Hindawi
Security and Communication Networks
Volume 2018, Article ID 1378583, 14 pages
https://doi.org/10.1155/2018/1378583

Research Article

Privacy-Preserving Data Aggregation Protocol for Fog Computing-Assisted Vehicle-to-Infrastructure Scenario

Yanan Chen,[1,2] Zhenyu Lu,[3] Hu Xiong,[3,4] and Weixiang Xu[1]

[1] *MOE Key Laboratory for Transportation Complex Systems Theory and Technology, School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China*
[2] *Basic Course Teaching Department, Jiangxi University of Science and Technology, Ganzhou, China*
[3] *School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China*
[4] *State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China*

Correspondence should be addressed to Hu Xiong; xionghu.uestc@gmail.com

Received 30 August 2017; Revised 20 October 2017; Accepted 9 November 2017; Published 18 April 2018

Academic Editor: Qi Jiang

Vehicle-to-infrastructure (V2I) communication enables moving vehicles to upload real-time data about road surface situation to the Internet via fixed roadside units (RSU). Thanks to the resource restriction of mobile vehicles, fog computation-enhanced V2I communication scenario has received increasing attention recently. However, how to aggregate the sensed data from vehicles securely and efficiently still remains open to the V2I communication scenario. In this paper, a light-weight and anonymous aggregation protocol is proposed for the fog computing-based V2I communication scenario. With the proposed protocol, the data collected by the vehicles can be efficiently obtained by the RSU in a privacy-preserving manner. Particularly, we first suggest a certificateless aggregate signcryption (CL-A-SC) scheme and prove its security in the random oracle model. The suggested CL-A-SC scheme, which is of independent interest, can achieve the merits of certificateless cryptography and signcryption scheme simultaneously. Then we put forward the anonymous aggregation protocol for V2I communication scenario as one extension of the suggested CL-A-SC scheme. Security analysis demonstrates that the proposed aggregation protocol achieves desirable security properties. The performance comparison shows that the proposed protocol significantly reduces the computation and communication overhead compared with the up-to-date protocols in this field.

1. Introduction

Most of road anomalies, that is, potholes, bumps and slipperiness, are potentially hazardous to the commuters and vehicles [1]. Naturally, the condition of road surface is considered to be an important criterion for assessing the quality of transportation infrastructure [2]. The continuous development of sensing technique provides a promising approach to build an autonomous system for monitoring road surface condition [3]. Specifically, the mobile sensors embedded in mobile vehicles are used to sense the real-time data about road surface condition [4]. With the V2I communication [5], the data collected by the vehicles can be uploaded to the backend server via the RSUs installed at road intersections. By collecting and analyzing these real-time road surface data, the congestion of traffic and car crashes can be reduced significantly. Thanks to the resource restraint of mobile vehicles, the vehicular cloud networking [6, 7] has been introduced to ease the cost of vehicles, where the sensed data are stored in the remote cloud centers. It is easy to observe that the delivery of these data to the cloud servers located in the core network is commonly considered to be cumbersome due to the unreliable latency and network congestion [8]. To address these issues, the fog computing [9–11] has been introduced as an alternative for cloud computing. Different from cloud computing, elastic and virtual cloud resources are extended to one or more collaborative edge devices in the fog computing. In this sense, the collected data can be preprocessed and aggregated by the edge devices, which are instantiated by the resource-abundant RSU, before

uploading to the data analytic center [12]. Therefore, the real-time road surface data can be efficiently processed with the support of fog computing-assisted V2I communication.

However, the fog computing-based V2I communication scenario cannot be accepted and deployed widely if the security of the transmitted data has not been considered appropriately. It is desirable to achieve data confidentiality such that the transmitted data can only be accessed by the intended RSU [13]. Otherwise, the collected data may be abused by the malicious adversary without any cost. Besides, it is also necessary to achieve message unforgeability such that the adversary is computationally infeasible to impersonate any vehicle [14]. Otherwise, the result about the analysis of collected data may be polluted by the forged data. To fulfill the mentioned security goals, it is naturally to introduce the public key encryption and signature to generate the ciphertext on the transmitted data. According to [15], signcryption is a promising primitive that achieves the security goals of encryption and signature simultaneously. It is realized by combining the public key encryption and digital signatures in one logical step. Moreover, this technique entails minimized computation and communication overhead compared with the sign-then-encrypt paradigm [16]. Since its introduction, the signcryption primitive has been studied in several cryptosystems, that is, traditional public key infrastructure- (PKI-) based cryptosystem [15], identity-based cryptosystem [17], and certificateless cryptosystem [16]. In the traditional PKI-based cryptosystem, the certificate management is a burdensome task. To alleviate the overhead of this task, the identity-based public key cryptosystem [18] has been introduced, where a trusted third party termed as private key generator is adopted to issue private keys for the users. This paradigm results in key escrow problem since the private key generator knows the private keys of all users in the system [18]. The certificateless cryptosystem [19] inherits from identity-based cryptosystem, whereas it eliminates the demand for the private key generator with key escrow capability. In this cryptosystem, a trusted third party named key generation center (KGC) is adopted to generate the private keys for users. Only a partial private key is issued by the KGC for each user. The full private key of a user is composed of the partial private key received from KGC and a secret value selected by his/herself. Because the full private key of a user is not held by the KGC, certificateless public key cryptosystem solves the key escrow problem of the identity-based cryptosystem. Thus, certificateless signcryption seems to be a promising primitive to ensure the security of the V2I communication.

Based on the idea of certificateless signcryption, Basudan et al. [20] proposed an anonymous aggregation protocol to secure the V2I communication recently. Unfortunately, in this paper, the protocol of [20] is demonstrated to be subject to the forgery attack, by which an adversary is able to forge a valid signcryption on any data. Besides, this protocol is constructed by utilizing the expensive bilinear pairings, which makes this protocol inefficient. Therefore, it is fair to regard the construction of anonymous aggregation protocol for the fog computing-based V2I communication scenario as an open issue.

Motivated by the practical needs, a privacy-preserving protocol for the V2I communication scenario with fog computing is proposed in this paper. The major contributions of this paper are summarized as follows:

(i) Firstly, Basudan et al.'s [20] protocol is demonstrated to be subject to the forgery attack, by which an adversary is able to forge a valid signcryption on any data. In this sense, the aggregation protocol in [20] does not provide unforgeability as they claimed.

(ii) Next, a light-weight and anonymous aggregation protocol for the V2I communication scenario with fog computing is proposed by elaborately combining a CL-A-SC scheme and the fog computing architecture. Specifically, the suggested protocol is realized without resorting to the costly bilinear pairings. Besides, the proposed protocol is proved secure under the standard computational Diffie–Hellman assumption and elliptic curve discrete logarithm problem in the random oracle model. Furthermore, the proposed aggregation protocol proved to be able to achieve desirable security properties including confidentiality, unforgeability, mutual authentication, anonymity, and key escrow resilience.

(iii) The practical performance of the proposed protocol and Basudan et al.'s protocol is presented through the experimental simulation. According to the simulation results, the proposed protocol outperforms Basudan et al.'s protocol in terms of computation and communication overhead.

The organization of this paper is summarized as follows: the next section describes the system model, mathematical background, design objectives, the notion, and the security model of CL-A-SC scheme. In Section 3, Basudan et al.'s protocol is briefly reviewed. After that, the forgery attack against this protocol is presented. The proposed protocol is introduced in Section 4. Furthermore, the security of the proposed protocol is discussed in Section 5, where the comparison of the practical performance of the proposed protocol and Basudan et al.'s protocol is also provided. Finally, Section 6 concludes this paper.

2. Preliminaries

The background information is introduced in this section.

2.1. System Model. The considered system is comprised of three types of entities: control center, mobile sensors and RSUs. For ease of understanding, the system model is depicted in Figure 1. The definitions of the entities are described as follows:

(i) Control center (CC): CC is considered to be a trustee which is able to initialize the whole system and generate the partial private key for mobile sensors and RSUs.

(ii) Mobile sensors: the devices are embedded into the vehicles to generate the report about the road event, that is, potholes, slipperiness and bumps.

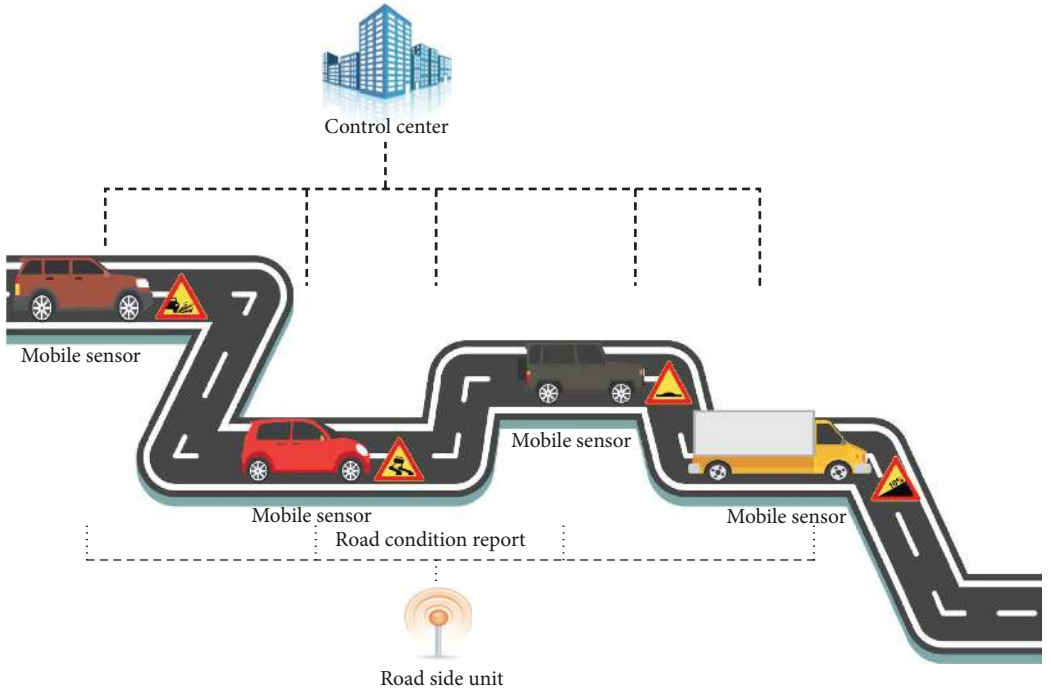

Figure 1: System model.

(iii) RSU: each RSU is able to receive and process the messages sent by the mobile sensors.

2.2. Mathematical Background

2.2.1. Elliptic Curve Group. Let an elliptic curve $\mathscr{E}$ over a prime finite field $\mathbb{F}_p$ denote a set of points (x, y), which are defined by the equation $\mathscr{E}(x, y) : y^2 = x^3 + ax + b$ with the discriminant $4a^3 + 27b^2 \neq 0 \bmod p$ $(a, b \in_R \mathbb{F}_p)$. This set of points and the point at infinity (denoted by $\mathcal{O}$) form a group $\mathscr{G} = \{(x, y) \mid (x, y) \in \mathbb{F}_p \cap \mathscr{E}(x, y) = 0\} \cup \{\mathcal{O}\}$. Particularly, $\mathscr{G}$ is an additive cyclic group formed by $\mathscr{E}$ and the point addition law, which is denoted by + and defined as follows. Let P, Q, and R be three elements in $\mathscr{G}$, where R is the intersection of the line l and $\mathscr{E}$. Specifically, l connects P and Q (tangent line to $\mathscr{E}$ if $P = Q$). Let l' be another line, which connects R and $\mathcal{O}$. The sum of $P + Q$ is denoted by the intersection of l' and $\mathscr{E}$. Moreover, the scalar multiplication on $\mathscr{E}$ is calculated as $mP = \underbrace{P + P + \cdots + P}_{m \text{ times}}$.

2.2.2. Bilinear Maps. Let $\mathscr{G}$ be an additive cyclic group of prime order q, $\mathscr{G}_T$ be a multiplicative cyclic group of the same order, $\hat{e} : \mathscr{G} \times \mathscr{G} \to \mathscr{G}_T$ be an admissible bilinear map and $P \in \mathscr{G}$ denote a generator of $\mathscr{G}$. $\hat{e}$ is considered to have the following features:

(1) Bilinearity: for all $P, Q \in \mathscr{G}$ and $a, b \in_R Z_q^*$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.

(2) Nondegeneracy: there exists $P, Q \in \mathscr{G}$ such that $\hat{e}(P, Q) \neq 1$.

(3) Computability: for all $P, Q \in \mathscr{G}$, there exists an efficient algorithm to calculate $\hat{e}(P, Q)$.

2.2.3. Cryptographic Assumptions. Given the mathematical background described above, the cryptographic assumptions are defined as follows.

Definition 1 (computational Diffie–Hellman assumption). This assumption is denoted as CDH. Given a tuple $\langle P, aP, bP \rangle \in \mathscr{G}$ $(a, b \in_R Z_q^*)$, the CDH assumption in $\mathscr{G}$ is to calculate abP.

Definition 2 (elliptic curve discrete logarithm problem). This assumption is denoted as ECDLP. Given a tuple $\langle P, aP \rangle \in \mathscr{G}$ $(a \in_R Z_q^*)$, the ECDLP assumption in $\mathscr{G}$ is to calculate a.

2.3. The CL-A-SC Scheme

2.3.1. Definition. Let $\mathscr{U} = \{U_1, U_2, \ldots, U_n\}$ denote a set of users. The user with identity ID_R is assumed to be the message receiver. The scheme consists of the following algorithms:

(i) CL-A-SC.Setup: on inputting the security parameter, this algorithm generates the public parameters *params* and the master private key *msk*.

(ii) CL-A-SC.Key-Generation: this algorithm is carried out by each U_i and KGC interactively.

(1) Given *params*, each U_i generates his/her user public/private key pair $(upk_{i,1}, usk_{i,1})$.

(2) Given *params*, *msk*, the identity ID_i of U_i, and its corresponding user public key $upk_{i,1}$, KGC

generates the partial public/private key pair $(upk_{i,2}, usk_{i,2})$.

(3) $(upk_{i,1}, upk_{i,2})$ and $(usk_{i,1}, usk_{i,2})$ are set to be the full public key and full private key of U_i, respectively.

(iii) CL-A-SC.Signcryption: this algorithm is carried out by each U_i. On inputting *params*, a message m_i, full private key $(usk_{i,1}, usk_{i,2})$ of U_i, and the public key $(upk_{R,1}, upk_{R,2})$ of the user with identity ID_R, this algorithm outputs signcryption C_i on m_i.

(iv) CL-A-SC.Aggregate: on inputting a set of signcryption schemes $(C_i)_{i=1,\ldots,n}$, this algorithm outputs aggregate signcryption C on messages $(m_i)_{i=1,\ldots,n}$.

(v) CL-A-SC.Aggregate-Verification: on inputting *params*, aggregate signcryption C and the set of users with its public keys, this algorithm outputs true if C is valid or false otherwise.

(vi) CL-A-SC.Designcryption: on inputting an aggregate signcryption C and the full private key $(usk_{R,1}, usk_{R,2})$ of the user with identity ID_R, this algorithm outputs a set of messages $(m_i)_{i=1,\ldots,n}$.

2.3.2. Security Model. There are two types of adversaries considered in the certificateless cryptosystem [19]. A Type I adversary $\mathscr{A}_1$ is able to replace the public key of a legitimate user with a bogus one but cannot access the master private key. A Type II adversary $\mathscr{A}_2$ is able to access the master private key but cannot execute the public key replacement. According to the protocol of [22], the security notions of data confidentiality and mutual authentication for the CL-A-SC scheme are captured by the indistinguishability and the existential unforgeability of the signcryption, respectively. By using the same security model provided in [22], the ability of the adversaries is modeled by the following four interactive games.

Game 3. This game is played by a challenger $\mathscr{C}$ and a Type I adversary $\mathscr{A}_1$.

(i) *Initializing*: $\mathscr{C}$ executes CL-A-SC.Setup algorithm to obtain the public parameters *params* and the master private key *msk*. After that, $\mathscr{C}$ sends params to $\mathscr{A}_1$.

(ii) *Training*: $\mathscr{A}_1$ is able to query the following oracles (these oracles model the capability of $\mathscr{A}_1$ in reality) in an adaptive manner:

(a) Secret-Value-Extraction(ID_i): on receiving the query on ID_i, this oracle returns the corresponding secret value $usk_{i,1}$ to $\mathscr{A}_1$.

(b) Partial-Private-Key-Extraction(ID_i): on receiving the query on ID_i, this oracle returns the corresponding partial private key $usk_{i,2}$ to $\mathscr{A}_1$.

(c) Public-Key-Extraction(ID_i): on receiving the query on ID_i, this oracle returns the corresponding public key $(upk_{i,1}, upk_{i,2})$ to $\mathscr{A}_1$.

(d) Public-Key-Replacement($\mathrm{ID}_i, upk'_{i,1}, upk'_{i,2}$): on receiving the query on ID_i, this oracle updates the public key $(upk_{i,1}, upk_{i,2})$ into $(upk'_{i,1}, upk'_{i,2})$.

(e) Signcryption($m_i, \mathrm{ID}_i, \mathrm{ID}_j$): on receiving the query on ID_i, ID_j, and a message m_i, this oracle prompts $\mathscr{C}$ to execute CL-A-SC.Signcryption algorithm to get signcryption C_i on m_i, where ID_i and ID_j are considered to be identity of the sender and the receiver, respectively. After that, $\mathscr{C}$ returns C_i to $\mathscr{A}_1$.

(f) Designcryption($C, (\mathrm{ID}_i)_{i=1,\ldots,n}, \mathrm{ID}_j$): on receiving the query on $(\mathrm{ID}_i)_{i=1,\ldots,n}$, ID_j and aggregate signcryption C, where $(\mathrm{ID}_i)_{i=1,\ldots,n}$ and ID_j are considered to be identity of the senders and the receiver, respectively. This oracle prompts $\mathscr{C}$ to execute the CL-A-SC.Aggregate-Verification algorithm on $(C, (\mathrm{ID}_i)_{i=1,\ldots,n}, \mathrm{ID}_j)$. If the output of this execution is false, this oracle returns "NULL" to $\mathscr{A}_1$; otherwise, $\mathscr{C}$ executes CL-A-SC.Designcryption algorithm on $(C, (\mathrm{ID}_i)_{i=1,\ldots,n}, \mathrm{ID}_j)$ and returns the output of this execution to $\mathscr{A}_1$.

(iii) *Challenging*: $\mathscr{A}_1$ sends $\{(m^*_{i,0}, m^*_{i,1}, \mathrm{ID}^*_i)_{i=1,\ldots,n}, \mathrm{ID}^*_j\}$ to $\mathscr{C}$. On receiving this message, $\mathscr{C}$ randomly chooses a bit $d \in \{0, 1\}$, generates the aggregate signcryption C^* on $((m^*_{i,d}, \mathrm{ID}^*_i)_{i=1,\ldots,n}, \mathrm{ID}^*_j)$, and then sends C^* to $\mathscr{A}_1$. After that, $\mathscr{A}_1$ adaptively queries the same oracles as the *Training* phase.

(iv) *Guessing*: a bit d' is outputted by $\mathscr{A}_1$.

$\mathscr{A}_1$ is considered to win this game iff

(1) $d' = d$, where d and d' are defined as above;

(2) The oracle Partial-Private-Key-Extraction(ID_j) has never been queried;

(3) The oracle Designcryption($C^*, (\mathrm{ID}'_i)_{i=1,\ldots,n}, \mathrm{ID}_j$) has never been queried, where there exists $i \in \{1, \ldots, n\}$ such that $\mathrm{ID}^*_i = \mathrm{ID}'_i$.

$\mathscr{A}_1$'s advantage to win this game is defined as $\mathrm{Adv}^{\text{IND-CCA-II}}_{\mathscr{A}_1} = |2\Pr[d = d'] - 1|$.

Game 4. This game is played by a challenger $\mathscr{C}$ and a Type II adversary $\mathscr{A}_2$.

(i) *Initializing*: this phase is the same as the first phase in Game 3, while $\mathscr{C}$ sends $(params, msk)$ to $\mathscr{A}_2$.

(ii) *Training*: in this phase, $\mathscr{A}_2$ queries the same oracles (except the Public-Key-Replacement oracle) and receives the same responses as the second phase in Game 3.

(iii) *Guess*: this phase is the same as the third phase in Game 3, where a bit d' is outputted by $\mathscr{A}_2$.

$\mathscr{A}_2$ is considered to win this game iff

(1) $d' = d$, where d and d' are defined as above;

(2) The oracle Secret-Value-Extraction(ID_j) has never been queried;

(3) The oracle Designcryption($C^*, (\mathrm{ID}'_i)_{i=1,\ldots,n}, \mathrm{ID}_j$) has never been queried, where there exists $i \in \{1, \ldots, n\}$ such that $\mathrm{ID}^*_i = \mathrm{ID}'_i$.

$\mathscr{A}_2$'s advantage to win this game is defined as $\mathrm{Adv}^{\text{IND-CCA-II}}_{\mathscr{A}_2} = |2\Pr[d = d'] - 1|$.

Definition 5. A CL-A-SC scheme is considered to be secure against the adaptively chosen ciphertext attacks if there is no adversary of Type I or Type II has a nonnegligible advantage to win Game 3 or Game 4, respectively.

Game 6. This game is played by a challenger $\mathscr{C}$ and a Type I adversary $\mathscr{A}_1$.

(i) *Initializing*: this phase is the same as the first phase in Game 3.

(ii) *Training*: in this phase, $\mathscr{A}_1$ queries the same oracles and receives the same responses as the second phase of Game 3.

(iii) *Forgery*: $\mathscr{A}_1$ sends a forged aggregate signcryption C^* on $\{(m^*_i, \mathrm{ID}^*_i)_{i=1,\ldots,n}, \mathrm{ID}^*_j\}$ to $\mathscr{C}$, where $(\mathrm{ID}^*_i)_{i=1,\ldots,n}$ and ID^*_j are considered to be identity of the senders and the receiver, respectively.

$\mathscr{A}_1$ is considered to win this game iff

(1) The output of the execution of Aggregate-Verification algorithm on $(C^*, (\mathrm{ID}^*_i)_{i=1,\ldots,n}, \mathrm{ID}^*_j)$ is true;

(2) There exists $i \in \{1, \ldots, n\}$ such that the Signcryption($m^*_i, \mathrm{ID}^*_i, \mathrm{ID}^*_j$) oracle or Partial-Private-Key-Extraction(ID^*_i) oracle has not been queried.

Game 7. This game is played by a challenger $\mathscr{C}$ and a Type II adversary $\mathscr{A}_2$.

(i) *Initializing*: this phase is the same as the first phase in Game 4.

(ii) *Training*: this phase is the same as the second phase in Game 4.

(iii) *Forgery*: this phase is the same as the third phase in Game 6.

$\mathscr{A}_2$ is considered to win this game iff

(1) The output of the execution of Aggregate-Verification algorithm on $(C^*, (\mathrm{ID}^*_i)_{i=1,\ldots,n}, \mathrm{ID}^*_j)$ is true;

(2) There exists $i \in \{1, \ldots, n\}$ such that the Signcryption($m^*_i, \mathrm{ID}^*_i, \mathrm{ID}^*_j$) oracle or Secret-Value-Extraction(ID^*_i) oracle has not been queried.

Definition 8. A CL-A-SC is considered to be existentially unforgeable against the adaptively chosen-message attack if there is no adversary of Type I or Type II has a nonnegligible advantage to win Game 6 or Game 7, respectively.

2.4. Objectives. The design goals of the proposed protocol are defined as follows:

(1) Data confidentiality and integrity: it is desirable to secure the transmitted data from revealing the sensitive information about the source mobile sensor. Besides, it is required to ensure the data has not been tampered [23].

(2) Mutual authentication: it is desirable that the RSU and the mobile sensor are allowed to authenticate each other [24].

(3) Anonymity: it is desirable to hide the real identity of the mobile sensor during the transmission [25, 26].

(4) Key escrow resilience: it is desirable that the adversary is unable to obtain the full private key of any mobile sensor even if CC has been compromised [27].

3. Cryptanalysis of Basudan et al.'s CL-A-SC Scheme

In this section, Basudan et al.'s CL-A-SC scheme is briefly reviewed. After that, their scheme is demonstrated to be insecure against the public-key-replacement attack.

3.1. Notations. To ensure the consistency, the notations are defined in the Symbols. Concretely, each sensor S_i is able to generate a real-time message $m_i = \{T_{i,j}, L_j, \mathrm{Sig}_i\}$ when sensing the road condition RC_j. After that, S_i generates signcryption on m_i to construct the road condition report $\mathrm{RCR}_{i,j}$ and then sends $\mathrm{RCR}_{i,j}$ to the nearest RSU.

3.2. Review of Basudan et al.'s CL-A-SC Scheme. The CL-A-SC scheme in the protocol of [20] consists of the following algorithms:

(i) Setup: let $\mathscr{G}$ be an additive cyclic group of prime order q, $\mathscr{G}_T$ be a multiplicative cyclic group of the same order, $\hat{e} : \mathscr{G} \times \mathscr{G} \to \mathscr{G}_T$ be an admissible bilinear map, and $P \in \mathscr{G}$ denote a generator of $\mathscr{G}$. Let H_1, H_2, H_3, and H_4 be four cryptographic hash functions such that $H_1 : \{0,1\}^* \to Z^*_q$, $H_2 : \{0,1\}^* \times \{0,1\}^* \times \mathscr{G}^5 \to \{0,1\}^n$, $H_3 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^n \times \mathscr{G}^5 \to \mathscr{G}$, and $H_4 : Z^*_q \to \mathscr{G}$, where n is assumed to be the bit length of messages. Randomly choose $msk \in Z^*_q$ as the master private key and calculates $mpk = mskP$. The public parameters $params = \{\mathscr{G}, \mathscr{G}_T, \hat{e}, P, q, mpk, H_1, H_2, H_3, H_4\}$.

(ii) Key-Generation:

(1) For i ranges from 1 to n, each mobile sensor S_i randomly chooses $usk_{i,1} \in_R Z^*_q$ and calculates $upk_{i,1} = usk_{i,1} \cdot P$, $h_i = H_1(\mathrm{ID}_i)$. After that, S_i sends ID_i to CC.

(2) Upon receiving ID_i from S_i, CC randomly chooses $y_i \in_R Z^*_q$ and calculates $upk_{i,2} = y_i \cdot P$, $usk_{i,2} = y_i + H_1(\mathrm{ID}_i)msk$. After that, CC sends $\{upk_{i,2}, usk_{i,2}\}$ to S_i.

(3) Upon receiving $\{upk_{i,2}, usk_{i,2}\}$ from CC, S_i checks if $usk_{i,2}P = upk_{i,2} + h_i \cdot mpk$. If the verification holds, $(upk_{i,1}, upk_{i,2})$ and $(usk_{i,1}, usk_{i,2})$ are set to be the full public key and full private key of S_i, respectively.

(iii) Signcryption: the RSU with identity ID_R is assumed to be the message receiver. For i ranges from 1 to n, S_i randomly chooses $r_i \in_R Z_q^*$ and calculates $v_{i,1} = r_i \cdot upk_{R,1}$, $v_{i,2} = r_i(upk_{R,2} + h_R \cdot mpk)$, $f_{i,1} = r_i \cdot P$, $h_{i,1} = H_2(\mathrm{ID}_R, \Delta, upk_{R,1}, upk_{R,2}, f_{i,1}, v_{i,1}, v_{i,2})$, $f_{i,2} = h_{i,1} \oplus m_i$, $h_{i,2} = H_3(\mathrm{ID}_R, \Delta, h_i, f_{i,2}, upk_{R,1}, upk_{R,2}, f_{i,1}, upk_{i,1}, upk_{i,2})$, $h_{i,3} = H_4(\Delta)$, and $f_{i,3} = usk_{i,2} \cdot h_{i,3} + r_i \cdot h_{i,2} + usk_{i,1} \cdot h_{i,3}$, where Δ is the state information and $h_R = H_1(\mathrm{ID}_R)$. After that, S_i constructs $C_i = \{f_{i,1}, f_{i,2}, f_{i,3}\}$, $\mathrm{RCR}_{i,j} = \{h_i, C_i\}$ and sends $\mathrm{RCR}_{i,j}$ to the RSU with identity ID_R.

(iv) Aggregate: upon receiving $(\mathrm{RCR}_{i,j})_{i=1,\ldots,n}$, the RSU with identity ID_R calculates $F_{R,3} = \sum_{i=1}^{n} f_{i,3}$.

(v) Aggregate-Verification: for i ranges from 1 to n, RSU calculates $h'_{i,2} = H_3(\mathrm{ID}_R, \Delta, h_i, f_{i,2}, upk_{R,1}, upk_{R,2}, f_{i,1}, upk_{i,1}, upk_{i,2})$ and $h'_{i,3} = H_4(\Delta)$, where $h_i = H_1(\mathrm{ID}_i)$. RSU checks if $\widehat{e}(F_{R,3}, P) = \widehat{e}(\sum_{i=1}^{n}(upk_{i,2} + h_i \cdot mpk), h'_{i,3}) \prod_{i=1}^{n} \widehat{e}(f_{i,1}, h'_{i,2}) \widehat{e}(\sum_{i=1}^{n} upk_{i,1}, h'_{i,3})$.

(vi) Designcryption: if the verification in the Aggregate-Verification algorithm holds, RSU calculates $v'_{i,1} = usk_{R,1} \cdot f_{i,1}$, $v'_{i,2} = usk_{R,2} \cdot f_{i,1}$, $h'_{i,1} = H_2(\mathrm{ID}_R, \Delta, upk_{R,1}, upk_{R,2}, f_{i,1}, v'_{i,1}, v'_{i,2})$, and $m'_i = f_{i,2} \oplus h'_{i,1}$ for i ranges from 1 to n.

3.3. Forgery Attack against Basudan et al.'s CL-A-SC Scheme. Basudan et al. [20] claimed that their CL-A-SC scheme proved to be able to achieve indistinguishability and unforgeability against the Type I and Type II adversary. However, the adversary $\mathscr{A}_1$ of Type I is able to forge signcryption on any message m^* by launching a public-key-replacement attack, which is described as follows:

(i) Public-Key-Replacement: given a mobile sensor S_i, $\mathscr{A}_1$ randomly chooses ${usk_{i,1}}^*, y_i^* \in Z_q^*$ and calculates ${upk_{i,1}}^* = {usk_{i,1}}^* P$, ${upk_{i,2}}^* = y_i^* P - h_i \cdot mpk$, where $h_i = H_1(\mathrm{ID}_i)$. After that, $({upk_{i,1}}^*, {upk_{i,2}}^*)$ is set to be the full public key of S_i.

(ii) Signature-Forgery: $\mathscr{A}_1$ randomly chooses $r_i^* \in_R Z_q^*$ and calculates $f_{i,1}^* = r_i^* P$, $v_{i,1}^* = r_i^* upk_{R,1}$, $v_{i,2}^* = r_i^*(upk_{R,2} + h_R \cdot mpk)$, $h_{i,1}^* = H_2(\mathrm{ID}_R, \Delta, upk_{R,1}, upk_{R,2}, f_{i,1}^*, v_{i,1}^*, v_{i,2}^*)$, $f_{i,2}^* = h_{i,1}^* \oplus m_i^*$, $h_{i,2}^* = H_3(\mathrm{ID}_R, \Delta, h_i, f_{i,2}^*, upk_{R,1}, upk_{R,2}, f_{i,1}^*, {upk_{i,1}}^*, {upk_{i,2}}^*)$, $h_{i,3} = H_4(\Delta)$, and $f_{i,3}^* = y_i^* h_{i,3} + r_i^* h_{i,2}^* + {usk_{i,1}}^* h_{i,3}$, where m_i^* is forged by $\mathscr{A}_1$ under the state information Δ and $h_R = H_1(\mathrm{ID}_R)$. After that, $\mathscr{A}_1$ constructs $C_i^* = \{f_{i,1}^*, f_{i,2}^*, f_{i,3}^*\}$, $\mathrm{RCR}_{i,j}^* = \{h_i, C_i^*\}$ and sends $\mathrm{RCR}_{i,j}^*$ to the RSU with identity ID_R.

(iii) Aggregate: the RSU calculates $F_{R,3} = \sum_{j=1}^{n,j\neq i} f_{j,3} + f_{i,3}^*$.

(iv) Aggregate-Verification: for j ranges from 1 to n, the RSU calculates $h'_{j,2} = H_3(\mathrm{ID}_R, \Delta, h_j, f_{j,2}, upk_{R,1}, upk_{R,2}, f_{j,1}, upk_{j,1}, upk_{j,2})$ and $h'_{j,3} = H_4(\Delta)$, where $h_j = H_1(\mathrm{ID}_j)$. After that, the RSU checks if $\widehat{e}(F_{R,3}, P) = \widehat{e}(\sum_{j=1}^{n}(upk_{j,2} + h_j \cdot mpk), h'_{j,3}) \prod_{j=1}^{n} \widehat{e}(f_{j,1}, h'_{j,2}) \widehat{e}(\sum_{j=1}^{n} upk_{j,1}, h'_{j,3})$.

The correctness of C_i^* can be easily verified since

$$
\begin{aligned}
\widehat{e}(F_{R,3}, P) &= \widehat{e}\left(\sum_{j=1}^{n} f_{j,3}, P\right) \\
&= \widehat{e}\left(\sum_{j=1}^{n}\left(usk_{j,2}H_4(\Delta) + r_j h_{j,2} + usk_{j,1}H_4(\Delta)\right), P\right) \\
&= \widehat{e}\left(\sum_{j=1}^{n,j\neq i}\left(usk_{j,2} + y_i^*\right)H_4(\Delta), P\right) \times \widehat{e}\left(\sum_{j=1}^{n,j\neq i} r_j h'_{j,2}\right. \\
&\left. + r_i^* h'_{i,2}, P\right) \times \widehat{e}\left(\left(\sum_{j=1}^{n,j\neq i} usk_{j,1} + {usk_{i,1}}^*\right)H_4(\Delta),\right. \\
&\left. P\right) = \widehat{e}\left(\sum_{j=1}^{n,j\neq i}\left(upk_{j,2} + h_j \cdot mpk\right) + upk_{i,2}^* + h_i\right. \\
&\left. \cdot mpk, H_4(\Delta)\right) \times \prod_{j=1}^{n,j\neq i} \widehat{e}\left(f_{j,1}, h'_{j,2}\right) \times \widehat{e}\left(f_{i,1}^*, h'_{i,2}\right) \\
&\times \widehat{e}\left(\sum_{j=1}^{n,j\neq i} upk_{j,1} + {upk_{i,1}}^*, H_4(\Delta)\right) \\
&= \widehat{e}\left(\sum_{j=1}^{n}\left(upk_{j,2} + h_j \cdot mpk\right), h'_{j,3}\right) \\
&\times \prod_{j=1}^{n} \widehat{e}\left(f_{j,1}, h'_{j,2}\right) \times \widehat{e}\left(\sum_{j=1}^{n} upk_{j,1}, h'_{j,3}\right).
\end{aligned} \tag{1}
$$

Thus, the verification holds. The message m^* is recovered by the RSU according to the specification of Designcryption algorithm.

Remark 9. The fundamental flaw of Basudan et al.'s CL-A-SC scheme against this forgery attack is due to the unreasonable position of the value $H_1(\mathrm{ID}_i) \cdot msk$. As described above, $\mathscr{A}_1$ is allowed to generate ${upk_{i,2}}^* = y_i^* P - h_i \cdot mpk$ to replace S_i's public key. According to the specification of the protocol in [20], $usk_{i,2}P = upk_{i,2} + H_1(\mathrm{ID}_i)mpk$, and thus $usk_{i,2}P = y_i^* P$. $\mathscr{A}_1$ calculates $f_{i,3}^* = y_i^* h_{i,3}^* + r_i^* h_{i,2}^* + {usk_{i,1}}^* h_{i,3}^*$ and then successfully forges the signcryption C_i^*. It is noted that this type of adversary has not been mentioned in their security proof. Hence, the proof fails.

4. Our Proposed Protocol

In this section, a concrete CL-A-SC scheme is proposed, which is the building block of our data aggregation protocol.

4.1. The Proposed CL-A-SC Scheme. This scheme consists of the following algorithms:

(i) CL-A-SC.Setup: let p and q be two large primes such that q divides $p-1$, $\mathscr{E}$ be an elliptic curve over a finite field $\mathbb{F}_p$, and $\mathscr{G}$ be an additive cyclic group formed by $\mathscr{E}$ with the point addition law. Let P be a generator of $\mathscr{G}$ and $H_1 : \{0,1\}^* \times \mathscr{G}^3 \to Z_q^*$, $H_2 : \mathscr{G} \times \{0,1\}^* \times \{0,1\}^n \to Z_q^*$, $H_3 : \mathscr{G} \times \mathscr{G} \to Z_q^*$ be three cryptographic hash functions. Randomly choose $msk \in_R Z_q^*$ as the master private key and calculates $mpk = mskP$. The system parameter $params = (p, q, P, mpk, H_1, H_2, H_3)$.

(ii) CL-A-SC.Key-Generation:

(1) The user randomly chooses $usk_{i,1} \in_R Z_q^*$ and calculates $upk_{i,1} = usk_{i,1} \cdot P$. After that, the user sends $\{\mathrm{ID}_i, upk_{i,1}\}$ to KGC.

(2) KGC randomly chooses $y_i \in_R Z_q^*$ and calculates $upk_{i,2} = y_i \cdot P$, $usk_{i,2} = y_i + H_1(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)msk$. After that, KGC sends $\{upk_{i,2}, usk_{i,2}\}$ to the user with identity ID_i.

(3) The user with identity ID_i checks if $upk_{i,2} + h_i \cdot mpk = usk_{i,2}P$, where $h_i = H_1(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)$. If the verification holds, $(upk_{i,1}, upk_{i,2})$ and $(usk_{i,1}, usk_{i,2})$ are set to be the full public key and full private key of the user, respectively.

(iii) CL-A-SC.Signcryption: the user ID_i randomly chooses $r_i \in_R Z_q^*$ and calculates $v_{i,1} = r_i \cdot upk_{R,1}$, $f_{i,1} = r_i \cdot P$, $f_{i,2} = r_i \cdot (usk_{i,1} + usk_{i,2} + f_{i,3})^{-1}$, and $c_i = H_3(v_{i,1}, v_{i,2}) \oplus m_i$, where $v_{i,2} = r_i(upk_{R,2} + h_R \cdot mpk)$ and $f_{i,3} = H_2(f_{i,1}, \mathrm{ID}_i, m_i) + H_2(v_{i,1}, \mathrm{ID}_R, m_i)$, where $h_R = H_1(\mathrm{ID}_R, upk_{R,1}, upk_{R,2}, mpk)$. After that, the user ID_i sends the ciphertext $C_i = \{f_{i,1}, f_{i,2}, f_{i,3}, c_i\}$ to the user with identity ID_R.

(iv) CL-A-SC.Aggregate: upon receiving $(C_i)_{i=1,\ldots,n}$, the user with identity ID_R calculates $F_{R,3} = \sum_{i=1}^{n} f_{i,3}$.

(v) CL-A-SC.Aggregate-Verification: for i ranges from 1 to n, the user with identity ID_R calculates $h_i' = H_1(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)$. After that, this user checks if $\sum_{i=1}^{n} upk_{i,1} + \sum_{i=1}^{n} upk_{i,2} + \sum_{i=1}^{n} h_i' \cdot mpk + F_{R,3} \cdot P = \sum_{i=1}^{n} f_{i,1} \cdot f_{i,2}^{-1}$.

(vi) CL-A-SC.Designcryption: if the verification in the Aggregate-Verification algorithm holds, the user with identity ID_R calculates $v_{i,1}' = usk_{R,1} \cdot f_{i,1}$, $v_{i,2}' = usk_{R,2} \cdot f_{i,1}$, and $m_i = H_3(v_{i,1}', v_{i,2}') \oplus c_i$ for i ranges from 1 to n.

4.2. The Data Aggregation Protocol. In this part, our data aggregation protocol is proposed, which involves the CC, RSU, and mobile sensors. The suggested protocol is comprised of four phases: system initialization, data generation and transmission, aggregate verification, and data retrieval.

4.2.1. System Initialization. In this phase, CC performs the CL-A-SC.Setup algorithm to initialize the system. The system parameter $params = (p, q, P, mpk, H_1, H_2, H_3)$. After that, the mobile sensors and the RSUs are allowed to register to CC by performing the following steps:

(1) For i ranges from 1 to n, each mobile sensor S_i randomly chooses $usk_{i,1} \in_R Z_q^*$ and calculates $upk_{i,1} = usk_{i,1} \cdot P$. After that, S_i sends $\{\mathrm{ID}_i, upk_{i,1}\}$ to CC.

(2) Upon receiving $\{\mathrm{ID}_i, upk_{i,1}\}$ from S_i, CC randomly chooses $y_i \in_R Z_q^*$ and calculates $upk_{i,2} = y_i \cdot P$, $usk_{i,2} = y_i + H_1(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)msk$. After that, CC sends $\{upk_{i,2}, usk_{i,2}\}$ to S_i.

(3) Upon receiving $\{upk_{i,2}, usk_{i,2}\}$ from CC, S_i checks if $upk_{i,2} + h_i \cdot mpk = usk_{i,2}P$, where $h_i = H_1(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)$. If the verification holds, $(upk_{i,1}, upk_{i,2})$ and $(usk_{i,1}, usk_{i,2})$ are set to be the full public key and full private key of S_i, respectively.

It is worth noting that the format of the road condition report is defined by CC in this phase. Concretely, each mobile sensor S_i is able to generate $m_i = \{T_{i,j}, L_j, \mathrm{Sig}_i\}$ when sensing the road condition RC_j, where $T_{i,j}$ is the time when S_i sensed RC_j, L_j is the location where RC_j occurred, and Sig_i is the action signal about RC_j. After that, S_i generates signcryption on m_i to construct the road condition report $\mathrm{RCR}_{i,j}$.

4.2.2. Data Generation and Transmission. In this phase, S_i is allowed to generate signcryption on m_i to construct $\mathrm{RCR}_{i,j}$. After that, $\mathrm{RCR}_{i,j}$ is sent to the nearest RSU. The identity of this RSU is assumed to be ID_R. This phase consists of the following steps:

(1) S_i randomly chooses $r_i \in_R Z_q^*$ and calculates $v_{i,1} = r_i \cdot upk_{R,1}$, $f_{i,1} = r_i \cdot P$, $f_{i,2} = r_i \cdot (usk_{i,1} + usk_{i,2} + f_{i,3})^{-1}$, and $c_i = H_3(v_{i,1}, v_{i,2}) \oplus m_i$, where $v_{i,2} = r_i(upk_{R,2} + h_R \cdot mpk)$, $f_{i,3} = H_2(f_{i,1}, \mathrm{ID}_i, m_i) + H_2(v_{i,1}, \mathrm{ID}_R, m_i)$, and $h_R = H_1(\mathrm{ID}_R, upk_{R,1}, upk_{R,2}, mpk)$.

(2) S_i sends $\mathrm{RCR}_{i,j} = \{f_{i,1}, f_{i,2}, f_{i,3}, c_i\}$ to the RSU with identity ID_R.

To protect private information of mobile sensors, the real identity of each S_i cannot be retrieved from $\mathrm{RCR}_{i,j}$. In this way, the anonymity of mobile sensors is preserved.

4.2.3. Aggregate Verification. Upon receiving the reports $(\mathrm{RCR}_{i,j})_{i=1,\ldots,n}$ from the sensors $(S_i)_{i=1,\ldots,n}$ on a road condition RC_j, the RSU is allowed to aggregate the ciphertexts and then verify the authenticity of the aggregate data. The identity of this RSU is assumed to be ID_R. The aggregation and verification procedures are carried out by performing the following steps:

(1) The RSU calculates $F_{R,3} = \sum_{i=1}^{n} f_{i,3}$.

(2) For i ranges from 1 to n, the RSU calculates $h_i' = H_1(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)$. After that, this RSU checks if $\sum_{i=1}^{n} upk_{i,1} + \sum_{i=1}^{n} upk_{i,2} + \sum_{i=1}^{n} h_i' \cdot mpk + F_{R,3} \cdot P = \sum_{i=1}^{n} f_{i,1} \cdot f_{i,2}^{-1}$.

If the equation holds, this RSU accepts the received reports and executes the next phase. Otherwise, this RSU aborts these reports.

4.2.4. Data Retrieval. If the verification in the previous phase holds, the RSU retrieves $(m_i)_{i=1,\ldots,n}$ as follows:

(1) For i ranges from 1 to n, the RSU calculates $v_{i,1}' = usk_{R,1} \cdot f_{i,1}$ and $v_{i,2}' = usk_{R,2} \cdot f_{i,1}$.

(2) This RSU calculates $m_i = H_3(v_{i,1}', v_{i,2}') \oplus c_i$ for i ranges from 1 to n.

5. Analysis and Comparison

The correctness and security properties of the proposed protocol are analyzed in this section. After that, the comparison in terms of efficiency and security properties of the proposed protocol and the related works is presented.

5.1. Correctness Analysis. The correctness of the decryption procedure is presented as follows:

$$\begin{aligned} m_i &= H_3\left(v_{i,1}', v_{i,2}'\right) \oplus c_i \\ &= H_3\left(usk_{R,1} \cdot f_{i,1}, usk_{R,2} \cdot f_{i,1}\right) \oplus c_i \\ &= H_3\left(r_i \cdot upk_{R,1}, r_i\left(y_R + h_R \cdot msk\right) P\right) \oplus c_i \\ &= H_3\left(r_i \cdot upk_{R,1}, r_i\left(upk_{R,2} + h_R mpk\right)\right) \oplus c_i \\ &= H_3\left(v_{i,1}, v_{i,2}\right) \oplus c_i = m_i. \end{aligned} \tag{2}$$

The correctness of the verification procedure is presented as follows:

$$\begin{aligned} &\sum_{i=1}^{n} upk_{i,1} + \sum_{i=1}^{n} upk_{i,2} + \sum_{i=1}^{n} h_i' \cdot mpk + F_{R,3} \cdot P \\ &\quad = \sum_{i=1}^{n} usk_{i,1} \cdot P + \sum_{i=1}^{n} usk_{i,2} \cdot P + \sum_{i=1}^{n} f_{i,3} \cdot P \\ &\quad = \sum_{i=1}^{n} \left(usk_{i,1} + usk_{i,2} + f_{i,3}\right) P \\ &\quad = \sum_{i=1}^{n} \frac{r_i\left(usk_{i,1} + usk_{i,2} + f_{i,3}\right) P}{r_i} = \sum_{i=1}^{n} \frac{f_{i,1}}{f_{i,2}}. \end{aligned} \tag{3}$$

5.2. Security Proof. In this part, the security proof of the proposed protocol is given under the random oracle model [28].

Lemma 10. *The proposed protocol is indistinguishable against the chosen ciphertext attacks* (Ind-CCA-II) *of the* Type I *adversary* $\mathscr{A}_1$ *in the random oracle model under the* CDH *assumption.*

Proof. See Appendix A. □

Lemma 11. *The proposed protocol is indistinguishable against the chosen ciphertext attacks* (Ind-CCA-II) *of the* Type II *adversary* $\mathscr{A}_2$ *in the random oracle model under the* CDH *assumption.*

Proof. The proof of this lemma is omitted since it follows the proof of Lemma 10. □

Theorem 12. *The proposed protocol achieves* IND-CCA *security under the* CDH *assumption.*

Proof. Theorem 12 is derived directly from Lemmas 10 and 11. □

Lemma 13. *The proposed protocol is existentially unforgeable against adaptive chosen-message attacks* (EUF-CMA-II) *of the* Type I *adversary* $\mathscr{A}_1$ *in the random oracle model under the* ECDLP *assumption.*

Proof. See Appendix B. □

Lemma 14. *The proposed protocol is existentially unforgeable against adaptive chosen-message attacks* (EUF-CMA-II) *of the* Type II *adversary* $\mathscr{A}_2$ *in the random oracle model under the* ECDLP *assumption.*

Proof. The proof of this lemma is omitted since it follows the proof of Lemma 13. □

Theorem 15. *The proposed protocol achieves* EUF-CMA *security under the* ECDLP *assumption.*

Proof. Theorem 15 is derived directly from Lemmas 13 and 14. □

5.3. Security Strength

(1) Data confidentiality and integrity: each c_i is calculated as $c_i = H_3(v_{i,1}, v_{i,2}) \oplus m_i$, where $v_{i,1}$, $v_{i,2}$ can only be recovered by the RSU. The confidentiality of the data is proved in Theorem 12. Moreover, the RSU is able to decrypt and verify the received data. Thus, the integrity of the data is ensured.

(2) Mutual authentication: each mobile sensor S_i authenticates itself by sending $\mathrm{RCR}_{i,j}$ to the RSU. Only the RSU which keeps the private key $(usk_{R,1}, usk_{R,2})$ can recover m_i. Besides, the RSU authenticates each sensor by verifying the received data. The unforgeability of the data is proved in Theorem 15.

(3) Anonymity: according to the specification of the proposed protocol, the real identity of each mobile sensor S_i cannot be retrieved from the ciphertext. Thus, the proposed protocol achieves anonymity.

(4) Key escrow resilience: the proposed protocol is designed under the certificateless cryptosystem. Specifically, CC is only allowed to issue the partial private key $usk_{i,2}$ for each mobile sensor S_i. The adversary

Table 1: Comparison of security properties.

Protocols	DCI	MA	AN	KER	TAR
Basudan et al.'s protocol [20]	√	–	√	√	√
Xiong and Qin's protocol [21]	√	√	√	√	√
Our protocol	√	√	√	√	√

is unable to obtain the full private key $(usk_{i,1}, usk_{i,2})$ of S_i even if CC is compromised. Thus, this protocol achieves key escrow resilience.

5.4. Comparison. The comparison of the security properties is presented in Table 1, which includes data confidentiality and integrity (DCI), mutual authentication (MA), anonymity (AN), key escrow resilience (KER), and timing attack resilience (TAR). The timing attack is considered as a kind of side-channel attack [29]. In the execution of the cryptographic protocols, variations of the executing timing can leak some information if sensitive data is involved. By measuring the time which the sensors take to perform the cryptographic operations, the adversary is able to obtain some secret parameters of the sensors. It is required to reduce the computation overhead of the sensors. According to this comparison, it can be concluded that the proposed protocol is able to achieve all of the security goals, while Basudan et al.'s [20] protocol fails to achieve mutual authentication.

The comparison of the communication overhead is presented in Figure 2. To get an intuitive comparison of the efficiency, the practical performance of the protocols is presented in Figures 3–5, respectively. To ensure the consistency, the 80-bit security level (RSA-1024 bit, ECC-160 bit equivalent) is adopted for both protocols. The implementation is based on a common hardware platform with Intel Core i5-4460 CPU at 3.2 GHz using the PBC library [30]. According to this comparison, it can be concluded that the proposed protocol outperforms the related works in terms of communication and computation overhead.

6. Conclusion

The security and privacy concerns are essential and challenging issues in road surface condition monitoring system. In this paper, the security flaw of a certificateless data aggregation protocol in [20] for monitoring system is pointed out. After that, a light-weight and anonymous data aggregation protocol is introduced, which is constructed by combining a CL-A-SC scheme and the fog computing architecture. The proposed protocol is proved secure under the random oracle model and achieves desirable security properties including data confidentiality, mutual authentication, anonymity and key escrow resilience. Besides, an experimental simulation of the proposed protocol and the protocol in [20] is presented. According to the comparison results, the proposed protocol is efficient and more practical for the road surface condition monitoring system.

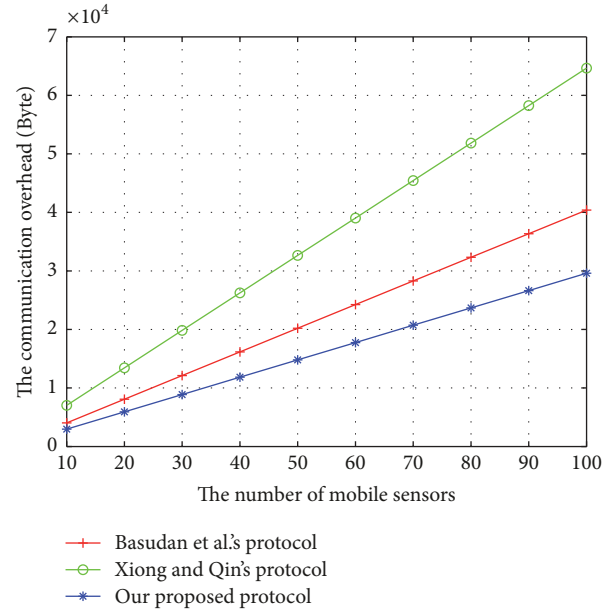

Figure 2: Comparison of communication overhead.

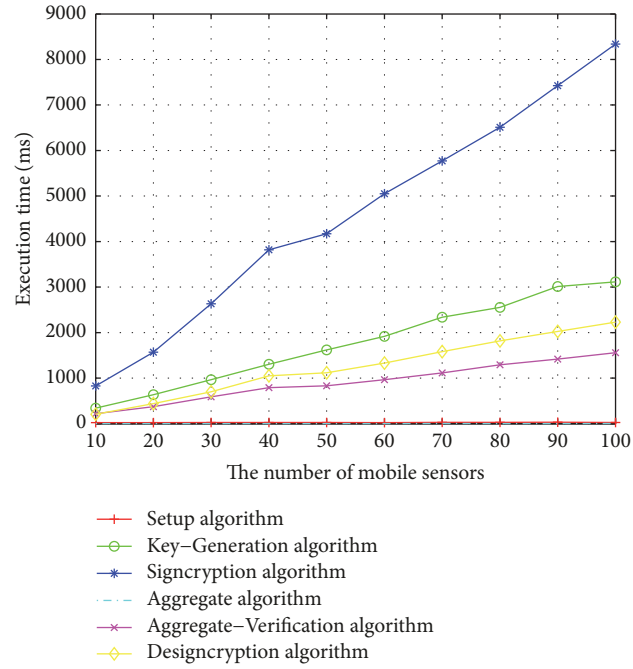

Figure 3: Computation overhead of Basudan et al.'s protocol.

Appendix

A. Proof of Lemma 10

Given an input $\langle P, aP, bP\rangle \in \mathcal{G}$ of the CDH assumption, the task of the challenger $\mathcal{C}$ is to calculate $abP \in \mathcal{G}$ with the support of Type I adversary $\mathcal{A}_1$. Assume $\mathcal{A}_1$ is able to break the Ind-CCA-II security with the advantage ϵ.

A.1. Setup. $\mathcal{C}$ randomly chooses $b \in_R Z_q^*$ and sets $mpk = bP$ and the public parameters $params = \{p, q, P, mpk, H_1, H_2, H_3\}$, where H_1, H_2, and H_3 are considered to be random

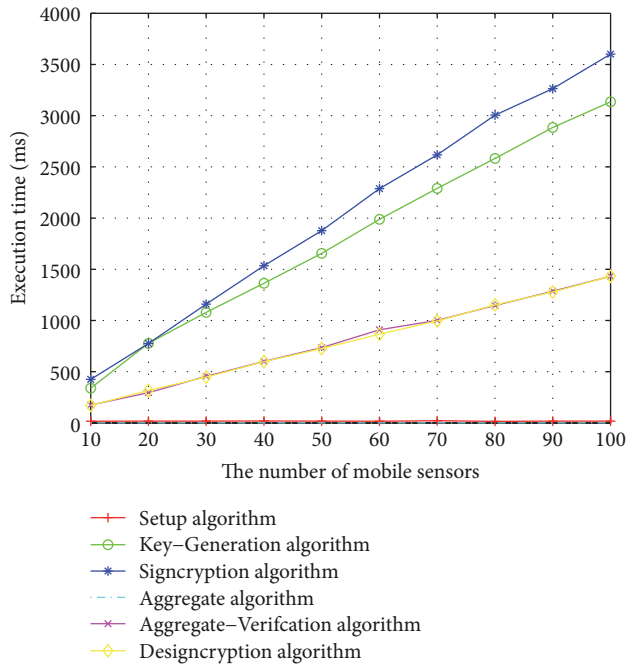

Figure 4: Computation overhead of the proposed protocol.

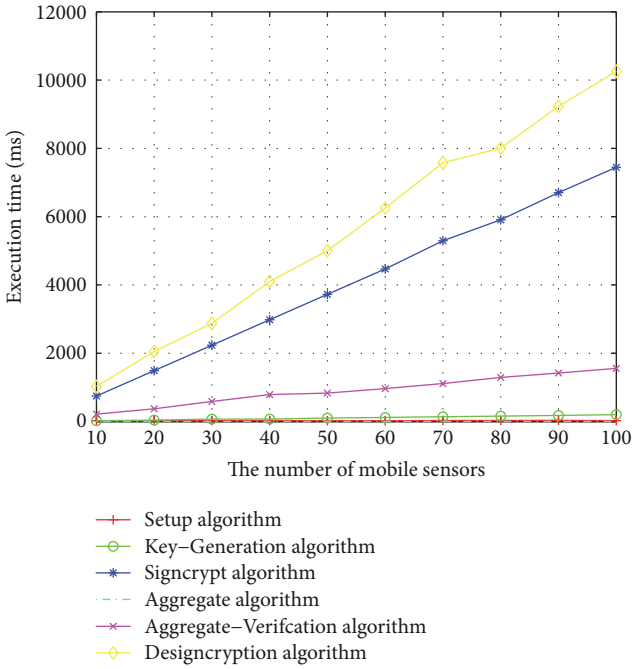

Figure 5: Computation overhead of the Xiong and Qin's protocol.

oracles. Let q_{H_1} denote the maximum number of queries on H_1. $\mathscr{C}$ randomly chooses $I \in [1, q_{H_1}]$.

A.2. Training. $\mathscr{C}$ and $\mathscr{A}$ interactively play the game as follows:

(i) $H_1(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)$ query: an initially empty list L_1 associated with this query is maintained by $\mathscr{C}$. If there is a tuple $\langle \mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk, h_{1,i}\rangle$ in L_1, $\mathscr{C}$ returns $h_{1,i}$ to $\mathscr{A}_1$ as the response of the input $(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)$. Otherwise, $\mathscr{C}$ randomly chooses $h_{1,i} \in_R Z_q^*$ and adds the tuple $\langle \mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk, h_{1,i}\rangle$ into L_1. After that, $\mathscr{C}$ returns $h_{1,i}$ to $\mathscr{A}_1$.

(ii) $H_2(w_i, \mathrm{ID}_i, m_i)$ query: let q_{H_2} denote the maximum number of queries on H_2. An initially empty list L_2 associated with this query is maintained by $\mathscr{C}$. If there is a tuple $\langle w_i, \mathrm{ID}_i, m_i, h_{2,i}\rangle$ in L_2, $\mathscr{C}$ returns $h_{2,i}$ to $\mathscr{A}_1$ as the response of the input $(w_i, \mathrm{ID}_i, m_i)$. Otherwise, $\mathscr{C}$ randomly chooses $h_{2,i} \in_R Z_q^*$ and adds the tuple $\langle w_i, \mathrm{ID}_i, m_i, h_{2,i}\rangle$ into L_2. After that, $\mathscr{C}$ returns $h_{2,i}$ to $\mathscr{A}_1$.

(iii) $H_3(v_{i,1}, v_{i,2})$ query: let q_{H_3} denote the maximum number of queries on H_3. An initially empty list L_3 associated with this query is maintained by $\mathscr{C}$. If there is a tuple $\langle v_{i,1}, v_{i,2}, h_{3,i}\rangle$ in L_3, $\mathscr{C}$ returns $h_{3,i}$ to $\mathscr{A}_1$ as the response of the input $(v_{i,1}, v_{i,2})$. Otherwise, $\mathscr{C}$ randomly chooses $h_{3,i} \in_R Z_q^*$ and adds the tuple $\langle v_{i,1}, v_{i,2}, h_{3,i}\rangle$ into L_3. After that, $\mathscr{C}$ returns $h_{3,i}$ to $\mathscr{A}_1$.

(iv) Secret-Value-Extraction(ID_i): let q_S denote the maximum number of queries on this oracle. An initially empty list L_S associated with this query is maintained by $\mathscr{C}$. Upon receiving this query on ID_i such that $\mathrm{ID}_i = \mathrm{ID}_I$, $\mathscr{C}$ aborts this simulation. Otherwise, $\mathscr{C}$ performs as follows: If there is a tuple $\langle \mathrm{ID}_i, usk_{i,1}, upk_{i,1}\rangle$ in L_S, $\mathscr{C}$ returns $usk_{i,1}$ to $\mathscr{A}_1$ as the response of the input (ID_i). Otherwise, $\mathscr{C}$ randomly chooses $usk_{i,1} \in_R Z_q^*$ and calculates $upk_{i,1} = usk_{i,1}P$. After that, $\mathscr{C}$ adds the tuple $\langle \mathrm{ID}_i, usk_{i,1}, upk_{i,1}\rangle$ into L_S and returns $usk_{i,1}$ to $\mathscr{A}_1$.

(v) Partial-Private-Key-Extraction(ID_i): an initially empty list L_P associated with this query is maintained by $\mathscr{C}$. If $\mathrm{ID}_i = \mathrm{ID}_I$, $\mathscr{C}$ randomly chooses $y_I \in_R Z_q^*$ and calculates $upk_{I,2} = y_I P$. After that, $\mathscr{C}$ adds the tuple $\langle \mathrm{ID}_I, upk_{I,1}, \perp, upk_{I,2}\rangle$ into L_P and returns $\perp$ to $\mathscr{A}_1$ as the response of the input (ID_i). Otherwise, $\mathscr{C}$ performs the following steps. If there is a tuple $\langle \mathrm{ID}_i, upk_{i,1}, usk_{i,2}, upk_{i,2}\rangle$ in L_P, $\mathscr{C}$ returns $usk_{i,2}$ to $\mathscr{A}_1$. Otherwise, $\mathscr{C}$ randomly chooses $usk_{i,2}, h_{1,i} \in_R Z_q^*$ and calculates $upk_{i,2} = usk_{i,2}P - h_{1,i} \cdot mpk$. After that, $\mathscr{C}$ adds the tuple $\langle \mathrm{ID}_i, upk_{i,1}, usk_{i,2}, upk_{i,2}\rangle$ into L_P and $\langle \mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk, h_{1,i}\rangle$ into L_1 and returns $usk_{i,2}$ to $\mathscr{A}_1$.

(vi) Public-Key-Extraction(ID_i): if there is a tuple $\langle \mathrm{ID}_i, upk_{i,1}, usk_{i,2}, upk_{i,2}\rangle$ in L_P, $\mathscr{C}$ returns $(upk_{i,1}, upk_{i,2})$ to $\mathscr{A}_1$ as the response of the input (ID_i). Otherwise, $\mathscr{C}$ queries the Partial-Private-Key-Extraction($\mathrm{ID}_i, upk_{i,1}$) and returns $(upk_{i,1}, upk_{i,2})$ to $\mathscr{A}_1$.

(vii) Public-Key-Replacement($\mathrm{ID}_i, upk'_{i,1}, upk'_{i,2}$): if there is a tuple $\langle \mathrm{ID}_i, upk_{i,1}, usk_{i,2}, upk_{i,2}\rangle$ in L_P, $\mathscr{C}$ also updates $\langle \mathrm{ID}_i, upk_{i,1}, usk_{i,2}, upk_{i,2}\rangle$ into $< \mathrm{ID}_i, upk'_{i,1}, \perp, upk'_{i,2} >$ in L_P.

(viii) Signcryption($m_i, \mathrm{ID}_i, \mathrm{ID}_j$): let q_{SC} denote the maximum number of queries on this oracle. ID_i, ID_j are

considered to be the identity of sender and receiver, respectively. This query is executed as follows:

(1) If $\mathrm{ID}_i \neq \mathrm{ID}_I$ and $\mathrm{ID}_i \neq \mathrm{ID}_j$, the Signcryption algorithm is executed by $\mathscr{C}$, who knows $(usk_{i,1}, usk_{i,2})$.

(2) Else if $\mathrm{ID}_i = \mathrm{ID}_I$ and $\mathrm{ID}_i \neq \mathrm{ID}_j$, $\mathscr{C}$ randomly chooses $f_{i,2}, f_{i,3} \in_R Z_q^*$, calculates $f_{i,1} = f_{i,2}(upk_{i,1} + upk_{i,2} + h_{1,i} \cdot mpk + f_{i,3}P)$, and sets $f_{i,3} = H_2(f_{i,1}, \mathrm{ID}_i, m_i) + H_2(usk_{i,1} f_{i,1}, \mathrm{ID}_j, m_i)$, where $h_{1,i}$ is obtained by either searching $\langle \mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk, h_{1,i} \rangle$ in L_1 or asking the $H_1(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)$ query. Note that, if such query has been responded with a different value before, $\mathscr{C}$ aborts this simulation. The tuples $\langle \mathrm{ID}_j, usk_{j,1}, upk_{j,1} \rangle$ and $\langle \mathrm{ID}_j, upk_{j,1}, usk_{j,2}, upk_{j,2} \rangle$ are outputted by searching L_S and L_P, respectively. $\mathscr{C}$ calculates $v_{i,1} = usk_{j,1} f_{i,1}$, $v_{i,2} = usk_{j,2} f_{i,1}$ and searches $\langle v_{i,1}, v_{i,2}, h_{3,i} \rangle$ in L_3. If there is no such tuple in L_3, $\mathscr{C}$ randomly chooses $h_{3,i} \in_R Z_q^*$ and adds the tuple $\langle v_{i,1}, v_{i,2}, h_{3,i} \rangle$ into L_3. After that, $\mathscr{C}$ calculates $m_i' = h_{3,i} \oplus m_i$ and returns $C_i = \{f_{i,1}, f_{i,2}, f_{i,3}, m_i'\}$ to $\mathscr{A}_1$.

(ix) Designcryption$(C_i, \mathrm{ID}_i, \mathrm{ID}_j)$: ID_i, ID_j are considered to be the identity of sender and receiver, respectively. This query is executed as follows:

(1) If $\mathrm{ID}_j \neq \mathrm{ID}_I$ and $\mathrm{ID}_j \neq \mathrm{ID}_i$, the Designcryption algorithm is executed by $\mathscr{C}$, who knows $(usk_{j,1}, usk_{j,2})$.

(2) If $\mathrm{ID}_j = \mathrm{ID}_I$ and $\mathrm{ID}_j \neq \mathrm{ID}_i$, $\mathscr{C}$ queries $H_1(\mathrm{ID}_i, upk_{i,1}, upk_{i,2}, mpk)$ and searches the tuple $\langle v_{i,1}, v_{i,2}, h_{3,i} \rangle$ in L_3 such that $m_i = h_{3,i} \oplus m_i'$. $\mathscr{C}$ checks if $H_2(f_{i,2}(upk_{i,1} + upk_{i,2} + h_{1,i} \cdot mpk + f_{i,3}P), \mathrm{ID}_i, m_i) + H_2(v_{i,1}, \mathrm{ID}_j, m_i) = f_{i,3}$. If the verification holds, $\mathscr{C}$ returns m_i to $\mathscr{A}_1$. Otherwise, $\mathscr{C}$ aborts this simulation.

A.3. Challenge. Eventually, $\mathscr{A}_1$ sends $S = \{(m_{i,0}^*, m_{i,1}^*, \mathrm{ID}_i^*)_{i=1,\ldots,n}, \mathrm{ID}_j^*\}$ to $\mathscr{C}$. It is required that there exists $i \in \{1, \ldots, n\}$ such that $\mathrm{ID}_i^* = \mathrm{ID}_I$. Thus, the solution of the CDH problem is calculated by $\mathscr{C}$ as follows:

(1) $\mathscr{C}$ randomly chooses $d \in \{0, 1\}$. For i ranges from 1 to n, $\mathscr{C}$ queries $H_1(\mathrm{ID}_i^*, upk_{i,1}^*, upk_{i,2}^*, mpk)$ to get $h_{1,i}^*$. After that, $\mathscr{C}$ randomly chooses $f_{i,2}^*, f_{i,3}^* \in_R Z_q^*$ and calculates $h_{1,j}^* = H_1(\mathrm{ID}_j^*, upk_{j,1}^*, upk_{j,2}^*, mpk)$, $f_{i,1}^* = f_{i,2}^*(upk_{i,1}^* + upk_{i,2}^* + h_{1,i}^* \cdot mpk + f_{i,3}^* P)$, where $f_{i,3}^*$ is defined as $f_{i,3}^* = H_2(f_{i,1}^*, \mathrm{ID}_i^*, m_{i,d}^*) + H_2(x_j^* f_{i,1}^*, \mathrm{ID}_j^*, m_{i,d}^*)$. $\mathscr{C}$ sets $f_{i,1}^* = aP$, $\eta^* = upk_{j,2}^* + h_{1,j}^* \cdot mpk$ and constructs $C_i^* = (f_{i,1}^*, f_{i,2}^*, f_{i,3}^*, m_{i,d}^*)$.

(2) $\mathscr{C}$ generates the aggregate signcryption C^* on $((m_{i,d}^*, \mathrm{ID}_i^*)_{i=1,\ldots,n}, \mathrm{ID}_j^*)$ and then sends C^* to $\mathscr{A}_1$.

$\mathscr{A}_1$ adaptively queries the same oracles as the Training phase. Note that, $\mathscr{A}_1$ is not allowed to query the Designcryption oracle on C^* with the receiver whose identity is ID_I.

A.4. Guess

$\mathscr{A}_1$ Returns d' to $\mathscr{C}$. If $d' = d$, $\mathscr{C}$ calculates the solution of the CDH instance as

$$\begin{aligned} abP &= \frac{v_{i,2}^* - y_j^* f_{i,1}^*}{h_{1,j}^*} = \frac{usk_{j,2}{}^* f_{i,1}^* - y_j^* f_{i,1}^*}{h_{1,j}^*} \\ &= \frac{aP\left(usk_{j,2}{}^* - y_j^*\right)}{h_{1,j}^*} = \frac{a \cdot msk \cdot h_{1,j}^* P}{h_{1,j}^*} \\ &= a \cdot mpk = abP. \end{aligned} \tag{A.1}$$

To solve the CDH problem, it is required that the following events are executed successfully by $\mathscr{C}$:

(i) Σ_1: any of $\mathscr{A}_1$'s Secret-Value-Extraction query is not aborted by $\mathscr{C}$.

(ii) Σ_2: any of $\mathscr{A}_1$'s Signcryption query is not aborted by $\mathscr{C}$.

(iii) Σ_3: there exists $i \in \{1, \ldots, n\}$ such that $\mathrm{ID}_i^* = \mathrm{ID}_I$ in S.

(iv) Σ_4: the C^* returned by $\mathscr{C}$ is valid.

The probability that $\mathscr{C}$ solves the CDH problem is denoted as $P(\Sigma_1\Sigma_2\Sigma_3\Sigma_4)$, which is decomposed as

$$\begin{aligned} P\left(\Sigma_1\Sigma_2\Sigma_3\Sigma_4\right) &= P\left(\Sigma_1\right) P\left(\Sigma_2 \mid \Sigma_1\right) P\left(\Sigma_3 \mid \Sigma_1\Sigma_2\right) \\ &\quad \cdot P\left(\Sigma_4 \mid \Sigma_1\Sigma_2\Sigma_3\right). \end{aligned} \tag{A.2}$$

Claim A.1. The probability that any of $\mathscr{A}_1$'s Secret-Value-Extraction query is not aborted by $\mathscr{C}$ is at least $(1 - 1/q_{H_1})^{q_S}$. Thus, $P(\Sigma_1) \geq (1 - 1/q_{H_1})^{q_S}$.

Proof. Upon receiving a Secret-Value-Extraction query, $\mathscr{C}$ aborts the simulation if the query is on $\mathrm{ID}_i = \mathrm{ID}_I$. Obviously, the probability that $\mathscr{C}$ does not abort a Secret-Value-Extraction query is $1 - 1/q_{H_1}$. Since $\mathscr{A}_1$ is able to ask at most q_S times of such queries, the probability of this event is as least $(1 - 1/q_{H_1})^{q_S}$. □

Claim A.2. The probability that any of $\mathscr{A}_1$'s Signcryption query is not aborted by $\mathscr{C}$ is at least $(1-(1/q_{H_1})(1-1/q_{H_1}))^{q_{SC}}$. Thus, $P(\Sigma_2 \mid \Sigma_1) \geq (1 - (1/q_{H_1})(1 - 1/q_{H_1}))^{q_{SC}}$.

Proof. Upon receiving a Signcryption query, $\mathscr{C}$ aborts the simulation if the included query on H_1 oracle has been responded with a different value before. Obviously, the probability that $\mathscr{C}$ does not abort a Signcryption query is $1-(1/q_{H_1})(1-1/q_{H_1})$. Since $\mathscr{A}_1$ is able to ask at most q_{SC} such queries, the probability of this event is as least $(1-(1/q_{H_1})(1-1/q_{H_1}))^{q_{SC}}$. □

Claim A.3. The probability that there exists $i \in \{1, \ldots, n\}$ such that $\mathrm{ID}_i^* = \mathrm{ID}_I$ in S is at least $1/q_{H_1}$.

Proof. If Σ_1, Σ_2 occur, $\mathscr{C}$ aborts the simulation unless there exists $\mathrm{ID}_i^* = \mathrm{ID}_I$ in S. Thus $P(\Sigma_3 \mid \Sigma_1\Sigma_2) \geq 1/q_{H_1}$. □

Claim A.4. The probability that the C^* is valid is at least ϵ. Thus, $P(\Sigma_4 \mid \Sigma_1\Sigma_2\Sigma_3) \geq \epsilon$.

Proof. Because $\mathscr{A}_1$ is able to break the Ind-CCA-II security with the advantage ϵ, $\mathscr{A}_1$ can check the validity of C^* with advantage ϵ. Thus, the probability of this event is as least ϵ. □

In this way, the probability that $\mathscr{C}$ solves the CDH problem is calculated as

$$\begin{aligned}\epsilon' &= P\left(\Sigma_1\Sigma_2\Sigma_3\Sigma_4\right)\\ &\geq \left(1-\frac{1}{q_{H_1}}\right)^{q_S}\left(1-\frac{1}{q_{H_1}}\left(1-\frac{1}{q_{H_1}}\right)\right)^{q_{SC}}\frac{\epsilon}{q_{H_1}}.\end{aligned} \tag{A.3}$$

B. Proof of Lemma 13

Given an input $\langle P, aP\rangle \in \mathscr{G}$ of the ECDLP assumption, the task of the challenger $\mathscr{C}$ is to calculate a with the support of Type I adversary $\mathscr{A}_1$. Assume $\mathscr{A}_1$ is able to forge a valid signcryption with advantage ϵ.

B.1. Setup. $\mathscr{C}$ randomly chooses $a \in_R Z_q^*$ and calculates $mpk = aP$. After that, $\mathscr{C}$ sets the public parameters $params = \{p, q, P, mpk, H_1, H_2, H_3\}$ and sends $params$ to $\mathscr{A}_1$, where H_1, H_2, and H_3 are considered to be random oracles. Let q_{H_1} denote the maximum number of queries on H_1. $\mathscr{C}$ randomly chooses $I \in [1, q_{H_1}]$.

B.2. Training. In this phase, $\mathscr{A}_1$ queries the same oracles and receives the same responses as the proof of Lemma 10.

B.3. Forgery. Eventually, $\mathscr{A}_1$ returns a forged aggregate signcryption $\sigma^{*(1)} = \{(f_{i,1}^{*(1)}, f_{i,2}^{*(1)}, m_i^{*\prime(1)})_{i=1,\ldots,n}, F_{j,3}^{*(1)}\}$ on n messages with n mobile sensors, where $F_{j,3}^{*(1)} = \sum_{i=1}^{n} f_{i,3}^{*(1)}$. Note that identities and public keys of the mobile sensors form the lists $L_{ID}^* = \{\mathrm{ID}_1^*, \ldots, \mathrm{ID}_n^*\}$, $L_{PK}^* = \{(upk_{1,1}{}^*, upk_{1,2}{}^*), \ldots, (upk_{n,1}{}^*, upk_{n,2}{}^*)\}$, respectively, and the messages form the list $L_m^{*(1)} = \{m_1^{*(1)}, \ldots, m_n^{*(1)}\}$. It is required that there exists $i \in \{1, \ldots, n\}$ such that $\mathrm{ID}_i^* = \mathrm{ID}_I$ and the oracle Signcryption$(m_i^{*(1)}, \mathrm{ID}_i^*, \mathrm{ID}_j^*)$ has not been queried. Moreover, the forged aggregate signcryption must be valid. Thus, the solution of the ECDLP problem is calculated by $\mathscr{C}$ as follows:

(1) If these is a tuple $\langle v_{i,1}, v_{i,2}, h_{3,i}\rangle$ in L_3 such that $m_i^{*(1)} = h_{3,i} \oplus m_i^{*\prime(1)}$, $v_{i,1} = usk_{j,1}{}^* f_{i,1}^{*(1)}$, $v_{i,2} = usk_{j,2}{}^* f_{i,1}^{*(1)}$, $\mathscr{C}$ retrieves $m_i^{*(1)}$. Otherwise, $\mathscr{C}$ aborts this game.

(2) If the verification holds in Aggregate-Verification algorithm by inputting $\sigma^{*(1)}$, $\mathscr{C}$ replays this game with the same random tape with different choices of H_1, H_2 oracles. Thus, another two forged aggregate signcryption schemes $\{(f_{i,1}^{*(2)}, f_{i,2}^{*(2)}, m_i^{*\prime(2)})_{i=1,\ldots,n}, F_{j,3}^{*(2)}\}$ and $\{(f_{i,1}^{*(3)}, f_{i,2}^{*(3)}, m_i^{*\prime(3)})_{i=1,\ldots,n}, F_{j,3}^{*(3)}\}$ are returned by $\mathscr{A}_1$. The following equations hold if these signatures are valid:

$$f_{I,2}^{(k)}\left(upk_{I,1} + upk_{I,2} + h_{1,I}^{(k)} \cdot mpk + f_{I,3}^{(k)} P\right) = f_{I,1}^{(k)}. \tag{B.1}$$

In this equation, $k = 1, 2, 3$. Because $upk_{I,1} = usk_{I,1}P$, $upk_{I,2} = y_I P$, $mpk = aP$, $f_{I,1}^{(k)} = rP$, (B.1) is denoted as

$$f_{I,2}^{(k)}\left(usk_{I,1} + y_I + h_{1,I}^{(k)} a + f_{I,3}^{(k)}\right) = r, \quad k = 1, 2, 3. \tag{B.2}$$

In (B.1) and (B.2), the values of $usk_{I,1}$, r, and a are unknown to $\mathscr{C}$. The value of $usk_{I,1}$ is calculated as follows:

$$\begin{aligned}usk_{I,1} = &\left(f_{I,2}^{(1)} - f_{I,2}^{(2)}\right)^{-1}\left(a\left(h_{1,I}^{(2)} f_{I,2}^{(2)} - h_{1,I}^{(1)} f_{I,2}^{(1)}\right)\right.\\ &\left. + f_{I,2}^{(2)} f_{I,3}^{(2)} - f_{I,2}^{(1)} f_{I,3}^{(1)} - y_I\left(f_{I,2}^{(1)} - f_{I,2}^{(2)}\right)\right).\end{aligned} \tag{B.3}$$

Consequently, the value of a is calculated as

$$\begin{aligned}a = &\left(f_{I,2}^{(3)} h_{1,I}^{(3)} - f_{I,2}^{(2)} h_{1,I}^{(2)}\right)^{-1}\left(usk_{I,1} + y_I\right)\left(f_{I,2}^{(2)} - f_{I,2}^{(3)}\right)\\ &+ f_{I,2}^{(2)} f_{I,3}^{(2)} - f_{I,2}^{(3)} f_{I,3}^{(3)}.\end{aligned} \tag{B.4}$$

To solve the ECDLP problem, it is required that the following events are executed successfully by $\mathscr{C}$:

(i) Σ_1: any of $\mathscr{A}_1$'s Secret-Value-Extraction query is not aborted by $\mathscr{C}$.

(ii) Σ_2: any of $\mathscr{A}_1$'s Signcryption query is not aborted by $\mathscr{C}$.

(iii) Σ_3: the forged aggregate signcryption generated by $\mathscr{A}_1$ is valid.

(iv) Σ_4: if Σ_3 occurs, there exists $i \in \{1, \ldots, n\}$ such that $\mathrm{ID}_i^* = \mathrm{ID}_I$.

The probability that $\mathscr{C}$ solves the ECDLP problem is denoted as $P(\Sigma_1\Sigma_2\Sigma_3\Sigma_4)$, which is decomposed as

$$\begin{aligned}&P\left(\Sigma_1\Sigma_2\Sigma_3\Sigma_4\right)\\ &= P\left(\Sigma_1\right)P\left(\Sigma_2 \mid \Sigma_1\right)P\left(\Sigma_3 \mid \Sigma_1\Sigma_2\right)P\left(\Sigma_4 \mid \Sigma_1\Sigma_2\Sigma_3\right).\end{aligned} \tag{B.5}$$

Here, $P(\Sigma_1)P(\Sigma_2 \mid \Sigma_1)$ is calculated as the proof of Lemma 10.

Claim B.1. The probability that $\mathscr{A}_1$ generates a valid forged aggregate signcryption is at least ϵ. Thus, $P(\Sigma_3 \mid \Sigma_1\Sigma_2) \geq \epsilon$.

Proof. If $\mathscr{A}_1$'s Secret-Value-Extraction query and Signcryption query are not aborted by $\mathscr{C}$, $\mathscr{A}_1$ is able to forge signcryption with advantage ϵ. Thus, the probability of this event is as least ϵ. □

Claim B.2. The probability that $\mathscr{C}$ does not abort the simulation on receiving a valid forged signcryption is at least $1/q_{H_1}$.

Proof. If Σ_1, Σ_2, and Σ_3 occur, $\mathscr{C}$ aborts the simulation unless $\mathscr{A}_1$ returns a forgery such that $\mathrm{ID}_i^* = \mathrm{ID}_I$. Thus $P(\Sigma_4 \mid \Sigma_1\Sigma_2\Sigma_3) \geq 1/q_{H_1}$. □

In this way, the probability that $\mathscr{C}$ solves the ECDLP problem is calculated as

$$\begin{aligned}\epsilon' &= P\left(\Sigma_1\Sigma_2\Sigma_3\Sigma_4\right)\\ &\geq \left(1-\frac{1}{q_{H_1}}\right)^{q_S}\left(1-\frac{1}{q_{H_1}}\left(1-\frac{1}{q_{H_1}}\right)\right)^{q_{SC}}\frac{\epsilon}{q_{H_1}}.\end{aligned} \tag{B.6}$$

Symbols

$\mathrm{RCR}_{i,j}$: The road condition report generated by S_i on RE_j
RC_j: The jth road condition
S_i: The ith mobile sensor with identity ID_i
$T_{i,j}$: The time when S_i sensed RC_j
L_j: The location where RC_j occurred
Sig_i: The action signal about RC_j
m_i: The realtime message generated by S_i on RC_j.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Science Foundation of China (nos. 61370026, U1401257, 61672002, 61672135, 61602096, 61502087), Science and Technology Project of Guangdong Province (no. 2016A010101002), 13th Five-Year Plan of National Cryptography Development Fund for Cryptographic Theory of China (MMJJ20170204), Sichuan Science-Technology Support Plan Program (nos. 2016JZ0020, 2016GZ0065, 2016GZ0063), and Fundamental Research Funds for the Central Universities (nos. ZYGX2016J091, ZYGX2015J072).

References

[1] M. Perttunen, O. Mazhelis, F. Cong et al., "Distributed road surface condition monitoring using mobile phones," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6905, pp. 64–78, 2011.

[2] M. Ndoye, A. M. Barker, J. V. Krogmeier, and D. M. Bullock, "A recursive multiscale correlation-averaging algorithm for an automated distributed road-condition-monitoring system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 795–808, 2011.

[3] L. Zhou, R. Q. Hu, Y. Qian, and H.-H. Chen, "Energy-spectrum efficiency tradeoff for video streaming over mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 5, pp. 981–991, 2013.

[4] Q. Jiang, Z. Chen, B. Li, J. Shen, L. Yang, and J. Ma, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Ambient Intelligence and Humanized Computing*, 2017.

[5] G. Mauri, M. Gerla, F. Bruno, M. Cesana, and G. Verticale, "Optimal Content Prefetching in NDN Vehicle-to-Infrastructure Scenario," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2513–2525, 2017.

[6] S. Bitam, A. Mellouk, and S. Zeadally, "VANET-cloud: A generic cloud computing model for vehicular Ad Hoc networks," *IEEE Wireless Communications Magazine*, vol. 22, no. 1, pp. 96–102, 2015.

[7] C.-C. Lee, Y.-M. Lai, and P.-J. Cheng, "An efficient multiple session key establishment scheme for VANET group integration," *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 35–43, 2016.

[8] C.-T. Li, C.-C. Lee, and C.-Y. Weng, "A Secure Cloud-Assisted Wireless Body Area Network in Mobile Emergency Medical Care System," *Journal of Medical Systems*, vol. 40, no. 5, article no. 117, 2016.

[9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC 2012*, pp. 13–15, Finland, August 2012.

[10] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the Workshop on Mobile Big Data (Mobidata '15)*, pp. 37–42, ACM, Hangzhou, China, June 2015.

[11] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, and M. Dlodlo, "From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.

[12] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proceedings of the 2014 Australasian Telecommunication Networks and Applications Conference, ATNAC 2014*, pp. 117–122, Australia, November 2014.

[13] C.-C. Lee, I.-E. Liao, and M.-S. Hwang, "An efficient authentication protocol for mobile communications," *Telecommunication Systems*, vol. 46, no. 1, pp. 31–41, 2011.

[14] X. Li, J. Peng, S. Kumari, F. Wu, M. Karuppiah, and K. Raymond Choo, "An enhanced 1-round authentication protocol for wireless body area networks with user anonymity," *Computers & Electrical Engineering*, 2017.

[15] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) ≪ cost(signature) + cost(encryption)," in *Advances in Cryptology—CRYPTO '97. Proceedings 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997*, vol. 1294 of *Lecture Notes in Computer Science*, pp. 165–179, 1997.

[16] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASIACCS '08)*, pp. 369–372, ACM, March 2008.

[17] S. S. D. Selvi, S. S. Vivek, J. Shriram, S. Kalaivani, and C. P. Rangan, "Identity based aggregate signcryption schemes," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5922, pp. 378–397, 2009.

[18] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Advances in cryptology—ASIACRYPT 2002*, vol. 2501 of *Lecture Notes in Comput. Sci.*, pp. 548–566, Springer, Berlin, 2002.

[19] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology-ASIACRYPT*, vol. 2894 of *Lecture Notes in Computer Science*, pp. 452–473, Springer, 2003.

[20] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing based road surface condition monitoring system using fog computing," *IEEE Internet of Things Journal*, no. 99, pp. 1-1, 2017.

[21] H. Xiong and Z. Qin, "Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1442–1455, 2015.

[22] Z. Eslami and N. Pakniat, "Certificateless aggregate signcryption: Security model and a concrete construction secure in the random oracle model," *Journal of King Saud University - Computer and Information Sciences*, vol. 26, no. 3, pp. 276–286, 2014.

[23] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[24] C.-C. Lee, Y.-M. Lai, C.-T. Chen, and S.-D. Chen, "Advanced Secure Anonymous Authentication Scheme for Roaming Service in Global Mobility Networks," *Wireless Personal Communications*, vol. 94, no. 3, pp. 1281–1296, 2017.

[25] X. Li, A. K. Sangaiah, S. Kumari, F. Wu, J. Shen, and M. K. Khan, "An efficient authentication and key agreement scheme with user anonymity for roaming service in smart city," *Personal and Ubiquitous Computing*, vol. 21, no. 5, pp. 791–805, 2017.

[26] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity," *Information Sciences*, vol. 321, Article ID 11496, pp. 162–178, 2015.

[27] H. Xiong, "Cost-effective scalable and anonymous certificateless remote authentication protocol," *IEEE Transactions on Information Forensics & Security*, vol. 9, no. 12, pp. 2327–2339, 2014.

[28] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, November 1993.

[29] C. Chen, T. Wang, and J. Tian, "Improving timing attack on RSA-CRT via error detection and correction strategy," *Information Sciences*, vol. 232, pp. 464–474, 2013.

[30] B. Lynn, Pbc library, Online: http://crypto. stanford. edu/pbc.