applied sciences

MDPI

Article

Practical V2I Secure Communication Schemes for Heterogeneous VANETs

Fuxiao Zhou [1], Yanping Li [1,*] and Yong Ding [2,3]

[1] School of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710119, China
[2] School of Computer Science and Information Security, Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
[3] Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518000, China
* Correspondence: lyp@snnu.edu.cn; Tel.: +86-02985310232

Received: 30 June 2019; Accepted: 26 July 2019; Published: 1 August 2019

Abstract: Since the roadside infrastructure and vehicles come from different manufacturers, vehicular ad hoc networks (VANETs) now are extremely heterogeneous. It is difficult to communicate securely for heterogeneous facilities in VANETs because secure communication needs to concurrently realize confidentiality, authentication, integrity, and non-repudiation. To meet the above security attributes in one logical step, four bi-directional signcryption schemes are proposed for specific heterogeneous vehicle to infrastructure (V2I) communication in this paper. The first scheme supports batch verification, which allows multiple vehicles registered in a public key infrastructure (PKI) system to transmit messages to a receiver in an identity-based cryptosystem (IBC), both which are the mainstream public key cryptosystems. The second scheme supports a sender in a PKI to securely broadcast a message to multiple vehicles in an IBC. The communication direction of the latter two schemes is opposite to the former two schemes (i.e., from IBC to PKI). All these schemes can be proved to satisfy confidentiality and unforgeability based on the assumptions of decisional and computational Diffie-Hellman problems in the random oracle model. Furthermore, numerical analyses and simulation results demonstrate the computation costs, communication costs, storage, and the aggregate ciphertext length of our schemes are better than the existing ones.

Keywords: vehicular ad hoc networks (VANETs); smart transportation; identity-based cryptosystem (IBC); public key infrastructure (PKI); signcryption

1. Introduction

Smart transportation is one of the most important aspects of smart city, and vehicular ad hoc networks (VANETs) have recently been regarded as a promising smart transportation technique that can provide road safety, traffic management, and so on. VANETs consist of smart vehicles with on-board units (OBUs) and roadside units (RSUs). In VANETs, the vehicles can be regarded as mobile nodes and the communication between these nodes can be conducted by dedicated short-range communications (DSRC) technology [1]. There are two main communication ways: Vehicle to infrastructure (V2I) and vehicle to vehicle (V2V) [2]. V2V communication can exchange information among neighboring vehicles. V2I communication allows vehicles send messages to RSUs in driving. Vehicles or drivers can obtain the traffic information of a region or even a wider scope. After information integration, V2I provides comprehensive driving guidance and early warning services for each driver, which is more advantageous than V2V communication from this point of view. For example, if a traffic accident happened, the accident vehicles will firstly transmit the accident information to a nearby RSU; next, the RSU broadcasts the information to other RSUs and more vehicles in its management region. All kinds of messages can spread faster to prevent traffic jams because RSUs have higher power and stronger

information broadcasting ability. Hence, V2I secure communication is a promising technology and attracts more and more attention.

However, a critical problem is the security of V2I communication. Generally speaking, confidentiality, authentication, integrity, and non-repudiation are the key security requirements for secure V2I communication. Confidentiality can keep messages secret except for those who are authorized. Authentication and integrity can ensure the messages are not tampered by any unauthorized user. Non-repudiation can prevent the denial of prior behaviors. Generally, encryption can be used to realize confidentiality and digital signature can be used to realize authentication, integrity, and non-repudiation. Due to the openness of wireless communication in V2I, an adversary can easily control the communication links and modify session messages, which may lead to serious consequences. Hence, the above security requirements are the main concerns for V2I secure communication and are worth studying urgently.

In the paper, considering the aforementioned security attributes of V2I communication, we construct four practical bi-directional signcryption schemes, which offer heterogeneous V2I secure and efficient communication, in which the vehicles and the RSUs are registered in public key infrastructure (PKI) or identity-based cryptosystem (IBC), respectively. For this purpose, we mainly combine the signcryption and aggregation techniques to construct our schemes and permit the communicating parties sharing the same system parameters to achieve high efficiency according to current practices and traditions. To be specific, two many-to-one signcryption schemes can realize n vehicles in PKI/IBC system transmit messages to an RSU in the IBC/PKI system, which fit into the scenario where an RSU might receive multiple messages from different vehicles at almost the same time. In turn, two one-to-many signcryption schemes can support an RSU in PKI/IBC system transmitting one message m to n vehicles in IBC/PKI system, which applies to the scenario where an RSU broadcasts one message to multiple vehicles. All schemes have been proved to be indistinguishable against adaptive chosen ciphertext attacks (IND-CCA2) and existential unforgeability against adaptive chosen messages attacks (EUF-CMA) based on two difficulty assumptions of decisional Diffie–Hellman problem (DDHP) and computational Diffie–Hellman problem (CDHP), respectively. The main contributions of this work are summarized as follows:

(1) Two many-to-one heterogeneous signcryption schemes are proposed, which allow a large number of vehicles registered in the PKI/IBC freely communicating with a RSU registered in IBC/PKI. Two other one-to-many heterogeneous signcryption schemes are proposed, which allow a sender RSU in PKI/IBC freely broadcasts a message to many vehicles in IBC/PKI, which are especially designed for a RSU broadcasting message to vehicles in V2I communication scenario of VANETs.
(2) Inspired by the idea of signcryption and aggregation, our proposed many-to-one heterogeneous aggregate signcryption schemes support batch verification, which not only can save a lot of time compared with the traditional sequential verifications, but also can save the computational, communicational, and storage cost. In addition, all the proposed schemes can realize confidentiality, authentication, integrity, and non-repudiation at the same time.
(3) Numerical analyses and simulation results show the computation costs, communication costs, storage, and the aggregate ciphertext length of our proposed schemes are preferable to the existing ones.

The organization of our article is shown below: A survey of related work is described in Section 2. Some preliminaries are provided in Section 3. Four practical, heterogeneous signcryption proposals are presented in Section 4. The security proof and performance analysis are presented in Sections 5 and 6, respectively. In the end, we summarize the conclusions in Section 7.

2. Related Work

Considering the security requirements of V2I secure communication in VANETs, it is natural to utilize encryption to realize confidentiality and digital signature to realize authentication, integrity.

and non-repudiation. The traditional approach, named the signature-then-encryption method, is first to sign a message and then to encrypt it, or first to encrypt a message and then to sign it, named the encryption-then-signature method. Zheng first presented a new cryptographic primitive termed as signcryption [3], which simultaneously realizes the functions of encryption and digital signature. In addition, its cost is much smaller than the encryption-then-signature or signature-then-encryption techniques. An et al. proposed the general constructions of signcryption [4] and Baek et al. studied the formal proofs for the security of signcryption [5]. In addition, Baek et al. proved the security of the scheme in [3]. The performance advantage of signcryption makes it widely studied and used. Up to now, signcryption and its extension schemes have been put forward in several cryptosystem, such as the PKI-based cryptosystem [6,7], identity-based cryptosystem (IBC) [8–12], and certificateless cryptosystem [13–15]. Aggregate signcryption is one of the latest extensions of signcryption, which can combine multiple signcryption ciphertexts and verify them in batch. Compared with the traditional sequential verifications, it not only can save a lot of time, but also can save the computational, communicational, and storage cost. Hence, aggregate signcryption is very suitable for many-to-one mode of distributed communication, such as VANETs and routing protocol.

Today, vehicles and RSUs are extremely heterogeneous and may be registered with different public key cryptosystems. To ensure the secure V2I communication between these extreme heterogeneous devices, cryptographic schemes should be constructed to provide authentication, confidentiality, integrity, and non-repudiation, so signcryption naturally becomes the first choice. So far, a few signcryption schemes for heterogeneous environments have been proposed [16–26]. Many of these schemes consider two mainstream public key cryptosystems PKI and IBC. In the PKI system, a certificate authority (CA) issues the digital certificates for users, which bundle the public keys and users' real identities. However, certificates management has become a burden due to the storage, revocation, and distribution of certificates. However, in IBC, the email addresses, telephone numbers, or social security numbers of users are used to produce the public keys. The users' private keys are provided by a trusted third party named private key generator (PKG). IBC solves the problem of certificates management and becomes a more promising public key cryptosystem.

Sun and Li put forward two heterogeneous signcryption schemes. The first one supports a sender in a PKI system to transmit one message to a receiver in an IBC and the second is opposite (i.e., from IBC to PKI) [21]. In addition, the paper gives a discussion on the multi-receiver constructions from PKI to IBC. Unfortunately, their schemes do not support non-repudiation and cannot resist insider attacks. Later, a heterogeneous signcryption scheme against insider attacks is proposed in [22]. However, it simply permits a user in an IBC to transmit one message to the recipient in a PKI system. Schemes presented in [23,24] also do not allow users in a PKI system to send messages to recipient in an IBC. Recently, two new signcryption schemes are presented to support V2I mutual communication for heterogeneous PKI and IBC cryptosystem [25]. However, both schemes cannot protect the privacy of the senders. The scheme put forward in [26] gives a provable aggregate signcryption for heterogeneous PKI and IBC systems to improve the efficiency of computation and transmission. Unfortunately, it simply permits a user in a PKI system to transmit one message to the recipient in an IBC. Actually, it is not easy to design practical mutual many-to-one or one-to-many signcryption schemes for heterogeneous V2I communication because it needs to meet the security requirements and maintain high efficiency.

Similar to the above schemes, we mainly consider how to realize the secure and efficient V2I communication of heterogeneous devices registered in PKI and IBC. Different from the aforementioned schemes, we consider the most common scenario that a static RSU often receives multiple messages from different mobile vehicles almost at the same time; it needs to verify the integrity and authenticity of these messages in a batch way for the potential traceability, which can save a lot of time compared with the traditional sequential verifications. That is the reason that we design many-to-one aggregate signcryption schemes (as in Figure 1). In turn, two one-to-many signcryption schemes are designed to meet an application scenario of an RSU that broadcasts a message to a great number of vehicles. Therefore, our schemes are quite suitable for the heterogeneous secure V2I communication.

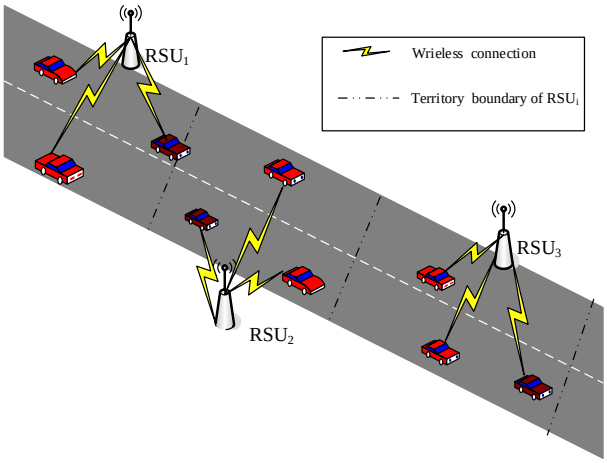

Figure 1. Communication model of our schemes.

3. Preliminaries

3.1. Mathematical Background

Let G_1/G_2 be the additive/multiplicative group of prime order q and P is a generator of G_1; $e: G_1 \times G_1 \rightarrow G_2$ is referred as a bilinear map if it meets the attributes as follows:

(1) Bilinearity: $\forall P, Q \in G_1$ and $a, b \in Z_q^*$, $e(aP, bQ) = e(P, Q)^{ab}$.

(2) Non-degeneracy: There exist $P, Q \in G_1$, s.t. $e(P, Q) \in 1_{G_2}$.

(3) Computability: There is an algorithm to calculate $e(P, Q) \in G_2$, $\forall P, Q \in G_1$.

Decisional Diffie–Hellman problem (DDHP): For a tuple $(P, aP, bP, cP) \in G_1$, where P as the generator of G_1 having order q and $a, b, c \in Z_q^*$, it is difficult to decide whether $ab = c \bmod q$ is held.

Computational Diffie–Hellman problem (CDHP): For a tuple $(P, aP, bP) \in G_1$, where P as the generator of G_1 having order q and $a, b \in Z_q^*$, it is hard to calculate abP.

3.2. Formal Definitions

(a) A many-to-one heterogeneous signcryption scheme contains six algorithms, as follows:

Setup: After taking a security parameter l, PKG chooses a master secret key msk and outputs the public system parameters **params**.

PKI-KG: A user in PKI selects a secret key sk and computes the corresponding public key pk. The pk has a certificate issued by the CA.

IBC-KG: A user in IBC sends his identity ID to the PKG. The PKG calculates the corresponding secret key sk and sends it to the user via a secure channel. Under these circumstances, the user's identity ID naturally served as his public key pk.

Signcrypt: Input **params**, a message m, a sender's secret key sk_i, a recipient's public key pk_r emits a signcryption ciphertext σ_i.

Aggregate-Verify: The algorithm firstly aggregate n ciphertexts $\{\sigma_i = (R_i, c_i, S_i)\}_{i=1}^n$ to a final ciphertext σ. Then, it verifies the validity of aggregate signcryption σ and outputs *true* or *false*.

Unsigncrypt: Input σ, the sender's public key pk_i, the recipient's secret key sk_r, outputs $\{m_i\}_{i=1}^n$ or $\perp$ that means decryption failure.

(b) A one-to-many heterogeneous signcryption scheme contains five algorithms, as follows:

The algorithms of **Setup, PKI-KG,** and **IBC-KG** are the same as those in (a).

Signcrypt: Input **params**, a message m, the sender's secret key sk, and multiple receivers' public keys $\left\{pk_{r_i}\right\}_{i=1}^n$, then the algorithm computes a signcryption ciphertext σ and sends it to receivers.

Unsigncrypt: Each receiver takes σ and the corresponding secret keys sk_{r_i} as inputs, then computes the broadcasting message m after a series of verifications.

Here, we omit the security model of heterogeneous signcryption scheme because of the limited space.

4. Four Heterogeneous Signcryption Schemes for V2I Communication Scenarios

In this section, four heterogeneous signcryption schemes are presented to support secure heterogeneous V2I communication. For brevity, we make PKI→IBC to indicate a sender in a PKI system transmits a message to a recipient in an IBC. In turn, IBC→PKI indicates a sender registered in an IBC transmits a message to a receiver in a PKI system. Many-to-one or one-to-many means the sender is multiple or single, and the receiver is single or multiple, which correspond to the two most common scenarios extracted from real V2I communication. The first scheme is called MOHSC-I (many-to-one heterogeneous signcryption), which is suitable for an RSU in an IBC to receive n ciphertexts of $\{m_i\}_{i=1}^n$ from n vehicles who are in a PKI system. In turn, the second construction is named OMHSC-I (one-to-many heterogeneous signcryption), which is suitable for the condition that an RSU in a PKI system to broadcast the cipher of m to n vehicles who are in IBC. The direction of the latter two schemes is opposite to the former schemes i.e., from IBC to PKI. The third scheme is named MOHSC-II (many-to-one heterogeneous signcryption) and the fourth construction is called OMHSC-II (one-to-many heterogeneous signcryption).

4.1. PKI→IBC Many-to-One Signcryption (MOHSC-I)

Setup: Input a security parameter l, PKG selects the additive/multiplicative group G_1/G_2 of prime order q (P be a generator of G_1), a bilinear map $e : G_1 \times G_1 \to G_2$, and three cryptographic hash functions $H_1 : \{0,1\}^* \to G_1$, $H_2 : G_2 \to \{0,1\}^n$, $H_3 : G_1^3 \times \{0,1\}^n \to Z_q^*$. Then, it selects $s \in Z_q^*$ randomly as the master secret key and computes the master public key $P_{pub} = sP$. Finally, PKG will publish **params** = $\{q, G_1, G_2, e, P, P_{pub}, H_1, H_2, H_3\}$ while keeping $\boldsymbol{s}$ secret.

PKI-KG: A vehicle V in the PKI selects $x_V \in Z_q^*$ randomly as the secret key sk_V and calculates $pk_V = x_V P$ as the public key. Let V_i's public/secret key is $pk_i = x_i P / sk_i = x_i$ below.

IBC-KG: An RSU in an IBC first sends its identity ID_r to the PKG, then PKG calculates the secret key $sk_{ID_r} = sH_1(ID_r) = sQ_{ID_r}$, and transmits sk_{ID_r} to RSU via a secure channel. Let the identity of RSU be ID_r and the public/secret key $pk_r = H_1(ID_r) = Q_r / sk_r = sQ_r$ below.

Signcrypt: Taking **params**, a random message m_i, the sender V_i's secret key sk_i, the receiver RSU's public key pk_r as inputs, the algorithm performs the following steps:

1. Randomly choose $r_i \in Z_q^*$ and calculate $R_i = r_i pk_i$;
2. Calculate $k_i = e(r_i sk_i P_{pub}, pk_r)$, $c_i = m_i \oplus H_2(k_i)$;
3. Calculate $h_i = H_3(pk_i, pk_r, R_i, c_i)$, $S_i = sk_i h_i P_{pub}$;
4. Output the signcryption ciphertext $\sigma_i = (R_i, c_i, S_i)$ to a nearby RSU.

Aggregate-Verify: A receiver RSU firstly act as an aggregate signcryption generator to save the verification costs. After receiving multiple ciphertexts $\sigma_i = (R_i, c_i, S_i)$ $(i = 1, 2, \ldots, n)$, the RSU computes $S = \sum_{i=1}^n S_i$, and get the final aggregate signcryption ciphertext $\sigma = (R_1, \ldots, R_n, c_1, \ldots, c_n, S)$. Then, the RSU performs the following procedures to verify the ciphertext σ by using n sender vehicles' public keys $\{pk_i\}_{i=1}^n$, the aggregate signcryption ciphertext σ and the receiver RSU's public key pk_r.

1. Compute $h_i = H_3(pk_i, pk_r, R_i, c_i)$,$i = 1, 2, \ldots, n$;
2. Verify

$$e(S, P) = e(\sum_{i=1}^{n} h_i pk_i, P_{pub}); \quad (1)$$

3. If (1) is held, emits true, which means σ is valid. Otherwise, emits false and abort.

Unsigncrypt: If **Aggregate-Verify** algorithm emits *true*, the receiver RSU's performs the following steps based on the aggregate signcryption σ and its secret key sk_r.

1. Calculate
$$k'_i = e(R_i, sk_r); \tag{2}$$
2. Calculate $m_i = c_i \oplus H_2(k'_i)$, and get the message m_i.

4.2. PKI→IBC One-to-Many Signcryption (OMHSC-I)

The **Setup** is the same as the above MOHSC-I algorithm, except the **PKI-KG** and **IBC-KG** algorithms have slight changes, i.e., the sender RSU's public/secret key $pk = xP/sk = x$ in PKI and n receivers V_i's key pairs $\{pk_{ri} = H_1(ID_{ri}) = Q_{ri}, sk_{ri} = sQ_{ri}\}_{i=1}^{n}$ in IBC.

Signcrypt: Taking **params**, a random message m, the RSU's secret key sk, multiple receivers' public keys $\{pk_{ri}\}_{i=1}^{n}$ as input, the algorithm performs the following steps (repeat 2, 3 steps n times):

1. Choose a random $r \in Z_q^*$ and compute $R = r \cdot pk$;
2. Compute $k_i = e(r \cdot sk \cdot P_{pub}, pk_{ri})$, $c_i = m \oplus H_2(k_i)$;
3. Calculate $h_i = H_3(pk, pk_{ri}, R, c_i)$, $S_i = sk \cdot h_i P_{pub}$.
4. Broadcast $\sigma = (R, c_1, \ldots, c_n, S_1, \ldots, S_n, \Omega)$ to $\{V_i\}_{i=1}^{n}$, where Ω is a label which includes message about how c_i and S_i are associated with the receivers.

Unsigncrypt: Any receiver V_i takes (R, c_i, S_i) from σ according to the label Ω. Input V_i's secret keys sk_{ri}, the sender's pk and **params**, then the algorithm executes the following procedures:

1. Compute $h_i = H_3(pk, pk_{ri}, R, c_i)$;
2. Verify
$$e(S_i, P) = e(h_i \cdot pk, P_{pub}); \tag{3}$$
If (3) is established, emits *true* which means σ is valid. Or else, emits *false* and abort.
3. Calculate
$$k'_i = e(R, sk_{ri}); \tag{4}$$
4. Compute $m = c_i \oplus H_2(k'_i)$, and get the message m.

4.3. IBC→PKI Many-to-One Signcryption (MOHSC-II)

The **Setup** is the same as the above MOHSC-I algorithm. **PKI-KG** and **IBC-KG** algorithms have slight changes, i.e., the sender V_i's key pair ($pk_i = H_1(ID_i) = Q_i$,$sk_i = sQ_i$) in an IBC, the recipient RSU's public/secret key ($pk_r = x_rP/sk_r = x_r$) in a PKI system.

Signcrypt: Taking **params**, a random message m_i, V_i's secret key sk_i, and RSU's public key pk_r as inputs, the algorithm executes the following procedures:

1. Select a random $r_i \in Z_q^*$ and compute $R_i = r_i P_{pub}$;
2. Compute $k_i = e(pk_r, r_i sk_i)$, $c_i = m_i \oplus H_2(k_i)$;
3. Calculate $h_i = H_3(pk_i, pk_r, R_i, c_i)$, $S_i = sk_i h_i$;
4. Transmit the signcryption ciphertext $\sigma_i = (R_i, c_i, S_i)$ to a nearby RSU.

Aggregate-Verify: Upon receiving the ciphertexts $\{\sigma_i\}_{i=1}^{n}$, the RSU computes $S = \sum_{i=1}^{n} S_i$ to get the final aggregate signcryption ciphertext $\sigma = (R_1, \ldots, R_n, c_1, \ldots, c_n, S)$. Then, the RSU executes the following procedures to verify σ by inputting the aggregate signcryption σ, n vehicles' public keys $\{pk_i\}_{i=1}^{n}$ and the receiver RSU's public key pk_r:

1. Compute $h_i = H_3(pk_i, pk_r, R_i, c_i), 1 \le i \le n$;
2. Verify

$$e(S, P) = e(\sum_{i=1}^{n} h_i pk_i, P_{pub}); \tag{5}$$

3. If (5) is held, emits true, which means σ is valid. Otherwise, emits false and abort.

Unsigncrypt: If the **Aggregate-Verify** algorithm outputs *true*, the receiver RSU takes the aggregate signcryption ciphertext σ, its secret key sk_r and n vehicles' public keys $\{pk_i\}_{i=1}^{n}$ as inputs, then performs as follows:

1. Compute

$$k_i' = e(pk_i, sk_r R_i); \tag{6}$$

2. Compute $m_i = c_i \oplus H_2(k_i')$, and get the message m_i.

4.4. IBC→PKI One-to-Many Signcryption (OMHSC-II)

The **Setup** is the same as the above MOHSC-I algorithm. **PKI-KG** and **IBC-KG** algorithms have slight changes, i.e., the sender RSU's key pair is ($pk = H_1(ID) = Q$,$sk = sQ$) in IBC and the receivers V_i's $(i = 1, 2, \ldots, n)$ key pairs ($pk_{ri} = x_{ri}P, sk_{ri} = x_{ri}$) in PKI.

Signcrypt: Taking **params**, a random message m, RSU's secret key sk, and multiple receivers' public keys $\{pk_{ri}\}_{i=1}^{n}$ as input, the algorithm performs as follows (repeat 2, 3 steps n times):

1. Randomly choose $r \in Z_q^*$ and compute $R = r \cdot P_{pub}$;
2. Calculate $k_i = e(pk_{ri}, r \cdot sk)$, $c_i = m \oplus H_2(k_i)$;
3. Calculate $h_i = H_3(pk, pk_{ri}, R, c_i)$, $S_i = sk \cdot h_i$;
4. Broadcast $\sigma = (R, c_1, \ldots, c_n, S_1, \ldots, S_n, \Omega)$ to multiple receivers $\{V_i\}_{i=1}^{n}$, where Ω is a label, which includes a message about how c_i and S_i are associated with the receivers.

Unsigncrypt: Any receiver V_i takes (R, c_i, S_i) from σ. Take V_i's secret keys sk_{ri}, the sender's pk, and **params** as inputs, then this algorithm performs the following steps:

1. Compute $h_i = H_3(pk, pk_{ri}, R, c_i), 1 \le i \le n$;
2. Verify

$$e(S_i, P) = e(h_i \cdot pk, P_{pub}); \tag{7}$$

If (7) is held, emits *true,* which means σ is valid. Or else, emits *false* and abort.

3. Calculate

$$k_i' = e(pk, sk_{ri} \cdot R); \tag{8}$$

4. Calculate $m = c_i \oplus H_2(k_i')$, and get the message m.

5. Security Proof

The correctness and security of our proposed schemes will be discussed in this section.

5.1. Correctness

A. The correctness of the Equations (1) and (2) in HMOSC-I are proven below.

$$\begin{aligned} e(S, P \quad) &= e(\sum_{i=1}^{n} sk_i h_i P_{pub}, P) \\ &= e(\sum_{i=1}^{n} h_i sk_i P, P_{pub}) \\ &= e(\sum_{i=1}^{n} h_i pk_i, P_{pub}) \end{aligned} \tag{9}$$

$$\begin{aligned} k_i &= e(r_i sk_i P_{pub}, pk_r) \\ &= e(r_i x_i sP, pk_r) \\ &= e(r_i x_i P, sQ_r) \\ &= e(r_i pk_i, sQ_r) \\ &= e(R_i, sk_r) = k_i' \end{aligned} \tag{10}$$

B. Equations (3)–(8) can be easily proved. Here we omit them for the limit space.

5.2. Security Proof

In the subsection, we will demonstrate our proposed schemes are secure. For each proposed signcryption scheme, we must prove its confidentiality (i.e., indistinguishability against adaptive chosen ciphertext attacks, short for IND-CCA2) and unforgeablity (i.e., existential unforgeability against adaptive chosen messages attacks, short for EUF-CMA) in a random oracle model due to its encryption and signature functions, respectively, which will make our paper very long because we proposed four schemes in all. Therefore, we mainly prove the confidentiality and unforgeability of MOHSC-I (PKI→IBC many-to-one signcryption) scheme as an example to illustrate our reduction idea. In the following, t_m and t_p indicate the time to calculate one scalar multiplication and a bilinear pairing in G_1, respectively, and n is the number of messages.

Theorem 1. *(Confidentiality of MOHSC-I scheme): **A** is a probabilistic polynomial-time (PPT) adversary with an advantage ε against the IND-CCA2 security within running time t, and asking at most q_i times H_i ($i = 1, 2, 3$) queries, q_k times key-generation queries, q_u times unsigncrypt queries, then there exists an algorithm **C** that can solve a DDHP instance with probability $\varepsilon' \geq \tau(1-\tau)^{q_k+q_u}\varepsilon$ in a time $t' \leq t + O(q_u)t_p + O(2q_{H_1} + 2q_k)t_m$.*

Proof. Here, we show how ***C*** uses ***A*** to settle a given DDHP example (P, aP, bP, cP). □

Initial: ***C*** firstly executes the **Setup** algorithm to set P_{pub}=aP and **PKI-KG** algorithm to get n senders' $\left\{pk_i^*, sk_i^*\right\}_{i=1}^n$, then sends the system parameters and $\left\{pk_i^*, sk_i^*\right\}_{i=1}^n$ to ***A***.

Phase 1: ***C*** keeps the lists L_1, L_2, and L_3 to simulate H_1, H_2, and H_3 oracles. Assume that H_1 queries are different and the challenged identity ID_r^* is sent to H_1 sometime. ***A*** queries $H_1(ID)$ before ID is applied to other inquiries.

- H_1 queries: The list L_1 with structure $\{ID_r, \alpha_r, Q_r, sk_r, \xi_r\}$ is maintained by ***C***. When ***A*** performs the query with ID_r, ***C*** examines whether $\{ID_r, \alpha_r, Q_r, sk_r, \xi_r\}$ is already in L_1. If so, ***C*** returns Q_r to ***A***. Otherwise, ***C*** flips a coin $\xi_r \in \{0, 1\}$ that returns 0 with possibility τ (which will be determined later) and 1 with possibility $1 - \tau$:
  - (1) If $\xi_r = 1$, ***C*** computes $Q_r = bP, \alpha_r = \perp, sk_r = \perp$;
  - (2) Otherwise, ***C*** chooses a random $\alpha_i \in Z_p^*$, computes $Q_r = \alpha_r P$, $sk_r = \alpha_r aP$, adds $\{ID_r, \alpha_r, Q_r, sk_r, \xi_r\}$ to L_1, and returns Q_r to ***A***.
- H_2 queries: The list L_2 has the tuples of $\{k_i, \rho_i\}$, which is maintained by ***C***. When ***A*** submits a k_i and issues H_2 query, the same answer from L_2 will be given if the query has been queried before. Otherwise, ***C*** chooses $\rho_i \in \{0, 1\}^n$ at random, then adds $\{k_i, \rho_i\}$ into L_2 and sends ρ_i to ***A***.
- H_3 queries: The list L_3 has the tuples of $\{pk_i, pk_r, R_i, c_i, h_i\}$. When ***A*** issues a query $\{pk_i, pk_r, R_i, c_i\}$ to H_3, ***C*** examines whether $\{pk_i, pk_r, R_i, c_i, h_i\}$ is already in L_3; if so, ***C*** returns h_i to ***A***. Otherwise, ***C*** chooses a random value $h_i \in Z_q^*$ as answer and adds $\{pk_i, pk_r, R_i, c_i, h_i\}$ to L_3.
- Key-generation queries: When ***A*** performs the query with IDr, if $ID_r = ID_r^*$, ***C*** returns $\perp$. Otherwise, ***C*** requests a H_1 query at first and gets $\{ID_r, \alpha_r, Q_r, sk_r, \xi_r\}$ from L_1 list. Then, ***C*** returns D_r.

– Unsigncrypt queries: $\boldsymbol{A}$ gives the recipient V_r's identity ID_r and a ciphertext σ. If $ID_r = ID_r^*$, $\boldsymbol{C}$ returns $\perp$. Otherwise, $\boldsymbol{C}$ performs **Unsigncrypt** $(\sigma, \{pk_i^*\}_{i=1}^n, sk_{ID_r})$ and returns the corresponding results.

Challenge: $\boldsymbol{A}$ produces two equal length plaintexts ($m_{i0,}$ m_{i1}) and a receiver's identity ID_r^*, which will be challenged. If $ID_r \neq ID_r^*$, $\boldsymbol{C}$ outputs $\perp$. Otherwise, $\boldsymbol{C}$ selects a bit $\beta \in \{0,1\}$ and $r_i^* \in Z_q^*$ randomly, calculates $R_i^* = r_i^* pk_i^*$, $c_i^* = H_2(e(r_i^* pk_i^*, cP)) \oplus m_{i\beta}$, $h_i^* = H_3(pk_i^*, pk_r, R_i^*, c_i^*)$, and $S_i^* = sk_i^* h_i^* P_{pub}$. Then, $\boldsymbol{C}$ computes $S^* = \sum_{i=1}^{n} S_i^*$, returns $\sigma^* = (R_1^*, \ldots, R_n^*, c_1^*, \ldots, c_n^*, S^*)$ to $\boldsymbol{A}$.

Phase 2: $\boldsymbol{A}$ can make a mass of queries as Phase 1. However, $\boldsymbol{A}$ cannot submit the key-generation query on ID_r^* and the unsigncrypt query on σ^* to obtain the plaintexts.

Guess: $\boldsymbol{A}$ outputs the bit β' after enough inquiries. If $\beta = \beta'$, $\boldsymbol{C}$ emits 1, which means (P, aP, bP, cP) are DH tuples. Otherwise, it outputs 0, which means the (P, aP, bP, cP) are random tuples. If $\boldsymbol{A}$'s guess is correct, $\boldsymbol{A}$ should have asked H_2 oracle with $e(r_i^* pk_i^*, abP)$ and $\boldsymbol{C}$ added $\{e(r_i^* pk_i^*, abP), \rho_i^*\}$ into L_2 list. As can be seen from the above, abP is equal to cP.

Further, we will analyze the probability of $\boldsymbol{C}$ success. Define the events E_1, E_2 E_3, and E_4 in the following:

E_1: $\boldsymbol{A}$ does not perform the key-generation query with identity ID_r^*.

E_2: $\boldsymbol{C}$ does not abort the unsigncryption queries.

E_3: $\boldsymbol{A}$ selects ID_r^* as the recipient's identity during the challenge phase.

E_4: $\boldsymbol{A}$ can successfully guess $\beta = \beta'$.

$\boldsymbol{C}$ succeeds if the aforementioned events happen. It is easy to get $\Pr[E_1] = (1-\tau)^{q_k}$, $\Pr[E_2|E_1] = (1-\tau)^{q_u}$, $\Pr[E_3|E_1E_2] \geq \tau$, and $\Pr[E_4|E_1E_2E_3] \geq \varepsilon$, so $\Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \geq \tau(1-\tau)^{q_k+q_u}\varepsilon$.

The computation time of $\boldsymbol{C}$ comes from $\boldsymbol{A}$'s computation time. We can get that 2, 2 scalar multiplications and 1 pairing calculation are needed in the H_1 query, key generation query, unsigncrypt query. So, the time of $\boldsymbol{C}$ solving the DDHP instance is $t' \leq t + O(q_u)t_p + O(2q_{H_1} + 2q_k)t_m$.

Theorem 2. *(Unforgeability of MOHSC-I scheme): $\boldsymbol{F}$ is a forger with a non-negligible advantage ε to forge an aggregate signcryption of the MOHSC-I scheme within running time t, and $\boldsymbol{F}$ requests q_i queries to H_i (i = 1, 2, 3) oracles, q_k queries to key-generation oracle, q_s queries to signcrypt oracle, then the CDHP will be settled by an algorithm $\boldsymbol{C}$ with probability $\varepsilon' \leq \varepsilon\tau(1-\tau)^{q_k+n-1}$ in a time $t' \leq t + O(2q_{H_1} + q_k + 2q_s + n + 1)t_m + O(q_s)t_p$.*

Proof. Here, we show that how $\boldsymbol{C}$ uses $\boldsymbol{F}$ to settle a given CDHP example (P, aP, bP). □

Initial: $\boldsymbol{C}$ executes the **Setup** algorithm to sets $Ppub = aP$, then transmits **params** and s to $\boldsymbol{F}$. $\boldsymbol{C}$ also executes the PKI-KG algorithm and sends the senders' public keys $\{pk_i^*\}_{i=1}^n$ to $\boldsymbol{F}$.

Attack: $\boldsymbol{C}$ keeps four lists L_1, L_2, L_3, and L_k to simulate the hash oracles H_1, H_2, H_3, and the key-generation oracle.

– H_1 queries: The list L_1 with structure $\{ID_r, \alpha_r, Q_r, D_r\}$ is maintained by$\boldsymbol{C}$. $\boldsymbol{C}$ randomly chooses $\alpha_r \in Z_p^*$ and computes $Q_r = \alpha_r P$,$D_r = \alpha_r aP$. Then, it adds $\{ID_r, \alpha_r, Q_r, D_r\}$ to L_1 list and returns $Q_{r,}$ D_r.

– H_2 queries and H_3 queries are the same as in Theorem 1, so we will not describe the details.

– Key-generation queries: The list L_k with structure $\{ID_i, x_i, pk_i, sk_i, d_i\}$ is maintained by $\boldsymbol{C}$. When $\boldsymbol{F}$ requests the query with ID_r, $\boldsymbol{C}$ examines whether $\{ID_i, x_i, pk_i, sk_i, d_i\}$ is already in L_k. If so, $\boldsymbol{C}$ transmits pk_i and sk_i to $\boldsymbol{F}$. Otherwise, $\boldsymbol{C}$ flips a coin $d_i \in \{0,1\}$ that returns 0 with probability τ and 1 with probability $1-\tau$. If $d_i = 0$, C sets $sk_i = bP$,$sk_i = \perp$, adds $\{ID_i, \perp, \perp, sk_i, d_i\}$ to L_k. Otherwise, $\boldsymbol{C}$ randomly picks $x_i \in Z_p^*$, sets $sk_i = x_i$,$pk_i = x_i P$, adds $\{ID_i, x_i, pk_i, sk_i, d_i\}$ to L_k, transmits pk_i and sk_i to $\boldsymbol{F}$.

– Signcrypt queries: $\boldsymbol{F}$ submits one message m_i, the sender V_i's identity ID_i, the receiver's identity ID_r to $\boldsymbol{C}$. If $d_i = 0$, $\boldsymbol{C}$ returns $\perp$. Otherwise, $\boldsymbol{C}$ selects randomly $r_i \in Z_p^*$, computes

$R_i = r_i x_i P_{pub}$,$k_i = e(R_i, Q_r)$. Then, ***C*** makes a H_2 query on k_i and gets (ρ_i, k_i) from L_2 list, ***C*** computes $c_i = m_i \oplus \rho_i$, $S_i = sk_i h_i P_{pub} = x_i h_i aP$, and returns $\sigma_i = (c_i, R_i, S_i)$ to ***F***.

Forgery: ***F*** returns n senders' identities $\left\{ID_i^*\right\}_{i=1}^n$, a receiver's identity ID_r^*, and a new aggregate signcryption ciphertext σ^* on messages $m^* = (m_1^*, m_2^*, \ldots, m_n^*)$. ***F*** wins the game if and only if:

(1) The output of **Unsigncrypt** $(\sigma^*, \{pk_i\}_{i=1}^n, sk_{ID_r^*})$ is valid. The advantage of the forger ***F*** can be defined as its probability of winning the game.
(2) At least one sender, without losing generality, let ID_1^* has not been requested in the key-generation query. In addition, $(\left\{pk_i^*\right\}_{i=1}^n, \left\{m_i^*\right\}_{i=1}^n, ID_r^*)$ have never been asked in the Signcrypt queries.

For all $1 \le i \le n$, C gets tuples $\left\{ID_i^*, x_i^*, sk_i^*, pk_i^*, d_i^*\right\}$,$\left\{pk_i^*, pk_r^*, R_i^*, c_i^*, h_i^*\right\}$ from L_k and $L_{3,}$ respectively. If $d_1^* = 0$ and $d_i^* = 1 (i = 2, \ldots, n)$, ***C*** continues. Otherwise, ***C*** aborts. Since σ^* meets the Equation (1), then

$$e(h_1^* pk_1^*, P_{pub}) = e(S^*, P) e(\sum_{i=2}^{n} h_i^* pk_i^*, -P_{pub}).$$

Since $pk_1^* = bP$, $P_{pub} = aP$ and $pk_i^* = x_i^* P$ for all $2 \le i \le n$, it can be transformed into:

$$e(h_1^* abP, P) = e(S^*, P) e(-\sum_{i=2}^{n} h_i^* x_i^* aP, P).$$

Hence, ***C*** can compute

$$abP = (h_1^*)^{-1} (S^* - \sum_{i=2}^{n} x_i^* h_i^* aP).$$

Further, we will analyze the possibility of ***C*** success. Define the events E_1, E_2, and E_3 in the following:

E_1: ***C*** does not abort all queries of key-generation.
E_2: ***F*** produces a valid and nontrivial forged aggregate ciphertext.
E_3: E_2 happens, and $d_1^* = 0, d_i^* = 1$ $(2 \le i \le n)$.
C is successful as long as the above events happened. The probability is $\Pr[E_1 \wedge E_2 \wedge E_3]$.
We know that

$$\Pr[E_1] \ge (1-\tau)^{q_k}, \Pr[E_2 | E_1] \ge \varepsilon, \Pr[E_3 | E_1 \wedge E_2] \ge \tau (1-\tau)^{n-1}.$$

So that

$$\Pr[E_1 \wedge E_2 \wedge E_3] \ge (1-\tau)^{q_k} \varepsilon \tau (1-\tau)^{n-1} = \varepsilon \tau (1-\tau)^{q_k + n - 1}.$$

The computation time of ***C*** comes from ***F***'s computation time, which contains the time ***C*** responses queries and the time that ***C*** calculates the CDHP example. We can get that 2, 1, 2 scalar multiplications are needed in the H_1 query, key-generation query and signcrypt query, respectively. In addition, 1 pairing calculation is needed in the signcrypt query. n+1 scalar multiplication is needed in ***C*** calculating the CDHP example. Therefore, the CDHP example will be settled within time $t' \le t + O(2q_{H_1} + q_k + 2q_s + n + 1)t_m + O(q_s)t_p$.

The proof process of confidentiality and unforgeability MOHSC-II, OMHSC-I, and OMHSC-II are very similar to **Theorem 1** and **Theorem 2**, respectively. Therefore, here we omit the detailed proof due to the limited space.

6. Performance Analysis

Figure 2 gives a specific application scenario of the proposed schemes. If vehicle V_A collides with vehicle V_B, V_A, V_B, and V_C signcrypt the traffic information including collision messages to

a nearby RSU by our many-to-one signcyption schemes (MOHSC-I or MOHSC-II) just in order to avoid traffic jams, and these steps are repeated in a short time interval according to DSRC protocol [1]. Then, the nearby RSU unsigncrypted the messages from V_A, V_B, and V_C after authenticating the messages integrity and vehicles' identities, which is just to ensure that the vehicles will be responsible for messages. Further, by using our one-to-many signcryption schemes (OMHSC-I or OMHSC-II), the RSU signcrypts and broadcasts the integrated information to other adjacent vehicles in time and makes them go around early. As a result, other cars can avoid joining this traffic congestion and make traffic management more convenient. Compared with the other current broadcast technology, our schemes can guarantee the integrity and tamper-resistance of message and the authentication of message sources, which can improve the credibility of message.

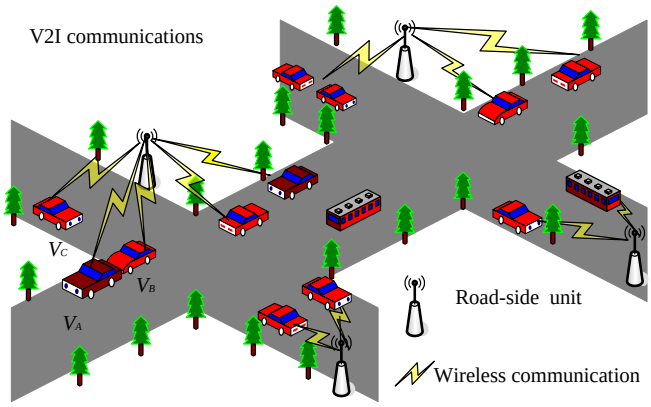

Figure 2. Communication scenarios of our many-to-one and one-to-many schemes.

To guarantee the authenticity and confidentiality of message, the sender signcrypts message in our schemes. Upon receiving messages from many vehicles, the RSU first verifies the authenticity of these messages and then discards the error or distorted messages. Although some methods in [21–26] achieve heterogeneous communication, they have different disadvantages, which are given in **Related Work**. The function comparisons of all the schemes are depicted in Table 1. In addition, Tables 2 and 3 and Figures 3 and 4 mainly focus on the comparison of computation costs, Figures 5 and 6 give the comparison of energy consumption because both vehicles and RSUs are computation-limited and energy-constrained devices, and the computation costs and energy consumption directly affect the practicability of our schemes.

Table 1. Function comparisons.

Scheme	Cryptosystem	Provable Security	n Ciphertexts Length
HOOSC [23]	IBC→PKI	Yes	$n(\|m\| + 3\|G_1\|)$
SEDT [24]	IBC→PKI	Yes	$n\|m\| + (n^2 + 3n)\|G_1\|$
HSC-I [25]	**PKI→IBC**	Yes	$n(\|m\| + 2\|G_1\|)$
HSC-II [25]	**IBC→PKI**	Yes	$n(\|m\| + 2\|G_1\|)$
MHSC [26]	PKI→IBC	Yes	$n\|m\| + (n + 1)\|G_1\|$
MOHSC-I	**PKI→IBC**	Yes	$n\|m\| + (n + 1)\|G_1\|$
MOHSC-II	**IBC→PKI**	Yes	$n\|m\| + (n + 1)\|G_1\|$

Table 2. Computation comparisons (public key infrastructure (PKI)→ identity-based cryptosystem (IBC)).

Scheme	PKI Setup	IBC Setup	Signcryption	Unsigncryption	Total
HSC-I [25]	$n(2t_m + t_{inv})$	$n(t_m + t_{inv})$	$n(t_e + 3t_m)$	$n(2t_p + t_e + t_{inv})$	$2nt_p + 2nt_e + 3nt_{inv} + 6nt_m$
MHSC [26]	nt_m	$n(t_m + t_{inv})$	$n(t_e + 4t_m)$	$(n + 2)t_p + nt_m$	$(n + 2)t_p + nt_e + nt_{inv} + 7nt_m$
MOHSC-I	nt_m	nt_m	$n(t_p + 3t_m)$	$(n + 2)t_p + nt_m$	$(2n + 2)t_p + 6nt_m$

Table 3. Computation comparisons (IBC→ PKI).

Scheme	PKI Setup	IBC Setup	Signcryption	Unsigncryption	Total
HOOSC [23]	$n(2t_m + t_{inv})$	$n(t_m + t_{inv})$	$n(t_e + 2t_m + t_{inv})$	$n(2t_p + t_e + 2t_m + t_{inv})$	$2nt_p + 2nt_e + 4nt_{inv} + 7nt_m$
HSC-II [25]	$n(t_m + t_{inv})$	$n(2t_m + t_{inv})$	$n(t_e + 2t_m)$	$n(2t_p + t_e + t_m + t_{inv})$	$2nt_p + 2nt_e + 3nt_{inv} + 6nt_m$
MOHSC-II	nt_m	nt_m	$n(t_p + 3t_m)$	$(n + 2)t_p + nt_m$	$(2n + 2)t_p + 6nt_m$

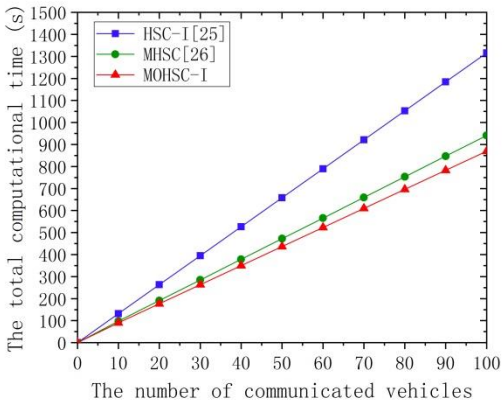

Figure 3. Comparisons of the computational time (PKI→IBC).

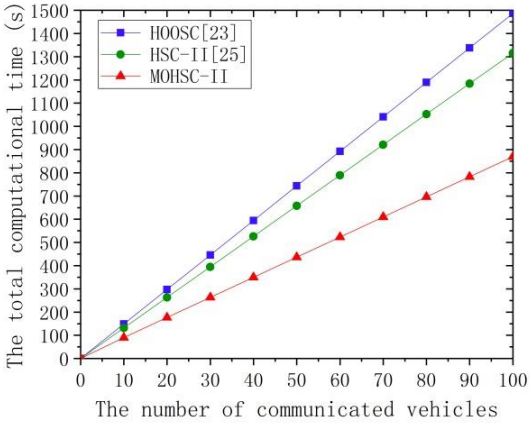

Figure 4. Comparisons of the computational time (IBC→PKI).

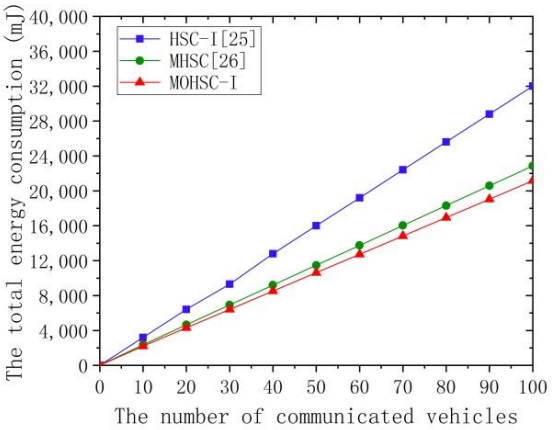

Figure 5. Comparisons of total energy consumption (PKI→IBC).

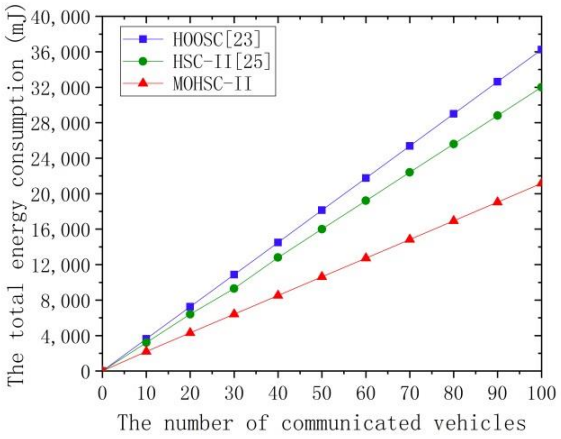

Figure 6. Comparisons of total energy consumption (IBC→PKI).

From Table 1, we see only the schemes from [25] and our schemes can achieve bi-directional heterogeneous communication. Since our MOHSC-I and MOHSC-II schemes simultaneously send n messages but the schemes in [23–25] send one message to the receiver, their ciphertexts length should be multiplied by n just to ensure the fairness of comparisons. Obviously the aggregate ciphertext length from n messages in our MOHSC-I and MOHSC-II schemes is the shortest, regardless of the size of m and G_1 that are selected. So, the communication costs and storage in our schemes have also been reduced.

In Tables 2 and 3, t_m, t_p, t_{inv}, and t_e, respectively, represent the time of performing a scalar multiplication, one pairing calculation in G_1, an inverse operation in Z_q^*, and an exponent operation in G_2 individually. We do not consider other less time-consuming operations, such as the XOR operation. For the fairness of comparison, we extend schemes in [23,25,26] to n senders sending n messages to a receiver in Tables 2 and 3. Since other schemes in [23,26] only provide one-way heterogeneous communication, the comparisons are divided into PKI→IBC in Table 2 and IBC→PKI in Table 3 just for more scientific and elaborate results.

From Tables 2 and 3, we can see our MOHSC-I and MOHSC-II schemes require a smaller total computation time (the sum of PKI setup, IBC setup, signcryption, and unsigncryption) when n messages are involved. The more intuitive analyses are given in Figures 3 and 4 for schemes in [23,25,26] and our schemes. We implement the experiment on MICA2 platform (same as [24]). We can get t_p, t_e, and t_m takes 1.9 s, 0.9 s, and 0.81 s, respectively [24]. Note that a t_{inv} operation needs roughly 0.9 s, although, theoretically, a t_{inv} operation is more time-consuming than a t_e operation. Finally, according to Tables 2 and 3, we can compute the total computation time of HSC-I, MHSC, MOHSC-I, HOOSC, HSC-II, and MOHSC-II are $2n \times 1.9 + (2n + 3n) \times 0.9 + 6n \times 0.81 = 13.16n$s, $(n + 2) \times 1.9 + 2n \times 0.9 + 7n \times 0.81 = 9.37n + 3.8$s, $(2n + 2) \times 1.9 + 6n \times 0.81 = 8.66n + 3.8$s, $2n \times 1.9 + (2n + 4n) \times 0.9 + 7n \times 0.81 = 14.87n$s, $2n \times 1.9 + (2n + 3n) \times 0.9 + 6n \times 0.81 = 13.16n$s, and $(2n + 2) \times 1.9 + 6n \times 0.81 = 8.66n + 3.8$s, respectively. The comparisons of total computational time are shown in Figures 3 and 4.

Since vehicles and RSUs are both energy-constrained devices, we must consider the energy consumption. According to [24], a t_p operation consumes 45.6 mJ, a t_e operation consumes 21.6 mJ, and a t_m operation consumes 19.44 mJ; here, we suppose t_{inv} also consumes 21.6 mJ. So, the computational energy consumption of HSC-I, MHSC, HMOSC-I, HOOSC, HSC-II, and HMOSC-II schemes are $2n \times 45.6 + (2n + 3n) \times 21.6 + 6n \times 19.44 = 315.84n$ mJ, $(n + 2) \times 45.6 + 2n \times 21.6 + 7n \times 19.44 = 224.88n + 91.2$ mJ, $(2n + 2) \times 45.6 + 6n \times 19.44 = 207.84n + 91.2$ mJ, $2n \times 45.6 + (2n + 4n) \times 21.6 + 7n \times 19.44 = 356.88n$ mJ, $2n \times 45.6 + (2n + 3n) \times 21.6 + 6n \times 19.44 = 315.84n$ mJ, and $(2n + 2) \times 45.6 + 6n \times 19.44 = 207.84n + 91.2$ mJ, respectively. For the communication energy consumption, as in [24], a sensor consumes 0.052 mJ and 0.019 mJ to transmit and receive a 1-byte message (namely the total energy consumption of communicating a one-byte message is 0.071 mJ). Combined with the ciphertext length in Table 1 and the common assumptions that $|G_1| = 160$ bits and $|m| = 160$ bits, which can be reduced to

20 bytes, the communication energy consumption of HSC-I, MHSC, MOHSC-I, HOOSC, HSC-II, and MOHSC-II schemes are $(20 + 2 \times 20) \times 0.071 \times n = 4.26n$ mJ, $20 \times 0.071 \times n + (n + 1) \times 20 \times 0.071 = 2.84n + 1.42$ mJ, $20 \times 0.071 \times n + (n + 1) \times 20 \times 0.071 = 2.84n + 1.42$ mJ, $(20 + 3 \times 20) \times 0.071 \times n = 5.68n$ mJ, $(20 + 2 \times 20) \times 0.071 \times n = 4.26$n mJ, and $20 \times 0.071 \times n + (n + 1) \times 20 \times 0.071 = 2.84n + 1.42$ mJ, respectively. In a word, the total energy consumption of HSC-I, MHSC, MOHSC-I, HOOSC, HSC-II, and MOHSC-II schemes are $315.84n + 4.26n = 320.1$n mJ, $224.88n + 91.2 + 2.84n + 1.42 = 227.72n + 92.62$ mJ, $207.84n + 91.2 + 2.84n + 1.42 = 210.68n + 92.62$ mJ, $356.88n + 5.68n = 362.56$n mJ, $315.84n + 4.26n = 320.1n$, and $207.84n + 91.2 + 2.84n + 1.42 = 210.68n + 92.62$ mJ, respectively. The comparisons of total energy consumption are shown in Figures 5 and 6.

As can be seen from the Tables 1–3 and Figures 3–6, our schemes have the minimum aggregate ciphertext length, total computation costs, and total energy consumption among these schemes. It is very viable and sound for the practical application of VANETs.

7. Conclusions

In the article, two many-to-one heterogeneous signcryption schemes and two one-to-many heterogeneous signcryption schemes for secure V2I communication in VANETs are proposed. These schemes can all construct a secure channel between heterogeneous vehicles and a RSU to support confidentiality, authentication, integrity, and non-repudiation services in a logical step. Specifically, the many-to-one signcryption schemes adopt the aggregate method to support batch verification when multiple vehicles in the PKI (IBC) system transmit messages to a nearby RSU in the IBC (PKI) system, and the one-to-many schemes support a RSU in the PKI (IBC) broadcasts a message to multiple vehicles registered in IBC (PKI) system. All the schemes can be proven to be IND-CCA2 and EUF-CMA secure. Furthermore, the numerical analyses and simulation results, which are shown in Tables 1–3 and Figures 3–6, can demonstrate the aggregate ciphertext length, communication costs, total computation costs, and total energy consumption of our schemes are better than the existing ones. The analyses show our schemes are more suitable for the practical heterogeneous V2I communication in VANETs.

Author Contributions: Writing—original draft: F.Z.; supervision: Y.L.; methodology: Y.D.

Funding: The work was are partly supported by the National Natural Foundation Science of China [grant numbers 61802243, 61602232, 61572246]; Key R&D Program in industry field of Shaanxi Province (grant numbers 2019GY-013), the Fundamental Research Funds for the Central Universities (2019CSLY002, GK201803005, GK201903011).

Acknowledgments: The authors thank all the received funds to support of the research work and the anonymous reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jiang, D.; Taliwal, V.; Meier, A.; Holfelder, W. Design of 5.9 GHZ dsrc-based vehicular safety communication. *IEEE Wirel. Commun.* **2006**, *13*, 36–43. [CrossRef]
2. Zhou, J.; Tian, D.; Wang, Y.; Sheng, Z.; Duan, X.; Leung, V. Reliability-optimal cooperative communication and computing in connected vehicle systems. *IEEE Trans. Mob. Comput.* **2019**, *99*, 1–18. [CrossRef]
3. Zheng, Y. Digital signcryption or how to achieve cost (signature & encryption) ≪ cost (signature) + cost (encryption). In *Advances in Cryptology—Crypt'97*; LNCS 1294; Springer: Berlin/Heidelberg, Germany, 1997; pp. 165–179.
4. An, J.; Dodis, Y.; Rabin, T. On the security of joint signature and encryption. In *Advances in Cryptology—Eurocrypt 2002*; LNCS 2332; Springer: Berlin/Heidelberg, Germany, 2002; pp. 83–107.
5. Baek, J.; Steinfeld, R.; Zheng, Y. Formal proofs for the security of signcryption. *Cryptology* **2007**, *20*, 203–235. [CrossRef]
6. Li, C.; Yang, G.; Wong, D.; Deng, X.; Chow, S. An efficient signcryption scheme with key privacy and its extension to ring signcryption. *Comput. Secur.* **2010**, *18*, 451–473. [CrossRef]
7. Malone-Lee, J.; Mao, W. Two birds one stone: Signcryption using RSA. In *Proc. CT-RSA*; LNCS 2612; Springer: Berlin/Heidelberg, Germany, 2003; pp. 211–226.

8. Enos, G.; Zheng, Y. An ID-based signcryption scheme with compartmented secret sharing for unsigncryption. *Inf. Process. Lett.* **2015**, *115*, 128–133. [CrossRef]
9. Sun, Y.; Li, H. ID-based signcryption KEM to multiple recipients. *Chin. J. Electron.* **2011**, *20*, 317–322.
10. Selvi, S.; Vivek, S.; Shriram, J.; Kalaivani, S.; Rangan, C. Identity based aggregate signcryption schemes. In *Progress in Cryptology—INDOCRYPT 2009*; LNCS 5922; Roy, B., Sendrier, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 378–397.
11. Han, Y.; Lu, W.; Zhang, J. Identity based aggregate signcryption scheme. *Lect. Notes Electr. Eng.* **2014**, *273*, 383–389.
12. Qi, Z.; Ren, X.; Yang, G. Provably secure general aggregate signcryption scheme in the random oracle model. *China Commun.* **2012**, *9*, 107–116.
13. Li, F.; Han, Y.; Jin, C. Cost-Effective and anonymous access control for Wireless Body Area Networks. *IEEE Syst. J.* **2016**, *12*, 747–758. [CrossRef]
14. Yu, H.; Yang, B. Provably secure certificateless hybrid signcryption. *J. Comput.* **2015**, *38*, 804–813.
15. Su, J.; Liu, J. Efficient certificateless aggregate signcryption scheme without bilinear pairings. *J. Comput. Appl.* **2018**, *38*, 374–378, 385.
16. Wang, C.; Liu, C.; Li, Y.; Qiao, H.; Chen, L. Multi-message and multi-receiver heterogeneous signcryption scheme for ad-hoc networks. *Inf. Secur. J. A Glob. Perspect.* **2017**, *26*, 1–17. [CrossRef]
17. Wang, C.; Liu, C.; Niu, S.; Chen, L.; Wang, X. An authenticated key agreement protocol for cross-domain based on heterogeneous signcryption scheme. In Proceedings of the 2017 13th International Wireless Communications & Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017.
18. Liu, J.; Zhang, L.; Sun, R.; Du, X.; Guizani, M. Mutual heterogeneous signcryption schemes for 5G network slicings. *IEEE Access* **2018**, *6*, 7854–7863. [CrossRef]
19. Li, Y.; Lu, L.; Zhang, K. A Novel Cross-Domain Many-to-one V2I for Hetergeneous VANETs. *J. Inf. Sci. Eng. JISE* **2018**, *34*, 869–884.
20. Li, F.; Han, Y.; Jin, C. Practical Signcryption for secure communication of Wireless Sensor Networks. *Wirel. Pers. Commun.* **2016**, *89*, 1391–1412. [CrossRef]
21. Sun, Y.; Li, H. Efficient signcryption between TPKC and IDPKC and its multi-receiver construction. *Sci. China Inf. Sci.* **2010**, *53*, 557–566. [CrossRef]
22. Huang, Q.; Wong, D.S.; Yang, G. Heterogeneous signcryption with key privacy. *Comput. J.* **2011**, *54*, 525–536. [CrossRef]
23. Li, F.; Xiong, P. Practical secure communication for integrating wireless sensor networks into the internet of things. *IEEE Sens. J.* **2013**, *13*, 3677–3684. [CrossRef]
24. Li, F.; Zheng, Z.; Jin, C. Secure and efficient data transmission in the Internet of Things. *Telecommun. Syst.* **2016**, *62*, 111–122. [CrossRef]
25. Li, F.; Zhang, H.; Takagi, T. Efficient signcryption for heterogeneous systems. *IEEE Syst. J.* **2013**, *7*, 420–429. [CrossRef]
26. Niu, S.; Niu, L.; Wang, C.; Du, X. A provable aggregate signcryption for heterogeneous systems. *J. Electron. Inf. Technol.* **2017**, *39*, 1213–1218.