The proposed scheme is based on our reputation based selection mechanism, which aims to improve the efficiency and robustness of cooperative downloading. In our mechanism, we use expected downloading capacities to evaluate the reliability of proxy vehicles. The expected downloading capacities of vehicles are determined by the reputation values and the maximum downloading capacities of vehicles. The reputation value of a vehicle is managed by a trusted authority and reflects the activeness and honesty of the vehicle in participating in a cooperative downloading task. If a vehicle often involves in cooperative downloading tasks and behaves honest, then the vehicle will have high reputation value. Else if a vehicle behaves maliciously in cooperative downloading tasks, then the vehicle will be punished by reducing its reputation value. The maximum downloading capacity of a vehicle is related to the vehicle's position, speed, etc. With the reputation values and the maximum downloading capacities of the vehicles, a client vehicle may calculate the expected downloading capacities of the vehicles and choose the proxy vehicles with the highest expected downloading capacities. Based on the reputation based selection mechanism, we then propose a cooperative downloading scheme for highway VANETs.

For the security, our scheme satisfies privacy preservation, message confidentiality and (traditional) authentication. Specifically, we introduce the notion of *process authentication* which authenticates the order of file blocks and the order of the assisting vehicles, and show that our scheme also realize *process authentication*. In our scheme, each vehicle is pre-loaded a pool of short term anonymous certificates (and the corresponding private-public key pairs). The anonymous certificate is used to bind the reputation value and public key with the pseudonym of a vehicle. In each new run of our scheme, each vehicle has to use a fresh anonymous certificate. By using this mechanism, the privacy preservation property of our scheme is achieved. The message confidentiality is achieved by using key agreement and symmetric encryption. Traditional authentication is achieved by using signatures and MACs. In particular, the new notion named process authentication is proposed. Screening of signatures and ordered multi-signatures are used to efficiently verify the order of file blocks and the order of the assisting vehicles respectively. If there is a malicious behavior, it can be detected by using the process authentication. In other words, it can help the client vehicle recover the full file and award or punish the assisting vehicles if the process authentication is realized. To the best of our knowledge, we address the issue of process authentication for the first time.

The proposed scheme is robust and flexible. In our scheme, a file is divided into relative small blocks, so that each proxy vehicle can download flexible amount of file data based on its own downloading capacity. The robustness of our scheme is realized by using contract, reputation, credit and multi-path forwarding mechanisms. When a client vehicle launches a cooperative downloading task, it has to publish a contract which defines how the proxy/assistance vehicles will be awarded and punished, etc. All the candidate proxy vehicles have to sign the contract. The client vehicle will select the proxy vehicles using our reputation based selection mechanism to avoid choosing low reliable vehicles. A proxy vehicle then may download file blocks flexibly and deliver the downloaded file blocks using multi-path forwarding mechanism to ensure reliable forwarding. Finally, by applying the reputation and credit mechanisms, a honest vehicle will be rewarded by increasing its reputation value and obtaining e-cash (credit), while a malicious vehicle will be punished by lowing down its reputation value. Simulation shows that our scheme has low download delay and high download success rate.

The remainder of this paper is organized as follows. Section II is the background. In Section III, we introduce our reputation based selection mechanism. Our proposal is proposed in Section IV. Security and performance analyses are given in Section V. In Section VI, we perform several simulations to evaluate the efficiency of our scheme. Finally, Section VII concludes the paper.

II. BACKGROUND

In this section, we introduce the system architecture and design goals of our scheme, as well as aggregate and multi-signature schemes.

A. SYSTEM ARCHITECTURE

Figure 1 illustrates our system architecture, which consists of following entities:

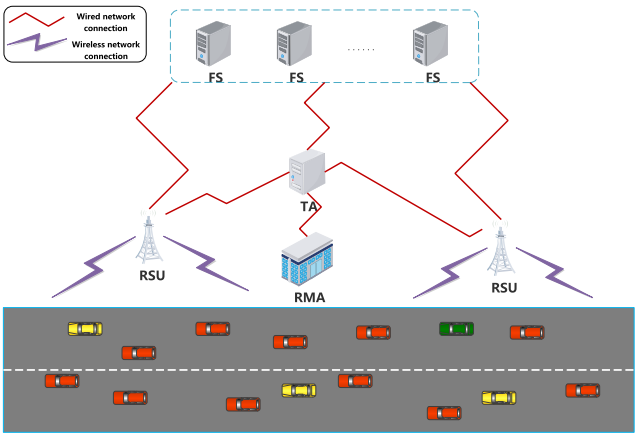


FIGURE 1. System architecture.

- Trusted Authority (TA): This entity is a trusted third party, and is tasked with the generation of the system's master key and parameters. All the other entities in the system must be enrolled by the TA to obtain their certificates.
- Reputation Management Authority (RMA): This entity is a trusted third party, and is tasked with the management of enrolled vehicles' reputation values.

- Roadside units (RSUs): RSUs are located along the roadsides, and embedded with sensory, processing, and wireless communication modules.
- File Servers (FSs): FSs are trusted and offer various download services (e.g., music, software and video) for all vehicles enrolled in the system.
- Vehicles: Each vehicle is equipped with an onboard unit (OBU). It is assumed that an OBU is a device with sufficient computational capability and installed with a wireless communication module, and secure storage medium. The vehicles in our cooperative downloading scheme are divided into client, proxy and assisting vehicles [23]. A client vehicle is the initiator of a cooperative downloading task. A file requested by the client vehicle will be downloaded from a FS with the help of proxy vehicles that will forward the file to the client vehicle with the help of assisting vehicles.

B. DESIGN GOALS

Our goal is to design a secure, robust and flexible cooperative downloading scheme for highway VANETs. The security requirements include authentication, privacy preservation and message confidentiality, as follows:

Authentication: All the entities in the system should be legitimate entities. A receiver may verify whether a message is from a legitimate entity in the system. Further, in our scheme, a file will be separated into several file blocks and forwarded to a client vehicle with the help of assisting vehicles. It is desirable if a client may check whether the received file blocks are in order, and if the RMA (and probably other entities that are interested in) may learn the order of the assisting vehicles so that it may reward the assisting vehicles according to this order. We say that a cooperative downloading scheme holds *process authentication* if the order of the file blocks and the order of the assisting vehicles can be verified.

Privacy preservation: It requires that an attacker cannot determine the true identity of a message sender and distinguish whether two messages in two independent executions of the cooperative downloading scheme are from the same sender.

Message confidentiality: Suppose a client $\mathcal{V}_i$ wants to download a file from FS $\mathcal{S}_j$. It requires that, with the exception of $\mathcal{V}_i$ and $\mathcal{S}_j$, no other entity can learn the content of the file as well as the file name.

In addition to above requirements, this paper also aims to address the following requirements:

Robustness: Malicious vehicles may exist in VANET. If a malicious vehicle is selected as a proxy vehicle, this may lead to download failure. Therefore, to ensure that a file can be downloaded successfully, we have to choose reliable vehicles as proxy vehicles. Further, a file will be forwarded to a client vehicle with the help of assisting vehicles. Reliable forwarding mechanism has to be designed to help a client vehicle to receive the file. We note that robustness also implies that the vehicles in the system are willing to participate in cooperative downloading tasks and honestly execute the tasks.

Flexibility: The downloading capacity of each vehicle is not fixed. A vehicle may determine the amount of file data that it can download according to its position, speed, network condition, etc.

C. AGGREGATE SIGNATURES AND MULTI-SIGNATURES

An aggregate signature scheme [33], [34] allows n signers to sign distinct messages while the resulting signatures can be aggregated into a single short signature. Instead of verifying n signatures, a verifier just needs to verify the validity of the aggregated signature to determine whether the n signers have truly signed the messages. This property not only significantly reduces the signature length to be transmitted but also greatly speeds up the signature verification procedure. We note that if the n signatures are from the same signer, then the scheme is referred to as screening of signatures [35]. Screening of signatures also allows the signatures to be aggregated into a single signature (i.e., screened signature) and verified in a batch.

A multi-signature scheme [36] can be viewed as a specific aggregate signature scheme. The main difference is that, in a multi-signature scheme, all the signers have to sign the same message. Multi-signature schemes can be further divided into broadcast multi-signature schemes and ordered multi-signature schemes. The difference is that the latter requires that the set of signers is ordered (i.e., each signer must aggregate his/her signature into the aggregate signature formed by all the previous signers), while the former has no such restriction.

III. REPUTATION BASED SELECTION MECHANISM

A key problem in a cooperative downloading scheme is how to choose proxy vehicles. This is due to the fact that the vehicles in VANET may be dishonest, selfish or even malicious. If selfish or malicious vehicles are chosen, this may lead to low download success rate even the communication breakdown. In order to overcome this problem, we propose a reputation based selection mechanism.

In our mechanism, we assume each vehicle has a reputation value managed by the RMA. The reputation value of a vehicle is smaller than 100% and reflects the download reliability of the vehicle. If a vehicle often involves in cooperative downloading tasks and behaves honest, then the vehicle will be awarded and have high reputation value. Else if a vehicle behaves maliciously in cooperative downloading tasks, then the vehicle will be punished by reducing the reputation value of the vehicle.

When a client vehicle wants to choose proxy vehicles, two factors need to be considered: the reputation value of a candidate proxy vehicle cr_i (for $0 \leqslant cr_i \leqslant 100\%$) and maximum downloading capacity of the candidate proxy vehicle omd_i. The later factor can be calculated based on the current position, driving direction and the speed of the candidate proxy vehicle. The client vehicle may calculate the expected

downloading capacity of the candidate proxy vehicle $f_i = cr_i \cdot omd_i$. The client vehicle chooses K vehicles with the highest expected downloading capacities as the proxy vehicles, which satisfy $f_1 + f_2 + \ldots + f_K \geqslant F_{size}$, where F_{size} is the size of downloading file and K is the minimum number of vehicles for which this equation is hold.

IV. OUR COOPERATIVE DOWNLOADING SCHEME

In this section, we propose our concrete cooperative downloading scheme.

A. HIGH-LEVEL DESCRIPTION

Our scheme consists of four stages: *System Setup, Enrollment, Cooperative Downloading* and *Adjudication*. Figure 2 simply shows the process of our scheme. In the first stage, the TA generates the system master key and the public system parameters. The master key is used to issue certificates for the entities in the systems. In the second stage, FSs and vehicles are enrolled by the TA. Each FS/vehicle will generate a private-public key pair and obtain a certificate issued by the TA corresponding to the private-public key pair. In the third stage, a client vehicle that wants to download a file may send a download request to the corresponding FS. With the help of nearby vehicles, it may download the file from the FS securely. As mentioned previously, the help vehicles are classified into proxy vehicles and assisting vehicles. Malicious and/or non-cooperative vehicles may exist in our system. The final stage shows how the vehicles (assisting and proxy vehicles) will be rewarded or punished. The key notations are shown in Table 1.

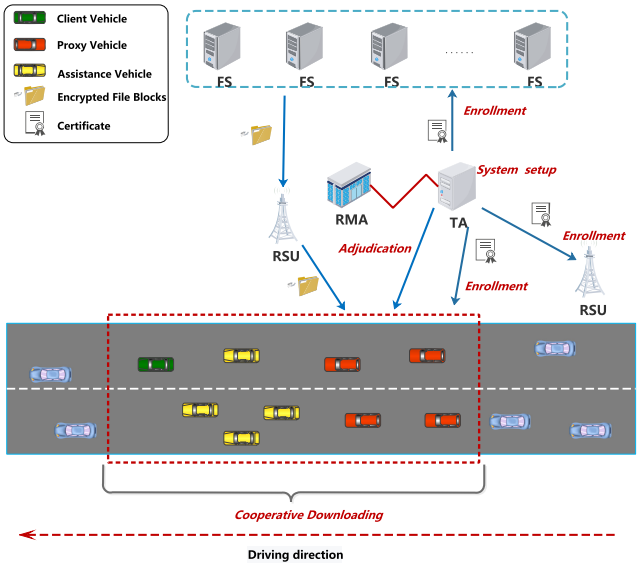


FIGURE 2. High-level description of our scheme.

B. SYSTEM SETUP

On input a security parameter λ, the TA performs the following steps:

TABLE 1. Key notations.

Notation	Definition
V_I	the client vehicle
V_{I_l}	a proxy vehicle
V_{a_x}	a assisting vehicle
S_J	the file server
v_i/u_i	the public/private key of a vehicle V_i
s_i/η_i	the public/private key of a file server S_i
ct_{V_i}	a short term anonymous certificate of V_i
ct_{S_i}	the certificate of S_i
sk_I/sk_{IJ_l}	the session key between V_I and S_J
sk_{I_l}	the session key between V_{I_l} and S_J
$info_i$	the status information of V_i (e.g., location, direction, speed,reputation value)
m_I	the contract of cooperative downloading
ec_I	the e-cash
EC_I	the encrypted e-cash
aux_I	any auxiliary message
F	the downloading file
F_{size}	the size of file F
F_{Name}	the name of file F
er	the encrypted file's name
F_u	a file block
T_u	a encrypted file block
$\tilde{T}_l'$	the data that should be forwarded
a_{J_l}	the key random number to calculate sk_{IJ_l}
Ω_{I_l}	the aggregate signature to help authenticate the order of file blocks
Υ_{a_x}	the ordered multi-signature to help record assisting vehicles

1) Select bilinear groups $\mathbb{B} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ of type 3[1] [37], [38], $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are three cyclic multiplicative groups of prime order q along with a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that satisfies $\hat{e}(g_1^{\mu}, g_2^{\nu}) = \hat{e}(g_1, g_2)^{\mu\nu}$ for all $\mu, \nu \in \mathbb{Z}_q^*$.
2) Select the master key $\alpha \in \mathbb{Z}_q^*$, set $p_1 = g_1^{\alpha}, p_2 = g_2^{\alpha}$.
3) Select hash functions $H_1 : \{0,1\}^* \to \mathcal{K}$, $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_3 : \{0,1\}^* \to \mathbb{G}_1$.
4) Select a symmetric encryption scheme $\mathcal{E}_Y(.)/\mathcal{D}_Y(.)$ and a keyed hash $H_Z(.)$. For simplicity, we assume $\mathcal{E}_Y(.)/\mathcal{D}_Y(.)$ and $H_Z(.)$ have the same key space $\mathcal{K}$.
5) Output the system parameters $\Omega = (\mathbb{B}, g_1, g_2, p_1, p_2, H_1 \sim H_3, H_Z(.), \mathcal{E}_Y(.)/\mathcal{D}_Y(.), \mathcal{K})$.

C. ENROLLMENT

The vehicles and FSs have to be enrolled by the TA to obtain their (anonymous) certificates [33]. The concrete procedures come as follows:

For a vehicle $\mathcal{V}_i$, to protect the privacy of a vehicle, the vehicle has to periodically update its private-public key pair. A pool of short term private-public key pairs will be pre-loaded by $\mathcal{V}_i$ and the corresponding anonymous certificates will issued by the TA. For the l-th private-public key pair, $\mathcal{V}_i$ selects $\mu_{il} \in \mathbb{Z}_q^*$ as its short term private key, sets $v_{il} = g_2^{\mu_{il}}$ as its short term public key. The TA also issues a short term anonymous certificate $ct_{V_{il}}$ corresponding to v_{il} for $\mathcal{V}_i$. To generate the certificate, the TA first requests the

[1]In type 3 bilinear groups, $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable homomorphism exists between $\mathbb{G}_1$ and $\mathbb{G}_2$ in either direction [40].

RMA for the current credit rating cr_i of $\mathcal{V}_i$, then generates a pseudonyme pid_{il} and sets $ct_{V_{il}}$ to be $(pid_{il}, vp, cr_i, v_{il}, \sigma_{il})$, where σ_{il} is the signature on $(pid_{il}, vp, cr_i, v_{il})$ signed using α, vp is a validity period. We note that it is not recommended to set vp to a very high value since cr_i may change over time.

For an FS $\mathcal{S}_i$, we do not need to consider its privacy. Assume the identity of $\mathcal{S}_i$ is id_{S_i}. $\mathcal{S}_i$ selects $\eta_i \in \mathbb{Z}_q^*$ as its private key, sets $s_i = g_2^{\eta_i}$ as it's public key. The TA also issues a certificate $ct_{S_i} = (id_{S_i}, vp_i, s_i, \sigma_i)$ corresponding to s_i for $\mathcal{S}_i$, where σ_i is the signature on (id_{S_i}, vp_i, s_i), vp_i is a validity period.

D. COOPERATIVE DOWNLOADING

Assume the client vehicle is $\mathcal{V}_I$ and the current corresponding private-public pair is $(\mu_I = \mu_{Il}, v_I = v_{Il})$, the server is $\mathcal{S}_J$ with private-public pair (η_J, s_J), the file to be downloaded is F with file name F_{Name}. If F is too large, $\mathcal{V}_I$ cannot download the file by itself. In this case, the vehicle needs to launch a cooperative downloading task. The *Cooperative Downloading* stage consists of three protocols: **Agreement**, **File Downloading** and **Message Delivery**. Fig. 3 shows the basic ideas in this stage. The agreement protocol is mainly used for a client vehicle to choose proxy vehicles. During the file downloading protocol, F is downloaded by the chosen proxy vehicles from the FS via an RSU. Each proxy vehicle will download several file blocks of F. Finally, the proxy vehicles forward the file blocks to the client vehicle with the help of assisting vehicles. Next, we show the details of the three protocols.

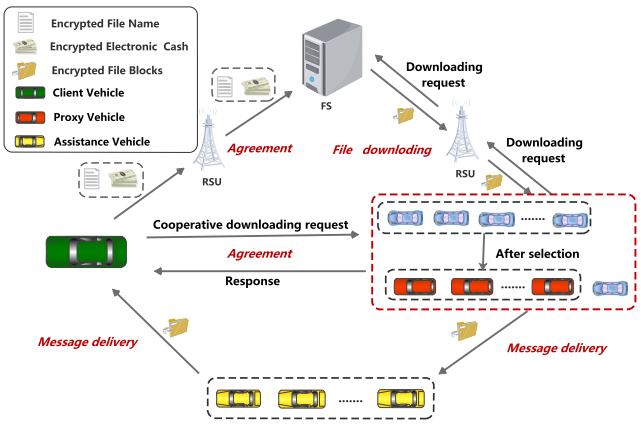


FIGURE 3. Basic ideas in cooperative downloading stage.

Protocol 1 (Agreement): In this protocol, $\mathcal{V}_I$ will broadcast a cooperative downloading request to the potential proxy vehicles and choose proper proxy vehicles based on the responses from the potential proxy vehicles. The protocol has following steps:

Step 1: $\mathcal{V}_I$ broadcasts a *unique* cooperative downloading request $req = (F_{size}, vp, Aux, \sigma_{\mathcal{V}_I}, ct_{V_I})$ to the nearby vehicles and waits for the responses, where vp is the valid period of the request, Aux is any additional information and $\sigma_{\mathcal{V}_I} = H_3(F_{size}, ct_{V_I})^{\mu_I}$ is the signature on (F_{size}, vp, Aux).

Step 2: We assume that the vehicles driving at the same direction will response this request. Each potential proxy vehicle that wants to join the cooperative downloading task has to verify the validity of $\sigma_{\mathcal{V}_I}$ by checking $\hat{e}(H_3(F_{size}, ct_{V_I}), v_I) \stackrel{?}{=} \hat{e}(\sigma_{\mathcal{V}_I}, g_2)$. If it is valid, it generates a signed response. To enable fast signature verification, we use the aggregate signature scheme in [35] to generate the signature in a signed response. In this case, the signatures from different potential proxy vehicles can be aggregated into a short aggregate signature and verified in a batch by $\mathcal{V}_I$. Suppose a potential proxy vehicle that wants to join the cooperative downloading task is $\mathcal{V}_{I_i}$ with private-public key pair $(\mu_{I_i}, v_{I_i} = g_2^{\mu_{I_i}})$. $\mathcal{V}_{I_i}$ generates and broadcasts the signed response $res_i = (info_i, \sigma_{\mathcal{V}_{I_i}}, ct_{V_{I_i}})$, where $info_i$ is $\mathcal{V}_{I_i}$'s status information (e.g., location, direction, speed, reputation value) to $\mathcal{V}_I$, $\sigma_{\mathcal{V}_{I_i}}$ is the aggregatable signature on $info_i$ under reg which is signed by using Algorithm 1 (i.e., the signature generation algorithm of the aggregate signature scheme in [35]).

Algorithm 1 A.Sign()

Input: $\mu_{I_i}, req, info_i$.
Output:
1: Compute $a = H_3(req, 0), b = H_3(req, 1), \gamma_{I_i} = H_2(info_i, req)$.
2: Compute an aggregatable signature $\sigma_{\mathcal{V}_{I_i}} = a^{\mu_{I_i}} b^{\mu_{I_i}\gamma_{I_i}}$.
3: **return** $\sigma_{\mathcal{V}_{I_i}}$.

Algorithm 2 A.Verify()

Input: $req, \{res_i = (info_i, \sigma_{\mathcal{V}_{I_i}}, ct_{V_{I_i}})\}_{i\in\{1,\ldots,K+x\}}$.
Output:
1: Extract v_{I_i} from $ct_{V_{I_i}}$ for $i \in \{1, \ldots, K+x\}$.
2: Compute the aggregate signature $\sigma_{\mathcal{V}_p} = \prod_{i=1}^{K+x} \sigma_{\mathcal{V}_{I_i}}$.
3: Compute $a = H_3(req, 0), b = H_3(req, 1), \gamma_{I_i} = H_2(info_i, req)$ for $i \in \{1, \ldots, K+x\}$.
4: Check $\hat{e}(\sigma_{\mathcal{V}_p}, g_2) \stackrel{?}{=} \hat{e}(a, \prod_{i=1}^{K+x} v_{I_i}) \hat{e}(b, \prod_{i=1}^{K+x} v_{I_i}^{\gamma_{I_i}})$.
5: **return** $\sigma_{\mathcal{V}_p}$ if the above equation holds or abort.

Step 3: Once $\mathcal{V}_I$ receives enough responses, it does the following:

1) Select $K + x$ proxy vehicles based on our *reputation based selection mechanism*, where K is the estimated minimal number of proxy vehicles, x is the redundancy to ensure successful downloading. Without loss of generality, we assume the chosen proxy vehicles are $\mathcal{V}_{I_1}, \ldots, \mathcal{V}_{I_K}, \ldots, \mathcal{V}_{I_{K+x}}$.
2) Verify the validity of the signed responses from the chosen proxy vehicles using Algorithm 2 (i.e., the signature verification algorithm of the aggregate signature scheme in [35]). $req, \{(info_i, ct_{V_{I_i}})\}_{i\in\{1,\ldots,K+x\}}$, $\sigma_{\mathcal{V}_p}$ have to be stored by $\mathcal{V}_I$.
3) Generate a partially signed contract which defines the identities of the chosen proxy vehicles, how the task will be paid, etc. as follows:

a) Choose $\theta_I \in \mathbb{Z}_q^*$, compute $a_I = g_2^{\theta_I}, b_I = s_J^{\theta_I}$, $d_I = s_J^{\mu_I}$ and a session key $sk_I = H_1(ct_{V_I}, ct_{S_J}, a_I, b_I, d_I, vp_I)$, where vp_I is a valid period. To protect the privacy of F, the file name of F is encrypted by computing $er = E_{sk_I}(F_{Name})$.
b) Suppose the cooperative download task will be paid for with e-cash EC_I. Compute encrypted e-cash $ec_I = E_{sk_I}(EC_I)$. Set the contract to be $m_I = (\sigma_{\mathcal{V}_p}, \mathbb{C}, id_{S_J}, er, ec_I, a_I, vp_I, aux_I)$, where $\mathbb{C} = (ct_{V_{I_0}}, \ldots, ct_{V_{I_{K+x}}})$, $ct_{V_{I_0}} = ct_{V_I}$, aux_I is any auxiliary message. Generate a partially signed contract $con_I = (m_I, \sigma_0)$, where σ_0 is a partial multi-signature on m_I and is generated using Algorithm 3 (i.e., the signature generation algorithm of the multi-signature scheme in [39]).
c) con_I is broadcasted (to all the proxy vehicles). In the next step, the chosen proxy vehicles have to sign the contract using Algorithm 3 too. If a proxy vehicle violates the contract, it will be punished.

Algorithm 3 M.Sign()

Input: m_I, μ_{I_i}. We set $\mu_{I_0} = \mu_I$.
Output:
1: Compute the partial multi-signature $\sigma_i = H_3(m_I)^{\mu_{I_i}}$.
2: **return** σ_i.

Algorithm 4 M.Verify()

Input: $m_I, (\sigma_0, \ldots, \sigma_y), \mathbb{C} = (ct_{V_{I_0}}, \ldots, ct_{V_{I_y}})$.
Output:
1: Extract v_{I_i} from ct_{I_i} for $i \in \{0, \ldots, y\}$.
2: Compute $h = H_3(m_I)$.
3: Compute the multi-signature $\tilde{\sigma}_y = \prod_{i=0}^{y} \sigma_i$.
4: Check $\hat{e}(\tilde{\sigma}_y, g_2) \stackrel{?}{=} \hat{e}(h, \prod_{i=0}^{y} v_{I_i})$.
5: **return** $\sigma_{\mathcal{V}_p}$ if above equation holds or abort.

Step 4: When $\mathcal{V}_{I_l}, l \in \{1, \ldots, K+x\}$ receives con_I, it has to verify the validity of σ_0 using Algorithm 4 (i.e., the signature verification algorithm of the multi-signature scheme in [39]). Assume it is valid, $\mathcal{V}_{I_l}$ generates its partial multi-signature σ_l on m_I using Algorithm 3 and sends $(\sigma_l, ct_{V_{I_l}})$ to $\mathcal{V}_I$.

Step 5: Assume $\mathcal{V}_I$ receives y signatures from y vehicles in $\{\mathcal{V}_{I_l}\}_{l\in\{1,\ldots,K+x\}}$ before vp_I expires. Without loss of generality, we assume the vehicles are $\mathcal{V}_{I_1}, \ldots, \mathcal{V}_{I_y}, K \le y \le K+x$. $\mathcal{V}_I$ verifies received signatures by using Algorithm 4. If all the signatures are valid, $\mathcal{V}_I$ sends the fully signed contract $\tilde{con}_I = (m_I, \tilde{\sigma}_y, ind)$ to $\mathcal{S}_J$, where ind records the indexes of the certificates (in $\mathbb{C}$) corresponding to the proxy vehicles.

Step 6: When $\mathcal{S}_J$ receives $\tilde{con}_I$, it verifies the validity of $\tilde{\sigma}_y$ by using Algorithm 4. If the signature is valid, it computes $b_I = a_I^{\eta_J}$, $d_I = v_I^{\eta_J}$ and the session key $sk_I = H_1(ct_{V_I}, ct_{S_J}, a_I, b_I, d_I, vp_I)$, recovers $F_{Name} = D_{sk_I}(er)$ and $EC_I = D_{sk_I}(ec_I)$.

Protocol 2 (File Downloading): In this protocol, the proxy vehicles in $\{\mathcal{V}_{I_1}, \ldots, \mathcal{V}_{I_y}\}$ will help $\mathcal{V}_I$ to download F. Assume the file F is separated into L blocks $F_1, \ldots, F_L$. L is determined by the size of F. Assume a proxy vehicle $\mathcal{V}_{I_l}$ will download the file blocks $\tilde{F}_l = (F_{n_l}, \ldots, F_{n_l+j})$. $\mathcal{V}_{I_l}$ and S_J run the protocol as follows:

Step 1: When $\mathcal{V}_{I_l}$ designated by $\mathcal{V}_I$ passes through the nearby RSU, it sends a down request to $\mathcal{S}_J$ through the RSU as follows:

1) Choose $\theta_{I_l} \in \mathbb{Z}_q^*$, compute $a_{I_l} = g_2^{\theta_{I_l}}, b_{I_l} = s_J^{\theta_{I_l}}, d_I = s_J^{\mu_{I_l}}$, generate a session key $sk_{I_l} = H_1(ct_{V_{I_l}}, ct_{S_J}, a_{I_l}, b_{I_l}, d_{I_l}, tp_{I_l})$, where sk_{I_l} is used to generate message authentication codes and protect the communications between $\mathcal{V}_{I_l}$ and $\mathcal{S}_J$.
2) Send the download request $(a_{I_l}, tp_{I_l}, ct_{V_{I_l}})$ to $\mathcal{S}_J$.

Step 2: After receiving the download request, $\mathcal{S}_J$ does the following:

1) Check whether $ct_{V_{I_l}}$ is a valid certificate in $\mathbb{C}$. If it is valid, continue.
2) Compute $b_{I_l} = a_{I_l}^{\eta_J}, d_{I_l} = v_{I_l}^{\eta_J}$ and the session key $sk_{I_l} = H_1(ct_{V_{I_l}}, ct_{S_J}, a_{I_l}, b_{I_l}, d_{I_l}, tp_{I_l})$.
3) Choose $\theta_{J_l} \in \mathbb{Z}_q^*$, compute $a_{J_l} = g_2^{\theta_{J_l}}, b_{J_l} = v_I^{\theta_{J_l}}$ and a session key $sk_{IJ_l} = H_1(ct_{V_I}, ct_{S_J}, a_I, a_{J_l}, b_I, b_{J_l})$. sk_{IJ_l} can only be calculated by $\mathcal{S}_J$ and the client vehicle $\mathcal{V}_I$. It is used for $\mathcal{S}_J$ to encrypt file blocks such that only $\mathcal{V}_I$ may decrypt the resulting ciphertexts. When $\mathcal{V}_I$ receives a_{J_l}, it can compute $sk_{IJ_l} = H_1(ct_{V_I}, ct_{S_J}, a_I, a_{J_l}, b_I, b_{J_l})$, where $b_{J_l} = a_{J_l}^{u_I}$.

Step 3: Assume u is the index of a file block, $u \in \{n_l, \ldots, n_l + j\}$. $\mathcal{S}_J$ and $\mathcal{V}_{I_l}$ run an interactive protocol which is used for $\mathcal{V}_{I_l}$ to download the file blocks from $\mathcal{S}_J$. For $u \in \{n_l, \ldots, n_l + j\}$, $\mathcal{S}_J$ and $\mathcal{V}_{I_l}$ does the following two sub-steps:

Sub-step 1: $\mathcal{S}_J$ computes the encrypted file block $T_u = E_{sk_{IJ_l}}(F_u)$, generates a signature σ_{Ju} using Algorithm 5 (i.e., the signature generation algorithm of the screening of signatures in [35]), sends (u, T_u, σ_{Ju}) to $\mathcal{V}_{I_l}$.

Sub-step 2: $\mathcal{V}_{I_l}$ verifies the validity of σ_{Ju} by invoking Algorithm 6 (i.e., the signature verification algorithm of the screening of signatures in [35]). If σ_{Ju} is valid, $\mathcal{V}_{I_l}$ sends $MAC_1 = H_{sk_{I_l}}(1, u, T_u, \sigma_{Ju})$ to $\mathcal{S}_J$; else sends $MAC_0 = H_{sk_{I_l}}(0, u, T_u, \sigma_{Ju})$ to S_J. If MAC_1 is received by $\mathcal{S}_J$, it sets $u = u + 1$ and goto sub-step 1; else, (u, T_u, σ_{Ju}) has to be re-downloaded by $\mathcal{V}_{I_l}$.

Once $\mathcal{V}_{I_l}$ has received all the encrypted blocks $\tilde{T}_l = (T_{n_l}, \ldots, T_{n_l+j})$. $\mathcal{S}_J$ computes $C_{Jl} = E_{sk_{I_l}}(a_{J_l})$ and the signature $\bar{\sigma}_{Jl} = H_3(\tilde{T}_l, a_{J_l}, er)^{\eta_J}$, then sends $(C_{Jl}, \bar{\sigma}_{Jl})$ to $\mathcal{V}_{I_l}$. $\mathcal{V}_{I_l}$ decrypts $a_{J_l} = D_{sk_{I_l}}(C_{Jl})$ and has to verify the validity of the signature by checking $\hat{e}(H_3(\tilde{T}_l, a_{J_l}, er), s_J) \stackrel{?}{=} \hat{e}(\bar{\sigma}_{Jl}, g_2)$. If the signature is valid, it generates the aggregate signature $\Omega_{I_l} = \prod_{u=n_l}^{n_l+j} \sigma_{Ju}$.

Protocol 3 (Message Delivery): Assume the file to be forwarded is $T = (\tilde{T}'_1, \ldots, \tilde{T}'_K)$, where V_{I_l} holds $\tilde{T}'_l = (\tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl}, a_{J_l})$. V_{I_l} has to transmit $\tilde{T}'_l$ to the client

Algorithm 5 S.Sign()

Input: $\eta_J, u \in \{n_l, \ldots, n_l + j\}, T_u, er$.
Output:
1: Compute $\sigma_{Ju} = H_3(u, T_u, er)^{\eta_J}$.
2: **return** σ_{Ju}.

Algorithm 6 S.Verify()

Input: $(n_l, u), (T_{n_l}, \ldots, T_u), \Omega_{I_l} = \prod_{i=n_l}^{u} \sigma_{Ji}, er, ct_{S_J}$.
Output:
1: Compute $f_i = H_3(i, T_i, er)$ for $i \in \{n_l, \ldots, u\}$.
2: Check $\hat{e}(\prod_{i=n_l}^{u} f_i, s_J) \stackrel{?}{=} \hat{e}(\Omega_{I_l}, g_2)$.
3: **return** *true* if above equation holds or abort.

vehicle $\mathcal{V}_I$. Let $\mathcal{V}_{a_0} = \mathcal{V}_{I_l}$, $\mu_{a_0} = \mu_{I_l}, ct_{V_{a_0}} = ct_{V_{I_l}}$, $\mathcal{V}_{a_{S+1}} = \mathcal{V}_I, \varpi_l = H_2(\tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl})$. The protocol runs as follows:

Step 1: When $\mathcal{V}_{I_l}$ meets the next vehicle, if the vehicle is $\mathcal{V}_I$, $\mathcal{V}_{I_l}$ sends $(\tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl})$ together with an ordered multi-signature Υ_{a_0} on $(\tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl})$ generated using Algorithm 7 (i.e., the signature generation algorithm of the ordered multi-signature scheme in [40]) to $\mathcal{V}_I$ directly; else, assume the assisting vehicles will be $\mathcal{V}_{a_1}, \ldots, \mathcal{V}_{a_S}$, for $0 \leqslant x \leqslant S$, $\mathcal{V}_{a_x}$ does the following:

1) Generate an ordered multi-signature Υ_{a_x} on $(\tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl})$ using Algorithm 7. The ordered multi-signature is used to proof the order of these assisting vehicles. We note that if $x \neq 0$, it has to verify the validity of the ordered multi-signature generated by $\mathcal{V}_{a_{x-1}}$ using Algorithm 8 (i.e., the signature verification algorithm of the ordered multi-signature scheme in [40]).
2) Send $(\tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl}, \Upsilon_{a_x})$ to $\mathcal{V}_{a_{x+1}}$, set $x = x + 1$.

Step 2: Once $\mathcal{V}_{a_{S+1}}$ receives $(\tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl}, \Upsilon_{a_S})$ from $\mathcal{V}_{a_S}$, it verifies the validity and order of the file blocks by using Algorithm 6 and the validity of the ordered multi-signature generated using Algorithm 8. If they are valid, it generates an ordered multi-signature $\Upsilon_{a_{S+1}}$ by using Algorithm 7, sends $(ct_{V_{a_1}}, \ldots, ct_{V_{a_S}}, \Upsilon_{a_{S+1}})$ to $\mathcal{V}_{I_l}$ to inform that $\mathcal{V}_{a_{S+1}}$ has received $(\tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl})$.

Step 3: When $\mathcal{V}_{I_l}$ receives the above message from $\mathcal{V}_{a_{S+1}}$, it verifies the validity of $\Upsilon_{a_{S+1}}$ by invoking Algorithm 8, sends a_{J_l} to $\mathcal{V}_I$.

Step 4: We have to consider two cases. Case 1 ($\mathcal{V}_I$ receives a_{J_l} successfully): $\mathcal{V}_I$ verifies the validity of a_{J_l} by checking $\hat{e}(H_3(\tilde{T}_l, a_{J_l}, er), s_J) \stackrel{?}{=} \hat{e}(\bar{\sigma}_{Jl}, g_2)$, calculates the session key sk_{IJ_l} (using a_{J_l}) to decrypt the encrypted file blocks, submits $(ct_{V_{a_1}}, \ldots, ct_{V_{a_S}}, \Upsilon_{a_{S+1}}, 1)$ to $\mathcal{S}_J$ to inform $\mathcal{S}_J$ that it has successfully download all the file blocks. Case 2 (a_{J_l} is not received by $\mathcal{V}_I$): $\mathcal{V}_I$ submits $(ct_{V_{a_1}}, \ldots, ct_{V_{a_S}}, \Upsilon_{a_{S+1}}, 0)$ to $\mathcal{S}_J$. In both cases, if $\Upsilon_{a_{S+1}}$ is valid, $\mathcal{S}_J$ sends a_{J_l} to $\mathcal{V}_I$ as the confirmation message, and will reward the proxy and assistant vehicles according to the contract. If case 2 happens, $\mathcal{V}_I$ may decrypt the encrypted file blocks similar to Case 1.

Algorithm 7 OM.Sign()

Input: If $x = 0$, the input is $\mu_{a_x}, \varpi_l, \tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl}$; else it is $\mu_{a_x}, \varpi_l, \Upsilon_{a_{x-1}}, \tilde{T}_l, n_l, j, \Omega_{I_l}, \bar{\sigma}_{Jl}, (ct_{a_1}, \ldots, ct_{a_{x-1}})$.
Output:
1: Do the following:
- If $x = 0$, select $\beta \in \mathbb{Z}_q^*$, set $\Upsilon_{a_0} = (\Upsilon_{a_{01}}, \Upsilon_{a_{02}}) = (g_1^{\beta}, (p_1 g_1^{\varpi_l \mu_{a_x}})^{\beta})$.
- Else if $x > 0$ and OM.Verify$(\varpi_l, \Upsilon_{a_x}, (ct_{a_1}, \ldots, ct_{a_x-1})) = 0$, abort.
- Else, select $\beta \in \mathbb{Z}_q^*$, set $\Upsilon_{a_x} = (\Upsilon_{a_{y1}}^{\beta}, (\Upsilon_{a_{y2}} \Upsilon_{a_{y1}}^{\varpi_l \mu_{a_x}})^{\beta})$, where $y = x - 1$.

2: **return** Υ_{a_x}.

Algorithm 8 OM.Verify()

Input: $\varpi_l, \Upsilon_{a_x}, (ct_{a_0}, \ldots, ct_{a_x})$.
Output:
1: Parse Υ_{a_x} as $(\Upsilon_{a_{x1}}, \Upsilon_{a_{x2}})$.
2: Extract v_{a_l} from ct_{a_l} for $l \in \{1, \ldots, x\}$.
3: Check $\hat{e}(\Upsilon_{a_{x1}}, p_2(\prod v_{a_l})^{\varpi_l}) \stackrel{?}{=} \hat{e}(\Upsilon_{a_{x2}}, g_2)$.
4: **return** 1 if above equation holds or abort.

E. ADJUDICATION

If the protocols in the Cooperative Downloading stage are honestly performed, then the proxy and assisting vehicles will be rewarded according to the contract. Further, $\mathcal{S}_J$ will notify the RMA to reward those proxy and assisting vehicles by increasing their reputation values.

In our scheme, the client vehicles, proxy vehicles and assisting vehicles may behave maliciously. They may send fake messages to mislead other vehicles. In our scheme, all the messages sent by the client vehicles, proxy vehicles and assisting vehicles are signed by the senders. If a fake message is found, a receiver may send the message-signature pair and the corresponding anonymous certificate to the TA. The TA may recover the identity of the signature generator according to the anonymous certificate if the signature is valid. Then the malicious vehicle will be punished. For instance, the RMA may lower down the reputation value of the vehicle.

The client vehicles, proxy vehicles and assisting vehicles may be also uncooperative. If a client vehicle is uncooperative, this may directly lead to the failure of a cooperative downloading task. Further, the proxy vehicles and assisting vehicles have the evidences, i.e., the file blocks downloaded/forwarded by themselves and the corresponding signatures. They may submit the evidences to the RMA for adjudication. Then the uncooperative behaviors of the client vehicle will be punished. Therefore, a client vehicle is unlikely to violate the agreement in our scheme. For the proxy vehicles, since they have to sign the contract generated by a client vehicle, if a proxy vehicle behaves uncooperative, it will be punished according to the contract. For instance, the RMA may lower down the reputation value of the proxy vehicle. For the assisting vehicles, the unco-

operative behaviors may be alleviated by using multi-path forwarding method.

V. SECURITY AND PERFORMANCE ANALYSIS

In this section, we demonstrate that our scheme meets the security and performance requirements defined in Section II-B.

A. THE SECURITY

In our scheme, we mainly need to analyze the security of the cooperative downloading stage which consists of three protocols, i.e., agreement, file downloading and message delivery.

Obviously, our agreement protocol satisfies the authentication requirement, since every vehicular message is signed by using a (aggregate) signature scheme. As to the file downloading protocol, it has three steps. For Step 1 and 2, they are essentially authenticated key agreement protocols which may be used to authenticate the identities of the senders. For Step 3, the authentication requirement is guaranteed by signatures and MACs. In particular, the screening of signatures is used in this step. It not only guarantees the authentication of the encrypted file blocks but also the order of those blocks. Finally, for the message delivery protocol, it is easy to see that all the messages are signed by the legitimate entities. Specifically, the ordered multi-signatures are used to prove the order of the signers and the screening of signatures (in Step 2) are used to prove the order of the file blocks. Hence, the cooperative downloading stage holds the authentication requirement.

As to privacy preservation, this is guaranteed by using anonymous certificates. Obviously, an attacker cannot determine the true identity of a message sender. Further, for a vehicle, a fresh anonymous certificate will be used in each run of our scheme. Hence, an attacker cannot distinguish whether two messages in two independent executions of our scheme are from the same sender.

In terms of message confidentiality, the file name of a file is encrypted by a client vehicle using a session key. Only the corresponding file sever can recover the session key and decrypt the ciphertext to obtain the file name. Further, the file is separated into several file blocks. Each file block is encrypted using a session key negotiated by the client and the file sever. Without the private key (and the random input) of the client or the file sever, an attacker cannot even learn the content of a file block. Therefore, except the client and the file sever, any other entity cannot learn the content of the file as well as the file name.

B. THE PERFORMANCE

In this section, we show that our scheme satisfies the performance requirements defined in Section II-B.

The robustness of our scheme is ensured by reputation (especially our reputation based selection mechanism), credit, contract and the multi-path forwarding methods. For reliable proxy vehicle choosing, at first, the client vehicle and all the proxy vehicles have to sign the contract which restricts the behaviors of vehicles and prevent them from not executing the agreements in our scheme. Further, according to our reputation based selection mechanism, the vehicles with the highest expected downloading capacities close to a client vehicle will be selected by the client vehicle as the proxy vehicles. This guarantees that a file can be downloaded by the proxy vehicles with high success rate. Hence, reliable proxy vehicle choosing is guaranteed. As to reliable forwarding, the multi-path forwarding method may be employed to enable more reliable forwarding. This is proved by the simulations in Section VI-D.

For initiativeness, all the proxy vehicles and assisting vehicles will be rewarded by electronic cash (credit) if they honestly execute our scheme. Further, the RMA will reward their active behaviors by increasing their reputation values. Hence, the vehicles will actively and honestly participate in cooperative downloading tasks. In other words, our scheme holds initiativeness.

As to flexibility, the downloading capacity of a proxy vehicle is determined by its position and reputation value. Obviously, the downloading capacity of a proxy vehicle is not fixed. Hence, flexibility is realized.

VI. EVALUATION

In this section, we will evaluate our schemes from different aspects.

A. SECURITY AND PERFORMANCE

For security and performance, we compare our scheme with other three papers [21]–[23] which are also focused on secure cooperative downloading. An overall comparison is given in table 2. From the comparison, we can observe that our solution is better in security and performance. In addition to traditional security, we also meets the process authentication, cooperative vehicles selection mechanisms, and more flexible file blocks.

We will evaluate computation overhead and compare it with [23], which is also used aggregate signature technique to provide security protection in Cooperative Downloading. Since the aggregate signatures in the protocol are the main component, for the simplicity, we evaluate the total computation overhead of aggregate signature (including signing and verification) on a 3.6 GHz machine with 2 GB-memory, based on the MIRACL [41] and a BN curve with 128 bits security level was chosen. Figure 4 shows the comparison results. Then, we can find that our scheme has the lower computational overhead.

To further evaluate our scheme, we performed several simulations to show the efficiency of our proposal using MIRACL, VanetMobiSim and NS-3.

B. SIMULATION SETUP

For simulations, we used a real city topology which is a part of Washington state, USA. The map used is shown in Fig. 5. The New York Avenue, which has sufficient length and width, was selected to simulate highway. Since we simulate highway

TABLE 2. Comparison for security and performance.

Schemes	Process Authentication	Privacy	Message Confidentiality	Proxy Vehicles Selection	Flexible File Blocks
[21]	×	√	√	×	×
[22]	×	√	√	×	×
[23]	×	√	√	×	×
Ours	√	√	√	√	√

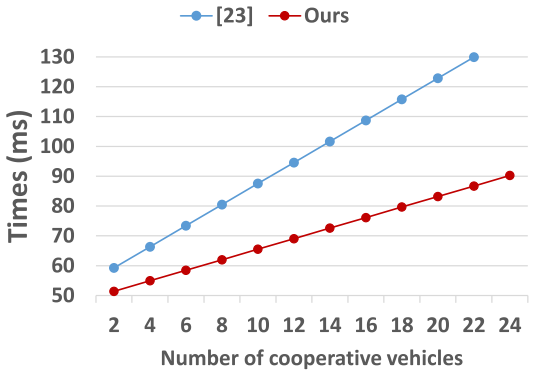


FIGURE 4. The comparison of computational overhead.

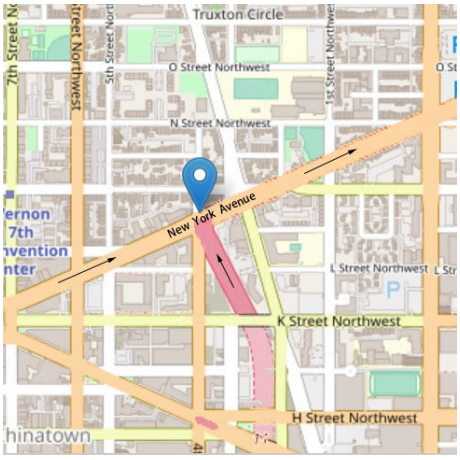


FIGURE 5. The map of the simulation scenario.

environment, the vehicles on New York Avenue are moving in the same direction. The simulations were run on a Linux machine using an Intel Core i7-4790 at a frequency of 3.6 GHz. A BN curve with 128 bits security level was chosen. AES-128 was chosen as the symmetric encryption scheme. More parameters are shown in Table 3.

The efficiency of our proposal is mainly dominated by the Cooperative Downloading stage which contains three protocols: Agreement, File Downloading and Message Delivery. For this stage, we simulated 50 times. For each simulation, the simulation time was 200 s. The target RSU is set on New York Avenue. In Agreement, the reputation based selection mechanism is adopted to select the appropriate proxy vehicles. We assume the reputation values of vehicles follow normal distribution. Next we show the efficiency of these protocols.

TABLE 3. More parameters.

Parameters	Setting
Average Data Transmission Rate	6 Mbps
Map Size	2 km × 2 km
Communication Range	300 m
Vehicle Speed	90-120 km/h
Routing Algorithm	AODV
Physical Layer Standard	IEEE 802.11a
Channel Bandwidth Bound	54 Mbps

C. EFFICIENCY OF AGREEMENT

The agreement protocol is mainly used for the client vehicle to choose the proxy vehicles. The efficiency of this protocol is related to the file size and the vehicle density. Fig. 6 shows the average execution time of the protocol. Obviously, the execution time increases with the file size F_{size}. This is because more proxy vehicles are required as the file size grows. According to our simulations, each vehicle may download 8-12 MB of data. When $F_{size} = 45$ MB, 4-6 vehicles will join a cooperative downloading task by using our *reputation based selection mechanism*. Similarly, when $F_{size} = 90/135/180$ MB, 8-11/12-17/15-22 vehicles will become the proxy vehicles. From the figure, one may also find that the execution time increases with the vehicle density in generally. This is due to the fact more vehicles will response the request of the client vehicle.

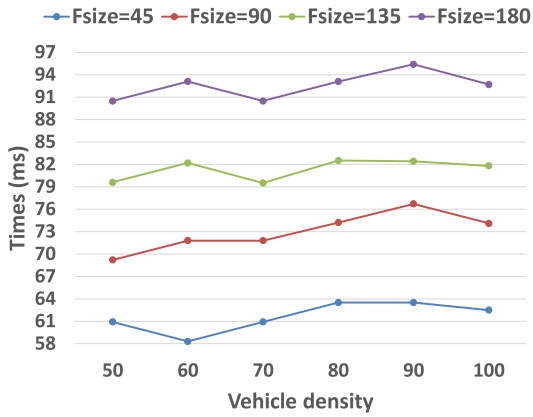


FIGURE 6. The average execution time of agreement protocol.

D. EFFICIENCY OF FILE DOWNLOADING AND MESSAGE DELIVERY

In this section, we first evaluate the average download delay by using our file downloading and message delivery protocols, then we show the average download success rate

using these two protocols. The former reflects the average download delay for a client vehicle to receive a file requested after the proxy vehicles are selected while the latter reflects the success rate for delivering a packet of a file from a file sever to a client vehicle using our protocols.

Fig. 7 and 8 show the average download delay and the average download success rate respectively. From the figures, it is easy to see that the average download delay is not high and each packet can be delivered to a client vehicle with high success rate for all the conditions. When the vehicle density grows, the average download delay decreases while the average download success rate increases in generally. This is because more high-quality candidate proxy vehicles can be selected when the vehicle density grows. We note that when the vehicle density is not high (lower than about 70), the average download delay slightly increases while the average download success rate decreases a little when the vehicle density grows. This is because the probability of choosing high-quality proxy vehicles is relatively low and other factors (e.g., the routing protocol) have more affect on the efficiency of the entire network. Further, it is easy to see that if the multi-path forwarding method is applied, the average download success rate is better than that the average download success rate using the single path one. However, due to network congestion, the average download delay using multi-path forwarding method is slightly larger than the one using single path forwarding method.

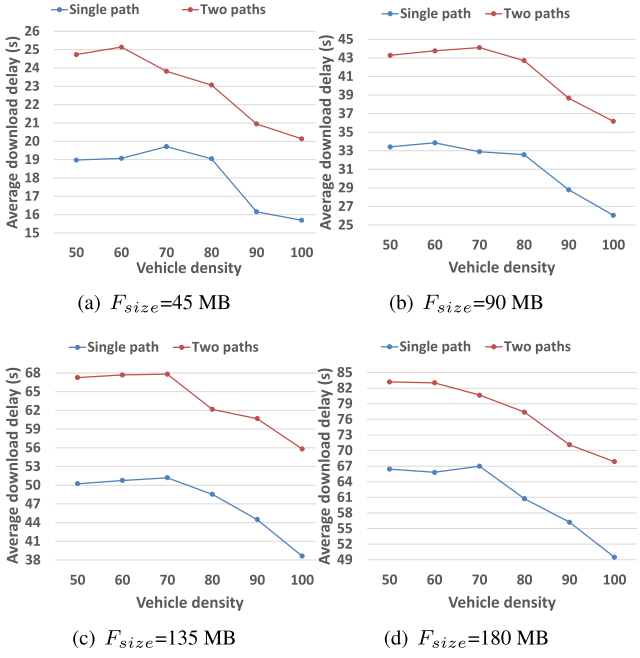


FIGURE 7. The average download delay.

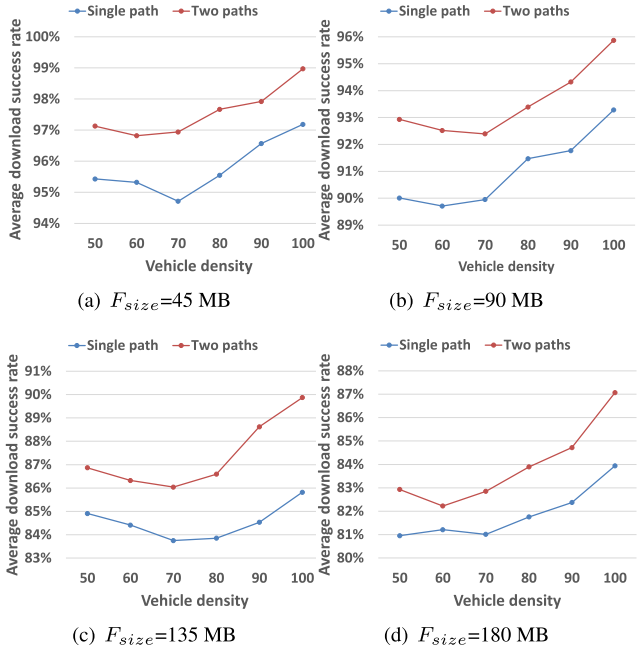


FIGURE 8. The average download success rate.

VII. CONCLUSION

We proposed a scheme for reliable cooperative downloading in VANETs, particularly those that are along highways. Our scheme realizes privacy preservation, message confidentiality, and traditional authentication, as well as our newly introduced process authentication property (i.e., authenticates the order of file blocks and the order of the assisting vehicles). Our scheme also satisfies robustness and flexibility, as demonstrated in the evaluations. For example, the simulation findings show that our scheme has low download delay and high download success rate.

Future research includes trialling an implementation of the prototype of our proposed scheme in collaboration with a county or city government. This will allow us to evaluate its real-world utility.

REFERENCES

[1] I. Ali, A. Hassan, and F. Li, "Authentication and privacy schemes for vehicular ad hoc networks (VANETs): A survey," *Veh. Commun.*, vol. 16, pp. 45–61, Apr. 2019, doi: 10.1016/j.vehcom.2019.02.002.

[2] L. Zhang, "OTIBAAGKA: A new security tool for cryptographic mix-zone establishment in vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 2998–3010, Dec. 2017.

[3] Q. Pei, B. Kang, L. Zhang, K.-K.-R. Choo, Y. Zhang, and Y. Sun, "Secure and privacy-preserving 3D vehicle positioning schemes for vehicular ad hoc network," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, pp. 1–12, Dec. 2018.

[4] J. Li, R. Xing, Z. Su, N. Zhang, Y. Hui, T. H. Luan, and H. Shan, "Trust based secure content delivery in vehicular networks: A bargaining game theoretical approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3267–3279, Mar. 2020.

[5] L. Zhang, M. Luo, J. Li, M. H. Au, K.-K.-R. Choo, T. Chen, and S. Tian, "Blockchain based secure data sharing system for Internet of vehicles: A position paper," *Veh. Commun.*, vol. 16, pp. 85–93, Apr. 2019.

[6] L. Zhang, X. Meng, K.-K.-R. Choo, Y. Zhang, and F. Dai, "Privacy-preserving cloud establishment and data dissemination scheme for vehicular cloud," *IEEE Trans. Depend. Sec. Comput.*, vol. 17, no. 3, pp. 634–647, May/Jun. 2020, doi: 10.1109/TDSC.2018.2797190.

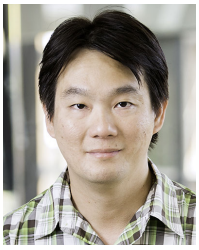
[7] S. Gao, G. Piao, J. Zhu, X. Ma, and J. Ma, "TrustAccess: A trustworthy secure ciphertext-policy and attribute hiding access control scheme based on blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5784–5798, Jun. 2020.

[8] J. Song, F. Yang, K.-K. R. Choo, Z. Zhuang, and L. Wang, "SIPF: A secure installment payment framework for drive-thru Internet," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 2, pp. 52-1–52-18, 2017,
[9] X. Zhuo, W. Gao, G. Cao, and S. Hua, "An incentive framework for cellular traffic offloading," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 541–555, Mar. 2014.
[10] A. Dua, N. Kumar, and S. Bawa, "A systematic review on routing protocols for vehicular ad hoc networks," *Veh. Commun.*, vol. 1, no. 1, pp. 33–52, Jan. 2014.
[11] T. Hao Luan, X. Ling, and X. Shen, "MAC in motion: Impact of mobility on the MAC of drive-thru Internet," *IEEE Trans. Mobile Comput.*, vol. 11, no. 2, pp. 305–319, Feb. 2012.
[12] W. Saad, Z. Han, A. Hjorungnes, D. Niyato, and E. Hossain, "Coalition formation games for distributed cooperation among roadside units in vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 48–60, Jan. 2011.
[13] H. Liang and W. Zhuang, "Cooperative data dissemination via roadside WLANs," *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 68–74, Apr. 2012.
[14] H. Zhou, B. Liu, T. H. Luan, F. Hou, L. Gui, Y. Li, Q. Yu, and X. Shen, "ChainCluster: Engineering a cooperative content distribution framework for highway vehicular communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2644–2657, Dec. 2014.
[15] W. Junjun, W. Shuqi, S. Yongjun, C. Yiqi, L. Wei, and W. Di, "An incentive mechanism for cooperative downloading method in VANET," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, Jul. 2013, pp. 125–130.
[16] M. Obaidat, M. Khodjaeva, J. Holst, and M. B. Zid, "Security and privacy challenges in vehicular ad hoc networks," in *Connected Vehicles in the Internet of Things*. Cham, Switzerland: Springer, 2020, pp. 223–251.
[17] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Y. Sanadidi, "Cooperative downloading in vehicular ad hoc wireless networks," in *Proc. 2nd Annu. Conf. Wireless On-Demand Netw. Syst. Services*, 2005, pp. 32–41.
[18] S. Yang, C. K. Yeo, and B. S. Lee, "MaxCD: Max-rate based cooperative downloading for drive-thru networks," in *Proc. 21st Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2012, pp. 1–7.
[19] Y. Sun, L. Xu, and Y. Tang, "Cooperative downloading in vehicular networks: A graph-based approach," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–5.
[20] K. Ota, M. Dong, S. Chang, and H. Zhu, "MMCD: Cooperative downloading for highway VANETs," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 1, pp. 34–43, Nov. 2014.
[21] W. Zhang, S. Jiang, X. Zhu, and Y. Wang, "Privacy-preserving cooperative downloading for value-added services in VANETs," in *Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst.*, Sep. 2013, pp. 358–362.
[22] Y. Hao, J. Tang, and Y. Cheng, "Secure cooperative data downloading in vehicular ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 523–537, Sep. 2013.
[23] C. Lai, K. Zhang, N. Cheng, H. Li, and X. Shen, "SIRC: A secure incentive scheme for reliable cooperative downloading in highway VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1559–1574, Jun. 2017.
[24] E. Evdokimova, A. Vinel, N. Lyamin, and D. Fiems, "Internet provisioning in VANETs: Performance modeling of drive-thru scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2801–2815, Jul. 2020.
[25] S. Xu, M. Li, Y. Chen, L. Shu, and X. Gu, "A cooperation scheme based on reputation for opportunistic networks," in *Proc. Int. Conf. Comput., Manage. Telecommun. (ComManTel)*, Jan. 2013, pp. 289–294.
[26] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.
[27] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.
[28] S.-B. Lee, G. Pan, J.-S. Park, M. Gerla, and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," in *Proc. 8th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2007, pp. 150–159.
[29] R. Lu, X. Lin, H. Zhu, C. Zhang, P.-H. Ho, and X. Shen, "A novel fair incentive protocol for mobile ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2008, pp. 3237–3242.
[30] Y. S. Sun, M.-L. Lu, Y.-C. Pan, and M. Chang Chen, "Optimal incentive-compatible pricing for dynamic bandwidth trading and allocation in efficient spectrum management," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–6.
[31] Z. Hasan and V. K. Bhargava, "Relay selection for OFDM wireless systems under asymmetric information: A contract-theory based approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 3824–3837, Aug. 2013.
[32] Y. Zhang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Contract-based incentive mechanisms for device-to-device communications in cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2144–2155, Oct. 2015.
[33] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu, "Distributed aggregate privacy-preserving authentication in VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 516–526, Mar. 2017.
[34] L. Zhang, C. Hu, Q. Wu, J. Domingo-Ferrer, and B. Qin, "Privacy-preserving vehicular communication authentication with hierarchical aggregation and fast response," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2562–2574, Aug. 2016.
[35] J. Camenisch, S. Hohenberger, and M. Ø. Pedersen, "Batch verification of short signatures," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2007, pp. 246–263.
[36] G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille, "Simple Schnorr multi-signatures with applications to bitcoin," *Des., Codes Cryptogr.*, vol. 87, no. 9, pp. 2139–2164, Sep. 2019.
[37] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Appl. Math.*, vol. 156, no. 16, pp. 3113–3121, Sep. 2008.
[38] S. Chatterjee and A. Menezes, "On cryptographic protocols employing asymmetric pairings—The role of Ψ revisited," *Discrete Appl. Math.*, vol. 159, no. 13, pp. 1311–1322, 2011.
[39] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, Sep. 2004.
[40] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *Proc. Cryptographers' Track RSA Conf.*, 2016, pp. 111–126.
[41] L. Zhang and J. Li, "Enabling robust and privacy-preserving resource allocation in fog computing," *IEEE Access*, vol. 6, pp. 50384–50393, 2018, doi: 10.1109/ACCESS.2018.2868920.
[42] X. Meng, L. Zhang, and B. Kang, "Fast secure and anonymous key agreement against bad randomness for cloud computing," *IEEE Trans. Cloud Comput.*, 2020, doi: 10.1109/TCC.2020.3008795.

YAN ZHANG received the B.S. degree (Hons.) from the School of Software Engineering, Jiangxi Normal University, China. She is currently pursuing the master's degree with the School of Software Engineering, East China Normal University, China. Her research interests include VANET security and data privacy.

LEI ZHANG (Member, IEEE) received the Ph.D. degree in computer engineering from Universitat Rovira i Virgili, Tarragona, Spain. Since then, he has been with Universitat Rovira i Virgili, as a Postdoctoral Researcher. He is currently a Full Professor with the School of Software Engineering, East China Normal University, Shanghai, China. He has been a holder/coholder of more than ten China/Spain-funded (key) projects. His fields of activity are information security, VANET security, cloud security, data privacy, and network security. He has authored over 80 publications. He has served in the program committee of more than 60 international conferences in information security and privacy. He is an Editor of several international journals.

DINGKAI NI is currently pursuing the degree in network security and cryptography with the School of Software Engineering, East China Normal University, China.



KIM-KWANG RAYMOND CHOO (Senior Member, IEEE) received the Ph.D. degree in information security from the Queensland University of Technology, Australia, in 2006. He currently holds the Cloud Technology Endowed Professorship with The University of Texas at San Antonio (UTSA). In 2015, he and his team won the Digital Forensics Research Challenge organized by the Germany's University of Erlangen-Nuremberg. He was a recipient of the 2008 Australia Day Achievement Medallion, the British Computer Society's Wilkes Award, in 2008, the Fulbright Scholarship, in 2009, the 2014 Highly Commended Award from the Australia New Zealand Policing Advisory Agency, the ESORICS 2015 Best Research Paper Award, the IEEE TrustCom 2018 Best Paper Award, the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, the Outstanding Associate Editor of 2018 for IEEE Access, the British Computer Society's 2019 Wilkes Award Runner-up, the 2019 EURASIP Journal on Wireless Communications and Networking (JWCN) Best Paper Award, the Korea Information Processing Society's Journal of Information Processing Systems (JIPS) Survey Paper Award (Gold) 2019, the IEEE Blockchain 2019 Outstanding Paper Award, the International Conference on Information Security and Cryptology (Inscrypt 2019) Best Student Paper Award, and the 2019 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence in Scalable Computing (Middle Career Researcher). He is an IEEE Computer Society Distinguished Visitor from 2021 to 2023 (inclusive), and included in Web of Science's Highly Cited Researcher in the field of Cross-Field – 2020.

BURONG KANG received the B.S. degree in computer science from Northwest Normal University, China. She is currently pursuing the Ph.D. degree with the School of Software Engineering, East China Normal University, China. Her research interests include information security, public key cryptography, and network security.