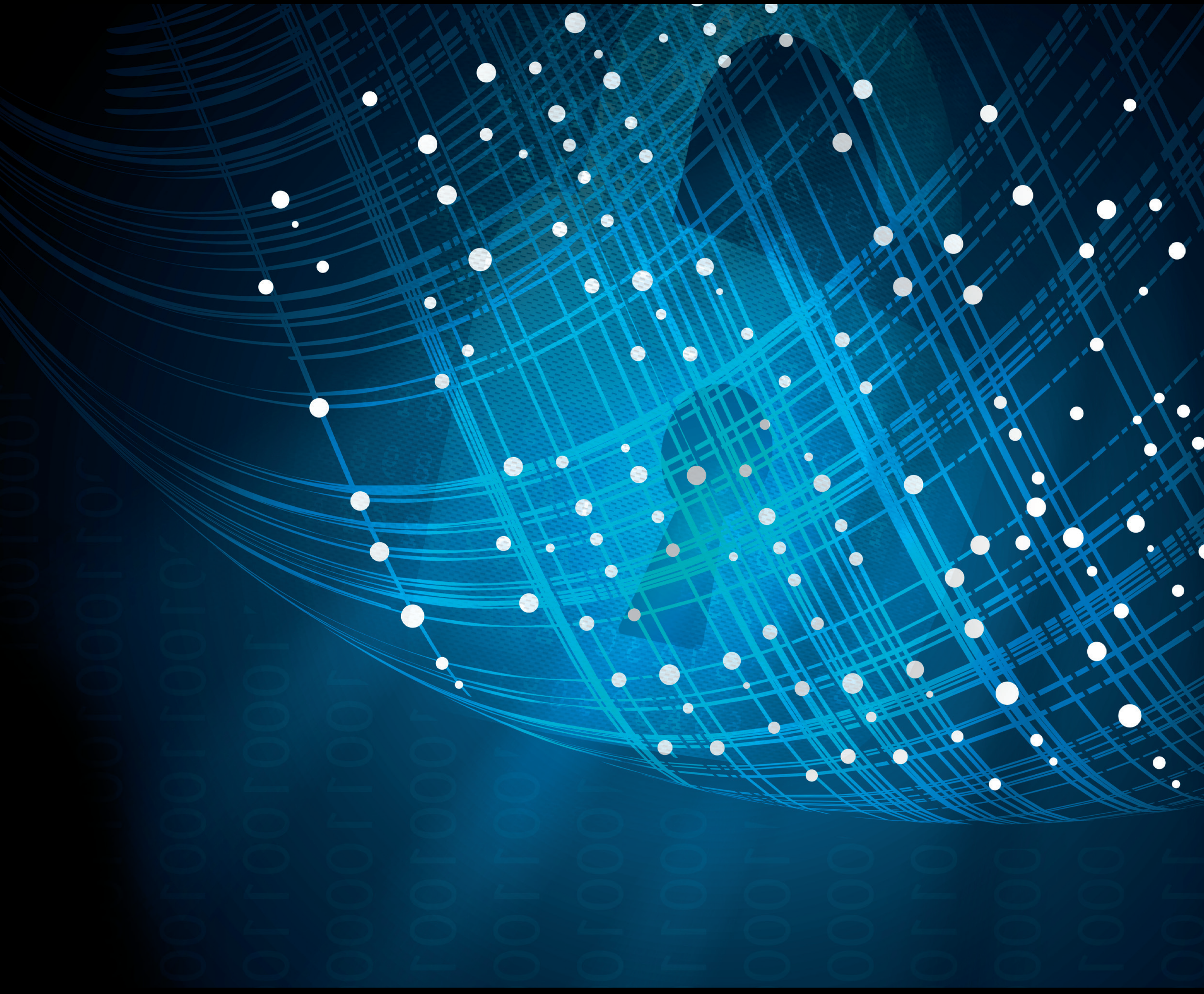
Security and Communication Networks

Security Threats and Defenses for Connected Vehicles

Lead Guest Editor: Jian Weng
Guest Editors: Rongxing Lu, Gerhard Hancke, Anjia Yang, and Leo Y. Zhang



Security Threats and Defenses for Connected Vehicles

Security and Communication Networks

Security Threats and Defenses for Connected Vehicles

Lead Guest Editor: Jian Weng
Guest Editors: Rongxing Lu, Gerhard Hancke,
Anjia Yang, and Leo Y. Zhang

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Security and Communication Networks
Volume 2021, Article ID 2847731, 11 pages
https://doi.org/10.1155/2021/2847731

Research Article

Efficient Identity-Based Broadcast Encryption Scheme on Lattices for the Internet of Things

Kai He,[1] **Xueqiao Liu**,[2] **Jia-Nan Liu**,[3] **and Wei Liu**[3]

[1]*School of Cyberspace Security, Dongguan University of Technology, DongGuan 523808, China*
[2]*Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong 2522, Australia*
[3]*The College of Cyber Security, Jinan University, Guangzhou 510632, China*

Correspondence should be addressed to Jia-Nan Liu; j.n.liu@foxmail.com

Received 11 June 2021; Accepted 22 November 2021; Published 13 December 2021

Academic Editor: David Megias

In an identity-based broadcast encryption (IBBE) scheme, the ciphertext is usually appended with a set of user identities to specify intended recipients. However, as IBBE is adopted in extensive industries, the demand of anonymity for specific scenarios such as military applications is urgent and ought no more to be ignored. On the contrary, how to optimize computation and communication is an unavoidable challenge in the IBBE scheme construction, especially in the large-scaled resource-limited wireless networks such as the Internet of Things (IoT), where the cost of computation and communication should be mitigated as much as possible since other functions including connectivity and privacy should be given the top priority. Thus, we present an IBBE scheme from the lattice, in which we employ the Chinese remainder theorem and lattice basis delegation in fixed dimensions to obtain several desirable characteristics, such as constant-size public parameter, private key, and ciphertext. In addition, our encryption and decryption algorithms are more efficient than broadcast encryption (BE) schemes based on number-theoretic problems. To be noticed, our scheme can simultaneously achieve confidentiality and outsider anonymity against the chosen-plaintext attack under the hardness of the learning with error (LWE) problem.

1. Introduction

IoT is a network of interconnected things/devices, in which sensors, software, network connections, and necessary electronic devices are integrated to collect and exchange information and respond to real-time data requests. IoT allows data accumulation from and exchange between the physical world and computer systems through existing network infrastructures. With these connected tiny and smart devices, one's life can be of higher quality, safer, smarter, more convenient, comfortable, and timely informed than ever before. Security is one of the main concerns mentioned by cybersecurity experts. They believe that even end device connectivity and information sharing can be exploited to have a negative impact on a person safety and well-being. Besides hacking IoT devices to compromise online data and privacy, it can also become the entry point of invading the entire network [1, 2].

Remote terminal unit (RTU) [1] is an electronic device, which is installed in a remote site (generally, few people supervise the distant site). It is used to monitor and control sensors and equipment remotely and widely adopted in the supervisory control and data acquisition (SCADA) system. RTU usually converts the measured state or signal into a data format that can be sent on the communication medium by using the Modbus protocol. It can also receive commands sent by the central monitor computer to execute functional control of the equipment. As the Modbus protocol does not apply data encryption mechanism, the data flow between the monitor center and RTU is in plaintext. As a consequence, the data transmitted in the open network may be eavesdropped or tampered with. What is worse, the data tampering may cause disorder in the automated production process or even serious accidents of equipment damage. To keep the confidentiality of data transmission, cryptographic modules can be embedded in data collection equipment such as RTU/DTU and effectively help prevent data theft and command tampering [2]. Once the concern of confidentiality is got rid of, such devices can be safely applied to industrial control industries such as oil and gas exploitation, environmental monitoring, power transmission and transformation, oil and gas pipeline networks, and hydrological monitoring.

Fiat and Naor [3] first introduced broadcast encryption, which allows a sender to send an encrypted message to a large number of receivers via public channels, and only authorized users can obtain the message, as shown in Figure 1. Compared with the public key encryption for a single recipient, BE significantly saves computing and communication costs. Therefore, BE has been promoted to numerous applications, such as key distributing [4], encrypted file sharing [5], satellite TV subscription [6], digital right management [7], and social network service [8]. Take pay service as an example. As shown in Figure 2, nonpaying user U_2 cannot enjoy the service or just is able to enjoy limited service, while paying users $(U_1, U_3, \ldots, U_n)$ can enjoy entire and high-quality service. There are a large number of related works that can be classified into the conventional BE [6, 9–13] since they are based on number-theoretic problems, such as big integer factoring and discrete logarithm problem, and rarely meet the requirements of industrial applications.

With the advent of quantum cryptography, the security of conventional BE schemes is heavily threatened. In FOCS$'$94, Shor [14] proposed a quantum algorithm to solve the problem of discrete logarithm and factorization in polynomial time. Thereafter, it becomes one of the most urgent topics to design BE schemes against quantum attacks.

Lattice cryptography can resist quantum-computing attacks [15] and has multiple advantages over the conventional cryptography. Firstly, lattice is a vector space composed of n linearly independent vectors $b_1, \ldots, b_n$ in $\mathbb{R}^m$, which only request lightweight operations such as modular addition and matrix multiplication. Thus, it is suitable for devices with limited computational ability such as smart cards. Secondly, lattice cryptography enjoys pretty strong security guaranteed by the worst-case hardness assumptions [16, 17], such as shortest vector problem (SVP) [18] and closest vector problem (CVP) [18]. Thirdly, lattice cryptography can be adopted to comparable extensive industries as its conventional cryptography was, given almost all conventional public key encryption (PKE) schemes based on big integer factoring or discrete logarithm problems can also be realized in lattice cryptography.

A desirable BE scheme on lattices should keep not only confidentiality but also anonymity as anonymity is an extremely favourable characteristic for diverse BE systems [19]. To distinguish authorized receivers from the unauthorized, BE ciphertext usually includes the intended recipients' identities. This means users' identity information is revealed. Specifically, such identity exposure is expected to be avoided when users' identities are sensitive. For instance, in the military field, the set of broadcast receiver identities undoubtedly implies specific military objectives or personnel. Meanwhile, to support a large number of receivers in a BE system, the public key of every receiver can be conveniently chosen as a meaningful string, which is their unique identification, such as a passport number or an e-mail address. This is exactly the motivation of proposing an IBBE system that is capable to support exponential user scale.

1.1. Our Results.

Each BE system involves multiple recipients. Thus, it is intricate to construct a BE scheme in a lattice context. Our main contributions include the construction of an anonymous IBBE from the lattice and the security reduction to the LWE problem. Our design is inspired by the lattice-based BE scheme of Wang et al. [20], which depends on the Chinese remainder theorem to achieve the dynamic anonymity. In this work, we rely on the Chinese remainder theorem to construct an IBBE scheme, and the core idea is as follows.

The Chinese remainder theorem offers one-dimensional linear congruence equation $x \equiv a_i \,(\mathrm{mod} q_i)$ that has and only has one solution $x = Q_1 Q_1' a_1 + \cdots + Q_k Q_k' a_k \,(\mathrm{mod} Q)$. In order to construct a BE scheme on lattices, we combine the Chinese remainder theorem with the LWE hardness assumption.

(i) Firstly, we extend the Chinese remainder theorem to a matrix form, such as x and a_i are extended to matrices X and A_i with dimension $n \times m$, respectively. Thus, the system of linear congruence equations has the similar solution; that is, $X = Q_1 Q_1' A_1 + \cdots + Q_k Q_k' A_k \,(\mathrm{mod} Q)$, where X is close to uniform distribution [20] if A_i is a random matrix over Z_q^n.

(ii) Then, choose a random vector s, which is to blind X. Blind results are used to encapsulate symmetric keys K, e.g., $C_2 = (\sum_{i=1}^{k} Q_i Q_i' A_i)(\sum_{i=1}^{k} Q_i Q_i' s) + 2e + K)$ $(\mathrm{mod} Q)$, where A_i and q_i are receiver i's public keys and e is an error vector. Since the key encapsulation is constructed by the Chinese remainder theorem, its distribution is indistinguishable from the uniform distribution [20].

(iii) Thirdly, when authorized receiver i decrypts the ciphertext, he does not need to know the other users' identities. He firstly computes C_2 mod q_i, where qi is his public key. and then C_2 is transformed to a LWE instance vector related to his public key A_i, i.e., $C_2 \,(\mathrm{mod} q_i) = (A_i^{\top} s + 2e + K) \,(\mathrm{mod} q_i)$. Now, authorized receiver i uses his private key to decrypt $(A_i^{\top} s + 2e + K) \,(\mathrm{mod} q_i)$ to obtain the symmetric key K and then gets the broadcast message.

(iv) To obtain an IBBE scheme, we need to connect users' public keys A_i and q_i to identity ID_i. Firstly, we use an encoding function $H_1 \colon \mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n \times n}$ to map identities $ID_i \in \mathbb{Z}_q^n$ to matrices $A_i \in \mathbb{Z}_p^{n \times n}$, i.e., $A_i = H_1(ID_i)$ [21], and a division intractable hash function $H_2 \colon \{0, 1\}^* \longrightarrow \mathbb{Z}_q^n$ to map identities $ID_i \in \mathbb{Z}_q^n$ to integer $q_i \in \mathbb{Z}_p^n$. Note that integer q_i is a prime with an overwhelming probability [22] so

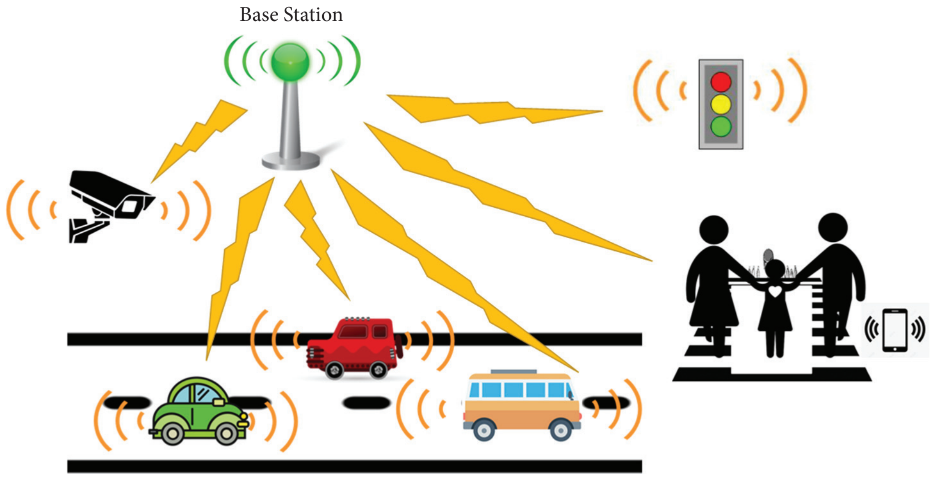


Figure 1: Broadcast in the Internet of Vehicles network.

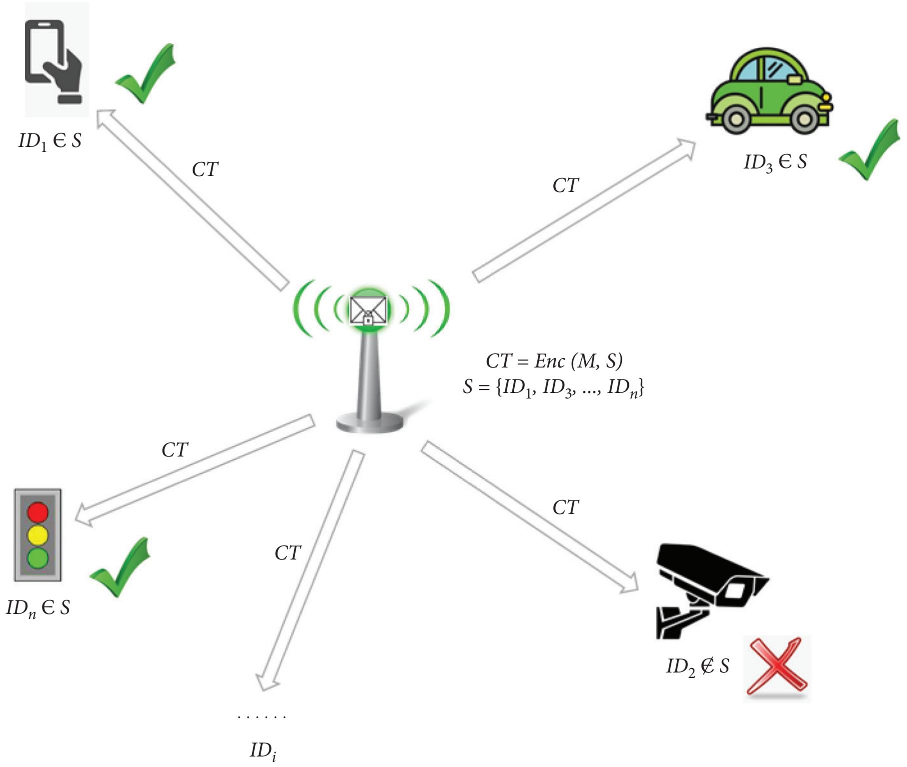


Figure 2: IBBE for pay service.

that user i's public key q_i and user j's public key q_j are ensured mutually prime.

(v) Lattice basis delegation mechanisms were proposed by Cash et al. [23] and Agrawal et al. [21]. Given a matrix $A \in \mathcal{Z}_q^{n\times m}$ and a lattice basis T_A of $\Lambda_q^{\perp}(A)$, a matrix B from A and a random basis T_B for $\Lambda_q^{\perp}(B)$ can be generated. However, in [21, 23], the dimension of matrix B is larger than the dimension of the given matrix A. So, the ciphertext and private key sizes of their HIBE schemes increase as the hierarchy deepens. Thus, in terms of private key generation of our scheme, we employ lattice basis delegation with constant dimension technology [21] to generate the user private key, where $B = AR^{-1} \in \mathcal{Z}_q^{n\times m}$ and B has the same dimension as A. Thence, the private key size of our scheme is constant, and the size of the ciphertext has nothing to do with the number of recipients.

1.2. Related Work. Identity-based encryption (IBE) [24] is a special kind of BE. There is one receiver set specifying intended receivers, and in an IBE scheme, the user public key can be any string as long as the string can be a uniquely identified user, such as a passport number and e-mail address. In 2008, Craig et al. [25] proposed the technology of lattice-based one-way trapdoor function and constructed an IBE scheme whose security is based on the LWE problem [26] in the random oracle model. In their scheme, trapdoor sampling algorithm [27] is used for generating the master public key and master secret key. Then, the preimage sampler [25] takes the master secret key as the input to generate the user's secret key. Finally, both the master public key and the user's identity are used to generate two separate pseudorandom LWE instances as the ciphertexts.

Hierarchical identity-based encryption (HIBE) [28, 29] is also a special kind of BE. Users in the broadcast set have a hierarchical structure, and the lower-level users' keys are generated by the higher-level users. In 2010, Cash et al. [23, 30] proposed a new concept of cryptography, called bonsai tree, and constructed an HIBE scheme based on the LWE problem by utilizing the lattice basis delegation technique, which allows one to use a short basis of a certain integer lattice L to generate a short random basis for a new lattice L_0 derived from L. However, in their HIBE scheme, the dimension of the child lattice L_0 is greater than that of the parent lattice L for the reason that, as the hierarchical structure increases, the private key and ciphertext also become longer. Shweta Agrawal and Boyen [31] proposed a lattice basis delegation technique which does not increase the dimension of the lattices involved and presented two HIBE schemes with shorter ciphertext and private keys with and without the random oracle based on the LWE problem, respectively.

Attribute-based encryption (ABE) [32] and BE are both one kind of one-to-many encryption. In the ABE system, the private key and the ciphertext are related to the attributes; when the attributes owned by the user match the ciphertext attributes, the user can obtain the ciphertext. Boyen [33] proposed an efficient ABE scheme and proved its security in the selective sense from LWE hardness assumption in the standard model. Nevertheless, BE needs to specify which users are authorized receivers.

Fiat and Naor first introduced BE [3]. In 2005, Boneh et al. [11] proposed the first fully collusion-resistant BE scheme with static security, and both the size of ciphertexts and private keys are constant, but the size of the public key is proportional to the number of receivers. In 2009, Craig and Waters [13] proposed a BE scheme with adaptive security in the random oracle model. In 2007, Delerablée [34] proposed the first IBBE scheme, which obtains adaptive chosen-ciphertext attack (CCA) in the random oracle model, as well as has constant-size ciphertexts and private keys. In 2009, Craig and Waters [13] presented the first IBBE scheme, which is against adaptively chosen-plaintext secure in the standard model. In 2014, Boneh et al. [35] proposed the first IBBE scheme, which obtains selectively CCA-secure from multilinear maps and has constant-size ciphertexts. In 2015, Jongkil Kim et al. [36] proposed an IBBE scheme, which is adaptively CCA-secure in the standard model, but uses dual encryption technique. In 2016, Dan and Zhandry [37] proposed a BE scheme, which obtains adaptive security by using indistinguishability obfuscation technique and has short ciphertexts, secret keys, and public keys.

Anonymity is a good security property; however, the aforementioned scheme cannot be obtained because the recipients' identities are broadcasted as ciphertext. Thus, the identities' information is exposed. In 2006, Adam et al. [12] presented two fully anonymous BE constructions; both of them obtain CCA security. The first one is a generic construction, and the decryption cost has a linear relationship with the number of receivers. The second is a specific construction, requiring a certain number of decryption operations, and the security proof relies on a random oracle model. In 2012, Libert et al. [19] proposed some fully anonymous BE schemes, which are fully anonymous and have adaptive CCA security in the standard model; at the same time, the formal security definition of the anonymous BE scheme is given. In 2012, two anonymous BE schemes with outsider anonymous were proposed by Fazio and Milinda Perera [38], and the two BE schemes have sublinear-size ciphertexts. In 2016, two anonymous BE schemes were proposed by He et al. [39]; the first one is the general scheme [39], and the second one is the specific scheme [39]. Both of these schemes are proven to be adaptive CCA-secure. However, all the aforementioned traditional BE/IBBE schemes cannot resist quantum attacks.

In 2010, Wang and Bi [40] proposed a secure lattice-based IBBE scheme using the basis delegation technique [23], and their scheme can be easily extended to a hierarchical IBBE. However, their lattice basis delegation technique increases the dimension of users' identity matrix. In 2013, Georgescu [41] used a tag-based hint system which is secure based on ring-LWE hardness and an IND-CCA-secure public key encryption scheme from LWE to construct a CCA-secure lattice-based anonymous BE scheme. In 2015, Wang et al. [20] used the Chinese remainder theorem to construct a dynamical and outsider-anonymous BE scheme over the lattice, which is proven semantic secure in the standard model under the hardness of the LWE problem. In 2020, Brakerski and Vaikuntanathan [42] proposed a lattice-based BE scheme where the size of the key and ciphertext has a logarithmic correlation with the number of users. However, their BE construction is based on a heuristic that allows to "invert" the key succinctness of the BGG + KP-ABE scheme [43] and does not have a security reduction for this heuristic; its security is an open problem. In 2020, Agrawal and Yamada [44] improved Boneh et al.'s [35] BE scheme which used multilinear maps by using LWE and bilinear mapping, and the parameters of the improved solution were also very small. Thus, in this paper, we construct an anonymous IBBE scheme on the lattice. We make a detailed function comparison between our scheme and other schemes in Table 1.

Table 1: Comparisons with related works.

Scheme	Identity-based	Anonymity
[41]	×	√
[20]	×	√
[40]	√	×
[42]	×	×
Ours	√	√

2. Preliminaries

Let us briefly introduce some of the symbols and definitions used throughout the paper.

2.1. Collision Intractability [22]. $\mathscr{H} = \{H_k\}_{k \in \mathscr{N}}$ is a family of hash functions. If it is difficult to find two inputs that hash to the same output, $\mathscr{H}$ is collision intractable. Formally, for every probability polynomial-time (PPT) adversary $\mathscr{A}$, there is a negligible function negl() such that

$$Pr^{\mathscr{A}}_{H \in H_k}[\mathscr{A}(H) = (x, x'), \text{s.t. } x \neq x' \text{ and } H(x) = H(x')] = \text{negl}(k)a. \tag{1}$$

2.2. Division Intractability [22]. $\mathscr{H}$ is a hashing family; if it is division intractable, it is hard to find distinct inputs $x_1, \ldots, x_n, y$ such that $h(y)$ divides $\prod_{i=1}^{n} h(x_i)$. Formally, for every PPT adversary $\mathscr{A}$, there is a negligible function negl() such that

$$Pr^{\mathscr{A}}_{h \in H_k}\left[\begin{array}{c} \mathscr{A}(h) = \left(\{x_i\}_{i\in[n]}, y\right), \\ \text{s.t. } y \neq \{x_i\}_{i\in[n]} \text{ and } h(y) \text{ divides } \prod\limits_{i=1}^{n} h(x_i) \end{array}\right] = \text{negl}(k). \tag{2}$$

It is not difficult to see that a hash family $\mathscr{H}$ that is division intractable must also be collision intractable, but the reverse is not true. Such a function is easy to obtain by setting $H_0(X) = H(X)|1$ (or only the lowest bit of $H(X)$) to be one.

2.3. Lattice and Lattice Problems

2.3.1. Lattice [45]. Lattice Λ is generated by a set of n linearly independent vectors $B = \{b_1, b_2, \ldots, b_n\}$ such that

$$\Lambda = \{Bc | Bc = c_1 b_1 + c_2 b_2 + \cdots + c_n b_n \in \mathbb{Z}\}. \tag{3}$$

2.3.2. q-ary Lattices [45]. (q, m, n) are some integers, $A \in \mathbb{Z}_q^{n\times m}$ is a parity check matrix, and q-ary lattices are defined as

$$\Lambda_q^{\perp}(A) = \left\{e \in \mathbb{Z}_q^m, Ae = 0 (\text{mod} q)\right\}. \tag{4}$$

In fact, all vectors in lattice $\Lambda_q^{\perp}(A)$ are orthogonal modulo q to the matrix A row vector.

2.3.3. Gaussian over Lattices [45]. Gaussian function $\rho_s\colon \mathbb{R}^m \longrightarrow (0, 1]$ is defined as

$$\rho_s(x) = \exp\left(-\pi \frac{\|x\|^2}{s^2}\right), \tag{5}$$

for any $s > 0$ and dimension $m \geq 1$. The discrete Gaussian distribution $D_{\Lambda_y^{\perp}(A),s}$ over the coset $L = t + \Lambda_y^{\perp}(A)$, $t \in \mathbb{Z}^m$, whose probability is proportional to $\rho_s(x) x \in \Lambda_y^{\perp}(A)$, and probability is zero elsewhere.

2.3.4. LWE Problem [26, 45]. Let $A \leftarrow_R \mathbb{Z}_q^{m\times n}$, $s \leftarrow_R \mathbb{Z}_q^n$, the error distribution χ be over $\mathbb{A}_{(s,\chi)}$, and e be distributed according to χ. Given $(A, As + e(\text{mod} q))$, the decision variant LWE problem is to distinguish $(A, As + e(\text{mod} q))$ from uniform distribution.

2.3.5. Gaussian Error Distributions Φ_α^m26. The standard error distribution Φ_α^m is the Gaussian distribution on $\mathbb{Z}_q^m$, and the deviation is $q\alpha > \sqrt{n}$. According to the distribution Φ_α^m, the error vector can be effectively sampled, as shown in the following:

(i) Sample $\eta_1, \eta_2, \ldots, \eta_m$ comes from the Gaussian distribution D_α on $\mathscr{R}$

(ii) Let $e_i = (q\eta_i)(\text{mod} q)$ where (x) is used to represent the integer closest to x

(iii) Let $e = (e_1, \ldots, e_m)$ be the error vector in the LWE problem instance

2.3.6. Trapdoor Sampling Algorithm [21]. Let $q \geq 3$ be odd and $m\colon = \lceil 6n \log q \rceil$. There exists a PPT algorithm (TrapGen) (q, n), and it outputs a matrix A and a full rank set $T \in \mathbb{Z}^{m\times m}$, where A's distribution is statistically close to a uniform distribution, T is a lattice basis of $\Lambda^{\perp}(A)$, which satisfies $\|\widetilde{T}\| \leq \mathscr{O}(\sqrt{(n \log q)})$, and $\|T\| \leq \mathscr{O}(n \log q)$ with almost negligible probability.

The trapdoor T can be utilized to solve the LWE problem; that is, given $y = A^t s + e(\text{mod} q)$ where e is any "short enough" vector, it can be used to recover s as follows [25]:

(i) Calculate $T^t y = T^t(A^t s + e) = (AT)^t s + T^t e = T^t e (\text{mod} q)$ and $e = (T^t)^{-1} T^t e (\text{mod} q)$. Now, since both T and e contain small entries, each entry of the vector $T^t e$ is less than q, so $T^t e (\text{mod} q) = T^t e$.

(ii) LWE secret s can be recovered via A, e, y.

2.3.7. Algorithm Basis Delegation [21]. The basic delegation algorithm *BasisDel* (A, R, T_A, σ) will not increase the dimension of the basic matrix [21]. On inputting a rank n matrix A in $\mathbb{Z}_q^{n\times m}$, a $\mathbb{Z}_q$-invertible matrix R in $\mathbb{Z}^{m\times m}$ sampled from $\mathscr{D}_{m\times m}$, a basis T_A of $\Lambda_q^{\perp}$, and the parameter $\sigma \in \mathbb{R}_{>0}$, output a basis T_B of $\Lambda_q^{\perp}(B)$, where $B = AR^{-1}$ in $\mathbb{Z}_q^{n\times m}$.

2.3.8. Algorithm Sample R (1^n)21. Our security proof uses algorithm sample R. The sample matrix in $\mathbb{Z}^{m\times m}$ comes from a distribution that is statistically close to $\mathscr{D}_{m\times m}$ [21]. On the canonical basis T of the lattice $\mathbb{Z}^m$, run $r_i \leftarrow_R$ Sample Gaussian $(\mathbb{Z}^m, T, \sigma, 0)$ for $i = 1, \ldots, m$. If R is $\mathbb{Z}_q-$invertible, then output R; otherwise, run the sample Gaussian algorithm repeatedly.

Algorithm sample R with basis is used in our security proof, which gives a random rank n matrix A in $\mathbb{Z}^q_{n\times m}$ and generates a "low-norm" matrix R from $\mathbb{D}_{m\times m}$ and the short base of $\Lambda_q^{\perp}(AR^{-1})$ as follows.

2.3.9. Algorithm Sample R with Basis (A) 21. $a_1, \ldots, a_m \in \mathbb{Z}_q^n$ are the m columns of the matrix $A \in \mathbb{Z}_q^{n\times m}$.

(i) Run TrapGen(q, n) to generate a matrix $B \in \mathbb{Z}_q^{n\times m}$ with random rank n, as well as lattice $\lambda_q^{\perp}(B)$ base T_B, where

$$\|\widetilde{T_B}\| \leq \widetilde{L}_{TG} = \frac{\sigma_R}{w(\sqrt{\log m})}. \tag{6}$$

(ii) For $i = 1, \ldots, m$, do

(1) Sample r_i by running SamplePre$(B, T_B, a_i, \ldots)$, and we have $Br_i = a_i \text{mod} q$, $Br_i = a_i \text{mod} q$, where r_i is sampled from a distribution statistically close to $D_{\Lambda_q^{a_i}(B), \rho_R}$

(2) Repeat Step (1) until r_i is linearly independent of $r_1, \ldots, r_{i-1}$

(iii) Let $R \in \mathbb{Z}^{m\times n}$ be the matrix with columns $r_1, \ldots, r_m$. Then, R has rank m. Output R and T_B.

2.3.10. Chinese Remainder Theorem [46]. If q_i and q_j are integers, gcd $(q_i, q_j) = 1$, and $\{a_i\}_{1\leq i\leq k}$ are arbitrary integers, a system of linear congruence

$$\begin{cases} x \equiv a_1 (\text{mod} q_1), \\ \vdots, \\ x \equiv a_k (\text{mod} q_k), \end{cases} \tag{7}$$

equations has only one solution:

$$x = Q_1 Q_1' a_1 + \cdots + Q_k Q_k' a_k (\text{mod} Q), \tag{8}$$

where $Q = q_1 q_2 \ldots q_k$, $Q_i = Q/q_i$, and $Q_i Q_i' = 1 (\text{mod} q_i)$ for $1 \leq i \leq k$.

We can also extend the Chinese remainder theorem to a matrix form, such as x and a_i are extended to matrices X and A_i with dimension $n \times m$, respectively; the system of linear congruence has the same solution; that is,

$$X = Q_1 Q_1' A_1 + \cdots + Q_k Q_k' A_k (\text{mod} Q), \tag{9}$$

where X is close to uniform distribution [20].

3. Identity-Based Broadcast Encryption

(i) Init: adversary $\mathscr{A}$ outputs two receiver subsets S_0 and S_1 that he wants to attack; it is required that $|S_0| = |S_1|$ in order to avoid trivial attacks

(ii) Setup: challenger $\mathscr{C}$ first runs Setup to generate the public parameters params and a master secret key *msk*, then gives params to adversary $\mathscr{A}$, and keeps *msk* to itself

(iii) Phase 1: adversary $\mathscr{A}$ adaptively issues the private key for identity $I\ D \notin S_0 \cup S_1$ query, and challenger $\mathscr{C}$ runs $sk_{I\ D} \leftarrow$ Extract $(msk, I\ D)$ and returns $sk_{I\ D}$ to adversary $\mathscr{A}$

(iv) Challenge: adversary $\mathscr{A}$ selects two equal-length messages $M_0, M_1 \in \mathscr{M}$ and sends to challenger $\mathscr{C}$, and challenger $\mathscr{C}$ flips a random coin $\beta \in \{0, 1\}$ and returns the challenge ciphertext $CT^* \leftarrow$ Encrypt (params, S_β, M_β) to adversary $\mathscr{A}$

(v) Phase 2: adversary $\mathscr{A}$ continues to adaptively issue queries as in Phase 1

(vi) Guess: adversary $\mathscr{A}$ outputs a guess $\beta' \in \{0, 1\}$

Definition 1. An IBBE scheme consists of four algorithms (Setup, Extract, Enc, Dec) [19, 34] as follows:

(i) Setup (1^λ): intake a security parameter λ, and output the public parameters params and a master secret key *msk*

(ii) Extract (msk, ID): intake a master secret key *msk* and an identity ID, and output a private key $sk_{I\ D}$ for identity ID

(iii) Enc (params, S, M): intake the public parameters params, a receiver set S, and a message $M \in \mathscr{M}$, and output a ciphertext CT

(iv) Dec $(sk_{I\ D}, CT)$: intake a private key $sk_{I\ D}$ and a ciphertext CT, and output either a message M or an error symbol $\perp$

The correctness property requires that, for all $I\ D \in S$, if (params, msk) $\leftarrow$ Setup (1^λ), $sk_{I\ D} \leftarrow$Extract $(msk, I\,D)$, and $CT \leftarrow Enc$(params, S, M), then Dec $(sk_{I\ D}, CT) = M$ with overwhelming probability.

In the above definition, the set S is not required to intake the decryption algorithm which keeps the anonymity of an IBBE system.

We now present the security requirements for an IBBE scheme to be outsider anonymous against the chosen-plaintext attack (CPA). In an outsider-anonymous IBBE scheme, when the adversary receives a ciphertext of which he is not a legal recipient, he will be unable to learn anything about the identities of the legal recipients, but for those ciphertexts for which the adversary is in the authorized set of recipients, he might also learn the identities of some other legal recipients. First, we define the CPA of an outsider-anonymous IBBE scheme as a game, which we term oAIBBE-IND-CPA, played between a probabilistic polynomial-time (PPT) adversary $\mathscr{A}$ and a challenger $\mathscr{C}$. Meanwhile, we present a selective indistinguishable chosen-plaintext security game (sIND-CPA), where selective security is a weaker notion which forces the adversary $\mathscr{A}$ to announce ahead of time the identities it will target.

Definition 2. The oAIBBE-sIND-CCA game defined for an oAIBBE scheme $\Pi = (\text{Setup, Extract, Enc, Dec})$, a PPT adversary $\mathscr{A}$, and a challenger $\mathscr{C}$ is as follows:

Definition 3. Define adversary $\mathscr{A}$'s advantage in the above oAIBBE-sIND-CPA game as $A\,dv_{\mathscr{A},IBBE}^{\text{oAIBBE}-\text{sIND}-\text{CPA}} = |\Pr[\beta' = \beta] - 1/2|$. We say that an IBBE scheme is oAIBBE-sIND-CPA secure if for any PPT adversary $\mathscr{A}$, the advantage $A\,dv_{\mathscr{A},IBBE}^{\text{oAIBBE}-\text{sIND}-\text{CPA}}$ is negligible in the above oAIBBE-sIND-CPA game.

4. Construction

Our lattice-based IBBE scheme is designed by translating the lattice-based BE scheme of Wang et al. [20] into an identity-based environment. The private key generation depends on the lattice basis delegation without increasing the dimension [21].

(i) Setup (n): intake a secure parameter n, set $q \geq 3$ to be odd and $m\colon = \lceil 6n \log q \rceil$, and let $H_1\colon \{0,1\}^* \longrightarrow \{0,1\}^n$ be a division intractability hash function and $H_2\colon \{0,1\}^* \longrightarrow \mathbb{Z}_q^{m\times m}$ be a hash function. Invoke trapdoor sampling algorithm *TrapGen* (q, n) to generate a uniformly random matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ with a basis $T_0 \in \mathbb{Z}^{m\times m}$ satisfying $A_0 T_0 = 0 \,(\text{mod} q)$ such that $\|T_0\| \leq \mathcal{O}(n \log q)$. Output public parameters

$$mpk = (n, m, q, A_0, H_1, H_2). \tag{10}$$

and a master key $msk = T_0$.

(ii) Extract (mpk, msk, ID): intake public parameters mpk, a master key msk, and an identity $ID \in \{0,1\}^*$, and compute $R_{ID} = H_2(ID) \in \mathbb{Z}_q^{m\times m}$ and $A_{ID} = A_0 R_{ID}^{-1} \in \mathbb{Z}_q^{n\times m}$. Evaluate

$$SK_{ID} \leftarrow \text{Basis Del}(A_0, R_{ID}, T_0, \sigma) \tag{11}$$

to obtain a short random basis $SK_{I\ D}$ for $\Lambda_q^{\perp}(A_{ID})$. Output identity $I\ D$'s private key SK_{ID}.

(iii) Encrypt (mpk, S, M): intake public parameters mpk, a broadcast set $S = \{ID_1, \ldots, ID_d\}$, and message $M \in \{0,1\}^m$, and compute $q_{ID_i} = H_1(ID_i)$ for $ID_i \in S$. Moreover, to ensure the correctness of decryption, we need $q_{ID_i} > q$. According to the Chinese remainder theorem, it needs to compute $Q = q_{ID_1} \cdots q_{ID_d}$ and $Q_{ID_i} = Q/q_{ID_i}$, where $Q_{ID_i} Q_{ID_i}' \equiv 1 \,(\text{mod} q_{ID_i})$. Calculate

$$A_{ID_i} = A_0 H_2(ID_i)^{-1} \in \mathbb{Z}_q^{n\times m}, \tag{12}$$

for $ID_i \in S$, choose random vector $s \in \mathbb{Z}_q^n$ and $e \in \overline{\Phi}_\alpha^m$ and a symmetric key $K \in \{0,1\}^m$, and compute the ciphertext (C_1, C_2) as follows:

$$\begin{aligned} C_1 &= K + M \,(\text{mod}2), \\ C_2 &= \left(\sum_{i=1}^{k} Q_{ID_i} Q_{ID_i}' A_{ID_i}^{\top}\right)\left(\sum_{i=1}^{k} Q_{ID_i} Q_{ID_i}' s\right) + 2e + K)\,(\text{mod}Q). \end{aligned} \tag{13}$$

(iv) Decrypt $(mpk, SK_{I\ D}, I\ D)$: user with identity $I\ D$ in the broadcast set S uses his private key to decrypt ciphertext (C_1, C_2) as follows:

$$\begin{aligned} q_{ID} &= H_1(ID), \\ A_{ID} &= A_0 H_2(ID)^{-1}, \\ K &= (SK_{ID}^{\top})^{-1} (SK_{ID}^{\top} (C_2 \,(\text{mod} q_{ID})) \\ &\quad (\text{mod} q))\,(\text{mod}2) \\ &= (SK_{ID}^{\top})^{-1} (SK_{ID}^{\top} (A_{ID}^{\top} s + 2e + K) \\ &\quad (\text{mod} q))\,(\text{mod}2) \\ &= (SK_{ID}^{\top})^{-1} (SK_{ID}^{\top} (2e + K))\,(\text{mod}2) \\ &= (2e + K)\,(\text{mod}2), \\ M &= C_1 + K \,(\text{mod}2). \end{aligned} \tag{14}$$

5. Analysis of the Proposed Anonymous IBBE Construction

5.1. Parameters and Correctness. Given the security parameter n, the analysis of parameters and correctness for our scheme is as follows.

(i) To ensure that *TrapGen* (q, n) can operate, the following requirements should be met: $m > 6n \log q$ and $q = \text{poly}\,(n)$ [21].

(ii) To guarantee the decryption of the ciphertext, the error term should be less than $q_{ID_i}/2$, and let α, q_{ID_i}, and σ be set as [23, 47]

$$\begin{aligned} \alpha &< \frac{1}{\sigma m \omega (\log m)}, \\ q_{ID_i} &> \alpha m^{3/2} \omega (\log m), \\ \sigma &= m^{3/2} \omega\left(\log^2 n\right). \end{aligned} \tag{15}$$

(iii) Parameters q should always satisfy $q < q_{ID_i}$ and $q = m \log m$.

To ensure that decryption works, we first note that C_2 is designed according to the Chinese remainder theorem, and recall that $Q = q_{ID_1} \cdots q_{ID_n}$ and $Q_{ID_i} = Q/q_{ID_i}$; then, $Q_{ID_i} Q_{ID_i}' \equiv 1 \,(\text{mod} q_{ID_i})$ and $Q_{ID_i} Q_{ID_i}' \equiv 0 \,(\text{mod} q_{ID_j})$ for $i \neq j$. Hence, it would be valid.

$$C_2\left(\mathrm{mod} q_{ID_i}\right) = \left(\left(\sum_{i=1}^{k} Q_{ID_i} Q'_{ID_i} A_{ID_i}^{\top}\right)\left(\sum_{i=1}^{k} Q_{ID_i} Q'_{ID_i} s\right) + 2e + K\right)\left(\mathrm{mod} q_{ID_i}\right)$$
$$= \left(A_{ID_i}^{\top} s + 2e + h\right)\left(\mathrm{mod} q_{ID_i}\right). \tag{16}$$

By the properties of basis delegation, $A_{ID_i} \cdot SK_{ID_i} = 0\,(\mathrm{mod} q)$, where $q < q_{ID_i}$; therefore,

$$\begin{aligned} & SK_{ID_i}^{\top}\left(\left(A_{ID_i}^{\top} s + 2e + h\right)\left(\mathrm{mod} q_{ID_i}\right)\right)(\mathrm{mod} q) \\ & = SK_{ID_i}^{\top}\left(A_{I\ D}^{\top} s + 2e + K\right)(\mathrm{mod} q) \\ & = SK_{ID_i}^{\top}(2e + K)(\mathrm{mod} q). \end{aligned} \tag{17}$$

Finally, we know $K \in \{0,1\}^m$,

$$\begin{aligned} & \left(SK_{ID_i}^{\top}\right)^{-1}\left(SK_{ID_i}^{\top}(2e + K)(\mathrm{mod} q)\right)(\mathrm{mod} 2) \\ & = \left(SK_{ID_i}^{\top}\right)^{-1} SK_{ID_i}^{\top}(2e + K) \\ & = K(\mathrm{mod} 2) \\ & = K. \end{aligned} \tag{18}$$

5.2. Security

Theorem 1. *The above scheme is oAIBBE-sIND-CPA secure if the LWE problem is hard and H_2 is simulated as a random oracle.*

Proof. Suppose there exists a PPT adversary that is able to distinguish the above scheme's ciphertext from random elements with advantage ε. Then, there is a challenger $\mathscr{C}$ with advantage at least $\varepsilon + 1/2$ that distinguishes (A_0, y_0) between the two distributions

$$\begin{aligned} & \left\{(A_0, y_0) | y_0 = A_0^t s_0 + e_0 (\mathrm{mod} q)\right): A_0 \leftarrow \mathbb{Z}_q^{n\times m}, s_0 \leftarrow \mathbb{Z}_q^n, e_0 \leftarrow \Phi_\alpha^m\}, \\ & \left\{\mathrm{Unif}\left(\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m\right)\right\}. \end{aligned} \tag{19}$$

(i) Init: adversary $\mathscr{A}$ outputs two different subsets $S_1^* = \{ID_1^{1\,*}, \ldots, ID_d^{1\,*}\}$ and $S_2^* = \{ID_1^{2\,*}, \ldots, ID_d^{2\,*}\}$ that he wants to distinguish. Challenger $\mathscr{C}$ samples 2 d random matrices $R_1^*, \ldots, R_{2\ d}^* \sim \mathbb{D}_{m\times m}$ by running $R_i^* \leftarrow \mathrm{Sample}\, R(1^n)$ (described in Section 3.1), where all R_i^* are invertible mod q.

(ii) Setup: challenger $\mathscr{C}$ chooses two collision-intractable hash functions H_1 and H_2. H_1 is a division intractability hash function, and H_2 is simulated as a random oracle. Let Q_H be the number of H_2 queries made by $\mathscr{A}$. Let the master public key be A_0, and the master secret key is unknown to $\mathscr{C}$. The system parameters $mpk = (n, m, q, A_0, H_1, H_2)$ are given to $\mathscr{A}$.

(iii) Phase 1: adversary $\mathscr{A}$ adaptively issues queries as follows:

(iv) Random oracle hash queries: $\mathscr{A}$ may adaptively query the random oracle H_2 on any identity ID_i of its choice at any time. $\mathscr{C}$ answers the query as follows.

If $ID_i \in S_1^* \cup S_2^*$, define $H_2(ID_i) \leftarrow R_i^*$, return R_i^* to adversary $\mathscr{A}$, and save the tuple (ID_i, R_i^*) in a list $\mathscr{L}$.

If $ID_i \notin S_1^* \cup S_2^*$, sample a random matrix $\widetilde{R}_i \leftarrow \mathrm{Sample}\, R(1^n)$, where $\widetilde{R}_i$ is invertible mod q, compute $A_i = A_0 \cdot (\widetilde{R}_i)^{-1}(\mathrm{mod} q)$, and then run sample R with basis (A_i) (described in Section 3.1) to obtain a random matrix $\widehat{R}_i \sim \mathbb{D}_{m\times m}$ and a short basis T_{B_i} for

$$\begin{aligned} B_i & = A_i \widehat{R}_i^{-1}(\mathrm{mod} q) \\ & = A_0 \cdot \widetilde{R}_i^{-1} \widehat{R}_i^{-1}(\mathrm{mod} q) \\ & = A_0 \cdot \widetilde{R}_i^{-1} \widehat{R}_i^{-1}(\mathrm{mod} q) \\ & = A_0 \cdot R_i^{-1}(\mathrm{mod} q). \end{aligned} \tag{20}$$

Save the tuple $(ID_i, R_i, B_i, T_{B_i})$ in a list $\mathscr{L}$ for future use, and return $H_2(ID_i) \leftarrow R_i$ to adversary $\mathscr{A}$.

(i) Secret key queries: $\mathscr{A}$ makes interactive key extraction queries on arbitrary identity ID_i. $\mathscr{C}$ answers a query on ID_i as follows:

If $ID_i \in S_1^* \cup S_2^*$, $\mathscr{C}$ aborts and fails.
If $ID_i \notin S_1^* \cup S_2^*$, $\mathscr{C}$ retrieves the saved tuple $(ID_i, R_i, B_i, T_{B_i})$ from the hash oracle query list $\mathscr{L}$; else, it runs the random oracle hash query on ID_i. Let $B_i = A_0 \cdot R_i^{-1}(\mathrm{mod} q)$ and T_{B_i} be a short basis for $\Lambda q^{\perp}(B_i)$, and return T_{B_i} to adversary $\mathscr{A}$.

Notice that B_i is exactly the encryption matrix for ID_i, and therefore, T_{B_i} is a trapdoor for $\Lambda_q^{\perp}(B_i)$.

Challenge: adversary $\mathscr{A}$ chooses two equal-length messages $M_1, M_2 \in \{0,1\}^m$ and sends to challenger $\mathscr{C}$. Challenger $\mathscr{C}$ chooses at random a symmetric key $K \in \{0,1\}^m$ and a random bit $\beta \in \{0,1\}$; challenger $\mathscr{C}$ computes $Q^{\beta *} = q_{ID_1^{\beta *}} \cdots q_{ID_d^{\beta *}}$, where $q_{ID_j^{\beta *}} = H_1(ID_j^{\beta\,*})$ and $j \in S_\beta^*$, and then returns the challenge ciphertext to adversary $\mathscr{A}$.

$$\begin{aligned} C_1 & = M_\beta + K(\mathrm{mod} 2), \\ C_2 & = 2y_0 + K\left(\mathrm{mod} Q^{\beta *}\right). \end{aligned} \tag{21}$$

(ii) Phase 2: adversary $\mathscr{A}$ adaptively issues queries as Phase 1.

(iii) Guess: adversary $\mathscr{A}$ outputs a guess bit β', and $\mathscr{A}$ wins the game if $\beta = \beta'$.

(iv) Analysis: in the following, we analyse the correctness of the challenge ciphertext.

(v) On the one hand, if y_0 is a uniformly random matrix, then the challenge ciphertext is also uniformly random, regardless of the choice of β. Hence, in this case, $\mathscr{C}$ outputs 1 with probability at most 1/2.

(vi) On the other hand, if $y_0 = A_0 s_0 + e_0 \; (\mathrm{mod} q)$, then the challenge ciphertext is $2y_0 + K(\mathrm{mod} Q) = 2(A_0 s_0 + e_0) + K(\mathrm{mod} Q)^{\beta *} = 2 \quad (A_{ID_i^{\beta *}} \cdot R_i^* \cdot s_0 + e_0) + K(\mathrm{mod} Q^{\beta *}) = 2(A_{ID_i^{\beta *}} \cdot R_i^* \cdot s_0 + e_0) + K(\mathrm{mod} Q^{\beta *}) = (A_{ID_i^{\beta *}} \cdot 2R_i^* \cdot s_0 + 2e_0 + K)(\mathrm{mod} Q^{\beta *}) = (A_{ID_i^{\beta *}} \cdot s' + 2e_0 + K) \quad (\mathrm{mod} Q^{\beta *}) = (\sum_{i=1}^{k} Q_{ID_i^{\beta *}} Q_{ID_i^{\beta *}}{}' A_{ID_i^{\beta *}}) \quad (\sum_{i=1}^{k} Q_{ID_i^{\beta *}} Q_{ID_i^{\beta *}}{}' s') + 2e_0 + K) (\mathrm{mod} q_{ID_i^{\beta *}}) = (\sum_{i=1}^{k} Q_{ID_i^{\beta *}} Q_{ID_i^{\beta *}}{}' A_{ID_i^{\beta *}})(\sum_{i=1}^{k} Q_{ID_i^{\beta *}} Q_{ID_i^{\beta *}}{}' s') + 2e_0 + K)(\mathrm{mod} Q^{\beta *})$.

$s' = R_i^* \cdot 2s_0 (\mathrm{mod} q_{ID_i^{\beta *}})$ is uniformly distributed (since Q and 2 are relatively prime). This is identical to the output distribution of the real ciphertext.

Hence, if adversary $\mathscr{A}$ succeeds in guessing the right M_β and S_β with probability $1/2 + \varepsilon$, then challenger $\mathscr{C}$ will correctly guess the nature of the LWE oracle with probability at least $1/2 + \varepsilon/2$. This concludes the proof of the security reduction.

Remark. The above scheme cannot achieve the anonymity for the insider attacker. Because any authorized receiver can obtain the private information *s, e*, and *K*, he/she uses *s, e*, and *K* to decrypt C_2. The decryption process is similar to Thrapdoor Sampling Algorithm of Section 2.1 Therefore, in order to ensure whether or not ID is an authorized receiver, adversary $\mathscr{A}$ only needs to calculate whether $C_2(\mathrm{mod} H_1(ID))$ and $(A_{I\,D}^{\top} s + 2e + h) \; (\mathrm{mod} H_1(ID))$ are equal. If yes, ID is an authorized receiver; otherwise, ID is not an authorized receiver.

6. Conclusions

We propose a lattice-based anonymous IBBE scheme employing the Chinese remainder theorem and lattice basis delegation in fixed dimensions. Our scheme achieves chosen-plaintext security in the random oracle model and is with multiple attractive properties, such as constant-size private/public key and ciphertext and constant encryption/decryption overhead.

Data Availability

All the data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Science Foundation of China (NSFC) (Grant nos.61902067 and 62102166), Foundation for Young Innovative Talents in Ordinary Universities of Guangdong (2018KQNCX255), Opening Project of Guangdong Province Key Laboratory of Information Security Technology (Grant no.2020B1212060078), Dongguan Science and Technology of Social Development Program (2020507140146), Dongguan University of Technology (2021KTSCX134), Key-Area Research and Development Program of Guangdong Province (Grant no. 2020B0101360001), Guangdong Basic and Applied Basic Research Foundation (Grant no. 2020A1515111175), and Guangdong Natural Science Key Field Project (2019KZDZX1008).

References

[1] A. Lekbich, A. Belfqih, C. Zedak, J. Boukherouaa, and F. El Mariami, "A secure wireless control of remote terminal unit using the internet of things in smart grids," in *Proceedings of the 6th International Conference on Wireless Networks and Mobile Communications, WINCOM 2018*, pp. 1–6, IEEE, Marrakesh, Morocco, October 2018.

[2] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying iot security: an exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[3] A. Fiat and M. Naor, "Broadcast encryption," in *Proceedings of the Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference*, pp. 480–491, Santa Barbara, CA, USA, August 1993.

[4] X. Du, Y. Wang, J. Ge, and Y. Wang, "An id-based broadcast encryption scheme for key distribution," *IEEE Transactions on Broadcasting*, vol. 51, no. 2, pp. 264–266, 2005.

[5] B. Malek and M. Ali, "Adaptively secure broadcast encryption with short ciphertexts," *International Journal Network Security*, vol. 14, no. 2, pp. 71–79, 2012.

[6] C Delerablée, P. Pascal, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in *Proceedings of the Pairing-Based Cryptography - Pairing 2007, 1st International Conference*, pp. 39–59, Tokyo, Japan, July 2007.

[7] X. Xiaodong Lin, X. Xiaoting Sun, P.-H. Pin-Han Ho, and X. Xuemin Shen, "GSIS: a secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007.

[8] Y. Jung, Y. Nam, J. Kim, W. Jeon, H. Lee, and D. Won, "Key management scheme using dynamic identity-based broadcast encryption for social network services," *Lecture Notes in Electrical Engineering*, vol. 279, pp. 435–443, 2014.

[9] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proceedings of the Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference*, pp. 41–62, Santa Barbara, CA, USA, August 2001.

[10] Y. Dodis and N. Fazio, "Public key broadcast encryption for stateless receivers," in *Proceedings of the Security And Privacy In Digital Rights Management, ACM CCS-9 Workshop, DRM 2002*, pp. 61–80, Springer, Washington, DC, USA, November 2002.

[11] D. Boneh, G. Craig, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proceedings of the Advances in Cryptology - CRYPTO 2005:*

25th Annual International Cryptology Conference, pp. 258–275, Santa Barbara, CA, USA, August 2005.

[12] B. Adam, D. Boneh, and B. Waters, “Privacy in encrypted content distribution using private broadcast encryption,” in *Proceedings of the Financial Cryptography and Data Security, 10th International Conference, FC 2006*, pp. 52–64, Anguilla, West Indies, February 2006.

[13] G. Craig and B. Waters, “Adaptive security in broadcast encryption systems (with short ciphertexts),” in *Proceedings of the Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 171–188, Cologne, Germany, April 2009.

[14] W. Peter, “Shor. Algorithms for quantum computation: discrete logarithms and factoring,” in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, Santa Fe, NM, USA, November 1994.

[15] R. Bendlin, “Lattice-based cryptography,” *Lecture Notes in Computer Science*, vol. 4117, no. 1-2, pp. 131–141, 2013.

[16] M. Ajtai, “Generating hard instances of lattice problems (extended abstract),” in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pp. 99–108, Philadelphia, PA, USA, May 1996.

[17] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 84–93, Baltimore, MD, USA, May 2005.

[18] D. Micciancio and S. Goldwasser, “Complexity of lattice problems: a cryptographic perspective,” *Kluwer International Series in Engineering and Computer Science*, Kluwer Academic Publishers, Boston, MA, USA, 2002.

[19] B. Libert, K. G. Paterson, and E. A. Quaglia, “Anonymous broadcast encryption: adaptive security and efficient constructions in the standard model,” in *Proceedings of the Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography*, pp. 206–224, Darmstadt, Germany, May 2012.

[20] F. Wang, A. Wang, and C. Wang, “Lattice-based dynamical and anonymous broadcast encryption scheme,” in *Proceedings of the 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2015*, pp. 853–858, Krakow, Poland, November 2015.

[21] S. Agrawal, D. Boneh, and X. Boyen, “Efficient lattice (H)IBE in the standard model,” in *Proceedings of the Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 553–572, Monaco, French Riviera, May 2010.

[22] R. Gennaro, S. Halevi, and T. Rabin, “Secure hash-and-sign signatures without the random oracle,” in *Proceedings of the Advances in Cryptology - EUROCRYPT ’99, International Conference on the Theory and Application of Cryptographic Techniques*, J. Stern, Ed., Springer, Prague, Czech Republic, pp. 123–139, May 1999.

[23] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, “Bonsai trees, or how to delegate a lattice basis,” in *Proceedings of the Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 523–552, French Riviera, France, May 2010.

[24] Adi Shamir, “Identity-based cryptosystems and signature schemes,” *Lecture Notes in Computer Science*, vol. 196, no. 2, pp. 47–53, 1985.

[25] G. Craig, C. Peikert, and V. Vaikuntanathan, “Trapdoors for hard lattices and new cryptographic constructions,” in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pp. 197–206, Victoria, Canada, May 2008.

[26] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” *Journal of the ACM*, vol. 56, no. 6, pp. 1–40, 2009.

[27] M. Ajtai, “Generating hard instances of the short basis problem,” in *Proceedings of the Automata, Languages and Programming, 26th International Colloquium, ICALP’99*, pp. 1–9, Prague, Czech Republic, July 1999.

[28] G. Craig and Alice Silverberg, “Hierarchical id-based cryptography,” in *Proceedings of the Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 548–566, Queenstown, New Zealand, December 2002.

[29] J. Horwitz and B. Lynn, “Toward hierarchical identity-based encryption,” in *Proceedings of the Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 466–481, Amsterdam, The Netherlands, April 2002.

[30] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, “Bonsai trees, or how to delegate a lattice basis,” in *Advances in Cryptology –EUROCRYPT*, pp. 523–552, Springer Berlin Heidelberg, Berlin, Germany, 2010.

[31] D. B. Shweta Agrawal and X. Boyen, “Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE,” in *Proceedings of the Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference*, pp. 98–115, Santa Barbara, CA, USA, August 2010.

[32] S. Amit and B. Waters, “Fuzzy identity-based encryption,” in *Proceedings of the Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, R. Cramer, Ed., Springer, Aarhus, Denmark, pp. 457–473, May 2005.

[33] X. Boyen, “Attribute-based functional encryption on lattices,” in *Theory of Cryptography*, S. Amit, Ed., in *Proceedings of the Theory Of Cryptography - 10th Theory Of Cryptography Conference, TCC 2013*, pp. 122–142, Springer, Tokyo, Japan, March 2013.

[34] C. . Delerablée, “Identity-based broadcast encryption with constant size ciphertexts and private keys,” in *Proceedings of the Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security*, Kuching, Malaysia, December 2007.

[35] D. Boneh, B. Waters, and M. Zhandry, “Low overhead broadcast encryption from multilinear maps,” *Proceedings, Part I*, in *Proceedings of the Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference*, Santa Barbara, CA, USA, August 2014.

[36] J. Jongkil Kim, W. Susilo, and J. Seberry, “Adaptively secure identity-based broadcast encryption with a constant-sized ciphertext,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 679–693, 2015.

[37] B. Dan and M. Zhandry, “Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation,” *Algorithmica*, vol. 8616, no. 4, pp. 1–53, 2016.

[38] N. Fazio and I. Milinda Perera, “Outsider-anonymous broadcast encryption with sublinear ciphertexts,” in *Proceedings of the Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography*, Darmstadt, Germany, May 2012.

[39] K. He, J. Weng, M. H. Au, Y. Mao, R. H. Deng, and Deng, “Generic anonymous identity-based broadcast encryption with chosen-ciphertext security,” in *Information Security and*

Privacy, pp. 207–222, Springer International Publishing, Berlin, Germany, 2016.

[40] J. Wang and J. Bi, "Lattice-based identity-based broadcast encryption scheme," *IACR Cryptology ePrint Archive*, vol. 288, 2010.

[41] A. Georgescu, "Anonymous lattice-based broadcast encryption," in *Proceedings of the Information and Communicatiaon Technology - International Conference, ICT-EurAsia 2013*, Yogyakarta, Indonesia, March 2013.

[42] Z. Brakerski and V. Vaikuntanathan, "Lattice-inspired broadcast encryption and succinct ciphertext-policy ABE," *IACR Cryptol. ePrint Arch.*vol. 191, 2020.

[43] D. Boneh, G. Craig, S. Gorbunov et al., "Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits," in *Proceedings of the Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, P. Q. Nguyen and E. Oswald, Eds., , Copenhagen, Denmark, May 2014.

[44] S. Agrawal and S. Yamada, "Optimal broadcast encryption from pairings and LWE," in *Proceedings of the Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, A. Canteaut and Yuval Ishai, Eds., , Zagreb, Croatia, May 2020.

[45] D. Micciancio and S. Goldwasser, "Complexity of lattice problems - a cryptograhic perspective," *Kluwer International Series in Engineering and Computer Science*, Springer, Berlin, Germany, 2002.

[46] W. Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall, Hoboken NJ, USA, 2003.

[47] G. Craig, S. Halevi, and V. Vaikuntanathan, "A simple bgn-type cryptosystem from lwe," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 506–522, French Riviera, France, June 2010.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9951777, 26 pages
https://doi.org/10.1155/2021/9951777

Research Article

BusCount: A Provable Replay Protection Solution for Automotive CAN Networks

Daniel Zelle and Sigrid Gürgens

Fraunhofer Institute for Secure Information Technology, Darmstadt, Germany

Correspondence should be addressed to Daniel Zelle; daniel.zelle@sit.fraunhofer.de

Received 15 March 2021; Revised 26 July 2021; Accepted 21 August 2021; Published 30 November 2021

Academic Editor: Anjia Yang

Information technology has become eminent in the development of modern cars. More than 50 Electronic Control Units (ECUs) realize vehicular functions in hardware and software, ranging from engine control and infotainment to future autonomous driving systems. Not only do the connections to the outside world pose new threats, but also the in-vehicle communication between ECUs, realized by bus systems such as Controller Area Network (CAN), needs to be protected against manipulation and replay of messages. Multiple countermeasures were presented in the past making use of Message Authentication Codes and time stamps and message counters, respectively, to provide message freshness, most prominently AUTOSAR's Secure Onboard Communication (SecOC). In this paper, we focus on the latter ones. As one aspect of this paper, using an adequate formal model and proof, we will show that the currently considered solutions exhibit deficiencies that are hard if not impossible to overcome within the scope of the respective approaches. We further present a hardware-based approach that avoids these deficiencies and formally prove its freshness properties. In addition, we show its practicability by a hardware implementation. Finally, we evaluate our approach in comparison to counter-based solutions currently being used.

1. Introduction

Information technology has become an integral part of modern vehicles. More than 50 interconnected Electronic Control Units (ECUs) realize vehicular functions in hardware and software ranging from engine control and connected infotainment systems to future autonomous driving systems. The in-vehicle communication between ECUs is realized with bus systems like CAN (Controller Area Network Bus [1]). Further, vehicles communicate with the outside world (e.g. with their manufacturer's backend systems, with the garage's On-Board-Diagnose (OBD) devices) via different communication interfaces. Usually, these interfaces are not strictly separated from the in-vehicle network (the OBD port for example must have access to a car's ECUs to extract error codes). This poses serious security threats, one of the possible attack vectors being in-vehicle communication. The vehicle owner can for example install a tuning box to suppress or inject messages that control engine operations in order to achieve more horsepower. This in turn may damage the engine and violate the warranty. Moreover, third-party devices connected to the OBD port can inject messages to the regular in-vehicle network. In [2], Koscher et al. have shown various attack techniques like *Packet Sniffing and Targeted Probing*, *Fuzzing*, and *Reverse-Engineering*.

Multiple countermeasures were presented in the past to protect in-vehicle networks (see Section 2.4). Early work can be traced back to EVITA [3] that introduced Message Authentication Code (MAC) truncation in order to cope with the small bandwidth of field buses such as CAN. This approach has been adapted by AUTomotive Open System ARchitecture AUTOSAR in SecOC [4]. Including a freshness value in a message's MAC can in principle prohibit fuzzing or replay attacks. Most of the current approaches consider a monotonic counter value.

In this paper, we discuss our new counter-based approach BusCount based on our ideas introduced in [5], present its full formal verification, discuss its implementation and provide a practical evaluation. We further oppose it to a generic system that captures the principles of today's counter-based approaches for freshness protection. The principle idea of our approach is to use the messages that are sent on a specific bus as a pulse generator for the counter of this bus, resulting in only one counter per bus. To cope with the loss of counter values e.g. caused by technical problems or an attack, our approach includes counter synchronization. Further, it requires the sending and reception of messages to be processed simultaneously to MAC generation and verification. Therefore, we propose a hardware-based solution: The CAN controller is enhanced by the functionality to maintain a counter and to manage MAC generation and verification while the main ECU processors can be inactive at times.

In the next section, we present the principles of in-vehicle communication based on CAN, our attack model, the protection goals we will address, current work concerning the security of CAN-based communication, and finally the characteristics of the counter-based approaches currently being discussed. Section 3 then describes the details of our approach BusCount. The following Section 4 briefly introduces our Security Modeling Framework SeMF that is then used in Sections 5 and 6, respectively, to formally model and verify both the generic counter approach and our bus counter approach. Based on these results, in Section 7.1 we present a comparison of the security aspects of both approaches while Section 7.2 introduces our proof of concept implementation showing its practicability and design decisions that substantiate our formal proof. Finally, we conclude with Section 8.

2. Principles of In-Vehicle Communication

CAN bus is the core technology for onboard communication in vehicles. Brakes, acceleration, and many further essential features are controlled by ECUs that communicate using CAN bus messages. An overview of different network structures is given in [6]. The CAN network is accessible via the OBD port allowing repair shops to access the car network. Modern vehicles also have connections through infotainment systems as well as telematic control units (TCUs) connecting the CAN bus to the outside world. By connecting the in-vehicle communication with the outside world, the necessity arose to protect its messages against malicious entities.

In this section, we describe the basics of CAN bus communication, the attacker model we take as a basis, the most relevant protection goals, and the current approaches for protecting these goals.

2.1. Basics of CAN Communication in Vehicles.

The CAN bus, specified in [1], is a field bus where each entity connected to it is able to send messages and listen to every message sent on the bus. The maximum transfer rate of the highspeed-CAN is 1 Mbit/s.

A standard CAN message consists of 7 segments (Figure 1): "Start of frame" bit, a message identifier, a control field, a data field, a checksum, a confirmation field, and an "end of frame" sequence.

The 11 bit identifier which is the second section of a CAN message also represents the message's priority which is used to handle collisions. The CAN bus uses Carrier Sense Multiple Access/Collision Resolution (CSMA/CR) to prevent collisions: All ECUs start sending a CAN message simultaneously and monitor its identifier while sending. In case a dominant 0 overwrites a 1 the ECU with the lower priority stops its transmission, thereby avoiding collisions.

During the transmission of the message every ECU calculates the CRC (cyclic redundancy check) over the message and checks the correctness of it as soon as it gets transmitted by the sender. In case of a problem (e.g. if the CRC check has failed) an ECU interrupts a transmission with an error frame that invalidates the message for all receivers. Furthermore, an error counter is increased by 1 for every receiver and by 8 for the sender. Every successfully transmitted message decrements the counter. If a counter reaches 128, the ECU disables its CAN connection. This mechanism ensures that damaged ECUs do not block the entire bus communication.

Successors of CAN, like CAN FD or CAN XL, differ mainly in the frequency of transmitting data payload. CAN FD can transmit up to 64 bytes while CAN XL can handle 2048 bytes.

2.2. Our Attack Model.

Attacks on in-vehicle communication have been presented first by Kocher et al. in [2]. These attacks concern manipulation of brake control and vehicle acceleration via CAN Bus by message injection. In the real world, attacks on vehicle networks have been observed that manipulate in-vehicle communication by attaching devices to the bus system, like tuning devices, AdBlue emulators [7], unauthorized OBD dongles [8], etc.

In our attack model, an attacker can send arbitrary messages on any bus she has access to. Moreover, she can overhear and record all communication on a bus she is connected to and replay all recorded messages. Finally, an attacker is able to flip bits of messages or send an error frame. This enables her to invalidate messages for other ECUs after having recorded them herself. In our scenario, an attacker does not have access to any cryptographic keys. This also includes that the attacker cannot manipulate ECUs by e.g. corrupting firmware which in turn implies that legitimate ECUs always act correctly. Furthermore, the attacker is not able to manipulate processes or storage of ECUs by physical attacks. In addition, the attacker is not able to produce a counter overflow as a sufficient counter length is chosen to prohibit this sort of attacks.

2.3. Protection Goals.

A secure CAN bus communication in a vehicle needs to fulfill a set of requirements to prevent previously introduced attacks. These include e.g. data integrity, confidentiality, and availability. In the following, we explain those requirements we specifically address in this paper.

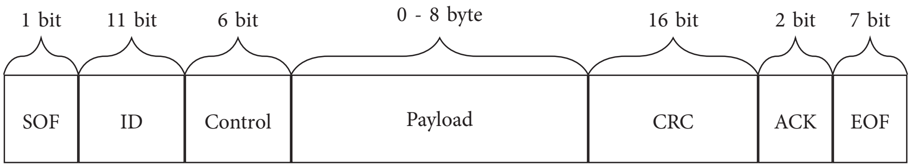


Figure 1: A single CAN frame.

Data Origin Authenticity: A message in a vehicle network should be accepted if and only if it has authentically for the recipient been sent by another valid member of the network. This property prevents attackers from manipulating messages or sending messages on the bus system from additional devices or replaced components without the intended recipient noticing.

Immediacy: Contrary to many other IT-systems, it is important for an automotive system to receive and process messages within a certain time frame. Once this time frame has passed, messages might be authentic, but can still cause fatal results, e.g. if breaking signals sent by the anti-lock braking system are processed too late. Immediacy expresses the fact that a message sent at time t_1 is accepted until t_2 if and only if $t_2 - t_1$ does not exceed a specified limit.

Non-repeatability: The last important property for an automotive network is non-repeatability: If a message is accepted at time t_1, the same message is not accepted at any later point in time. Thus, an attacker cannot eavesdrop on a message and successfully replay it at a later point in time.

Many articles do not distinguish between the two above properties and use the more abstract concept of "message freshness" with "message replay" seeking to violate freshness. We adapt to this notation and distinguish the specific characteristics when necessary.

2.4. State of the Art CAN Bus Security.

The security of bus communication in current vehicle networks has already been discussed in literature and standardization. Early work on MAC truncation for secure CAN bus communication can be traced back to [3]. In this section, we give an overview of state of the art with a focus on replay protection in CAN bus systems and compare the techniques.

A lot of CAN bus security approaches introduce message authentication mechanisms, but not all introduce replay protection. The latest example of an approach without freshness values is TOUCAN: A proTocol tO secUre Controller Area Network presented by Bella et al. [9], which introduces a 24 bit truncated Chaskey MAC and a SPECK64 encryption for each CAN message.

A more exotic approach for replay protection is used in LCAP by Hazem et al. [10]. LCAP appends a truncated element of a hash chain to the CAN message and encrypts the resulting message. An HMAC secures the transmission of the last element of the hash chain to initialize the communication. Woo et al. [11] periodically change HMAC keys to prevent replay attacks.

Nürnberger and Rossow [12] developed VatiCAN, an HMAC based authentication procedure that sends a MAC in a separate message following the original CAN message. The MAC is then validated with a delay of about 4 ms. Replay protection is implemented with a monotonically incremented counter, its starting value being a random nonce generated by a central component for every message ID. The authors recommend this procedure only for a few CAN messages since it increases the bus load. Van Bulck et al. improved this approach in [13] by introducing software isolation and attestation as well as key update mechanisms.

Hartkopp et al. presented a further approach to introduce freshness to CAN messages. MaCAN [14] formally verified in [15] introduces a central trusted time server which distributes time information over the network. This information is used as freshness value for message authentication.

AUTOSAR specifies the protection of communication in vehicle networks based on a MAC and a freshness value. The specification of the Secure Onboard Communication (SecOC) [4] module suggests to add a truncated timestamp or message counter and a truncated authenticator to every message. The specific counter mechanism is based on splitting the counter (with a maximal length of 96 bits) into three different parts: the so-called "trip counter" that only changes essentially with every new trip, a "reset counter" that is reset periodically, and the actual "message counter". Only the trip counter is stored in non-volatile memory, thus mitigating loss of counter values in case of sudden ECU shutdown. The truncated freshness value has a length between 0 and 8 bit. The truncated authenticator consists of the first 24 to 28 bits of the MAC covering the full freshness value and the message.

Similar to SecOC many approaches in literature use counters and an application-level protocol to ensure replay protection. Kurachi et al. [16] suggest attaching a truncated MAC (8 bit) and a truncated monotonic counter (4 bit) to a message. A monitoring node verifies messages during transmission and overwrites invalid messages with an error frame. ECUs do not verify messages. Groll et al. [17] suggest an initialization phase to form groups of ECUs. These groups generate a shared symmetric key using an asymmetric key exchange. Within these groups, ECUs use the shared secret for authentic and confidential encryption. A counter should be part of the message to protect against replay attacks. Lin et al. presented an approach in [18] with symmetric keys for message authentication. A sender calculates a MAC for every receiver. Every ECU also holds two counters for replay protection per message ID, the last counter it has sent and the last one it has received. Every receiver can verify the MAC and process its corresponding message. The LeiA protocol by Radu et al. [19] is another solution that transfers MAC and counter value in a separate message. Every ECU has a session key for each relevant message ID derived from a long-term symmetric key and renewed after a certain period.

VeCure [20] is a CAN authentication framework similar to VatiCAN. The authentication value is also transmitted via a separate message, but contrary to VatiCAN the second message includes a *Node-ID* besides a *Message Counter* and a four byte HMAC value.

Alternatively, several approaches suggest the use of CAN+ [21], a protocol extension for CAN allowing to transport 120 bit additional data. The first approach is CANAuth presented by Van Herrewege et al. [22]. Another one is LiBrA-CAN [23]. LiBrA-CAN introduces (Linearly) Mixed MAC, which mixes multiple MACs of one message generated with different keys allowing receivers to verify a MAC even though they do not know all keys. The approach allows making sure receivers cannot impersonate a sender in a properly organized group. Both approaches send counter values in their messages to protect against replay attacks.

Some works are also considering the implementation of a secure CAN bus controller. Their approaches introduce the calculation of MACs, denial of service countermeasures, or intrusion prevention mechanisms. [24] implemented a CAN controller including a physical unclonable function implementation, key generation and storage, and encryption and decryption allowing authenticated communication over CAN. However, the approach does not consider replay protection. Ueda et al. presented a CAN controller with integrated HMAC in [25]. To ensure replay protection a truncated monotonic value of 4 bits is part of every message. Messages which are not authentic are destroyed while correct messages update the counter.

A new approach by Groza et al. [26] suggests replacing CAN IDs with a specific MAC-based algorithm that preserves the order of CAN IDs. In predefined time intervals the counter included in the MAC is incremented thus the IDs change. This approach increases the resistance against reverse engineering and denial of service attacks related to a specific ID. It does not provide data integrity and authenticity which needs an additional security protocol as mentioned in the paper. Moreover freshness is not guaranteed since the counter used in the MAC of the CAN ID does not change with every message. In case of constantly changing counter values (IDs) and if a significant limitation of ID range is acceptable this can be a viable alternative to transfer of fresh counter values.

We observed that most of the presented approaches (cf. Table 1) have similar ways to ensure replay protection and authentication of messages. All approaches add a MAC to a CAN message. While a MAC provides authenticity of a message, only in combination with a freshness value replay and delay attacks can be mitigated. Most approaches introduce a counter value to provide freshness since the usage of time or nonce values has disadvantages, discussed e.g. in [27]. The transmission of MAC and freshness values is either realized in an additional message or achieved by including truncated values in the same message. The verification of a complete message is performed by the receiver or an additional node. In the following section, we present a detailed generic model covering the characteristics of the current counter-based approaches for freshness. This model is then compared to our approach based on formal verifications of the security goals.

2.5. The Generic Counter Concept. Considering the recent research, we simplified the approaches in order to generate an abstract model to evaluate the security of software-based freshness techniques compared to our approach. Since a large majority of approaches favor counters for freshness values, we focus on this technique.

Figure 2 illustrates the abstract protocol we assume. To transmit a CAN message m which contains the $I\ D$ and payload data msg an ECU first calculates a MAC covering m and a local counter c_a derived by incrementing the previously used one (steps 1 and 2). For our analysis, the choice of the MAC algorithm is not important. In the next steps, m, c_a and the authentication tag are concatenated (step 3) and the values are transmitted (step 4). Note that this transmission is not necessarily processed with one CAN message only, different techniques could apply here. Finally, a receiver gets the message, verifies the MAC and tests if its local counter c_b is smaller than the counter in s. The check is not necessarily performed by the same entity which later processes m. If both checks are successful, the local counter is set to the counter in s and the message can be processed. Otherwise, the message is discarded.

Only some approaches consider an explicit synchronization of counter values which is necessary in case an ECU loses its counter value, e.g. due to a software error, a power loss (engine stop or malfunction) or an ECU without persistent memory. Most approaches that use synchronization introduce a central system sending an authenticated message containing the current counter value. In case a sender has an incorrect counter, the value needs to be provided by a dedicated entity or some client. In both cases the counter value is transmitted in the payload CAN message with a reserved ID. This message is secured identically to regular messages.

Even though the generic counter protocol is fairly simple it represents the characteristic properties of all above mentioned protocols that increment counters after successful validation of the message. The fact that the local counters of message recipients only change when a message is accepted is a very important characteristic property. Consequently, these protocols cannot prohibit so-called delay attacks, as will be formally shown in Section 5.2. For such an attack, the adversary with the abilities described in Section 2.2 stores and then invalidates a message and all subsequent ones related to the same counter. The reinserted message will then be accepted by the intended recipients at any later point in time if no further countermeasures are taken.

3. BusCount: A Hardware Based Bus Counter Solution

In this section we describe BusCount, our approach for a hardware based secure CAN bus communication, which eliminates the possibilities of attackers with the abilities presented in Section 2.2. We first introduce our approach to ensure immediacy, non-repeatability, and authenticity of messages on the bus and then elaborate on the synchronization mechanism for freshness values.

Table 1: Comparison of different authentication approaches for CAN-Bus (HW: Hardware, SW: Software, C : Counter, T : Timestamp, N : Nonce, *: not described)

	HW change	SW change	Central component	Freshness technique	MAC	Encryption	Transfer techniques for MAC	Syncronisation of freshness value
AUTOSAR [2]	-	✓	-	C / T	✓	-	28 bit data field	✓
CaCAN [24]	✓	✓	✓	C	✓	-	8 bit data field	-
CANAuth [35]	✓	✓	-	C	✓	-	CAN+	(✓)
Groll et al. [14]	-	✓	✓	C	✓	✓	*	-
LeiA [31]	-	✓	-	C	✓	-	sep. message	✓
LibrA-CAN [16]	-	✓	✓	C	✓	-	CAN+	-
Lin et al. [25]	-	✓	-	C	✓	-	*	(✓)
Ueda et al. [34]	✓	✓	✓	C	✓	-	8 bit data field	-
VeCure [37]	-	✓	-	C	✓	-	separate message	-
MaCAN [20]	-	✓	✓	T	✓	-	32 bit data field	✓
VatiCAN [28]	-	✓	✓	C + N	✓	-	separate message	✓
vulCAN [7]	✓	✓	✓	C + N	✓	-	separate message	(✓)
Woo et al. [38]	-	(✓)	(✓)	key refresh	✓	✓	CAN-FD	-
LCAP [21]	-	✓	-	hash chain	-	✓	16 bit extended ID	✓
TouCAN [3]	-	✓	-	-	✓	✓	24 bit data field	-
Siddiqui et al. [33]	✓	✓	✓	-	✓	✓	data field	-
CAN-TORO [15]	✓	✓	✓	Authenticated ID	-	-	-	(✓)

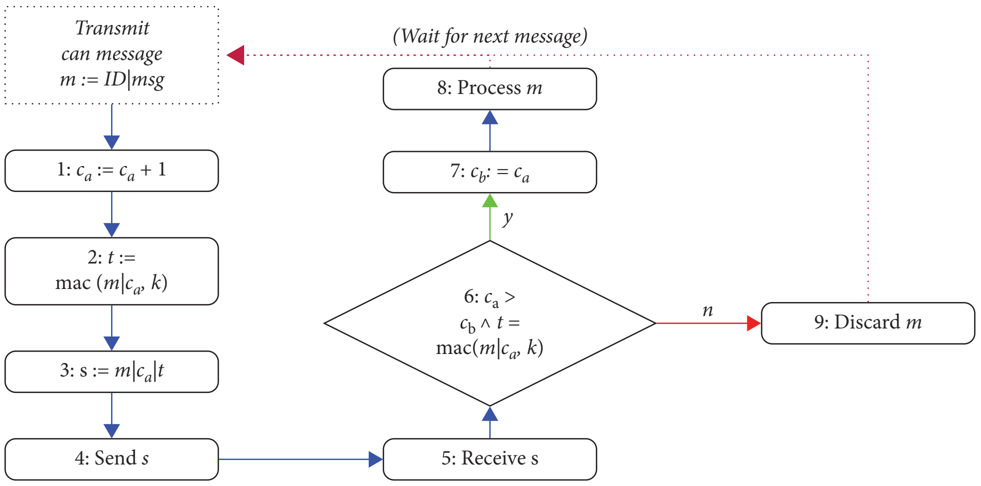


Figure 2: Process of Generic Counter Communication.

In a bus system, each participant can see and thus count every message written to the bus. Hence, the number of messages sent on a particular bus is an inherent part of the system that can serve as a bus specific counter [5] known by all devices connected to it. Consequently, there is no need to send counters. Each bus of a vehicle system is equipped with

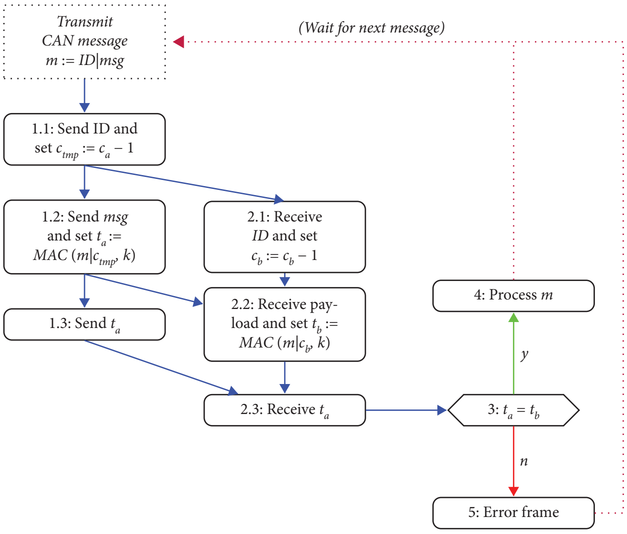

Transmit
CAN message
$m := ID|msg$
(Wait for next message)
1.1: Send ID and set $c_{tmp} := c_a - 1$
1.2: Send msg and set $t_a :=$ $MAC\ (m|c_{tmp}, k)$
2.1: Receive ID and set $c_b := c_b - 1$
1.3: Send t_a
2.2: Receive payload and set $t_b :=$ $MAC\ (m|c_b, k)$
4: Process m
y
2.3: Receive t_a
3: $t_a = t_b$
n
5: Error frame

Figure 3: Process of CAN message transmission of BusCount.

its own counter. Its value changes automatically with each new message: An ECU sending/receiving a message reduces its local counter value by 1 and authenticates/verifies the message including the counter with a MAC. This idea can be also applied to any other bus network.

The procedure of BusCount is described in more detail as follows (see Figure 3 for a schematic representation):

First, a sender ECU starts transmitting a message m which is composed of a message ID and the payload msg. As soon as the transmission starts, the local counter c_a of the sender is temporarily decremented (step 1.1). The counter is decremented instead of the usual incrementation because 0 is the dominant bit on the CAN bus thus a lower counter can overwrite a larger counter. This property is used for the synchronization of the counter explained in the next section. A receiver ECU, when receiving the start of the message, decrements its local counter c_b and uses the result as its new counter value (step 2.1). Since sending and receiving of messages is processed simultaneously and thus the sender also receives its own message, it decrements its counter analogously. After the counters are decremented the sender starts sending mgs and both receiver and sender start calculating the MAC over the message m and their respective local counter using a shared key k. Finally, the tag t_a of the sender's MAC is transmitted (step 1.3) and received (step 2.3). All ECUs now evaluate the tag.

If the evaluation is positive m can be processed. Otherwise, the receivers whose verification failed immediately transmit an error frame which has the effect that m is discarded by every ECU. Note that this effects also ECUs that do not implement the protocol: They will discard messages overwritten with an error frame. In case multiple errors occur, a synchronization is necessary.

3.1. Synchronization. Multiple transmission failures may indicate that an ECU is not synchronized. This situation can occur e.g. if the ECU is switched off without having been able to store the current correct counter value in persistent storage. Consequently, a synchronization between all entities of a bus is necessary. Our synchronization concept utilizes the mechanism used for collision resolving which is based on the fact that sending a 0 always overwrites a 1 on the CAN bus. Hence, in BusCount counters are decremented instead of the usual incrementation.

The mechanism is illustrated in Figure 4. One ECU initializes the synchronization by sending a predefined synchronization ID (Step 1.1). At the same time, each receiving ECU including the initiator of the synchronization, receiving the start of message bit, decrements its local counter. Now all ECUs, having identified the message as synchronization message, simultaneously start to send their respective newly decremented local counter in the data frame (Step 2.1 and 2.2). The lowest counter will overwrite larger counters and ECUs with larger counters stop sending (Step 3). An ECU with the lowest counter value (the actual sender of the synchronization message) sends a MAC over the ID and the counter value (Step 4.1 or 4.2). Each ECU as a recipient of this message verifies the correctness of the MAC and compares the counter to its local one. In case one check by any of the receivers fails, the rest of the message is overwritten with an error frame and is discarded by all controllers. Otherwise, every receiver replaces its local counter with the counter of the synchronization message.

In case multiple ECUs have the lowest counter in the network, each sends the same message without noticing each other. This concept has been used for example in [28] to implement a key exchange on a CAN bus.

In contrast to other approaches that use a counter for every message ID, our approach allows to synchronize all participants of a bus communication with just one message. Further, no central entity is needed for the process, any ECU connected to the bus can initialize it. The only condition for it to work is that at least one ECU owns and processes the correct counter value.

In Section 7.2 we will discuss design decisions and introduce our proof of concept implementation. In the next section we will briefly introduce the Security Modeling Framework SeMF that is then used in Sections 5 and 6, respectively, to formally model and verify both the generic counter approach and our bus counter approach with respect to the desired security properties. The achieved results will then be discussed in Section 7.1.

4. The Security Modeling Framework SeMF

We use our Security Modeling Framework SeMF (see [29] for a detailed description) to formally model and verify the two counter systems discussed in this paper. SeMF is a powerful modeling framework that we have already successfully applied to a variety of different domains and abstraction levels. For example, we used it to verify that specific security properties of service based systems are preserved under composition [30]. We also applied it to model and verify the integration of device authentication based on TPM attestation with secure channel establishment via SSL [31]. Another example is [32] where we proved preservation of

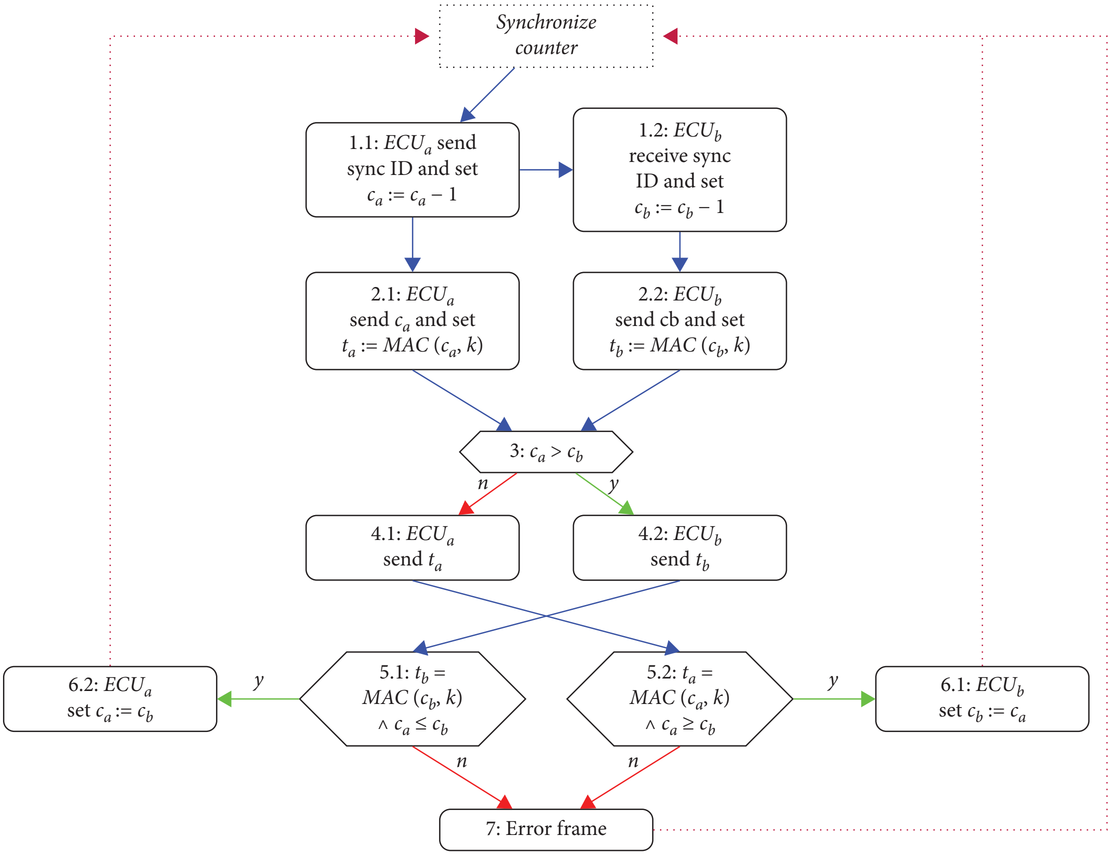

Synchronize counter
1.1: ECU_a send sync ID and set $c_a := c_a - 1$
1.2: ECU_b receive sync ID and set $c_b := c_b - 1$
2.1: ECU_a send c_a and set $t_a := MAC\,(c_a, k)$
2.2: ECU_b send cb and set $t_b := MAC\,(c_b, k)$
3: $c_a > c_b$
n
y
4.1: ECU_a send t_a
4.2: ECU_b send t_b
6.2: ECU_a set $c_a := c_b$
y
5.1: $t_b = MAC\,(c_b, k) \wedge c_a \le c_b$
5.2: $t_a = MAC\,(c_a, k) \wedge c_a \ge c_b$
y
6.1: ECU_b set $c_b := c_a$
n
n
7: Error frame

FIGURE 4: Synchronization of BusCount.

specific security properties for the composition of abstract security patterns.

The basic idea of SeMF is to describe the system behavior by sequences of actions that capture essential changes in the system. As underlying formal semantics SeMF uses prefix closed formal languages (see e.g. [33]) whose alphabet is composed of the actions in the system. More specifically, it uses a set of agents $\mathbb{P}$ (where the term "agent" can denote any entity acting in the system such as a human being, an ECU, etc.), and a set of actions Σ (e.g. specifying sending and receiving messages on a bus) performed by the agents. The system's behavior is then formally described by a prefix closed formal language $B \subseteq \Sigma^*$ (Σ^* denoting the set of all words composed of elements in Σ with $\varepsilon \in \Sigma^*$ denoting the empty word), i.e. by the set of its possible sequences of actions. A system model further comprises the agents' local views (denoted by λ_P for agent P). The local view of different agents usually differs since it describes which parts of the system behavior the agents can actually see (an ECU for example may see its own internal actions, but not those of other ECUs). A system model finally includes the so-called agents' "initial knowledge" $W_P \subseteq \Sigma^*$ which is defined to be prefix closed and to contain B. This concept is used in order to specify system constraints and assumptions.

Security properties are defined in terms of the system specification. The underlying formal semantics then allows formal validation, i.e. allows proving that a specific formal model satisfies specific security properties.

The following notations are used: For $\Upsilon \subseteq \Sigma^*$ and $\omega \in \Upsilon$, $\omega^{-1}(\Upsilon)$ denotes the set of all continuations of ω in Υ. For $\Gamma \subseteq \Sigma$ and $\omega \in \Sigma^*$, $\mathrm{card}(\Gamma, \omega)$ denotes the number of occurrences of any action of Γ in ω, $\mathrm{alph}(\omega)$ denotes its alphabet (i.e. the set of its actions), $\mathrm{pre}(\omega)$ is its set of prefixes, $\mathrm{pre}_1(\omega)$ denotes its first and $\mathrm{suf}_1(\omega)$ its last action. For $\omega = x_1 \ldots x_k \in \Sigma^*$ and $i \in \{1, \ldots, k\}$, $\mathrm{prevact}(x_i, P, \omega)$ denotes the last action before x_i in ω performed by agent P (in case x_i is P 's first action, $\mathrm{prevact}(x_i, P, \omega) = \varepsilon$). For $\omega \in \Sigma^*$, the function $\mathrm{actCnt}\colon \Sigma \times \Sigma^* \longrightarrow \mathbb{N}s$ enumerates strictly monotonically increasing the actions of ω in their order of occurrence: $\mathrm{actCnt}(a, \varepsilon) := 0$ for all $a \in \Sigma$, $\mathrm{act\,Cnt}(a, \omega) := 1$ for $\omega = a$, and for $\mathrm{card}(\Sigma, \omega) = k > 1$ we define $\mathrm{act\,Cnt}(\mathrm{suf}_1(\omega), \omega) := \mathrm{act\,Cnt}(\mathrm{suf}_1(\mathrm{pre}_{k-1}(\omega)), \mathrm{pre}_{k-1}(\omega)) + 1$.

We extend SeMF by a formal specification of actions and a homomorphism to extract any parameter of an action:

Definition 1 (Set of actions). *Let* $\mathtt{P} = \{\mathrm{par}_1, \ldots, \mathrm{par}_n\}$ *a set of parameters* ($n \in \mathbb{N}$) *and for* $j \in \{1, \ldots, n\}$ *let* V_j *the set of possible values of* par_j *with* $V_1 := \mathtt{A}$ *a set of action names and* $V_2 := \mathbb{P}$ *a set of agents. Then the set* Σ *of actions of a system* $\mathbb{S}$ *can be defined as follows:*

$$\Sigma \subseteq \bigcup_{\mathrm{par}_{i_1} \in \mathtt{A}, \mathrm{par}_{i_2} \in \mathbb{P}, \{i_3, \ldots, i_k\} \subseteq \{3, \ldots, n\}} \left(\mathrm{par}_{i_1}, \ldots, \mathrm{par}_{i_k}\right). \tag{1}$$

The sending of a message on a CAN bus can for example be formalized by (send, ECU, bus, msg). See Sections 5.1 and 6.1 for the concrete sets of actions of the two models introduced in this paper. In order to express relations between parameters of different actions, we need to extract them from the actions:

Definition 2 (Parameter extraction). *Let* Σ *be defined as in Definition* 1. *We define a homomorphism* $\widehat{\kappa}_{\mathrm{par}_\iota}\colon \Sigma \longrightarrow \mathsf{P} \cup \Sigma$ *by*

$$\widehat{\kappa}_{\mathrm{par}_\iota}((\mathrm{par}_1, \ldots, \mathrm{par}_k)) := \begin{cases} \mathrm{par}_i, & \text{if } \mathrm{par}_i = \mathrm{par}_\iota, \\ (\mathrm{par}_1, \ldots, \mathrm{par}_k), & \text{else.} \end{cases} \tag{2}$$

The security property provided by a MAC mechanism can be formally specified by the concept of authenticity introduced in [34] : A set of actions Γ is authentic for agent P after a sequence ω of actions has happened if in all sequences that P considers possible after ω, some time in the past an action in Γ must have happened. Formally:

Definition 3 (Authenticity). *A set of actions* $\Gamma \subseteq \Sigma$ *is authentic for* $P \in \mathbb{P}$ *after a sequence of actions* $\omega \in S$ *with respect to* W_P *if* $\mathrm{alph}(x) \cap \Gamma \neq \varnothing$ *for all* $x \in \lambda_P^{-1}(\lambda_P(\omega)) \cap W_P$.

The following weaker property describes that in all sequences of a language L that contain a specific action b, this action is preceded by one of the actions contained in $\Gamma \subseteq \Sigma$:

Definition 4 (Precedence). *For* $L \subseteq \Sigma^*, \Gamma \subseteq \Sigma, b \in \Sigma$ *the property* $\mathrm{prec}_L(\Gamma, b)$ *holds if for all* $\omega \in \mathrm{pre}(L)$ *with* $b \in \mathrm{alph}(\omega)$ *follows* $\Gamma \cap \mathrm{alph}(\omega) \neq \varnothing$. *We simply write* $\mathrm{prec}(\Gamma, b)$ *if from the context the language referred to is clear.*

Additionally to authenticity, we require the counter systems to provide immediacy and non-repeatability. In order to define a respective security property within SeMF we introduce the concept of a *phase class* that allows modeling that a particular action occurred within a particular period of the system. We base our definition on the concept of a phase introduced in [35]. Here a subset of Σ^* is a phase for B if it is a prefix closed language consisting only of words which, as long as they are not maximal, show the same continuation behavior within the phase as within B. Our definition transforms this to arbitrary subsets of Σ^*, not requiring them to be prefix closed:

Definition 5 (Phase class). *Let* $\Upsilon \subseteq \Sigma^*$. *A language* $\Phi(\Upsilon) \subseteq \Sigma^*$ *is a phase class for* Υ *if the following holds:*

1. $\Phi(\Upsilon) \cap \Sigma \neq \varnothing$
2. $\forall \omega, u \in \Upsilon$ *with* $\omega = uv$ *and* $v \in \Phi(\Upsilon) \backslash (\max(\Phi(\Upsilon)) \cup \{\varepsilon\})$ *holds:* $\omega^{-1}(\Upsilon) \cap \Sigma \subseteq v^{-1}(\Phi(\Upsilon)) \cap \Sigma$

Thus, a phase class is characterized by being closed with respect to concatenation. Maximal words in a phase class, denoted by $\max(\Phi(\Upsilon))$, are those $v \in \Phi(\Upsilon)$ for which holds $va \notin \Phi(\Upsilon)$ for all $a \in \Sigma$ (i.e. no matter whether or not exists $\omega = uva \in \Upsilon$).

A phase class can be a very complex construct. However, in many cases phase classes are of interest that can be defined by the actions that start and terminate, respectively, the words. The following definition takes into account that an action can occur more than once in a word. Each starting action occurring in a word $\omega \in \Upsilon$ starts a word of Φ. The word ends with the first j_i occurrences of an action in T_i:

Definition 6 ((S,T)-phase class). *Let* $\Upsilon \subseteq \Sigma^*$, $S \subseteq \Sigma$, $T = T_1 \cup \ldots \cup T_k \subseteq \Sigma$ $(k \in \mathbb{N})$ *with* $T_i \cap T_j = \varnothing$ *for all* $i \neq j$. *Then* , $\Phi := \Phi(\Upsilon, S, \{(T_1, j_1) \ldots, (T_k, j_k)\}) \subseteq \Sigma^*$ *is a phase class for* Υ *starting with* S *and terminating with respect to* $\{(T_1, j_1), \ldots, (T_k, j_k)\}$ *if*

1 Φ *is a phase class for* Υ,

2 $\Phi(\Upsilon) \cap \Sigma = S$

3 *for all* v *maximal in* Φ *the following holds: For* $\omega, u \in \Upsilon, z \in \Sigma^*$ *with* $\omega = uvz$ *it follows* $z = \varepsilon$ *or there exists* $i \in \{1, \ldots, k\}$ *such that* $\mathrm{suf}_1(v) \in T_i$, $\mathrm{card}(T_i, v) = j_i$, *and* $\mathrm{card}(T_l, v) < j_l$ *for all* $l \in \{1, \ldots, i-1, i+1, \ldots, k\}$.

We call such a phase class an $(S, \{(T_1, j_1), \ldots, (T_k, j_k)\})$ -phase class for Υ. If all words in the phase class terminate with the first occurrence of any $t \in T$, we simply call it an (S, T) -phase-class for Υ, denoted by $\Phi(\Upsilon, S, T)$.

It can easily be shown that an (S, T) -phase class for a prefix closed language is itself prefix closed. (S, T) -phase classes are a very useful concept for the concrete specification of freshness properties. We can further combine these two concepts with authenticity:

Definition 7 (Authenticity within a phase class). *Let* $B \subseteq \Sigma^*$ *be the behavior of a system,* $\omega \in B, b \in \mathrm{alph}(\omega)$, *and* $\Phi(W_P) \subseteq \Sigma^*$ *a phase class for agent P's initial knowledge* W_P. *A set of actions* $\Gamma \subseteq \Sigma$ *is authentic for P after* ω *within* $\Phi(W_P)$ *and with respect to* λ_P *and b if (i) it is authentic for P after* ω *and if (ii) for all* $x \in \lambda_P^{-1}(\lambda_P(\omega)) \cap W_P$ *for which exists* $u, z \in \Sigma^*$ *and* $v \in \Phi(W_P)$ *such that* $x = uvz$ *and* $b \in \mathrm{alph}(v)$ *it follows* $\mathrm{alph}(v) \cap \Gamma \neq \varnothing$. *If the property holds for all* $\omega \in B$, *we denote this property shortly by* $\mathrm{authWiPhase}(\Gamma, b, P, \Phi(W_P))$.

5. Formalization and Verification of the Generic Counter Approach

In this section, we introduce our SeMF model of the generic counter system described in Section 2.5 (denoted by GenCnt henceforth) and formally prove to which extend it satisfies the protection goals data origin authenticity, immediacy and non-repeatability of messages.

5.1. The Formal Generic Counter Model. Our SeMF model shall be as simple as possible. It needs to include ECUs connected to a bus whose messages shall be proven to be protected. It also needs to reflect our attack model introduced in Section 2.2, hence must include devices (e.g. ECUs) that an attacker can use to monitor, record, resent and manipulate messages sent on the bus. It is obvious that messages sent on one bus may be accepted by ECUs connected to another bus if the key used for protecting these messages is the same for both buses. Hence we disregard this aspect and restrict our model to one group of honest devices, all owning the same MAC key for message protection, and a further device Eve representing dishonest behavior. All devices are connected to the same bus. Our model can easily be extended, for example by adding more groups, keys and buses, in case other aspects than those addressed in this paper shall be investigated. A special honest device is the Fresh Value Master (FvM) that is responsible for the synchronization of ECUs regarding their counter. We assume all honest devices to act according to a given specification (see Section 5.1.3). FvM only sends synchronization messages, i.e. messages with msgid = `sync`. Other honest devices receive synchronization messages and send and receive functional messages with msgid = `fmsg`. We do not distinguish between different types of functional messages and use just one with its corresponding message counter. Eve can send and receive all types of messages but does not own the MAC key and can thus not generate MACs.

We use four different types of actions: sending and receiving (i.e. accepting) of messages, reading a message without processing it, and an action that models an ECU losing the correct message counter (denoted by genCnt henceforth). This action comprises any situation in which an ECU is not synchronized anymore, i.e. owns a counter smaller than the current correct counter value.

While in many systems (e.g. in [4]) messages only contain a truncated message counter and MAC, respectively, in our model the counter's complete value is included. This way we model the assumption that the recipients always succeed in determining the counter values used by the senders (after all, this aspect is not in the focus of our investigations).

5.1.1. Agents, Parameters and Actions. For the formal specification of actions according to Definition 1, we use the following sets:

(1) Set of agents:

$\mathbb{P}_{gCnt} := \mathscr{E}CU_{gC} \cup \{\text{Eve}\}$ with $\mathscr{E}CU_{gC} := \left\{\text{ECU}_1^{gC}, \text{ECU}_2^{gC}, \text{FvM}\right\}$ whose members are connected to the only bus of the system. FvM denotes the synchronization master and Eve denotes a further device being connected to the bus but not being member of $\mathscr{E}CU_{gC}$.

(2) set of action names: $\mathtt{A}_{\mathtt{gCnt}} = \left\{\text{send}_{gC}, \text{read}_{gC}, \text{recv}_{gC}, \text{loseCnt}_{gC}\right\}$

(3) set of parameters:

$\mathtt{P}_{\mathtt{gCnt}} := \{\text{aname}, \text{ecu}, \text{ecukey}, \text{ecucnt}, \text{prevcnt}, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt}\}$ with $\text{aname} \in \mathtt{A}_{\mathtt{gCnt}}$, $\text{ecu} \in \mathbb{P}_{gCnt}$, ecukey, $\text{mackey} \in \{\mathtt{key}\} \cup \mathbb{N}$, `key` being the key all honest ECUs use for MAC generation and verification, while $\text{ecukey} \in \mathbb{N}$ for ecu = Eve. Further, $\text{ecucnt}, \text{prevcnt}, \text{cnt} \in \mathbb{N} \cup \{\text{nocnt}\}$, $\text{bus} \in \{\text{bus}\}$, $\text{msgid} \in \{\text{sync}, \text{fmsg}\}$, and $\text{msg} \in \mathscr{M}$ ($\mathscr{M}$ being an arbitrary set of messages).

(4) The set of actions $\Sigma_{g\text{Cnt}}$ is then defined as follows:

(1) (send_{gC}, ecu, ecukey, ecucnt, prevcnt, bus, mackey, msgid, msg, cnt): $\text{ecu} \in P_{gCnt}$ sends a message on bus = `bus`. The message's MAC (not explicitly modelled by a parameter of this action) is generated with mackey and covers msgid, msg and cnt. $\text{ecu} \in \mathscr{E}CU_{gC}$ if none of the entire message bits as illustrated in Figure 1 has been written to the bus by Eve. ecu may or may not have generated the MAC. For $\text{ecu} \in \mathscr{E}CU_{gC}$, the parameter ecukey denotes ecu 's MAC generation and verification key `key`, ecucnt denotes its local counter value after having performed the send_{gC} action, and prevcnt denotes the counter value resulting from ecu's previous action (see Section 5.1.3 for the specific operations regarding an ECU's counter). The message can be a functional message, indicated by msgid = `fmsg`, in which case the counter contained in the message's payload is explicitly modelled by cnt, or a synchronization message with msgid = `sync`. In this case the message's payload msg only contains the counter determined by the sender and the parameter cnt contains the constant `nocnt`. Note that there is the possibility that the message is altered (by a technical error or by Eve) after having been sent and may thus only cause a read_{gC} action (see below).

(2) (read_{gC}, ecu, ecukey, ecucnt, prevcnt, bus, mackey, msgid, msg, cnt) denotesecu $\in \mathbb{P}_{bCnt}$ reading a message without processing it (i.e. without accepting it). The action does not change the local message counter ecucnt if $\text{ecu} \in \mathscr{E}CU_{gC}$ (see Prop.A10 below).

(3) (recv_{gC}, ecu, ecukey, ecucnt, prevcnt, bus, mackey, msgid, msg, cnt) denotes the successful reception and processing of a message by $\text{ecu} \in \mathbb{P}_{gCnt}$.

(4) With the action ($\text{loseCnt}_{\text{gC}}$, ecu, ecucnt, prevcnt, msgid) we model the fact that $\text{ecu} \in \mathscr{E}CU_{gC}$ has lost the correct counter value for some reason. This action comprises any situation in which an ECU is not synchronized anymore. As a consequence, its counter is set to a value smaller than the correct counter value.

The idea of a generic counter-based system is that counter values should be strictly monotonically increasing. However, in real systems message transmission may fail due to transmission errors of the bus (e.g. by flipping a bit). Such a message is not accepted in which case the sender simply repeats it, using the same counter as before. We abstract from this since incidents of this type are not security relevant and assume all messages sent by an honest ECU not to suffer from physical failures of the system. This leads to the following definition of the correct counter for a specific action.

Definition 8. Let $\omega = x_1 \ldots x_r \in \Sigma_{gCnt}^*$ and $k \in \{1, \ldots, r\}$. Then the correct counter for action x_k in ω is defined as follows:

$$\operatorname{cor}\operatorname{Cnt}_{gC}(x_k,\omega) := \begin{cases} 1, & \text{if } k = 1, \\ \operatorname{card}(\{x_i \in \operatorname{alph}(\omega) \mid i \in \{1,\ldots,k\} \wedge \\ \qquad \widehat{\kappa}_{\text{a name}}(x_i) = \text{send}_{gC} \\ \wedge \widehat{\kappa}_{cnt}(x_1) < \cdots < \widehat{\kappa}_{cnt}(x_k)\}, \omega), & \text{if } k > 1. \end{cases} \tag{3}$$

This definition assumes that the very first message of any action sequence of the system is sent by an honest ECU.

5.1.2. Introducing a Phase Class into the Model. In the GenCnt system, only messages with correct counters shall be received and accepted, i.e. their values shall be strictly monotonically increasing. Each counter therefore identifies a phase of the system that starts with sending the message containing it. Hence we use send actions to identify phase classes: Each send actions starts a new phase class, and the phase class ends with the next send action that in turn starts a new phase class. The formal definition of this particular (S, T)-phase class is based on Definition 6:

Definition 9. For $\Upsilon \subseteq \Sigma^*_{g\text{Cnt}}$ *and* $a \in \Sigma_{g\text{Cnt}}$ *with* $\widehat{\kappa}_{a\text{name}}(a) = \text{send}_{gC}$ *we define*

$$\Phi(a,\Upsilon) := \Phi\left(\Upsilon, \{a\}, \left\{a'. \in \sum_{gCnt} |\widehat{\kappa}_{\text{aname}}(a') = \text{send}_{gC}\right\}\right). \tag{4}$$

From a recipient's point of view, when having performed a receive action b containing a specific counter, the phase class "activating" this counter starts with the send action that writes this particular message onto the bus. Considering the characteristics of a CAN bus as described in Section 2.1, there cannot be any other send action between these two actions on the bus. Recall that a message manipulated by Eve is considered to have been sent by her. Consequently we do not have two consecutive send_{gC} actions. Hence for each receive action b occuring in a sequence of actions, the corresponding send action, denoted by $\sigma(b)$, is unique. Consequently, each b determines a unique phase class $\Phi(a,\Upsilon)$ with $a = \sigma(b)$. Thus for a specific receive action $b \in \omega$ we can rename the phase class it determines and denote it by $\Phi(\sigma(b),\Upsilon)$. For the sake of completeness, for a send action s we define $\sigma(s) := s$.

For the rest of the paper we will use the particular phase class $\Phi(\sigma(b), W_{gCnt})$ determined by a recv_{gC} action b with W_{gCnt} denoting all ECUs' initial knowledge that we assume to be identical. In Section 5.2 we will explain how this phase class can be used to model immediacy and non-repeatability.

5.1.3. Agents' Local View and Initial Knowledge. The definition of the ECUs' local view must take into account that they can see the messages sent on the bus they are connected to but cannot see who sent them nor the local parameters of the sender. Further, except for these send actions, ECUs can only see their own actions. Hence for all $P \in \mathbb{P}_{gCnt}$ and for all $a \in \Sigma_{gCnt}$ we define λ_P as follows:

(1) $\widehat{\kappa}_{\text{ecu}}(a) = \lambda_{P(a)} := a$

(2) $\widehat{\kappa}_{\text{ecu}}\ (a) \neq P \wedge \widehat{\kappa}_{\text{aname}}(a) \in \left\{\text{read}_{gC}, \text{recv}_{gC}, \text{loseCnt}_{gC}\right\}$ $\Rightarrow \lambda_P(a) := \varepsilon$

(3) $\widehat{\kappa}_{\text{ecu}}(a) \neq P \wedge \widehat{\kappa}_{\text{aname}}(a) = \text{send}_{gC} \Rightarrow \lambda_P(a) := (\text{send}_{gC},$ $\text{bus}, \widehat{\kappa}_{\text{mackey}}(a), \widehat{\kappa}_{\text{msgid}}\ (a), \widehat{\kappa}_{\text{msg}}(a), \widehat{\kappa}_{cnt}(a))$

Agents' Initial Knowledge The agents' initial knowledge captures the constraints and assumptions that we know to hold for our system. If not specified otherwise, the properties refer to $\omega \in W_{gCnt}$.

Prop. A1. A receive action on bus is always preceded by the corresponding send action that writes the message onto the bus. Obviously, the parameter values of mackey, msgid, msg and cnt in b and $\sigma(b)$ are identical (we forgo the formal specification of this statement). The only actions that can happen in between are read_{gC}, recv_{gC} and loseCnt_{gC} actions by ECUs other than sender and receiver. Formally:

For all $b \in \Sigma_{gCnt}$ with $\widehat{\kappa}_{\text{aname}} = \text{recv}_{gC}$ holds

(1) $\text{prec}_{W_{gCnt}}(\sigma(b), b)$

(2) $\forall v \in \Phi(W_{bCnt}, \{\sigma(b)\}, \{b\}) \forall a \in \operatorname{alph}(v)$: $\widehat{\kappa}_{\text{ecu}}(a) =$ $\widehat{\kappa}_{\text{ecu}}(\sigma(b)) \vee \widehat{\kappa}_{\text{aname}}(a) = \text{send}_{gC} \Rightarrow a = \text{pre}_1(v) = \sigma(b)$ and $\widehat{\kappa}_{\text{ecu}}\ (a) = \widehat{\kappa}_{\text{ecu}}(b) \Rightarrow a = \text{suf}_1(v) = b$

Prop. A2. Only members of $\mathscr{E}CU_{gC}$ own and can use key. Since Eve does not own this key and honest ECUs use only key to generate and verify a MAC, the MAC key contained in a send or receive action being equal to the ECU's key and this being equal to key is equivalent to the ECU being member of $\mathscr{E}CU_{gC}$.

$\forall a \in \operatorname{alph}(\omega) : \widehat{\kappa}_{\text{aname}}(a) \in \left\{\text{send}_{gC}, \text{recv}_{gC}\right\} \Rightarrow (\widehat{\kappa}_{\text{mackey}}(a) = \widehat{\kappa}_{\text{ecukey}}(a) = \text{key} \Leftrightarrow \widehat{\kappa}_{\text{ecu}}(a) \in \mathscr{E}CU_{gC})$.

Prop. A3. A recv_{gC} action performed by an honest ECU (i.e. a member of $\mathscr{E}CU_{gC}$) must be preceded by the respective send action of an agent having generated the MAC, i.e. owning the key used for MAC generation:

$\forall \text{ecu} \in \mathscr{E}CU_{gC}$: $\text{prec}_{W_{gCnt}}(\{(\text{send}_{gC}, \text{ecu}', \text{ecukey}', \text{ecucnt}', \text{prevcnt}', \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt}) | \text{ecukey}' = \text{mackey} = \text{key}\}, (\text{recv}_{gC}, \text{ecu}, \text{ecukey}, \text{ecucnt}, \text{prevcnt}, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt}))$

Again, obviously, the parameter values of mackey, msgid, msg and cnt in b and the send action are identical.

Prop. A4. The parameter prevcnt of an action performed by an honest ECU denotes the local message counter the ECU has used in its previous action. For the very first action of an ECU it is defined as the minimal value (which we assume without loss of generality to be equal to 1) of $bCnt$.

$\forall a \in \text{alph}(\omega){:}\widehat{\kappa}_{\text{ecu}} \in \mathscr{E}CU_{gC} \Rightarrow \widehat{\kappa}_{\text{prevcnt}}(a) = \widehat{\kappa}_{\text{ecucnt}}(\text{prevact}(a, \widehat{\kappa}_{\text{ecu}}(a), \omega))$. If for all $a_i \in \omega$ with $\text{actCnt}(a_i,\omega) < \text{actCnt}(a,\omega)$ holds $\widehat{\kappa}_{\text{ecu}}(a_i) \neq \widehat{\kappa}_{\text{ecu}}(a)$, it follows $\widehat{\kappa}_{\text{prevcnt}}(a) = 1$.

Prop. A5. FvM is the only ECU that sends synchronization messages. It does not perform any other action.

$\forall a \in \text{alph}(\omega){:}\ \widehat{\kappa}_{\text{ecu}}(a) = \text{FvM} \Leftrightarrow \widehat{\kappa}_{\text{aname}}(a) = \text{send}_{gC} \wedge \widehat{\kappa}_{\text{msgid}}(a) = \text{sync} \wedge \widehat{\kappa}_{\text{ecucnt}}(a) = \widehat{\kappa}_{\text{msg}}(a)$.

Prop. A6. It is obvious that synchronization messages including a wrong (i.e. too small) counter value open possibilities for all kinds of attacks. Hence we assume that a synchronization message sent by FvM always contains the correct counter value according to Definition 8.

$\forall x \in \text{pre}(\omega){:}\ \widehat{\kappa}_{\text{ecu}}(\text{suf}_1(x)) = \text{FvM} \Rightarrow \widehat{\kappa}_{\text{msg}}(\text{suf}_1(x)) = \text{corCnt}_{gC}(\text{suf}_1(x), x)$.

Prop. A7. When an honest agent different to FvM receives (i.e. accepts) a synchronization message, it verifies that the message's payload (which contains the counter) is greater than the local counter used in its previous action and then sets its local counter to the value of the message counter:

$\forall a \in \text{alph}(\omega){:}\ \widehat{\kappa}_{\text{aname}}(a) = \text{recv}_{gC} \ \wedge\ \widehat{\kappa}_{\text{ecu}}(a) \in \text{ECU}_{gC} \backslash \{\text{FvM}\} \Rightarrow \widehat{\kappa}_{\text{msg}}(a) = \widehat{\kappa}_{\text{ecucnt}}(a) \geq \widehat{\kappa}_{\text{prevcnt}}(a) + 1$.

Prop. A8. An honest agent other than FvM only sends functional messages. When doing so, it increments the counter used in its previous action by 1, uses this value as its new local counter value and as the value of cnt for MAC generation.

$\forall a \in \text{alph}(\omega){:}\widehat{\kappa}_{\text{aname}}(a) = \mathit{send}_{gC} \wedge \widehat{\kappa}_{ecu}(a) \in \text{ECU}_{gC} \backslash \ \{\text{FvM}\} \Leftrightarrow \widehat{\kappa}_{\text{msgid}}(a) = \text{fmsg} \wedge \widehat{\kappa}_{\text{ecucnt}}(a) = \widehat{\kappa}_{\text{cnt}}(a) = \widehat{\kappa}_{\text{prevcnt}}(a) + 1$.

Prop. A9. When an honest agent different to FvM receives (i.e. accepts) a functional message, it verifies that the message's counter is greater than the local counter used in its previous action and then sets its local counter to the value of the message counter:

$\forall a \in \text{alph}(\omega){:}\ \widehat{\kappa}_{\text{aname}}(a) = \text{recv}_{gC} \wedge \widehat{\kappa}_{\text{ecu}}(a) \in \text{ECU}_{gC} \backslash \{\text{FvM}\} \Rightarrow \widehat{\kappa}_{\text{cnt}}(a) = \widehat{\kappa}_{\text{ecucnt}}(a) \geq \widehat{\kappa}_{\text{prevcnt}}(a) + 1$.

Prop. A10. An important property of the generic counter system GenCnt is that an ECU increases its counter value only in case it has received and accepted a message with a bigger counter value. Hence an action read_{gC} by an honest ECU does not change ecu's local counter value: $\forall a \in \text{alph}(\omega){:}\ \widehat{\kappa}_{\text{aname}}(a) = \text{read}_{gC} \wedge \widehat{\kappa}_{\text{ecu}}(a) \in \text{ECU}_{gC} \Rightarrow \widehat{\kappa}_{\text{ecucnt}}(a) = \widehat{\kappa}_{\text{prevcnt}}(a)$

Prop. A11. An action loseCnt_{gC} performed by an honest ECU resets the ECU's counter value to a value smaller than the correct one:

$\forall x \in \text{pre}(\omega){:}\ \widehat{\kappa}_{\text{aname}}(\text{suf}_1(x)) = \text{loseCnt}_{gC} \wedge \widehat{\kappa}_{\text{ecu}}(\text{suf}_1(x)) \in \text{ECU}_{gC} \Rightarrow \widehat{\kappa}_{\text{ecucnt}}(\text{suf}_1(x)) < \text{corCnt}_{gC}(\text{suf}_1(x), x)$

Prop. A12. When an honest ECU performs two recv_{gC} actions with its local genCnt value of the first one being bigger than or equal to the local genCnt value of the second one, it must have performed a loseCnt_{gC} action in between.

For $\omega = x_1 \dots x_k, 1 \leq i < l < j \leq k$, if $\widehat{\kappa}_{\text{aname}}(x_i) = \widehat{\kappa}_{\text{aname}}(x_j) = \text{recv}_{gC}$ and $\widehat{\kappa}_{\text{ecu}}(x_i) = \widehat{\kappa}_{\text{ecu}}(x_j) \in \text{ECU}_{gC}$ and $\widehat{\kappa}_{\text{ecucnt}}(x_i) \geq \widehat{\kappa}_{\text{ecucnt}}(x_j)$ then there exists $x_l \in \text{alph}(\omega)$ with $\widehat{\kappa}_{\text{aname}}(x_l) = \text{loseCnt}_{gC}$ and $\widehat{\kappa}_{\text{ecu}}(x_l) = \widehat{\kappa}_{\text{ecu}}(x_i) = \widehat{\kappa}_{\text{ecu}}(x_j)$.

This concludes our system model specification. In the next section, we will show that the model allows certain states which violate a property that can be used for the specification of authenticity, immediacy and non-repeatability.

5.2. Formal Verification of the Generic Counter Concept.

As stated in Section 2.5, the security requirements the generic counter system (denoted by B_{gCnt}) shall satisfy are data origin authenticity, immediacy and non-repeatability. More precisely, an honest ECU shall accept only messages authentically generated and sent by another honest ECU, thus providing data origin authenticity. Further, the message must contain the correct counter which ensures that no counter is accepted twice (since the correct counter is strictly monotonically increasing), thus providing non-repeatability. In order to express this, we use the phase class $\Phi(\sigma(b), W_{gCnt})$ as defined in Definition 9 with b being a recv_{gC} action. Each time an honest ecu receives a message, the message must authentically for ecu have been sent by a member of the same group, and this send action must be the one to trigger ecu's recv_{gC} action b, i.e. must be the start action $\sigma(b)$ of the phase class determined by b. Since the time period between sending and receiving messages on a CAN bus is very short, we can assume that it never exceeds the specified limit which implies immediacy. This can be formalized as follows:

Theorem 1. *Let $\omega \in B_{gCnt}$ and $b := (\ recv_{gC}, ecu, ecukey, ecucnt,\ \ prevcnt,\ \ bus, mackey, msgid, msg, cnt) \in alph\ \ (\omega)$ with $ecu \in \mathscr{E}\ CU_{gC}$. Then the following property holds:*

$$\begin{aligned}&\text{authWiPhase}\Big(\ \{(\text{send}_{gC}, \text{ecu}', \text{ecukey}', \text{ecucnt}',\\ &\quad \text{prevcnt}', \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt}) | \text{ecu}' \in \text{ECU}_{gC}\},\\ &\quad b, \text{ecu}, \Phi(\sigma(b), W_{gCnt})\ \Big)\end{aligned} \tag{5}$$

Proof 1. Assume one of the honest ECUs different to FvM that is member of the group receives (i.e. accepts) a message. Without loss of generality assume it is $\text{ECU}_1^{gC} \in \mathscr{E}CU_{gC}$ and $b := \ (\text{recv}_{gC}, \text{ECU}_1^{gC}, \text{ecukey}, \text{ecucnt}, \text{prevcnt},\ \ \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt}) \in \text{alph}(\omega)$ for some $\omega \in B_{gCnt}$. By definition, $\lambda_{\text{ECU}_1^{gC}}$ keeps this action, thus b is also contained in each $x \in \lambda^{-1}_{\text{ECU}_1^{gC}}(\lambda_{\text{ECU}_1^{gC}}(\omega))$. Further, b is contained in $\omega \in B_{gCnt} \subseteq W_{gCnt}$. So let $x \in \lambda^{-1}_{\text{ECU}_1^{gC}}(\lambda_{\text{ECU}_1^{gC}}(\omega)) \cap W_{gCnt}$ arbitrarily chosen. Since $\text{ECU}_1^{gC} \in \mathscr{E}CU_{gC}$, Prop.A2 implies that ecukey = mackey = `key`. Further, by Prop.A3, there is an action $a_1 := (\text{send}_{gC}, \text{ecu}_1, \text{ecukey}_1, \text{ecucnt}_1, \text{prevcnt}_1\ \ , \text{bus}, \text{mackey},\ \text{msgid}, \text{msg}, \text{cnt}) \in \text{alph}(x)$ before ECU_1^{gC}'s receive action containing the same message, message ID and counter value and with $\mathrm{ecukey}_1 = \mathrm{mackey} = \mathrm{key}$. Applying again Prop.A2 it follows $\mathrm{ecu}_1 \in \mathrm{ECU}_{gC}$. Hence the message received in b has authentically for ECU_1^{gC} been sent by a member of ECU_{gC}, i.e. data origin authenticity is satisfied.

Therefore $\mathrm{ecu}_1 = \mathrm{FvM}$ and $\mathrm{msgid} = \mathrm{sync}$ (Prop.A5) or $\mathrm{ecu}_1 = \mathrm{ECU}_2^{gC}$ and $\mathrm{msgid} = \mathrm{fmsg}$ (Prop.A8) (we disregard the fact that in principle ECU_1^{gC} could itself be the originator of this message and assume this issue to be addressed by e.g. unique message IDs). By Prop.A1 b is preceded by $\sigma(b) = (\mathrm{send}_{gC}, \mathrm{ecu}', \mathrm{ecukey}', \mathrm{ecucnt}', \mathrm{bus}, \mathrm{mackey}, \mathrm{msgid}, \mathrm{msg}, \mathrm{cnt})$ that starts the phase class identified by b. By definition, the local view of ECU_1^{gC} does not reveal the sender, hence assume $\mathrm{ecu}' \neq \mathrm{ecu}_1$ and $\sigma(b) \neq a_1$, i.e. assume that the authentic action a_1 is not performed in the required phase class. Assume further that after having performed their respective last actions before a_1 (denoted by a_2 and a_3, respectively), ecu_1 and ECU_1^{gC} are synchronized, i.e. own the same counter which is the correct one for these actions. Let us assume ECU_1^{gC} performs a_3, ecu_1 performs a_2 and $\widehat{\kappa}_{\mathrm{ecucnt}}(a_2) = \widehat{\kappa}_{\mathrm{ecucnt}}(a_3) = k = \mathrm{corCnt}_{gC}(a_2, x) = \mathrm{corCnt}_{gC}(a_3, x)$.

Assume $\mathrm{ecu}_1 = \mathrm{ECU}_2^{gC}$ and $\widehat{\kappa}_{\mathrm{msgid}}(a_1) = \mathtt{fmsg}$. Then Prop.A8 implies $\widehat{\kappa}_{\mathrm{ecucnt}}(a_1) = \widehat{\kappa}_{\mathrm{prevcnt}}(a_1) + 1 = \widehat{\kappa}_{\mathrm{cnt}}(a_1)$. Since a_2 is the last action of ecu_1 before a_1, Prop.A4 implies that $\widehat{\kappa}_{\mathrm{prevcnt}}(a_1) = \widehat{\kappa}_{\mathrm{ecucnt}}(a_2)$. Prop.A3 implies $\mathrm{cnt} = \widehat{\kappa}_{\mathrm{cnt}}(b) = \widehat{\kappa}_{\mathrm{cnt}}(a_1)$ and it follows $\widehat{\kappa}_{\mathrm{cnt}}(a_1) = \widehat{\kappa}_{\mathrm{ecucnt}}(a_1) = \widehat{\kappa}_{\mathrm{ecucnt}}(a_2) + 1 = k + 1$. This situation, depicted in Table 2 , is the basis for the subsequent case-by-case analysis (note that it is irrelevant whether a_3 precedes a_2 or vice versa). □

5.2.1. Losing the counter. Assume that ECU_1^{gC} receives the message sent by ecu_1 in a_1 by performing an action a_4 (i.e. $\sigma(a_4) = a_1$). Then $\widehat{\kappa}_{\mathrm{cnt}}(a_4) = \widehat{\kappa}_{\mathrm{cnt}}(a_1) = k + 1$ (Prop.A1) and Prop.A9 implies $\widehat{\kappa}_{\mathrm{ecucnt}}(a_4) = \widehat{\kappa}_{\mathrm{cnt}}(a_4) = k + 1$. Prop.A9 also requires $\widehat{\kappa}_{\mathrm{ecucnt}}(a_4) \geq \widehat{\kappa}_{\mathrm{prevcnt}}(a_4) + 1$. This is the case, as by Prop.A4 we can conclude $\widehat{\kappa}_{\mathrm{prevcnt}}(a_4) = \widehat{\kappa}_{\mathrm{ecucnt}}(a_3)$ and thus $k + 1 = \widehat{\kappa}_{\mathrm{ecucnt}}(a_4) \geq \widehat{\kappa}_{\mathrm{ecucnt}}(a_3) + 1$ which by the assumption of ECU_1^{gC} and ecu_1 being synchronized before a_1 is equal to $\widehat{\kappa}_{\mathrm{ecucnt}}(a_2) + 1 = k + 1$, hence $\widehat{\kappa}_{\mathrm{ecucnt}}(a_4) = k + 1 \geq k + 1$ is satisfied. Since with action b, ECU_1 receives and accepts $\mathrm{cnt} = k + 1$, Prop.A12 implies that ECU_1^{gC} performs an action $a_5 \coloneqq (\mathrm{loseCnt}_{gC}, \mathrm{ECU}_1^{gC}, \mathrm{ecucnt}_5, \mathrm{prevcnt}_5, \mathrm{bus})$ between a_4 and b, and Prop.A11 implies that ecucnt_5 is smaller than the correct counter value for this action which in turn is equal to or bigger than $\widehat{\kappa}_{\mathrm{cnt}}(a_5) = k + 1$. Assume that between a_4 and a_5 there have been k' send actions by members of $\mathscr{E}CU_{gC}$ other than ECU_1^{gC} with correct counters, increasing its value to $k + 1 + k'$ without changing ECU_1^{gC}'s counter value (e.g. because it does not perform any action other than a_5 between a_4 and b). It follows $\mathrm{ecucnt}_5 < \mathrm{corCnt}_{gC}(a_5, x) = k + 1 + k'$. On the other hand, in b ECU_1^{gC} receives and accepts the message sent in $\sigma(b)$ with the counter $\mathrm{cnt} = k + 1$. So assuming ECU_1^{gC}'s $\mathrm{loseCnt}_{gC}$ action a_5 to be its last action before b, Prop.A4 and Prop.A9 imply $k + 1 = \widehat{\kappa}_{\mathrm{cnt}}(b) = \widehat{\kappa}_{\mathrm{ecucnt}}(b) \geq \widehat{\kappa}_{\mathrm{prevcnt}}(b) + 1 = \mathrm{ecucnt}_5$. Both inequalities are satisfied for $\mathrm{ecucnt}_5 \leq k + 1 - 1 = k$. Thus ECU_1^{gC} may very well receive and accept the message sent in $\sigma(b)$ by $\mathrm{ecu}\prime \notin \mathscr{E}CU_{gC}$.

The resulting sequence of actions is depicted in Table 3. While it satisfies data origin authenticity, it violates immediacy, assuming that only the time period between writing a message onto the bus and reading it does not exceed the specified limit. It also violates non-repeatability as the message sent in a_1 is accepted twice.

It is not surprising that counter loss without timely synchronization opens up attack possibilities. The same result can be shown in case $\mathrm{ecu}_1 = \mathrm{FvM}$ sends a synchronization message in a_1. We then need to consider the fact that between a_2 and a_1, ECU_2^{gC} may have sent n messages that increase the correct counter of a_1 accordingly. Further, instead of applying Prop.A8 we need to take into account that the counter is sent as the message's payload, i.e. modeled by the parameter msg, and apply Prop.A5.

We will now investigate whether sending of synchronization messages prohibits the above described attack. Assume therefore that with the $\mathrm{loseCnt}_{gC}$ action ECU_1^{gC} sets its local genCnt value to $k'\prime < k + 1 + k'$ and that between the $\mathrm{loseCnt}_{gC}$ action and b, ECU_1^{gC} receives one or more synchronization messages with a_6 being the last one before b. Since $\mathrm{cnt} = k + 1$ is the counter accepted by ECU_1^{gC} in b and since the counter sent in a synchronization message is contained in its payload, $\widehat{\kappa}_{\mathrm{msg}}(a_6) = \mathrm{msg}_6 \in [k'' + 1, k]$. Assume further these are the only messages sent on the bus. Then again by Prop.A4 and Prop.A7, $\mathrm{cnt} = k + 1 = \widehat{\kappa}_{\mathrm{cnt}}(b) = \widehat{\kappa}_{\mathrm{ecucnt}}(b) \geq \widehat{\kappa}_{\mathrm{prevcnt}}(b) + 1 = \widehat{\kappa}_{\mathrm{ecucnt}}(a_6) + 1 = \widehat{\kappa}_{\mathrm{msg}}(a_6) + 1 = \mathrm{msg}_6 + 1 \geq k'' + 2$ which implies $k'' \leq \mathrm{msg}_6 - 1 \leq k + 1 - 2 = k - 1$. According to Prop.A1, a_6 is preceded by an action $\sigma(a_6)$ in which the synchronization message received by ECU_1^{gC} is written to the bus. The property further implies that between $\sigma(a_6)$ and a_6, ECU_1^{gC} does not perform any further action, hence $\sigma(a_6)$ happens after ECU_1^{gC}'s $\mathrm{loseCnt}_{gC}$ action and in particular after a_1. Recall now that we have assumed k to be the correct counter value of both a_2 and a_3 and that $\widehat{\kappa}_{\mathrm{cnt}}(a_1) = k + 1$. If $\widehat{\kappa}_{\mathrm{ecu}}(\sigma(a_6))$ was FvM, Prop.A6 would imply $\widehat{\kappa}_{\mathrm{msg}}(\sigma(a_6)) \geq k + 1 + 1 = k + 2$. Yet what ECU_1^{gC} accepts in a_6 is $\mathrm{msg}_6 \leq k + 1 - 1 = k$. Thus $\widehat{\kappa}_{\mathrm{ecu}}(\sigma(a_6)) \neq \mathrm{FvM}$ and since by Prop.A8 an honest agent other than FvM only sends functional messages, it follows $\widehat{\kappa}_{\mathrm{ecu}}(\sigma(a_6)) = \mathrm{Eve}$.

While by Prop.A3, a_6 is preceded by an action $a_7 \coloneqq (\mathrm{send}_{gC}, \mathrm{FvM}, \ldots, \mathrm{sync}, \ldots \mathrm{msg}_6, \ldots)$, this does not necessarily interfere with the attack we are constructing here, assuming that ECU_1^{gC} does not receive (i.e. accept) this message but only performs a read_{gC} action which does not change its local counter. This attack is illustrated in Table 4.

This attack uses an important characteristic of the generic counter system GenCnt, captured in Prop.A10: It causes ECU_1^{gC} to keep the too small and thus incorrect counter since it does not correctly receive and accept the synchronization messages sent by FvM between a_5 and b. The attack again violates immediacy and non-repeatability.

5.2.2. Not losing the counter. Let us now consider the case in which ECU_1^{gC} only performs read_{gC} actions between a_3 and b and in particular does not perform an action a_4, i.e. does not receive and accept the message sent by ecu_1 in a_1. As above, by Prop.A1 and Prop.A3 we know $\widehat{\kappa}_{\mathrm{cnt}}(a_1) = \widehat{\kappa}_{\mathrm{cnt}}(b) = \widehat{\kappa}_{\mathrm{cnt}}$

Table 2: Developing possible sequences

a_3	last action by ECU_1^{gC} before a_1	$\widehat{\kappa}_{\text{ecucnt}}(a_2) = k$ (k is correct counter)
a_2	last action by ecu_1 before a_1	$\widehat{\kappa}_{\text{ecucnt}}(a_3) = k$
$\vdots$		
	no actions by ECU_1^{gC} and ecu_1	
$\vdots$		
a_1	$(\text{send}_{gC}, \text{ecu}_1, \text{ecukey}, \text{ecucnt}_1, \text{prevcnt}_1, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{ecucnt}_1 = \text{prevcnt}_1 + 1 = \widehat{\kappa}_{\text{ecucnt}}(a_2) + 1 = k + 1 = \text{cnt}$
$\vdots$		
$\sigma(b)$	$(\text{send}_{gC}, \text{ecu}', \text{ecukey}', \text{ecucnt}', \text{prevcnt}', \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{cnt} = k + 1$ by Prop.$A1$
b	$(\text{recv}_{gC}, \text{ECU}_1^{gC}, \text{ecukey}, \text{ecucnt}, \text{prevcnt}, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{ecucnt} = \text{cnt} = k + 1$

Table 3: A first attack sequence.

a_3	action by ECU_1^{gC}	$\widehat{\kappa}_{\text{ecucnt}}(a_2) = k$
a_2	action by ecu_1	$\widehat{\kappa}_{\text{ecucnt}}(a_3) = k$
$\vdots$		
	no actions by ECU_1^{gC} and ecu_1	
$\vdots$		
a_1	$(\text{send}_{gC}, \text{ecu}_1, \text{ecukey}, \text{ecucnt}_1, \text{prevcnt}_1, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{ecucnt}_4 = \text{cnt} = k + 1$ $\widehat{\kappa}_{\text{ecucnt}}(a_2) + 1 = k + 1 = \text{cnt}$
a_4	$(\text{recv}_{gC}, \text{ECU}_1^{gC}, \text{ecukey}, \text{ecucnt}_4, \text{prevcnt}_4, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{ecucnt}_4 = \text{cnt} = k + 1$
$\vdots$		
a_5	$(\text{loseCnt}_{gC}, \text{ECU}_1^{gC}, \text{ecucnt}_5, \text{prevcnt}_5, \text{bus})$	$\text{ecucnt}_5 < k + 1 + k'$
$\vdots$		
	no action by ECU_1^{gC}	
$\vdots$		
$\sigma(b)$	$(\text{send}_{gC}, \text{ecu}', \text{ecukey}', \text{ecucnt}', \text{prevcnt}', \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{cnt} = k + 1$
b	$(\text{recv}_{gC}, \text{ECU}_1^{gC}, \text{ecukey}, \text{ecucnt}, \text{prevcnt}, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{ecucnt} = \text{cnt} = k + 1$

Table 4: The second possible attack sequence.

a_3	action by ECU_1^{gC}	$\widehat{\kappa}_{\text{ecucnt}}(a_2) = k$
a_2	action by ecu_1	$\widehat{\kappa}_{\text{ecucnt}}(a_3) = k$
$\vdots$		
	no actions by ECU_1^{gC} and ecu_1	
$\vdots$		
a_1	$(\text{send}_{gC}, \text{ecu}_1, \text{ecukey}, \text{ecucnt}_1, \text{prevcnt}_1, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{ecucnt}_1 = \text{prevent}_1 + 1 =$ $\widehat{\kappa}_{\text{ecucnt}}(a_2) + 1 = k + 1 = \text{cnt}$
a_4	$(\text{recv}_{gC}, \text{ECU}_1^{gC}, \text{ecukey}, \text{ecucnt}_4, \text{prevcnt}_4, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{ecucnt}_4 = \text{cnt} = k + 1$
$\vdots$		
a_5	$(\text{loseCnt}_{gC}, \text{ECU}_1^{gC}, \text{ecucnt}_5, \text{prevcnt}_5, \text{bus})$	$\text{ecucnt}_5 < k + 1 + k'$
$\vdots$		
a_7	$(\text{send}_{gC}, \text{FvM}, \ldots, \text{sync}, \ldots, \text{msg}_6, \ldots,)$	
a_8	$(\text{read}_{gC}, \text{ECU}_1^{gC}, \ldots, \text{ecucnt}_8, \text{prevcnt}_8, \text{sync}, \ldots, \text{msg}_6, \ldots)$	$\text{ecucnt}_8 = \text{prevcnt}_8 = \text{ecucnt}_5$
$\vdots$		
$\sigma(a_6)$	$(\text{send}_{gC}, \text{Eve}, \ldots, \text{sync}, \ldots, \text{msg}_6, \ldots,)$	$\text{msg}_6 \in [k'' + 1, k + 1 - 1]$
a_6	$(\text{recv}_{gC}, \text{ECU}_1^{gC}, \text{ecukey}, \text{ecucnt}_6, \text{prevcnt}_6, \ldots, \text{sync}, \ldots, \text{msg}_6, \ldots,)$	$\text{ecucnt}_6 = \text{msg}_6 \leq k + 1 - 1$ $\text{prevent}_6 = \text{ecucnt}_8 = k''$ $\wedge \text{msg}_6 \geq k'' + 1$
$\sigma(b)$	$(\text{send}_{gC}, \text{ecu}', \text{ecukey}', \text{ecucnt}', \text{prevcnt}', \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{cnt} = k + 1$
b	$(\text{recv}_{gC}, \text{ECU}_1^{gC}, \text{ecukey}, \text{ecucnt}, \text{prevcnt}, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$	$\text{ecucnt} = \text{cnt} = k + 1$

$(\sigma(b))$. Since by Prop.A10 a read_{gC} action does not change ECU_1^{gC}'s counter, consecutive application of Prop.A4 to the sequence of these read_{gC} actions implies $\widehat{\kappa}_{\text{prevcnt}}(b) = \widehat{\kappa}_{\text{ecucnt}}(a_3)$ and by Prop.A9 it follows $k + 1 = \text{cnt} = \widehat{\kappa}_{\text{cnt}}(b) = \widehat{\kappa}_{\text{ecucnt}}(b) \geq \widehat{\kappa}_{\text{prevcnt}}(b) + 1 = \widehat{\kappa}_{\text{ecucnt}}(a_3) + 1 = k + 1$. This equation is always satisfied which means that there is no contradiction to $\widehat{\kappa}_{\text{ecu}}(\sigma(b)) = \text{ecu}' \neq \text{ecu}_1$. Therefore this sequence (illustrated in Table 5) is another example for violation of immediacy based on invalidation and replay of messages. Note that it does not violate non-repeatability as the message is only received and accepted once.

The only case in which the desired properties hold is if by performing a_4, ECU_1^{gC} receives and accepts the message sent in a_1 and does not lose the correct counter value, i.e. does not perform a loseCnt_{gC} action between a_4 and b. In this case ECU_1^{gC} sets its local counter $\widehat{\kappa}_{\text{ecucnt}}(a_4)$ to $k + 1$ (Prop.A9) and without losing the correct counter will not accept the same counter in action b anymore, as $\widehat{\kappa}_{\text{prevcnt}}(b) = \widehat{\kappa}_{\text{ecucnt}}$

Table 5: The third possible attack sequence.

a_3	action by ECU_1^{gC}	$\widehat{\kappa}_{\mathrm{ecucnt}}(a_2) = k$
a_2	action by ecu_1	$\widehat{\kappa}_{\mathrm{ecucnt}}(a_3) = k$
$\vdots$		
	only read_{gC} actions by ECU_1^{gC}	
$\vdots$		
a_1	$(\mathrm{send}_{gC}, \mathrm{ecu}_1, \mathrm{ecukey}, \mathrm{ecucnt}_1, \mathrm{prevcnt}_1, \mathrm{bus}, \mathrm{mackey}, \mathrm{msgid}, \mathrm{msg}, \mathrm{cnt})$	$\mathrm{ecucnt}_1 = \mathrm{prevcnt}_1 + 1 = \widehat{\kappa}_{\mathrm{ecucnt}}(a_2) + 1 = k + 1 = \mathrm{cnt}$
$\vdots$		
	only read_{gC} actions by ECU_1^{gC}	
$\vdots$		
$\sigma(b)$	$(\mathrm{send}_{gC}, \mathrm{ecu}', \mathrm{ecukey}', \mathrm{ecucnt}', \mathrm{prevcnt}', \mathrm{bus}, \mathrm{mackey}, \mathrm{msgid}, \mathrm{msg}, \mathrm{cnt})$	$\mathrm{cnt} = k + 1$
b	$(\mathrm{recv}_{gC}, \mathrm{ECU}_1^{gC}, \mathrm{ecukey}, \mathrm{ecucnt}, \mathrm{prevcnt}, \mathrm{bus}, \mathrm{mackey}, \mathrm{msgid}, \mathrm{msg}, \mathrm{cnt})$	$\mathrm{ecucnt} = \mathrm{cnt} = \mathrm{prevcnt} + 1 = \widehat{\kappa}_{\mathrm{ecucnt}}(a_2) = k + 1$

$(a_4) = k + 1$ and thus $\widehat{\kappa}_{\mathrm{cnt}}(b) = k + 1 \not> \widehat{\kappa}_{\mathrm{prevcnt}}(b) = k + 1$ as is required by Prop.A9.

Our proof indicating possible attacks is based on the fact that certain messages are not received but only read by ECU_1^{gC}. This can easily be accomplished by an attacker with the abilities described in Section 2.2. All she has to do is to monitor the respective message and then invalidate it and all following ones related to the relevant counter by changing a CRC bit or overwriting the message with an error frame. This will cause all ECUs connected to the bus to reject the messages. Since read_{gC} actions do not change the ECUs' local counter, any message containing a bigger counter will still be considered correct.

In Section 7.1 we will discuss the results achieved by the proof and compare them to the formal proofs of the bus counter-based system to be introduced in the next section.

6. Formalization and Verification of BusCount

In this section, we introduce the formal model and verification of our hardware-based counter approach.

6.1. The Formal Bus Counter Model. We model the bus counter-based system (denoted by BusCnt henceforth) as similar as possible to the generic counter model. One important difference is that it does not need a central freshness value master since all ECUs send synchronization messages simultaneously. Hence the set of agents is defined as $\mathbb{P}_{bCnt} = \mathscr{E}CU_{bC} \cup \{\mathrm{Eve}\}$ with $\mathscr{E}CU_{bC} := \left\{\mathrm{ECU}_1^{bC}, \mathrm{ECU}_2^{bC}, \mathrm{ECU}_3^{bC}\right\}$.

As in GenCnt, the BusCnt system has only one bus bus = bus all agents are being connected to. Further, members of $\mathscr{E}CU_{bC}$ are honest and own the key key, while Eve, not being member of this group, does not own this key. We use the same set of action parameters, but a different specification of agents' behavior (after all, we model a different system). The set of actions Σ_{bCnt} is defined as follows:

(i) $(\mathrm{send}_{bC}, \mathrm{ecu}, \mathrm{ecukey}, \mathrm{ecucnt}, \mathrm{prevcnt}, \mathrm{bus}, \mathrm{mackey}, \mathrm{msgid}, \mathrm{msg}, \mathrm{cnt})$ denotes a send action as described in Section 5.1.1, except that the counter value cnt is covered by the MAC but not transmitted. As in the GenCnt model, the message may be altered (by a technical error or by Eve) after having been sent and may thus only cause a read_{bC} action (see below).

(ii) $(\mathrm{read}_{bC}, \mathrm{ecu}, \mathrm{ecukey}, \mathrm{ecucnt}, \mathrm{prevcnt}, \mathrm{bust}, \mathrm{mackey}, \mathrm{msgid}, \mathrm{msg}, \mathrm{cnt})$ denotes agent $\mathrm{ecu} \in \mathbb{P}_{bCnt}$ reading a message without processing it afterwards. In contrast to the respective read_{gC} action of the GenCnt model, the read_{bC} action of the BusCnt model changes the state of an ecu being member of $\mathscr{E}CU_{bC}$ by decrementing its local bCnt value (stored with the last action and thus modeled by the parameter prevcnt) (see Prop.B13 in Section 6.1.1 below). This captures the fact that ecu reacts to the "start of message" bit on bus but discards the respective message (e.g. because the CRC verification fails) in which case it does not perform the receive action. Note that the sender of a message always reads its own action bey performing a read_{bC} action.

(iii) $(\mathrm{recv}_{bC}, \mathrm{ecu}, \mathrm{ecukey}, \mathrm{ecucnt}, \mathrm{prevcnt}, \mathrm{bus}, \mathrm{mackey}, \mathrm{msgid}, \mathrm{msg}, \mathrm{cnt})$ denotes the successful reception and processing of a message by $\mathrm{ecu} \in \mathbb{P}_{bCnt}$.

(iv) As in the GenCnt system, with $(\mathrm{loseCnt}_{bC}, \mathrm{ecu}, \mathrm{ecucnt}, \mathrm{prevcnt}, \mathrm{bus})$ we model the fact that $\mathrm{ecu} \in \mathbb{P}_{bCnt}$ has lost the correct counter value for some reason and thus is no longer synchronized. Since in the BusCnt system counter values decrease, its counter is set to a value bigger than the correct counter value (see Section 6.1.1 for more details).

6.1.1. Agents' Local View and Initial Knowledge. Again, the agents' local view is defined analogously to the generic counter model: All agents see their own actions completely and see the messages sent on the CAN bus they are connected to but cannot see who sent them nor the values of parameters stored locally by the sender. Further, agents cannot see actions read_{bC}, recv_{bC} and $\mathrm{loseCnt}_{bC}$ performed by other agents.

As already pointed out in Section 5.1.3, with specifying the agents' initial knowledge we capture the characteristics of our system. In the following, we list all properties we assume to be satisfied by the agents' initial knowledge W_{bCnt} with reference to the respective property in Section 5.1.3 (if any) in which case we omit the formalization. Analogously to Section 5.1.3, if not specified otherwise, the properties refer to $\omega \in W_{bCnt}$. We denote the correct counter for a specific action a in ω by $\mathrm{corCnt}_{bC}(a, \omega)$. In Lemma 2 (see Section 6.2) we will show how its value is determined.

Prop. B1 (analogous to first statement of Prop.A1). A read_{bC} and recv_{bC} action, respectively, on bus is always preceded by the corresponding send action that writes the message onto the bus. Obviously, the parameter values of mackey, msgid, msg and cnt in b and $\sigma(b)$ are identical (we forgo the formalization of the latter statement).

For all $b \in \Sigma_{bCnt}$ with $\widehat{\kappa}_{\text{aname}} \in \{\text{read}_{bC}, \text{recv}_{bC}\}$ holds $\text{prec}_{W_{bCnt}}(\sigma(b), b)$.

Prop. B2 (Prop.A2). Only members of $\mathscr{E}CU_{bC}$ own and can use key = `key`. Since Eve does not own this key and honest ECUs use only `key` to generate and verify a MAC, the MAC key contained in a send or receive action being equal to key = `key` is equivalent to the ECU being member of $\mathscr{E}CU_{bC}$.

Prop. B3 (Prop.A3). A recv_{bC} action performed by an honest ECU (i.e. a member of $\mathscr{E}CU_{bC}$) must be preceded by the respective send action of an agent having generated the MAC, i.e. owning the key used for MAC generation. Again, obviously, the parameter values of mackey, msgid, msg and cnt in b and $\sigma(b)$ are identical.

Prop. B4 (Prop.A4). The parameter prevcnt of an action performed by an honest ECU denotes the local bCnt value as result of the ECU's previous action. For the very first action of an ECU it is defined as the maximal value of *bCnt*, denoted by $bCnt_{\text{max}}$. Formally:

$\forall a \in \text{alph}(\omega)\colon \widehat{\kappa}_{\text{prevcnt}}(a) = \widehat{\kappa}_{\text{ecucnt}}\ (\text{prevact}(a, \widehat{\kappa}_{\text{ecu}}(a), \omega))$. If for all $a_i \in \omega$ with $\text{actCnt}(a_i, \omega) < \text{actCnt}(a, \omega)$ holds $\widehat{\kappa}_{\text{ecu}}(a_i) \neq \widehat{\kappa}_{\text{ecu}}(a)$, it follows $\widehat{\kappa}_{\text{prevcnt}}(a) = bCnt_{\text{max}}$.

Prop. B5. In Section 7.2.2 we will discuss which starting value of *bCnt* to choose in order to avoid counter overflow. Further, as explained in Section 2.2, we assume that memory failures and attacks cannot cause counter overflow. Hence we can assume that such a failure never results into a local counter value stored by an ECU being smaller than the correct one (see Prop.B17 below). For our formal model we assume that the local counter value of ECUs is always sufficiently large such that counter decrementation can result into the value 0 only in the last phase class of an action sequence. Formally:

$\forall a \in \text{alph}(\omega)\colon \widehat{\kappa}_{\text{prevcnt}}(a) > 0$

Prop. B6 (analog to Prop.A8). When an honest ECU sends a synchronization message, it includes as its message payload the local bCnt value of its previous action decremented by 1 but does not change the local bCnt value.

$\forall a \in \text{alph}(\omega)\colon \widehat{\kappa}_{\text{aname}}(a) = \text{send}_{bC} \wedge \widehat{\kappa}_{\text{ecu}}(a) \in \mathscr{E}CU_{bC} \wedge \widehat{\kappa}_{\text{msgid}}(a) = \text{sync} \Rightarrow \widehat{\kappa}_{\text{msg}}(a) = \widehat{\kappa}_{\text{prevcnt}}(a) - 1 \wedge \widehat{\kappa}_{\text{ecucnt}}(a) = \widehat{\kappa}_{\text{prevcnt}}(a)$.

Prop. B7. We assume that there always exists an ECU owning the correct counter value. Since our synchronization concept utilizes the mechanism used for collision resolving (a 0 written to the CAN bus always overwrites a 1), the correct counter always overwrites any incorrect one. Therefore a send_{bC} action containing a synchronization message that is actually performed by an honest ECU always contains the correct counter. Formally:

$a \in \text{alph}(\omega) \wedge \widehat{\kappa}_{\text{aname}}(a) = \text{send}_{bC} \wedge \widehat{\kappa}_{\text{ecu}}(a) \in \mathscr{E}CU_{bC} \wedge \widehat{\kappa}_{\text{msgid}}(a) = \text{sync} \Rightarrow \widehat{\kappa}_{\text{msg}}(a) = \text{corCnt}_{bC}(a, \omega)$.

Prop. B8. When monitoring a synchronization message being written to the bus, an honest ECU decrements its previously used counter value by 1 and verifies that the result is less or equal to the counter sent as the message's payload. The error frame parameter being equal to *no* indicates that this check has been successful (and that the MAC check that we do not formalize explicitly has been successful as well). It then uses this value as its new local counter.

$\forall a \in \text{alph}(\omega)\colon \widehat{\kappa}_{\text{aname}}(a) \in \{\text{read}_{bC}, \text{recv}_{bC}\} \wedge \widehat{\kappa}_{\text{ecu}}(a) \in \mathscr{E}CU_{bC} \wedge \widehat{\kappa}_{\text{msgid}}(a) = \text{sync} \wedge \widehat{\kappa}_{\text{errorFrame}}(a) = \text{no} \Rightarrow \widehat{\kappa}_{\text{ecucnt}}(a) = \widehat{\kappa}_{\text{msg}}(a) \leq \widehat{\kappa}_{\text{prevcnt}}(a) - 1$.

While a is actually a recv_{bC} action, our proofs do not depend on distinguishing between read_{bC} and recv_{bC} actions of synchronization messages.

Prop. B9 (analog to Prop.A8). When an honest ECU sends a functional message, it includes as its counter value the local bCnt value of its previous action decremented by 1 but does not change the local bCnt value. (It changes the value of ecucnt with the action of reading its own message that the ECU performs simultaneously to sending, see Prop.B10 and Prop.B13.)

$\forall a \in \text{alph}(\omega)\colon \widehat{\kappa}_{\text{aname}}(a) = \mathit{send}_{bC} \wedge \widehat{\kappa}_{\text{ecu}}(a) \in \mathscr{E}CU_{bC} \wedge \widehat{\kappa}_{\text{msgid}}(a) = \text{fmsg} \Rightarrow \widehat{\kappa}_{\text{cnt}}(a) = \widehat{\kappa}_{\text{prevcnt}}(a) - 1 \wedge \widehat{\kappa}_{\text{ecucnt}}(a) = \widehat{\kappa}_{\text{prevcnt}}(a)$.

Prop. B10 (analog to Prop.A9). When an honest ECU receives and accepts a functional message, it decrements its previously used counter value by 1 and verifies that the message's cnt value is equal to the result. It then sets its local bCnt value ecucnt to the message's counter.

$\forall a \in \text{alph}(\omega)\colon \widehat{\kappa}_{\text{aname}}(a) = \text{recv}_{bC} \wedge \widehat{\kappa}_{\text{ecu}}(a) \in \mathscr{E}CU_{bC} \wedge \widehat{\kappa}_{\text{msgid}}(a) = \text{fmsg} \Rightarrow \widehat{\kappa}_{\text{ecucnt}}(a) = \widehat{\kappa}_{\text{cnt}}(a) = \widehat{\kappa}_{\text{prevcnt}}(a) - 1$.

Prop. B11. All honest ECUs, when reading a message, check the message's MAC, independently of whether or not they accept it. In case of a functional message, this involves the ECU's local counter, more concretely the counter value used by the ECU in its previous action decremented by 1. If such a check succeeds which is a necessary condition for the error frame being set to no, this value is the one that was used to generate the message's MAC. Note that this assumes that an attacker that owns the correct counter and a MAC cannot guess the corresponding message.

Let $s \in \text{alph}(\omega)$ with $\widehat{\kappa}_{\text{aname}}(s) = \text{send}_{bC}$ and $v \in \Phi(s, W_{bCnt})$. Then the following holds:

$\forall b \in \{a \in \text{alph}(v) \mid \widehat{\kappa}_{\text{aname}}(a) \in \{\text{read}_{bC}, \text{recv}_{bC}\} \wedge \widehat{\kappa}_{\text{ecu}}(a) \in \mathscr{E}CU_{bC} \wedge \widehat{\kappa}_{\text{msgid}}(a) = \text{fmsg}\}\colon \widehat{\kappa}_{\text{errorFrame}}(b) = \text{no} \Rightarrow \widehat{\kappa}_{\text{cnt}}(b) = \widehat{\kappa}_{\text{ecucnt}}(b) = \widehat{\kappa}_{\text{prevcnt}}(a) - 1$.

Note that in our very simple model with only one type of functional message, a successful check indicated by errorFrame = no actually results into a recv_{bC} action. However, considering also read_{bC} actions with errorFrame = no allows to extend the model with respect to more types of functional messages without having to change the assumptions.

Prop. B12. As explained in Section 3, if an ECU's checks concerning for example a message's MAC fails and it therefore writes an error-frame, all other ECUs join in and write an error-frame as well, no matter whether or not their checks failed. Hence all read$_{bC}$ and recv$_{bC}$ actions induced by a specific send$_{bC}$ action have the same value for the parameter errorFrame. Since a message is only received and accepted if all checks have been successful, the error frame of a recv$_{bC}$ action is always set to no. Formally:

Let $s \in \mathrm{alph}(\omega)$ with $\widehat{\kappa}_{\mathrm{aname}}(s) = \mathrm{send}_{bC}$ and $v \in \Phi(s, W_{bCnt})$. Let further $R(v) := \{b \in \mathrm{alph}(v) | \widehat{\kappa}_{\mathrm{aname}}(b) \in \{\mathrm{read}_{bC}, \mathrm{recv}_{bC}\}\}$ denote the read$_{bC}$ and recv$_{bC}$ actions in v. Then for all $b_i, b_j \in R(v)$ the following holds:

$\widehat{\kappa}_{\mathrm{errorFrame}}(b_i) = \widehat{\kappa}_{\mathrm{errorFrame}}(b_j)$ and $\widehat{\kappa}_{\mathrm{aname}}(b_i) = \mathrm{recv}_{bC} \Rightarrow \widehat{\kappa}_{\mathrm{errorFrame}}(b_i) = \mathrm{no}$.

Prop. B13 (in contrast to Prop.A10). When an honest ECU reads a message, it always decrements its previously used bCnt value by 1 and uses the result as its new local bCnt value. This behavior is independent of whether or not its checks fail, i.e. independent of the errorFrame value. Formally:

$\forall a \in \mathrm{alph}(\omega): \widehat{\kappa}_{\mathrm{aname}}(a) = \mathrm{read}_{bC} \wedge \quad \widehat{\kappa}_{\mathrm{ecu}}(a) \in \mathscr{E}CU_{bC} \Rightarrow \widehat{\kappa}_{\mathrm{ecucnt}}(a) = \widehat{\kappa}_{\mathrm{prevcnt}}(a) - 1$.

Prop. B14. In a phase class $\Phi(s, W_{bCnt})$ with $\widehat{\kappa}_{\mathrm{aname}}(s) = \mathrm{send}_{bC}$ (i.e. a phase class that starts with a specific send$_{bC}$ action and ends with the next send$_{bC}$ action, see Definition 9), all honest ECUs including the sender either read or receive the message or perform a loseCnt action. They do not perform any other action.

Let $s \in \Sigma_{bCnt}$ with $\widehat{\kappa}_{\mathrm{aname}}(s) = \mathrm{send}_{bC}$ and v maximal in $\Phi(s, W_{bCnt})$. Then for all $\mathrm{ecu} \in \mathscr{E}CU_{bC}$ exists exactly one $c \in \mathrm{alph}(v)$ such that $\widehat{\kappa}_{\mathrm{ecu}}(c) = \mathrm{ecu}$ and $\widehat{\kappa}_{\mathrm{aname}}(c) \in \{\mathrm{read}_{bC}, \mathrm{recv}_{bC}, \mathrm{loseCnt}_{bC}\}$.

Prop. B15 (analog to Prop.A6). In every phase class $\Phi(s, W_{bCnt})$ with $\widehat{\kappa}_{\mathrm{aname}}(s) = \mathrm{send}_{bC}$ there is an honest ECU owning the correct counter and performing a read or recv$_{bC}$ action in this phase class, but no loseCnt action. Here, owning the correct counter means that the ECU has used and stored the correct counter value in its previous action and can thus use it in the next action.

Let $s \in \mathrm{alph}(\omega)$ with $\widehat{\kappa}_{\mathrm{aname}}(s) = \mathrm{send}_{bC}$. Then for all v maximal in $\Phi(s, W_{bCnt})$ exists an action $b \in alph(v)$ with $\widehat{\kappa}_{\mathrm{aname}}(b) \in \{\mathrm{read}, \mathrm{recv}\}, \mathrm{ECU}^*(s) := \widehat{\kappa}_{\mathrm{ecu}}(b) \in \mathscr{E}CU_{bC}$ and $\widehat{\kappa}_{\mathrm{prevcnt}}(b) = \mathrm{corCnt}_{bC}(b, \omega) + 1$.

Prop. B16 (analog to Prop.A12). If an ECU is not synchronized at a specific action, i.e. does not use the correct counter relevant for this action, it must have performed an action loseCnt$_{bC}$ before. Note that using a counter value refers to the parameter prevcnt.

$\forall a \in \mathrm{alph}(\omega): \widehat{\kappa}_{\mathrm{prevcnt}}(a) - 1 \neq \mathrm{corCnt}_{bC} \quad (a, \omega) \Rightarrow \exists a' \in \mathrm{alph}(\omega): \widehat{\kappa}_{\mathrm{aname}}(a') = \mathrm{loseCnt}_{bC} \wedge \widehat{\kappa}_{\mathrm{ecu}}(a') = \widehat{\kappa}_{\mathrm{ecu}}(a) \wedge \mathrm{actCnt}(a', \omega) < \mathrm{actCnt}(a, \omega)$

Prop. B17 (analog to Prop.A11). As explained in Prop.B5, memory failures never result into decrease of the counter value stored by an ECU. This implies that an action loseCnt$_{bC}$ performed by an honest ECU resets the ECU's counter to a value bigger than the correct one. For formal reasons we assign a counter value higher than the maximal value the system starts with to a loseCnt action if it is the first action of an action sequence.

$\forall a \in \mathrm{alph}(\omega)$:

(1) $a \neq \mathrm{pre}_1(\omega) \wedge \widehat{\kappa}_{\mathrm{aname}}(a) = \mathrm{loseCnt} \wedge \widehat{\kappa}_{\mathrm{ecu}}(a) \in \mathscr{E}CU_{bC} \Rightarrow \widehat{\kappa}_{\mathrm{ecucnt}}(a) > \mathrm{corCnt}_{bC}(a, \omega)$.

(2) $a = \mathrm{pre}_1(\omega) \wedge \widehat{\kappa}_{\mathrm{aname}}(a) = \mathrm{loseCnt} \wedge \widehat{\kappa}_{\mathrm{ecu}}(a) \in \mathscr{E}CU_{bC} \Rightarrow \widehat{\kappa}_{\mathrm{ecucnt}}(a) = bCnt_{\mathrm{max}} + 1$

Analogously to Section 5.1.2 we model immediacy and non-repeatability by the phase class $\Phi(\sigma(b), W_{bCnt})$ as defined in Definition 9 determined by a fixed but arbitrary recv$_{bC}$ action b. In the following section we will show that BusCnt satisfies both properties.

6.2. Formal Proof of bCnt System.

The idea of the BusCnt system is that counter values included in the MACs of sent messages are strictly monotonically decreasing (instead of strictly monotonically increasing as in the GenCnt system). However, in contrast to the GenCnt system, in BusCnt each send action inevitably induces a read$_{bC}$ or recv$_{bC}$ action and thus a decrement of the counter, no matter whether or not a check failed. In case a system does not suffer any anomalies (i.e. all actors act correctly and counter value change is never caused by physical irregularities), the counter used by the ECUs is always the correct one. Lemma 2 will show how it is determined. For its proof we need the following technical Lemma:

Lemma 1. *Let $\omega \in W^{cor}_{bCnt} := \{\omega \in W_{bCnt} | \widehat{\kappa}_{ecu}(a) \in \mathscr{E}CU_{bC} \wedge \widehat{\kappa}_{aname}(a) \neq loseCnt \, for \, all \, a \in alph(\omega)\}$. Let further $S(\omega) := \{a \in alph(\omega) | \widehat{\kappa}_{aname}(a) = send_{bC}\}$ with $actCnt(s_i, \omega) < actCnt(s_{(i+k)}, \omega)$ for all $s_i, s_{i+k} \in S(\omega) (i, k \in \mathbb{N})$. Then for all $b \in alph(\omega)$ with $\widehat{\kappa}_{aname}(b) \in \{read_{bC}, recv_{bC}\}$ the following holds:*

1. $\widehat{\kappa}_{msgid}(b) = fmsg \Rightarrow \widehat{\kappa}_{ecucnt}(b) = \widehat{\kappa}_{cnt}(b)$
2. $\widehat{\kappa}_{msgid}(b) = sync \Rightarrow \widehat{\kappa}_{ecucnt}(b) = \widehat{\kappa}_{msg}(b)$

Proof 2. If $\widehat{\kappa}_{\mathrm{aname}}(b) = \mathrm{recv}_{bC}$, the assertions follow directly by Prop.B10 and Prop.B12 together with Prop.B8, respectively. Let now $\widehat{\kappa}_{\mathrm{aname}}(b) = \mathrm{read}_{bC}$. We show the assertions of this case by induction over the number of consecutive phase classes $\Phi(s_i, W^{\mathrm{cor}}_{bCnt})$.

Induction basis: $i = 1$. Consider $v \in \Phi(s_1, W^{\mathrm{cor}}_{bCnt})$ and $b \in \mathrm{alph}(v)$. Let $\widehat{\kappa}_{\mathrm{msgid}}(b) = \mathrm{fmsg}$ and assume $\widehat{\kappa}_{\mathrm{ecu}}(s_1) = \widehat{\kappa}_{\mathrm{ecu}}(b)$. Then Prop.B13 and Prop.B4 imply

$\widehat{\kappa}_{\text{ecucnt}}(b) = \widehat{\kappa}_{\text{prevcnt}}(b) - 1 = \widehat{\kappa}_{\text{ecucnt}}(s_1) - 1$. Further, by Prop.B1, $\widehat{\kappa}_{\text{cnt}}(b) = \widehat{\kappa}_{\text{cnt}}(s_1)$. Now by Prop.B9, $\widehat{\kappa}_{\text{cnt}}(s_1) = \widehat{\kappa}_{\text{prevcnt}}(s_1) - 1 = \widehat{\kappa}_{\text{ecucnt}}(s_1) - 1$. Together this leads to $\widehat{\kappa}_{\text{cnt}}(b) = \widehat{\kappa}_{\text{ecucnt}}(s_1) - 1 = \widehat{\kappa}_{\text{ecucnt}}(b)$.

Assume now $\widehat{\kappa}_{\text{ecu}}(b) \neq \widehat{\kappa}_{\text{ecu}}(s_1)$. Since Prop.B1 requires a send action before any read_{bC} or recv_{bC} action and since s_1 is the first send action in ω, there is no other action in ω before s_1. Prop.B14 implies that b is the first $\widehat{\kappa}_{\text{ecu}}(b) \in \Omega\widehat{\kappa}_{\text{prevcnt}}(b) = bCnt_{\max}$ and thus Prop.B13 implies $\widehat{\kappa}_{\text{ecucnt}}(b) = \widehat{\kappa}_{\text{prevcnt}}(b) - 1 = bCnt_{\max} - 1$. s_1 being the first action in ω, it is the first action of $\widehat{\kappa}_{\text{ecu}}(s_1)$ as well and Prop.B4 implies $\widehat{\kappa}_{\text{prevcnt}}(s_1) = bCnt_{\max}$. By Prop.B9 it follows $\widehat{\kappa}_{\text{cnt}}(s_1) = \widehat{\kappa}_{\text{prevcnt}}(s_1) - 1 = bCnt_{\max} - 1$. Further, by Prop.B1, $\widehat{\kappa}_{\text{cnt}}(b) = \widehat{\kappa}_{\text{cnt}}(s_1)$. Together we can conclude $\widehat{\kappa}_{\text{cnt}}(s_1) = \widehat{\kappa}_{\text{cnt}}(s_1) = bCnt_{\max} - 1 = \widehat{\kappa}_{\text{ecucnt}}(b)$.

In case $\widehat{\kappa}_{\text{msgid}}(b) = \text{sync}$, we can argue analogously by replacing every occurrence of $\widehat{\kappa}_{\text{cnt}}$ by $\widehat{\kappa}_{\text{msg}}$ and applying Prop.B6 instead of Prop.B9.

Induction hypothesis: For $v \in \Phi(s_i, W^{\text{cor}}_{bCnt})$ and $b \in \text{alph}(v)$, let assertions 1 and 2 hold.

Induction step: Consider $v \in \Phi(s_{i+1}, W^{\text{cor}}_{bCnt}), b \in alph(v)$ with $\widehat{\kappa}_{\text{aname}}(b) \in \{\text{read}_{bC}, \text{recv}_{bC}\}$ and $\widehat{\kappa}_{\text{msgid}}(b) = \text{fmsg}$. First we again assume $\widehat{\kappa}_{\text{ecu}}(b) = \widehat{\kappa}_{\text{ecu}}(s_{i+1})$. b being a read_{bC} action, Prop.B13 implies $\widehat{\kappa}_{\text{ecucnt}}(b) = \widehat{\kappa}_{\text{prevcnt}}(b) - 1$ which by Prop.B4 and Prop.B14 is equal to $\widehat{\kappa}_{\text{ecucnt}}(s_{i+1}) - 1$. Further, by Prop.B9, $\widehat{\kappa}_{\text{cnt}}(s_{i+1}) = \widehat{\kappa}_{\text{prevcnt}}(s_{i+1}) - 1$ and $\widehat{\kappa}_{\text{ecucnt}}(s_{i+1}) = \widehat{\kappa}_{\text{prevcnt}}(s_{i+1})$. Since Prop.B1 implies $\widehat{\kappa}_{\text{cnt}}(s_{i+1}) = \widehat{\kappa}_{\text{cnt}}(s_{i+1})$, it follows $\widehat{\kappa}_{\text{cnt}}(b) = \widehat{\kappa}_{\text{prevcnt}}(s_{i+1}) - 1 = \widehat{\kappa}_{\text{ecucnt}}(s_{i+1}) - 1 = \widehat{\kappa}_{\text{ecucnt}}(b)$.

Assume now $\widehat{\kappa}_{\text{ecu}}(b) \neq \widehat{\kappa}_{\text{ecu}}(s_{i+1})$. As above, Prop.B13 implies $\widehat{\kappa}_{\text{ecucnt}}(b) = \widehat{\kappa}_{\text{prevcnt}}(b) - 1$. Since by Prop.B14 all ECUs perform a read_{bC} or recv_{bC} action in the previous phase class $\Phi(s_i, W^{\text{cor}}_{bCnt})$ (loseCnt_{bC} actions are excluded by definition), this holds in particular for $\widehat{\kappa}_{\text{ecu}}(b)$ and $\widehat{\kappa}_{\text{ecu}}(s_{i+1})$. Hence for all maximal v' in $\Phi(s_i, W^{\text{cor}}_{bCnt})$ there exist read_{bC} or recv_{bC} actions $b' \in \text{alph}(v')$ performed by $\widehat{\kappa}_{\text{ecu}}(b)$ and $a \in \text{alph}(v')$ performed by $\widehat{\kappa}_{\text{ecu}}(s_{i+1})$, being the previous actions of $\widehat{\kappa}_{\text{ecu}}(b)$ and $\widehat{\kappa}_{\text{ecu}}(s_{i+1})$, respectively. Prop.B4 implies $\widehat{\kappa}_{\text{ecucnt}}(b) = \widehat{\kappa}_{\text{prevcnt}}(b) - 1 = \widehat{\kappa}_{\text{ecucnt}}(b') - 1$ which by induction hypothesis is equal to $\widehat{\kappa}_{\text{cnt}}(b') - 1$. This in turn is equal to $\widehat{\kappa}_{\text{cnt}}(s_i) - 1 = \widehat{\kappa}_{\text{cnt}}(a) - 1$ by Prop.B1. Again by induction hypothesis, the latter expression is equal to $\widehat{\kappa}_{\text{ecucnt}}(a) - 1$. Prop.B4 implies equality to $\widehat{\kappa}_{\text{prevcnt}}(s_{i+1}) - 1$ which by Prop.B9 is equal to $\widehat{\kappa}_{\text{cnt}}(s_{i+1})$. Prop.B1 finally implies equality to $\widehat{\kappa}_{\text{cnt}}(b)$, hence $\widehat{\kappa}_{\text{cnt}}(b) = \widehat{\kappa}_{\text{ecucnt}}(b)$.

Again, in case $\widehat{\kappa}_{\text{msgid}}(b) = \text{sync}$, the analogous proof is achieved by replacing every occurrence of $\widehat{\kappa}_{\text{cnt}}$ by $\widehat{\kappa}_{\text{msg}}$ and applying Prop.B6 instead of Prop.B9. □

Lemma 2. *Let W^{cor}_{bCnt} and $S(\omega)$ as defined in Lemma 1. Let further $\omega \in W^{cor}_{bCnt}$ and $a \in alph(\omega)$. Then the following holds:*

1. $\widehat{\kappa}_{msgid}(a) = fmsg \Rightarrow \widehat{\kappa}_{cnt}(a) = bCnt_{\max} - card(\{s \in S(\omega) | actCnt(s, \omega) \leq actCnt(a, \omega)\}) \geq 0$
2. $\widehat{\kappa}_{msgid}(a) = sync \Rightarrow \widehat{\kappa}_{msg}(a) = bCnt_{\max} - card(\{s \in S(\omega) | actCnt(s, \omega) \leq actCnt(a, \omega)\}) \geq 0$

Further, for all $s_{i-1}, s_i \in S(\omega)$ (i.e. with $actCnt(s_{i-1}, \omega) < actCnt(s_i, \omega)$ and $i \in \mathbb{N}, i \geq 2$) holds

(i) $\widehat{\kappa}_{msgid}(s_{i-1}) = \widehat{\kappa}_{msgid}(s_i) = fmsg \Rightarrow \widehat{\kappa}_{cnt}(s_i) = \widehat{\kappa}_{cnt}(s_{i-1}) - 1$

(ii) $\widehat{\kappa}_{msgid}(s_{i-1}) = \widehat{\kappa}_{msgid}(s_i) = sync \Rightarrow \widehat{\kappa}_{msg}(s_i) = \widehat{\kappa}_{msg}(s_{i-1}) - 1$

(iii) $\widehat{\kappa}_{msgid}(s_{i-1}) = fmsg \wedge \widehat{\kappa}_{msgid}(s_i) = sync \Rightarrow \widehat{\kappa}_{msg}(s_i) = \widehat{\kappa}_{cnt}(s_{i-1}) - 1$

(iv) $\widehat{\kappa}_{msgid}(s_{i-1}) = sync \wedge \widehat{\kappa}_{msgid}(s_i) = fmsg \Rightarrow \widehat{\kappa}_{cnt}(s_i) = \widehat{\kappa}_{msg}(s_{i-1}) - 1$

Note that item 1 implies that the parameter cnt of actions concerning a functional message never reaches the value 0 unless there occur no more send_{bC} actions after the action a. The analogous statement holds for the parameter msg of actions concerning synchronization messages.

Proof 3. We prove assertions 1 and 2 by induction over the length $l \in \mathbb{N}$ of a word $\omega \in W^{\text{cor}}_{bCnt}$.

Induction basis: $l = 1$, i.e. $\omega = a_1$. Since Prop.B1 requires a send_{bC} action before any read_{bC} or recv_{bC} action, a_1 cannot be a read_{bC} or recv_{bC} action. Since further by definition ω does not contain any loseCnt action, $\widehat{\kappa}_{\text{aname}}(a_1) = \text{send}_{bC}$. Prop.B4 implies $\widehat{\kappa}_{\text{prevcnt}}(a_1) = bCnt_{\max}$. If $\widehat{\kappa}_{\text{msgid}}(a_1) = \text{fmsg}$, by Prop.B9 it follows $\widehat{\kappa}_{\text{cnt}}(a_1) = \widehat{\kappa}_{\text{prevcnt}}(a_1) - 1 = bCnt_{\max} - 1 = bCnt_{\max} - \text{card}(\{s \in S(a_1) | \text{actCnt}(s, a_1) \leq \text{actCnt}(a_1, a_1)\})$. Further, by Prop.B5 $\widehat{\kappa}_{\text{prevcnt}}(a_1) = bCnt_{\max} > 0$ which implies $\widehat{\kappa}_{\text{prevcnt}}(a_1) - 1 = bCnt_{\max} - 1 \geq 0$. Thus item 1 holds for $\omega = a_1$ containing a functional message. If on the other hand $\widehat{\kappa}_{\text{msgid}}(a_1) = \text{sync}$, by Prop.B6 it follows $\widehat{\kappa}_{\text{msg}}(a_1) = \widehat{\kappa}_{\text{prevcnt}}(a_1) - 1$ which as above implies the assertion, thus item 2 holds for $\omega = a_1$.

Induction hypothesis: Let $\omega_i = a_1 \ldots a_i$ $(i \geq 2, i \in \mathbb{N})$. Then for all $a \in \text{alph}(\omega_i)$ holds:

(1) $\widehat{\kappa}_{\text{msgid}}(a) = \text{fmsg} \Rightarrow \widehat{\kappa}_{\text{cnt}}(a) = bCnt_{\max} - \text{card}(\{s \in S(\omega_i) | \text{actCnt}(s, \omega_i) \leq \text{actCnt}(a, \omega_i)\}) \geq 0$

(2) $\widehat{\kappa}_{\text{msgid}}(a) = \text{sync} \Rightarrow \widehat{\kappa}_{\text{msg}}(a) = bCnt_{\max} - \text{card}(\{s \in S(\omega_i) | \text{actCnt}(s, \omega_i) \leq \text{actCnt}(a, \omega_i)\}) \geq 0$

Induction step: Consider $\omega_{i+1} := a_1 \ldots a_i a_{i+1}$.

(1) Assume $\widehat{\kappa}_{\text{aname}}(a_{i+1}) = \text{send}_{bC}$. By Prop.B14 and the fact that ω_{i+1} does not contain any loseCnt_{bC} actions, it follows that a_i is a recv_{bC} or read_{bC} action. This in turn is preceded by a send_{bC} action $\sigma(a_i)$ (see Prop.B1). So we have $\omega_{i+1} = a_1 \ldots \sigma(a_i) \ldots a_i a_{i+1}$ ($\sigma(a_i)$ may or may not be equal to a_1). Let v_i a maximal word in $\Phi(\sigma(a_i), W^{\text{cor}}_{bCnt})$, i.e. v_i starts with $\sigma(a_i)$ and ends with a_{i+1}, and all other actions in between are read_{bC} and recv_{bC} actions, the last one being a_i. By Prop.B1, for all these read_{bC} and recv_{bC} actions $b \in \text{alph}(v_i)$ holds $\widehat{\kappa}_{\text{cnt}}(b) = \widehat{\kappa}_{\text{cnt}}(\sigma(a_i)) = \widehat{\kappa}_{\text{cnt}}(a_i)$. Since all ECUs perform exactly one $read_{bC}$ or recv_{bC} action in v_i, there is exactly one recv_{bC} or read_{bC} action $b^* \in \text{alph}(v_i)$ performed by $\text{ecu}^* := \widehat{\kappa}_{\text{ecu}}(a_{i+1})$, being the last action performed by ecu^* before a_{i+1}.

(a) Assume $\widehat{\kappa}_{\text{msgid}}(a_{i+1}) = \text{fmsg}$. If $\widehat{\kappa}_{\text{msgid}}(b^*) = \text{fmsg}$ as well, Lemma 1 implies $\widehat{\kappa}_{\text{ecucnt}}(b^*) = \widehat{\kappa}_{\text{cnt}}(b^*)$.

Since a_{i+1} contains a functional message, Prop.B9 implies $\widehat{\kappa}_{\text{cnt}}(a_{i+1}) = \widehat{\kappa}_{\text{prevcnt}} a_{(i+1)} - 1 = \widehat{\kappa}_{\text{ecucnt}}(b^*) - 1 = \widehat{\kappa}_{\text{cnt}}(b^*) - 1$. By Prop.B1 this is equal to $\widehat{\kappa}_{\text{cnt}}(\sigma(a_i)) - 1 = \widehat{\kappa}_{\text{cnt}}(a_i) - 1$. Prop.B1 also implies $\widehat{\kappa}_{\text{msgid}}(a_i) = \widehat{\kappa}_{\text{msgid}}(b^*) = \text{fmsg}$, hence by induction hypothesis it follows $\widehat{\kappa}_{\text{cnt}}(a_{i+1}) = \widehat{\kappa}_{\text{cnt}}(a_i) - 1 = bCnt_{\max} - \text{card}(\{s \in S(\omega_{i+1}) | \text{actCnt}(s, \omega_{i+1}) \leq \text{actCnt}(a_i, \omega_{i+1})\}) - 1$. Finally, since a_{i+1} is the send$_{bC}$ action directly following a_i, $\widehat{\kappa}_{\text{cnt}}(a_{i+1}) = bCnt_{\max} - \text{card} \ \{s \in S(\omega_{i+1}) | \text{actCnt}(s, \omega_{i+1}) \leq \text{actCnt}(a_{i+1}, \omega_{i+1})\}$. Further, since $\widehat{\kappa}_{\text{cnt}}(a_{i+1}) = \widehat{\kappa}_{\text{prevcnt}}(a_{i+1}) - 1$ and by Prop.B5 $\widehat{\kappa}_{\text{prevcnt}}(a_{i+1}) > 0$, it follows $\widehat{\kappa}_{\text{cnt}}(a_{i+1}) \geq 0$.

If on the other hand $\widehat{\kappa}_{\text{msgid}}(b^*) = \text{sync}$, Lemma 1 implies $\widehat{\kappa}_{\text{ecucnt}}(b^*) = \widehat{\kappa}_{\text{msg}}(b^*)$. Using Prop.B9, we can then deduce $\widehat{\kappa}_{\text{cnt}}(a_{i+1}) = \widehat{\kappa}_{\text{ecucnt}}(b^*) - 1 = \widehat{\kappa}_{\text{msg}}(b^*) - 1$ which by Prop.B1 is equal to $\widehat{\kappa}_{\text{msg}}(a_i) - 1$. As above, the assertion follows.

(b) Assume $\widehat{\kappa}_{\text{msgid}}(a_{i+1}) = \text{sync}$. In this case we can use Prop.B6 to deduce $\widehat{\kappa}_{\text{msg}}(a_{i+1}) = \widehat{\kappa}_{\text{prevcnt}}(a_{i+1}) - 1$. If $\widehat{\kappa}_{\text{msgid}}(b^*) = \text{fmsg}$, the rest of the proof is identical to the case where this is combined with a functional message of a_{i+1}, if $\widehat{\kappa}_{\text{msgid}}(b^*) = \text{sync}$, the arguments regarding a_{i+1} being a synchronization message apply. This ends the proof for the case $\widehat{\kappa}_{\text{aname}}(a_{i+1}) = \text{send}_{bC}$.

(2) Assume $\widehat{\kappa}_{\text{aname}}(a_{i+1}) \in \{\text{read}_{bC}, \text{recv}_{bC}\}$. Then Prop.B1 implies that there is an action $\sigma(a_{i+1})$ which is either equal to a_i or occurs before a_i.

(a) Let $\sigma(a_{i+1}) = a_i$. By Prop.B1 it follows $\widehat{\kappa}_{\text{cnt}}(a_{i+1}) = \widehat{\kappa}_{\text{cnt}}(a_i), \widehat{\kappa}_{\text{msg}}(a_{i+1}) = \widehat{\kappa}_{\text{msg}}(a_i)$ and $\widehat{\kappa}_{\text{msgid}}(a_{i+1}) = \widehat{\kappa}_{\text{msgid}}(a_i)$. Since a_i is the last send action before the recv$_{bC}$/read$_{bC}$ action a_{i+1}, the number of send actions before these two actions including a_i is identical, i.e. $\text{card}(\{s \in S(\omega_{i+1}) | \text{actCnt}(s, \omega_{i+1}) \leq \text{actCnt}(a_i, \omega_{i+1})\}) = \text{card}(\{s \in S(\omega_{i+1}) | \text{actCnt}(s, \omega_{i+1}) \leq \text{actCnt}(a_{i+1}, \omega_{i+1})\})$. If $\widehat{\kappa}_{\text{msgid}}(a_i) = \text{fmsg}$, the induction hypothesis implies $0 \leq \widehat{\kappa}_{\text{cnt}}(a_i) = bCnt_{\max} - \text{card}(\{s \in S(\omega_{i+1}) | \text{actCnt}(s, \omega_{i+1}) \leq \text{actCnt}(a_i, \omega_{i+1})\}) = bCnt_{\max} - \text{card}(\{s \in S(\omega_{i+1}) | \text{actCnt}(s, \omega_{i+1}) \leq \text{actCnt}(a_{i+1}, \omega_{i+1})\}) = \widehat{\kappa}_{\text{cnt}}(a_{i+1})$. If $\widehat{\kappa}_{\text{msgid}}(a_i) = \text{sync}$, the induction hypothesis implies $0 \leq \widehat{\kappa}_{\text{msg}}(a_i) = bCnt_{\max} - \text{card}(\{s \in S(\omega_{i+1}) | \text{actCnt}(s, \omega_{i+1}) \leq \text{actCnt}(a_i, \omega_{i+1})\}) = bCnt_{\max} - \text{card}(\{s \in S(\omega_{i+1}) | \text{actCnt}(s, \omega_{i+1}) \leq \text{actCnt}(a_{i+1}, \omega_{i+1})\}) = \widehat{\kappa}_{\text{msg}}(a_{i+1})$.

(b) Let $\sigma(a_{i+1}) =: a'$ with $\text{actCnt}(a', \omega_{i+1}) > \text{actCnt}(a_i, \omega_{i+1})$. Then Prop.B14 and the fact that ω_{i+1} does not contain any loseCnt$_{bC}$ actions implies that a_i is a read$_{bC}$ or recv$_{bC}$ action. Again, by Prop.B1 it follows $\widehat{\kappa}_{\text{cnt}}(a_i) = \widehat{\kappa}_{\text{cnt}}(\sigma(a_i)) = \widehat{\kappa}_{\text{cnt}}(\sigma(a_{i+1})) = \widehat{\kappa}_{\text{cnt}}(a_{i+1})$ and the equivalent equations for the parameters msg and msgid. Now we can again conclude that the number of send actions before a_i is identical to those before a_{i+1} and as above, by induction hypothesis for the cases $\widehat{\kappa}_{\text{msgid}}(a_i) = \text{fmsgid}$ and $\widehat{\kappa}_{\text{msgid}}(a_i) = \text{sync}$, respectively, it follows the assertion. This concludes the proof of assertions 1 and 2.

We now prove assertions (i)–(iv) of the Lemma. So assume the first statement holds regarding functional messages. Then it holds in particular for two consecutive send$_{bC}$ actions s_{i-1} and s_i in a word $\omega \in W^{\text{cor}}_{bCnt}$, each sending a functional message. Then $\widehat{\kappa}_{\text{cnt}}(s_i) = bCnt_{\max} - \text{card}(\{s \in S(\omega) | \text{actCnt}(s, \omega) \leq \text{actCnt}(s_i, \omega)\}) = bCnt_{\max} - (\text{card}(\{s \in S(\omega) | \text{actCnt}(s, \omega) \leq \text{actCnt}(s_{i-1}, \omega)\}) + 1) = bCnt_{\max} - \text{card}(\{s \in S(\omega) | \text{actCnt}(s, \omega) \leq \text{actCnt}(s_{i-1}, \omega)\}) - 1 = \widehat{\kappa}_{\text{cnt}}(s_{i-1}) - 1$. The only difference between a send$_{bC}$ action s containing a synchronization message and one containing a functional message is that the value $bCnt_{\max}$ minus the cardinality of send$_{bC}$ actions happening before s and including s is assigned to the counter of s in the first case and to the message of s in the second case. Hence, the respective statements for all other combinations of synchronization and functional send$_{bC}$ actions can be shown analogously. □

The above Lemma uses properties that describe the behavior of honest agents when sending, receiving or reading a message and shows the resulting counter value. This behavior does not depend on whether or not any involved send action is performed by an honest ECU. Hence the counter value included in actions of sequences in W^{cor}_{bCnt} can be considered the correct one for actions in W_{bCnt}.

Definition 10. For $\omega \in W_{bCnt}$ and $a \in \text{alph}(\omega)$ we define

$$\text{corCnt}_{bC}(a, \omega) := bCnt_{\max} - \text{card}(\{s \in S(\omega) | \text{actCnt}(s, \omega) \leq \text{actCnt}(a, \omega)\}). \tag{9}$$

We now consider W_{bCnt} again and first show a Lemma whose important statement is that a synchronization message received by an ECU never contains a counter smaller than the correct one for this action. In general the Lemma states that an honest ECU owning the correct counter before processing a read$_{bC}$ or recv$_{bC}$ action (denoted by the parameter prevcnt of the action), also owns the correct counter afterwards (denoted by the parameter ecucnt).

Lemma 3. *Let $\omega \in W_{bCnt}$ and $a \in alph(\omega)$. Then the following holds:*

$$\widehat{\kappa}_{\text{aname}}(a) \in \{\text{read}_{bC}, \text{recv}_{bC}\} \wedge \widehat{\kappa}_{\text{prevcnt}}(a) = \text{corCnt}_{bC}(a, \omega) + 1 \Longrightarrow \widehat{\kappa}_{\text{ecucnt}}(a) = \text{corCnt}_{bC}(a, \omega). \tag{10}$$

Proof 4. Assume $\widehat{\kappa}_{\text{aname}}(a) = \text{read}_{bC}$. Then by Prop.B13 $\widehat{\kappa}_{\text{ecucnt}}(a) = \widehat{\kappa}_{\text{prevcnt}}(a) - 1 = \text{corCnt}_{bC}(a, \omega) + 1 - 1 = \text{corCnt}_{bC}(a, \omega)$. The same follows by Prop.B10 for a being a recv$_{bC}$ action of a functional message. So let $\widehat{\kappa}_{\text{aname}}(a) = \text{recv}_{bC}$ and $\widehat{\kappa}_{\text{msgid}}(a) = \text{sync}$. Prop.B12 implies $\widehat{\kappa}_{\text{errorFrame}}(a) = \text{no}$ and thus by Prop.B8 it follows $\widehat{\kappa}_{\text{ecucnt}}(a) \leq \widehat{\kappa}_{\text{prevcnt}}(a) - 1 = \text{corCnt}_{bC}(a, \omega) + 1 - 1$. In case of $\widehat{\kappa}_{\text{ecucnt}}(a) = \widehat{\kappa}_{\text{prevcnt}}(a) - 1$ the assertion holds. So let $\widehat{\kappa}_{\text{ecucnt}}(a) < \widehat{\kappa}_{\text{prevcnt}}(a) - 1$, i.e. $\widehat{\kappa}_{\text{ecucnt}}(a) < \text{corCnt}_{bC}(a, \omega)$. Prop.B16 implies that before a, $\widehat{\kappa}_{\text{ecu}}(a)$ performs a loseCnt action a' and by Prop.B17, if $a' \neq \text{pre}_1(\omega)$, it follows $\widehat{\kappa}_{\text{ecucnt}}(a')>\text{corCnt}_{bC}(a',\omega)$. For simplicity we assume that a' is the last action of $\widehat{\kappa}_{\text{ecu}}(a)$ in ω before a. By Prop.B14, every ECU performs only one action per phase class, hence a' happens in the phase class $\Phi(s',W_{bCnt})$ that ends with $\sigma(a)$, i.e. in the phase class directly occuring before the one including a. Lemma 2 implies that $\text{corCnt}_{bC}(a',\omega)=\text{corCnt}_{bC}(a,\omega)+1$. So we have $\text{corCnt}_{bC}(a,\omega)+1=\widehat{\kappa}_{\text{prevcnt}}(a)=\widehat{\kappa}_{\text{ecucnt}}(a')>\text{corCnt}_{bC}(a',\omega)=\text{corCnt}_{bC}(a,\omega)+1$. This constitutes a contradiction, hence $\widehat{\kappa}_{\text{ecucnt}}(a)=\text{corCnt}_{bC}(a,\omega)$ holds. In case a' is not the previous action of $\widehat{\kappa}_{\text{ecu}}(a)$, we can argue analogously.

Let $a' = \text{pre}_1(\omega)$. Since a' is the first action and not a send_{bC} action, Lemma 2 implies $\text{corCnt}_{bC}(a',\omega) = bCnt_{\max}$. As above, we have $\text{corCnt}_{bC}(a,\omega) + 1 = \widehat{\kappa}_{\text{prevcnt}}(a) = \widehat{\kappa}_{\text{ecucnt}}(a')$ which by Prop.B17 is equal to $bCnt_{\max} + 1 = \text{corCnt}_{bC}(a',\omega) + 1$. This implies $\text{corCnt}_{bC}(a,\omega) = \text{corCnt}_{bC}(a',\omega)$. However, since $\sigma(a)$ happens before the recv_{bC} action a (and after the loseCnt_{bC} action a'), by Lemma 2 $\text{corCnt}_{bC}(a) = bCnt_{\max} - 1 \neq bCnt_{\max} = \text{corCnt}_{bC}(a',\omega)$. So again this constitutes a contradiction, hence $\widehat{\kappa}_{\text{ecucnt}}(a) = \text{corCnt}_{bC}(a,\omega)$ always holds. □

We can now prove our main Theorem. As in Section 5.2, the property we want to prove is that whenever an honest ECU receives and accepts a message (action b), the send_{bC} action $\sigma(b)$ having triggered b and starting the phase class determined by b must have authentically for the ECU been performed by an agent being member of the same group, and the message must contain the correct counter. In contrast to the proof regarding the GenCnt system, for the BusCnt system (formally denoted by B_{bCnt}) we can show that this property is always satisfied, i.e. that both immediacy and non-repeatability hold.

Theorem 2. *Let $\omega \in B_{bCnt}$ and $b \coloneqq (recv_{bC}, ecu, ecukey, ecucnt, prevcnt, bus, mackey, msgid, msg, cnt) \in alph(\omega)$ with $ecu \in \mathcal{E}CU_{bC}$. Then the following property holds:*

$$\text{authWiPhase}\{(\text{send}, \text{ecu}', \text{ecukey}', \text{ecucnt}', \text{prevcnt}', \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt}) \mid ecu' \in \mathcal{E}CU_{bC}\}, b, \text{ecu}, \Phi(\sigma(b), W_{bCnt\cdot}) \tag{11}$$

Proof 5. Analogously to Section 5.2, without loss of generality we assume $\text{ECU}_1^{bC} \in \mathcal{E}CU_{bC}$ to perform a receive action $b \coloneqq (\text{recv}_{bC}, \text{ECU}_1^{bC}, \text{ecukey}, \text{ecucnt}, \text{prevcnt}, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt}) \in \text{alph}(\omega)$ and consider an arbitrary $x \in \lambda^{-1}_{\text{ECU}_1^{bC}}(\lambda_{\text{ECU}_1^{bC}}(\omega)) \cap W_{bCnt}$ (which by definition of $\lambda_{\text{ECU}_1^{bC}}$ and $B_{bCnt} \subseteq W_{bCnt}$ contains b). Since $\text{ECU}_1^{bC} \in \mathcal{E}CU_{bC}$, by Prop.B2 it follows mackey = key = ecukey. Further, by Prop.B3 there is an action $a_1 \coloneqq (\text{send}_{bC}, \text{ecu}_1, \text{ecukey}_1, \text{ecucnt}_1, \text{prevcnt}_1, \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt}) \in \text{alph}(x)$ before the receive action b by ECU_1^{bC} containing the same message ID, message and counter value and with $\text{ecukey}_1 = \text{mackey}$. Again by Prop.B2 it follows $\text{ecu}_1 \in \mathcal{E}CU_{bC}$ which proves that the message received in b has authentically for ECU_1^{bC} been generated and sent by a member of $\mathcal{E}CU_{bC}$.

Further, by Prop.B1 the receive action b by ECU_1^{bC} is preceded by a send action $\sigma(b) = (\text{send}_{bC}, \text{ecu}', \text{ecukey}', \text{ecucnt}', \text{prevcnt}', \text{bus}, \text{mackey}, \text{msgid}, \text{msg}, \text{cnt})$ triggering b.

Let v maximal in $\Phi(\sigma(b), W_{bCnt})$ with $b \in \text{alph}(v)$. Since $\widehat{\kappa}_{\text{aname}}(b) = \text{recv}_{bC}$, Prop.B12 implies $\widehat{\kappa}_{\text{errorFrame}}(b) = \text{no}$ and thus $\widehat{\kappa}_{\text{errorFrame}}(c) = \text{no}$ for all $c \in \text{alph}(v)$ with $\widehat{\kappa}_{\text{aname}}(c) \in \{\text{read}_{bC}, \text{recv}_{bC}\}$. By Prop.B15 one of the actions read_{bC}, recv_{bC} in v is performed by an ECU owning the correct counter, i.e. there exists $c^* \in \text{alph}(v)$ with $\widehat{\kappa}_{\text{aname}}(c^*) \in \{\text{read}_{bC}, \text{recv}_{bC}\}, \text{ECU}^*(\sigma(b)) \coloneqq \widehat{\kappa}_{\text{ecu}}(c^*) \in \mathcal{E}CU_{bC}$, $\widehat{\kappa}_{\text{errorFrame}}(c^*) = \text{no}$ and $\widehat{\kappa}_{\text{prevcnt}}(c^*) = \text{corCnt}_{bC}(c^*, x) + 1$. Lemma 3 implies $\widehat{\kappa}_{\text{ecucnt}}(c^*) = \text{corCnt}_{bC}(c^*, x)$. Recall that Prop.B1 and Prop.B3 imply that the values of the parameters msg, msgid and cnt in $a_1, \sigma(b), b$ and c^* are identical.

(1) Assume $\widehat{\kappa}_{\text{msgid}}(c^*) = \text{fmsg}$ and assume further that when sending the message, ecu_1 is synchronized, i.e. includes the correct counter as the message's counter value. By Prop.B9 it follows $\widehat{\kappa}_{\text{cnt}}(a_1) = \widehat{\kappa}_{\text{prevcnt}}(a_1) - 1 = \text{corCnt}_{bC}(a_1, x)$. Since all read_{bC} and recv_{bC} actions in v have errorFrame = no, Prop.B11 implies $\widehat{\kappa}_{\text{cnt}}(c^*) = \widehat{\kappa}_{\text{ecucnt}}(c^*)$, hence $\widehat{\kappa}_{\text{cnt}}(c^*) = \text{corCnt}_{bC}(c^*, x)$. It follows $\widehat{\kappa}_{\text{cnt}}(\sigma(b)) = \widehat{\kappa}_{\text{cnt}}(a_1) = \widehat{\kappa}_{\text{cnt}}(c^*) = \text{corCnt}_{bC}(c^*, x)$. Now if $a_1 \neq \sigma(b)$, the number of send_{bC} actions until a_1 is smaller than the number of send_{bC} actions until $\sigma(b)$. Since by Lemma 2 the correct counter minus any number of send_{bC} actions is always bigger or equal to 0, it follows that the correct counter for $\sigma(b)$ is different to the one for a_1. More specifically, it is smaller, i.e. $\text{corCnt}_{bC}(\sigma(b), x) \leq \text{corCnt}_{bC}(a_1, x) - 1$. Further, again by Lemma 2, the number of send actions having occurred until $\sigma(b)$ is equal to those having occurred until a read_{bC} or recv_{bC} action induced by $\sigma(b)$, i.e. $\text{corCnt}_{bC}(\sigma(b), x) = \text{corCnt}_{bC}(c^*, x)$. This implies $\widehat{\kappa}_{\text{cnt}}(a_1) = \widehat{\kappa}_{\text{cnt}}(\sigma(b)) = \widehat{\kappa}_{\text{cnt}}(c^*) = \text{corCnt}_{bC}(c^*, x) = \text{corCnt}_{bC}(\sigma(b), x) \leq \text{corCnt}_{bC}(a_1, x) - 1$. This constitutes a contradiction to our assumption that ecu_1 is synchronized in a_1 and sends $\widehat{\kappa}_{\text{cnt}}(a_1) = \text{corCnt}_{bC}(a_1, x)$. Thus $a_1 = \sigma(b)$.

(2) Assume $\widehat{\kappa}_{\text{msgid}}(c^*) = \text{sync}$ and ecu_1 is synchronized which by Prop.B7 implies that it sends the correct counter as the message's payload, i.e. $\widehat{\kappa}_{\text{msg}}(a_1) = \text{corCnt}_{bC}(a_1, x)$. Further, c^* being a read_{bC} or recv_{bC} action with $\widehat{\kappa}_{\text{errorFrame}}(c^*) = \text{no}$, Prop.B8 implies $\widehat{\kappa}_{\text{ecucnt}}(c^*) = \widehat{\kappa}_{\text{msg}}(c^*) \leq \widehat{\kappa}_{\text{prevcnt}}(c^*) - 1$. Hence it follows $\text{corCnt}_{bC}(c^*, x) = \widehat{\kappa}_{\text{ecucnt}}(c^*) = \widehat{\kappa}_{\text{msg}}(c^*) \leq$

$\widehat{\kappa}_{\text{prevcnt}}(c^*) - 1 = \text{corCnt}_{bC}(c^*, x)$ and thus $\widehat{\kappa}_{\text{msg}}(c^*) = \text{corCnt}_{bC}(c^*, x)$. Assume $a_1 \neq \sigma(b)$. As above, Lemma 2 implies $\text{corCnt}_{bC}(\sigma(b), x) \leq \text{corCnt}_{bC}(a_1, x) - 1$ and $\text{corCnt}_{bC}(\sigma(b), x) = \text{corCnt}_{bC}(c^*, x)$. Since by Prop.B1 $\widehat{\kappa}_{\text{msg}}(a_1) = \widehat{\kappa}_{\text{msg}}(\sigma(b)) = \widehat{\kappa}_{\text{msg}}(c^*)$, it follows $\text{corCnt}_{bC}(a_1, x) = \widehat{\kappa}_{\text{msg}}(a_1) = \widehat{\kappa}_{\text{msg}}(c^*) = \text{corCnt}_{bC}(c^*, x) = \text{corCnt}_{bC}(\sigma(b), x) \leq \text{corCnt}_{bC}(a_1, x) - 1$. This again constitutes a contradiction, thus it follows $a_1 = \sigma(b)$.

(3) Assume ecu_1 is not synchronized in a_1. By Prop.B16 it has performed an action loseCnt_{bC} before a_1, denoted by a_2. If $a_2 \neq \text{pre}_1(x)$, by Prop.B17, $\widehat{\kappa}_{\text{ecucnt}}(a_2)$ is set to a value bigger than the action's correct counter value: $\widehat{\kappa}_{\text{ecucnt}}(a_2) > \text{corCnt}_{bC}(a_2, x)$. If $a_2 = \text{pre}_1(x)$, Prop.B17 implies $\widehat{\kappa}_{\text{ecucnt}}(a_2) = bCnt_{\max} + 1 > \text{corCnt}_{bC}(a_2, x)$ as well. Assume for simplicity that ecu_1 's next action after a_2 is a_1. Then $\widehat{\kappa}_{\text{prevcnt}}(a_1) = \widehat{\kappa}_{\text{ecucnt}}(a_2) > \text{corCnt}_{bC}(a_2, x)$. Let $\widehat{\kappa}_{\text{msgid}}(a_1) = \text{fmsg}$. Then by Prop.B9, $\widehat{\kappa}_{\text{cnt}}(a_1) = \widehat{\kappa}_{\text{prevcnt}}(a_1) - 1 = \widehat{\kappa}_{\text{ecucnt}}(a_2) - 1 > \text{corCnt}_{bC}(a_2, x) - 1$. Now a_1 is the first send_{bC} action after a_2 since a send_{bC} action in between would imply another action by ecu_1 (Prop.B14). Lemma 2 implies $\text{corCnt}_{bC}(a_2, x) - 1 = \text{corCnt}_{bC}(a_1, x)$. Hence ecu_1 not being synchronized results into $\widehat{\kappa}_{\text{cnt}}(a_1) > \text{corCnt}_{bC}(a_1, x)$. If a_2 is not ecu_1 's last action before a_1 we can argue analogously only with a longer sequence of actions and counters in between a_2 and a_1 to consider, being decreased step by step. By Prop.B1 $\widehat{\kappa}_{\text{cnt}}(a_1) = \widehat{\kappa}_{\text{cnt}}(\sigma(b)) = \widehat{\kappa}_{\text{cnt}}(c^*)$ always holds. Hence $\text{corCnt}_{bC}(a_1, x) < \widehat{\kappa}_{\text{cnt}}(c^*) = \widehat{\kappa}_{\text{cnt}}(a_1)$. By Lemma 2 and the definition of corCnt_{bC}, $\text{corCnt}_{bC}(\sigma(b), x) = \text{corCnt}_{bC}(c^*, x)$, and since ECU^* owns the correct counter in c^*, by Prop.B15 it follows $\widehat{\kappa}_{\text{prevcnt}}(c^*) = \text{corCnt}_{bC}(c^*, x) + 1$ and Lemma 3 implies $\widehat{\kappa}_{\text{ecucnt}}(c^*) = \text{corCnt}_{bC}(c^*, x)$. Together these statements imply $\text{corCnt}_{bC}(\sigma(b), x) = \widehat{\kappa}_{\text{ecucnt}}(c^*)$. Further, Lemma 2 implies $\text{corCnt}_{bC}(a_1, x) \geq \text{corCnt}_{bC}(\sigma(b), x)$, no matter whether or not a_1 and $\sigma(b)$ are identical. Hence $\widehat{\kappa}_{\text{ecucnt}}(c^*) = \text{corCnt}_{bC}(\sigma(b), x) \leq \text{corCnt}_{bC}(a_1, x)$ and therefore $\widehat{\kappa}_{\text{ecucnt}}(c^*) \leq \text{corCnt}_{bC}(a_1, x) < \widehat{\kappa}_{\text{cnt}}(c^*)$. Prop.B11 implies $\widehat{\kappa}_{\text{errorFrame}}(c^*) = \text{yes}$, and by Prop.B12 it follows $\widehat{\kappa}_{\text{errorFrame}}(b) = \text{yes}$ and therefore $\widehat{\kappa}_{\text{aname}}(b) \neq \text{recv}_{bC}$, a contradiction to the assumption we started the proof with. Hence $\text{corCnt}_{bC}(a_1, x) = \widehat{\kappa}_{\text{cnt}}(a_1)$ in the case of $\widehat{\kappa}_{\text{msgid}}(a_1) = \text{fmsg}$.

If $\widehat{\kappa}_{\text{msgid}}(a_1) = \text{sync}$, we can decude $\widehat{\kappa}_{\text{errorFrame}}(b) = \text{yes}$ and thus the same contradiction by exchanging the parameter cnt by msg and applying Prop.B6 instead of Prop.B9 to deduce $\widehat{\kappa}_{\text{msg}}(a_1) > \text{corCnt}_{bC}(a_1, x)$. Further, Lemma 2 implies $\widehat{\kappa}_{\text{msg}}(a_1) = \widehat{\kappa}_{\text{msg}}(\sigma(b)) = \widehat{\kappa}_{\text{msg}}(c^*)$ and it follows $\widehat{\kappa}_{\text{msg}}(c^*) > \text{corCnt}_{bC}(a_1, x)$. With the same arguments as above, this implies $\widehat{\kappa}_{\text{msg}}(c^*) > \widehat{\kappa}_{\text{ecucnt}}(c^*)$. Prop.B8 implies $\widehat{\kappa}_{\text{errorFrame}}(c^*) = \text{yes}$ and again by by Prop.B12 it follows $\widehat{\kappa}_{\text{errorFrame}}(b) = \text{yes}$ and therefore $\widehat{\kappa}_{\text{aname}}(b) \neq \text{recv}_{bC}$. So again the assumption of ecu_1 not being synchronized leads to a contradiction. This concludes our proof.

The above, proof shows that our approach indeed satisfies data origin authenticity as well as immediacy and non-repeatability. In contrast, the generic counter system violates the latter ones. In the next section, we will discuss the security related differences in more detail. We will then introduce our proof of concept implementation showing its practicability and design decisions that substantiate our formal proof. □

7. Evaluation

In this section, we will evaluate both the security and the practicability of our bus counter approach. More specifically, in the next section we will discuss the formal proof results concerning the satisfaction of the security requirements immediacy and non-repeatability by the generic and the BusCnt system, respectively, and highlight the differences. In Section 7.2 we will then demonstrate the feasibility of our BusCnt approach based on a practical implementation and discuss design decisions.

7.1. Security Aspects. One fundamental difference between our approach and the generic counter-based approach is that in the BusCnt system the pulse generator is an integral component of the system itself: The very writing onto the bus causes a change of the local bus counter values of all ECUs connected to it as they inevitably read (part of) the message (even if not accepting it) and decrement their counters. By this read action, the message's counter and thus its MAC is invalidated. Hence, any subsequent message written onto the bus must use a smaller counter in order to be accepted. This prohibits message delay and replication.

Another important aspect is the assignment of computations to the controller that in traditional CAN communication systems is processed by the application layer. This concerns in particular MAC calculation and verification and checks regarding the size of the message's counter. Performing these checks on controller level enables to use the error frame mechanism of CAN in case of a failed check which in return results in invalidation of the respective message and prevention of its acceptance by any of the ECUs connected to the same bus.

These two aspects together allowed to formally prove that the BusCnt system satisfies both immediacy and non-repeatability, provided at least one of the ECUs owns the correct counter. The inevitable decrease of the counter value with every new message prohibits message delay, and the checks performed by the controllers allow immediate invalidation of any manipulated message.

One of the core assumptions of our proof is the correctness of at least one counter value in the bus network. It is based on two observations: First, ECUs are designed to be safe and thus hardware or software failures occur significantly less often compared to regular PC hardware. Second, an incorrect counter value may be the result of the shutdown process, as an ECU may not be able to store the counter value in time to persistent storage. However, the likelihood for this failure to occur is reduced in our approach since only a single value per bus needs to be stored. In contrast, traditional counter-based approaches use various counters for different types of messages. Moreover, a bus has a large number of ECUs connected to it and only one needs to store the correct counter value at the end of a ride. Therefore, the probability that our assumption does not hold is insignificant.

On the other hand, an ECU not being synchronized, i.e. not owning the correct counter, may in principle cause a safety problem: If it sends a functional message, MAC verification by ECUs owning the correct counter will fail and result into an error frame. If the ECU itself receives a functional message containing the correct counter, its own MAC verification will fail and cause an error frame as well. Too many error frame events will cause the ECU to change into an inactive state. This safety problem can be minimized by adequate synchronization approaches as discussed in Section 7.2.2 below.

In contrast to the BusCnt system, our proof of the generic counter-based approach indicates that it exhibits several weaknesses. First, it cannot ensure immediacy and non-repeatability (neither of synchronization nor of functional messages) in case an ECU loses its counter. Once being active again the ECU will accept any replayed message whose counter is still in the required range (i.e. bigger than its own counter). This violates immediacy, assuming that only the period between writing a message onto the bus and reading it does not exceed a specified limit. It also violates non-repeatability since the attack is possible even if the replayed message has already been accepted before.

The counter synchronization mechanism of GenCnt does not prohibit this attack as synchronization messages themselves are susceptible to delay attacks, i.e. recorded and then invalidated by an attacker by destroying their CRC or by interrupting the message with an error frame. These messages will then not be accepted by the ECUs with the consequence that an unsynchronized ECU cannot be synchronized. It will therefore accept delayed synchronization and functional messages at any later point in time the attacker chooses for a replay, thus violating immediacy. Hence, the synchronization mechanism of the generic counter-based system is no guarantee for ensuring immediacy and non-repeatability.

Even if no counter loss occurs, violation of immediacy by a delay of messages cannot be prohibited. This is due to the fact that the counter value stored by the intended recipients of a message does not change as long as all messages relevant for the respective counter are invalidated by the attacker and thus not accepted. Consequently, the intended recipients will accept any relayed message since it still contains a counter being valid from their point of view.

One question that comes to mind is whether it would be sufficient to equip the Fresh Value Manager FvM of the GenCnt system with the ability to perform all security checks by the controller in order to avoid the above-described attacks. However, it turns out that this measure alone is not enough. First, the FvM behavior would need to be changed as it must increment its own local counter value with every sent message, independently of whether or not the message is accepted. In other words, the determination of what is the correct counter for a message would need to be adapted to the one used for the BusCnt system. Otherwise, the FvM would not be able to detect the repetition of a message as it would still consider the old counter to be correct. Since the sender of a message that has caused an error frame normally simply resets its counter and resends the message, the FvM would additionally need to adjust the sender's counter. Secondly, the system would need to be changed to using one single counter per bus since otherwise the FvM is not able to assign an error frame to the counter used in a message that has been interrupted before writing the message ID to the bus. All these changes result in a system that is in some of its main aspects equivalent to the BusCnt system.

There are some assumptions in our proof regarding the satisfaction of immediacy and non-repeatability by the BusCnt system that need to be substantiated by specific design decisions. This concerns for example the size of the counter that must prohibit overflow. In the BusCnt system, an attacker can accelerate the pulse generator by inserting messages onto the bus. Independently of being accepted, they will cause the connected ECUs to decrement their counters faster than they normally would. However, in Section 7.2.2 below, we discuss the counter length necessary to avoid counter overflow even in the presence of such an attack and show how this length can be implemented in praxis. Other assumptions concern storage errors that we assume never to result in a counter being smaller than the correct one as this could lead to counter overflow as well. This is an issue for all counter-based approaches, adequate measures for detection of such incidents is out of the scope of this paper.

A final security aspect concerns the truncation of MACs which in principle enables an attacker to construct its own message and to determine the corresponding truncated MAC by brute force. This holds in particular for synchronization messages. So if an attacker was able to construct and insert a synchronization message containing e.g. the counter 0. . .0 with a correct MAC, it could take over the whole bus communication. This is an issue for all approaches using MAC truncation (e.g. for AUTOSARs SecOC). As a counter-measure, we have chosen a specific number of failed synchronization messages as indicator of a brute force attack which we deem small enough to recognize such attacks and on the other hand big enough to not cause unnecessary dysfunction of the bus. See Section 7.2.2 for more details on this aspect.

7.2. Practical Aspects. To evaluate the practical aspects of our work we implemented a proof of concept of BusCount to demonstrate the general feasibility of the mechanisms. Moreover, we want to show the suggested approach can be implemented at a low cost. We first introduce our development setup and describe the design decisions before presenting the evaluation we performed bsaed on this implementation.

7.2.1. Setup. For the implementation of our security enhanced CAN controller, we chose a low-cost FPGA (ICE40HX8K-B-EVN) with only 7680 logic cells and a maximum frequency of 12 MHz. The FPGA is connected with a CAN transceiver (MCP2561) that converts logical CAN messages into physical signals for a CAN bus.

We used an open-source implementation1 as a basis for our CAN controller. We extended the controller with an SLCAN2 protocol to communicate with a connected ECU. The ECU is simulated by a Raspberry Pi 2B running a default Linux SocketCAN3. The software does not need to be modified besides the fact that the payload is reduced by the length of the MAC. A second FPGA with the same setup was introduced to perform MAC verifications and synchronization tests. The correctness of the CAN implementation and the compatibility with regular CAN bus devices was evaluated with a remaining bus simulation using a Vector VN5610 in combination with CANoe v9. The hardware setup of our proof of concept (see Figure 5) contains:

7.2.2. Design Decisions. We decided to use a truncated MAC value with a size of 24 bit to be compatible with AUTOSAR SecOC. Since CAN has a very limited message size per package we decided to use the counter implicitly. This requires more explicit synchronizations yet does not disrupt the system's functionality due to the fast synchronization mechanism of our protocol. Corresponding to the 24 bit MAC the remaining payload of a CAN message is 40 bits. The counter transferred with the described synchronization mechanism would then also be restricted to these 40 bits. A CAN bus can transmit up to 17,543 messages per second [36], thus a 40 bit counter suffices for about 725.4 days ($2^{40}/17,543 \cdot 60 \cdot 60 \cdot 24$) of non-stop communication before an overflow occurs. An attacker may even reduce the duration by starting CAN messages and stopping them immediately with an error frame. This increases the number of messages sent by an attacker (15 bits per message) compared to a regular sender (minimum 44 bits per message) to about 51,459 messages per second ($17,543 \cdot 44/15$). We consider a counter value that could overflow after only 247.3 days ($2^{40}/51,459 \cdot 60 \cdot 60 \cdot 24$) not sufficient in an attack scenario, thus we increased the counter by additional 18 bits which can be transmitted using the extended message ID of the CAN specification for synchronization messages. The 58 bit counter is sufficient for about 82,884.75 years ($2^{58}/51,459 \cdot 60 \cdot 60 \cdot 24 \cdot 365$). In order to counter brute force attacks and to increase the security of the truncated MAC we suggest renewing the key regularly by deriving it from the current counter value.

Furthermore, we needed to make sure the synchronization cannot be attacked by a brute force attack. An attacker may try to forge a synchronization message setting the counter to 0 which would lead to an overflow or would establish the number 0 as the counter value of all subsequent messages. Since we suggest to transfer only a 24 bit MAC, attacks cannot be prevented by the key size. For this reason we count the number of failed synchronizations. If more than 16 failed synchronizations during a car ride or 128 failed synchronizations in total have been detected, the ECUs need to consider the bus no longer trustworthy and must enable the driver to safely stop the car. The 128 synchronizations give an attacker a 0.000977% $\left(\binom{128}{1}\binom{2^{24}}{127}\Big/\binom{2^{24}}{128}\right)$ chance to forge a synchronization message successfully. A recovery process for a car network is out of the scope of this paper. Compared to the generic counter-based approach, our synchronization solution has the advantage that it is independent of functional disruptions regarding a central entity (fresh value master): Any ECU can initialize synchronization, and all ECUs join in, sending their respective synchronization messages simultaneously. This mechanism allows the synchronization of all ECUs connected to one bus during the transmission time of only one message.

The bus counter-based security protocol does not change the CAN frame and thus it can work with default CAN controllers in one network. The adaptation to CAN FD and CAN XL is also possible. Further, it is compatible with SecOC with respect to the message structure.

The introduction of error frames in conjunction with security checks potentially changes the safety considerations of ISO26262 regarding CAN bus communication. A CAN controller sending a message which results in an error frame increments its error counter by 8 while receiving controllers increment their error counter by 1. Successful message transfer reduces the counter by 1. If an error counter exceeds the value 128, the respective controller changes into `bus-off` mode and is not able to send or receive data anymore and a hardware reset is necessary. However, there are only two cases that our security mechanisms could cause a `bus-off` state. In the first case, the attacker is the sender of unauthorized messages by delay, replay, or forging of messages. In this scenario, the attacker performs a denial of service attack which she could also perform in every other secure or non-secure CAN BUS by inserting error frames. The second scenario is the startup of a vehicle where every CAN controller is out of synchronization. In the worst case, ECUs might join the network one by one, each of them sending an error frame caused by a failed check based on its wrong counter value, and then initiate a synchronization. To prevent this we suggest that after waking up and before starting to send error frames, a CAN controller first initiates a synchronization in case of an invalid message. Thus, the ECU is in sync as soon as it joins the network.

7.2.3. Performance Evaluation. The CAN controller implementation we adapted to work on the ICE40HX8K consumed 4,483 of the 7,680 logic cells. The remaining logic cells need to be sufficient for processing a MAC algorithm, the synchronization, and the counter mechanism. Moreover, the MAC algorithm needs to be fast enough to calculate the MAC during the transmission of the truncated MAC value ($24 \cdot 4 = c$). The time consumed by the sending process is

CAN controller	ICE40HX8K-B-EVN
CAN transceiver	MCP2561
ECU	Raspberry Pi 2B
Remaining bus simulation	Vector VN5610

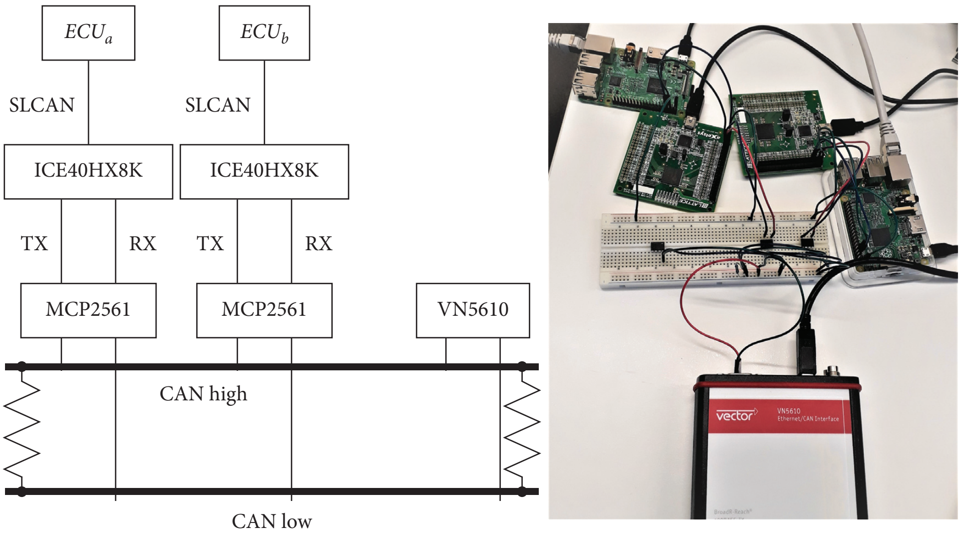


Figure 5: Evaluation setup for BusCount.

longer since much more bits are transmitted prior to the truncated MAC value. Finally, the MAC algorithm needs to calculate the MAC over at most 131 bits (58 bit counter value + 29 ID + 4 bit data length + 40 bit data) for classic CAN.

The remaining 3,197 cells are not sufficient to implement a regular cipher, like AES or HMAC with SHA2, with low latency together with the counter and synchronization mechanism. For this reason, we evaluated several lightweight ciphers regarding their number of rounds, state size, and table size. Based on these results we implemented our CAN controller in combination with the two CBC-MAC algorithms Present80 [37] and Prince [38] and the HMAC algorithm SipHash [39]. All three have a block size of 64 bit, so two blocks need to be processed at most. We evaluated the number of cycles each algorithm needs for this task. Table 6 shows the results of our evaluation. The number of cycles as well as the number of logical cells needed for the CAN controller including the different ciphers.

The smallest (in terms of cells needed) but also the slowest algorithm was Present80. Our implementation of the controller with Present80 needed 5,599 logic cells and 68 cycles for two blocks. Prince only needed 30 cycles while increasing the number of logic cells needed to 5,947. SipHash was only slightly larger with 6,024 logic cells, but needs only 13 cycles to compute a MAC over two blocks. Since the security of Present80 is lower and the number of blocks increases drastically regarding CAN FD (10 blocks for 579 bits of authentic data) and CAN XL (258 blocks for at most 16,466 bits of authentic data), we recommend SipHash for a fast and size efficient MAC algorithm in our approach. Additionally, cryptanalysises [40,41] of SipHash did not reveal problems with this cryptographic primitive.

Table 6: Evaluation of ciphers for secure CAN controller.

Algorithm	Cycles	Logic cells (total)
Plain CAN controller	-	4,483
SipHash [39]	13	6,024
Prince [38]	30	5,947
Present80 [37]	68	5,599

8. Conclusions and Future Work

In this paper, we have presented a detailed discussion and formal evaluation of our hardware-based approach BusCount for the security protection of automotive CAN networks. We further opposed this to the characteristics of counter-based approaches currently being used.

The fundamental difference between currently considered approaches and ours is that in ours the pulse generator is an integral component of the system itself: The very writing onto a bus causes a change of the counter values of all ECUs connected to it as they inevitably read the message (even if not accepting it) and decrement their counters. Since the number of messages sent on the bus cannot be manipulated, the correct counter value cannot be manipulated as well. By this read action, the message's counter and thus its MAC is invalidated. Another important aspect is the assignment of MAC calculation and verification and checks regarding the size of the message's counter to the controller. This enables to use the error frame mechanism of CAN communication in case of a failed check which results in invalidation of all messages whose MAC is not based on the correct counter value. These messages will then not be accepted by any of the ECUs. Software-based approaches do not have this possibility since they verify the MAC on application-level after the controller completely received the message.

We formally proved that our bus counter approach satisfies data origin authenticity as well as immediacy and non-repeatability (also denoted by message freshness), both during regular operation and in case an ECU loses its counter. Our proof is based on the assumption that simultaneous loss of counter values does not occur, i.e. that there is always at least one ECU per bus owning the correct value. This assumption seems appropriate, given the low possibility of it being violated. It enables our synchronization mechanism to take advantage of the physical characteristics of a CAN bus and ensures that always the correct counter value is sent.

On the other hand, current counter-based systems cannot assure message freshness if an ECU loses its counter. Immediacy is even violated in case an ECU does not lose its counter: An attacker can invalidate all messages relevant for a specific counter and insert them again at a later point in time without the ECU being able to notice this manipulation. This is due to the fact that the ECU's local counter value only changes if it actually accepts a message.

Compared to other approaches, our bus counter mechanism offers several practical advantages: It avoids the necessity to include (parts of) the counter in the messages which saves bandwidth, and it requires only one counter per bus to be stored instead of one counter per message ID favored by currently discussed approaches. This reduces both storage capacities and the risk of ECUs being unsynchronized.

The synchronization solution of BusCount has the advantage that it is independent of functional disruptions regarding a central fresh value master that is being used by other counter-based approaches: Any ECU can initialize synchronization, and all ECUs join in, sending their respective synchronization messages simultaneously. This mechanism allows the synchronization of all ECUs connected to the same bus during the transmission time of only one message.

It must be noted that our approach cannot be realized with currently available ECUs. On the other hand, it does not change the CAN frame and can thus work with default CAN controllers in one network. It can also be adapted to CAN FD and CAN XL and is compatible with SecOC with respect to the message structure. Moreover, our proof of concept implementation described in Section 7.2 shows that our approach is not only theoretically interesting, but is functionally working in a CAN network. However, an implementation needs to respect a couple of considerations. One of them is that our approach allows pulse acceleration, hence the counter must be sufficiently long in order to prohibit counter overflow during the lifetime of a car. We consider 58 bits as suggested in Section 7.2.2 adequate. Further, a concrete synchronization mechanism must for example prohibit brute force attacks on truncated MACs of synchronization messages and must ensure a synchronized network as soon as possible after the startup process.

In the future, we plan to extend our attack model and address an adversary that can manipulate devices connected to the bus. One possible countermeasure that at least reduces the severity of this attack is to assign specific roles to different ECUs and to enable ECUs to authentically identify the sender of messages. Hence, we plan to investigate how our approach can be extended by sender authentication mechanisms. Further, we will explore whether and how techniques to protect the devices' integrity can be integrated in order to prohibit manipulation of genuine ECUs.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research work has been partly funded by the German Federal Ministry of Education and Research (BMBF) and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE and by the BMBF project VITAF (ID 16KIS0835).

References

[1] Road Vehicles - Controller Area Network (CAN), *Standard*, International Organization for Standardization, Geneva, Switcherland, 2015.

[2] K. Koscher, A. Czeskis, F. Roesner et al., "Experimental security analysis of a modern automobile," in *Proceedings of the 31st IEEE Symposium on Security and Privacy*, pp. 447–462, IEEE Computer Society, Oakland, CA, USA, May 2010.

[3] H. Schweppe, "Deliverable D3.3: secure on-board protocols specifcation," Technical report, EVITA, Panaji, India, 2011.

[4] Autosar, "Specification of secure Onboard communication," 2018, https://www.autosar.org/standards/classic-platform/classic-platform-440.

[5] S. D. Gürgens and D. Zelle, "A hardware based solution for freshness of secure onboard communication in vehicles," in *Computers & Security*, Sokratis K. Katsikas et al., Ed., in *Proceedings of the Computer Security - ESORICS 2018 International Workshops, CyberICPS 2018 and SECPRE 2018*, vol. 11387, pp. 53–68, Springer, Barcelona, Spain, September 2018.

[6] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," in *Proceedings of the Black Hat USA*, Lag Vegas, NJ, USA, August 2014.

[7] D. Pohler, A. Tim, K. Chsristopher, H. Martin, L. Johannes, and P. Ulrich, "Real driving NOx emissions of European trucks and detection of manipulated emission systems," in *Proceedings of the EGU General Assembly Conference Abstracts*, vol. 19, p. 13991, Vienna, Australia, April 2017.

[8] I. D. Foster, A. Prudhomme, K. Koscher, and S. Savage, "Fast and vulnerable: a story of telematic failures," Edited by Aurelien Francillon and Thomas Ptacek. USENIX Association, Ed., in *Proceedings of the 9th USENIX Workshop on Offensive Technologies, WOOT 215*, vol. 2015, Washington, DC, USA, August, 2015.

[9] G. Bella, P. Biondi, G. Costantino, and I. Matteucci, "TOUCAN: a proTocol tO secUre Controller Area Network," in

Proceedings of the ACM Workshop on Automotive Cybersecurity, Ziming Zhao, Qi Alfred Chen, and Gail-Joon Ahn, Ed., vol. 2019, pp. 3–8, AutoSec@CODASPY, Richardson, TX, USA, March 2019.

[10] H. Ahmed and F. Ha, "Lcap-a lightweight can authentication protocol for securing in-vehicle networks"" in *Proceedings of the escar Embedded Security in Cars Conference*, vol. 10, isits AG International School of IT Security, Berlin, Germany, November 2012.

[11] S. Woo, H. J. Jo, I. S. Kim, and D. H. Lee, "A practical security architecture for in-vehicle CAN-FD," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 2248–2261, 2016.

[12] S. Nürnberger and C. Rossow, "vatiCAN - vetted, authenticated CAN bus," in *Proceedings of the Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference*, Benedikt Gierlichs and Axel Y. Poschmann, Ed., vol. 9813, pp. 106–124, Springer, Santa Barbara, CA, USA, August, 2016.

[13] J. Van Bulck, J. T. Mühlberg, and F. Piessens, "VulCAN," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, pp. 225–237, ACM, Orlando, FL, USA, December 2017.

[14] O. Hartkopp, C. Reuber, and R. Schilling, "MaCAN-message authenticated CAN," in *Proceedings of the escar Embedded Security in Cars Conference*, vol. 10, isits AG International School of IT Security, Berlin, Germany, November 2012.

[15] A. Bruni, M. Sojka, F. Nielson, and H. Riis Nielson, "Formal security analysis of the MaCAN protocol," in *Proceedings of the Integrated Formal Methods - 11th International Conference, IFM 2014*, Elvira Albert and Emil Sekerinski, Ed., vol. 8739, pp. 241–255, , Springer, Bertinoro, Italy, September, 2014.

[16] R. Kurachi, Y. Matsubara, H. Takada, H. Ueda, and S. Horihata, "CaCAN-centralized authentication system in CAN (controller area network)," in *Proceedings of the escar Embedded Security in Cars Conference*, vol. 14, isits AG International School of IT Security, Hamburg, Germany, November 2014.

[17] A. Groll and C. Ruland, "Secure and authentic communication on existing in-vehicle networks," in *Proceedings of the 2009 IEEE Intelligent Vehicles Symposium*, pp. 1093–1097, IEEE, Xi'an, China, June 2009.

[18] C.-W. Lin, L. Alberto, and S. Vincentelli, "Cyber-security for the controller area network (CAN) communication protocol," in *Proceedings of the 2012 ASE International Conference on Cyber Security*, pp. 1–7, IEEE Computer Society, Alexandria, VA, USA, December, 2012.

[19] A.-I. Radu, D. Flavio, and G. LeiA, "A lightweight Authentication protocol for CAN," in *Proceedings of the Computer Security - ESORICS 2016 - 21st European Symposium on Research in Computer Security*, Ioannis G. Askoxylakis et al., Ed., vol. 9879, pp. 283–300, Springer, Heraklion, Greece, September, 2016.

[20] Q. Wang and S. Sawhney, "VeCure: a practical security framework to protect the CAN bus of vehicles""vol. 2014, pp. 13–18, in *Proceedings of the 4th International Conference on the Internet of Things, IOT 2014*, vol. 2014, IEEE, Cambridge, MA, USA, October, 2014.

[21] T. Ziermann, S. Wildermann, and J. Teich, "CAN+: a new backward compatible Controller Area Network (CAN) protocol with up to 16x higher data rates," in *Proceedings of the Design, Automation and Test in Europe, DATE 2009*, L. Benini, Ed., vol. 2009, pp. 1088–1093, , IEEE, Nice, France, April 2009.

[22] A. Van Herrewege, S. Dave, and I. Verbauwhede, "CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus," in *Proceedings of the ECRYPT Workshop on Lightweight Cryptography*, vol. 2011, Louvain-la-Neuve, Belgium, November 2011.

[23] B. Groza, S. Murvay, A. V. Herrewege, and I. Verbauwhede, "LiBrA-CAN: a lightweight broadcast authentication protocol for controller area networks," in *Proceedings of the Cryptology and Network Security, 11th International Conference, CANS 2012*, Josef Pieprzyk, Ahmad-Reza Sadeghi, and Mark Manulis, Ed., vol. 7712, pp. 185–200, Springer, Darmstadt, Germany, December, 2012.

[24] A. S. Siddiqui, J. Plusquellic, Y. Gui, and F. Saqib, "Secure communication over CANBus," in *Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1264–1267, Boston, MA, USA, August 2017.

[25] H. Ueda, R. Kurachi, H. Takada, T. Mizuthani, M. Inoue, and S. Horihat, "Security authentication system for in-vehicle network," *SEI Technical Review*, vol. 81, pp. 5–9, 2015.

[26] B. Groza, L. Popa, and P.-S. Murvay, "Highly efficient authentication for CAN by identifier reallocation with ordered CMACs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6129–6140, 2020.

[27] Q. Zou, W. Keung Chan, K. C. Gui et al., "The study of secure CAN communication for automotive applications," in *SAE Technical Paper*SAE International, Warrendale, PA, United States, 2017.

[28] A. Muller and T. Lothspeich, "Plug-and-Secure communication for CAN," *CAN Newsletter*, vol. 4, pp. 10–14, 2015.

[29] A. Fuchs, S. Gürgens, and C. Rudolph, "Formal notions of trust and confidentiality- enabling reasoning about system security," *Journal of Information Processing*, vol. 19, pp. 274–291, 2011.

[30] L. Pino, G. Spanoudakis, A. Fuchs, and S. Gürgens, "Generating Secure Service Compositions," in *Proceedings of the International Conference on Cloud Computing and Services Science*, pp. 81–99, Springer, Lisbon, Portugal, May 2015.

[31] A. Fuchs, S. Gurgens, and C. Rudolph, "On the security validation of integrated security solutions," in *IFIP Advances in Information and Communication Technology*, Dimitris Gritzalis and Javier Lopez, Ed., in *Proceedings of the Emerging Challenges for Security, Privacy and Trust - 24th IFIP TC 11 International Information Security Conference, SEC 2009*, vol. 297, Springer, Cyprus, 2009.

[32] B. Hamid, S. Gurgens, and A. Fuchs, "Security patterns modeling and formalization for pattern-based development of secure software systems," *Innovations in Systems and Software Engineering - A NASA Journal ISSN*, vol. 12, pp. 1614–5046, 2015.

[33] S. Eilenberg, "Automata, languages, and machines," *A. Pure and Applied Mathematics*, Academic Press, Cambridge, MA, USA, 1974.

[34] S. Gürgens, P. Ochsenschläger, and C. Rudolph, "Authenticity and provability - a formal framework," in *Infrastructure Security*, George I. Davida,Yair Frankel, and Owen Rees, Ed., in *Proceedings of the Infrastructure Security, International Conference, InfraSec 2002*, vol. 2437, pp. 227–245, Springer, Bristol, UK, October 2002.

[35] R. Grimm and P. Ochsenschläger, "Binding telecooperation - a formal model for electronic commerce," *Computer Networks*, vol. 37, no. 2, pp. 171–193, 2001.

[36] W. Voss, *A Comprehensible Guide to Controller Area Network. Greenfield,Massachusetts*, Copperhill Technologies Corporation, Greenfield, MA, USA, 2008.

[37] A. Bogdanov, L. R. Knudsen, G. Leander et al., "PRESENT: an ultra-lightweight block cipher," in *Proceedings of the Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop*, Pascal Paillier and Ingrid Verbauwhede, Ed., vol. 4727, pp. 450–466, Springer, Vienna, Austria, September, 2007.

[38] J. Borghoff, A. Canteaut, T. Güneysu et al., "Prince - a low-latency block cipher for pervasive computing applications," in *Proceedings of the Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security*, Xiaoyun Wang and Kazue Sako, Ed., vol. 7658, pp. 208–225, Springer, Beijing, China, December, 2012.

[39] J.-P. Aumasson and D. J. Bernstein, "SipHash: a fast short-input prf," in *Proceedings of the In: Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India*, Steven D. Galbraith and Mridul Nandi, Ed., vol. 7668, pp. 489–508, Springer, Kolkata, India, December, 2012.

[40] C. Dobraunig, F. Mendel, and M. Schläffer, "Differential cryptanalysis of SipHash," in *Selected Areas in Cryptography - SAC 2014*, Antoine Joux and Amr Youssef, Ed., Springer International Publishing, New York, NY, USA, pp. 165–182, 2014.

[41] W. Xin, Y. Liu, B. Sun, and C. Li, "Improved cryptanalysis on SipHash," in *Cryptology and Network Security*, Y. Mu, R. H. Deng, and X. Huang, Eds., Springer International Publishing, New York, NY, USA, 2019.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9678409, 11 pages
https://doi.org/10.1155/2021/9678409

Research Article

Enabling Fairness-Aware and Privacy-Preserving for Quality Evaluation in Vehicular Crowdsensing: A Decentralized Approach

Zhihong Wang,[1] **Yongbiao Li**,[1] **Dingcheng Li**,[1] **Ming Li**,[1] **Bincheng Zhang**,[2] **Shishi Huang**,[1] **and Wen He**[1]

[1]*The College of Information Science and Technology and the College of Cyber Security, Jinan University, Guangzhou 510632, China*
[2]*The College of Information, Xiamen University, Xiamen 361005, China*

Correspondence should be addressed to Yongbiao Li; liyongbiao@jnu.edu.cn

Received 9 June 2021; Revised 22 September 2021; Accepted 16 October 2021; Published 12 November 2021

Academic Editor: Leo Y. Zhang

With the rapid development of vehicular crowdsensing, it becomes easier and more efficient for mobile devices to sense, compute, and measure various data. However, how to address the fair quality evaluation between the platform and participants while preserving the privacy of solutions is still a challenge. In the work, we present a fairness-aware and privacy-preserving scheme for worker quality evaluation by leveraging the blockchain, trusted execution environment (TEE), and machine learning technologies. Specifically, we build our framework atop the decentralized blockchain which can resist a single point of failure/compromise. The smart contracts paradigm in blockchain enforces correct and automatic program execution for task processing. In addition, machine learning and TEE are utilized to evaluate the quality of data collected by the sensors in a privacy-preserving and fair way, eliminating human subject judgement of the sensing solutions. Finally, a prototype of the proposed scheme is implemented to verify the feasibility and efficiency with a benchmark dataset.

1. Introduction

Recently, mobile crowdsensing paradigm has significantly attracted attention from both the academic and industrial area [1, 2]. It is an essentially distributed problem-solving mechanism that leverages various sensing devices to collect valuable data in order to obtain a solution. There exist numerous famous crowdsensing platforms that cover from the global positioning system (GPS) to the weather report [3]. Particularly, due to the efficient sensing capability under the 5G network, mobile crowdsensing has been adopted in the vehicular network to assist autonomous driving [4]. It can be seen that, with the rapid development of mobile crowdsensing technology, the mobile sharing data in crowdsensing will play a critical role in digital society in the near future.

Generally, the architecture of mobile crowdsensing is mainly composed of three entities: the *requester, worker,* and a *crowdsensing service provider* (CSP). Specifically, a requester refers to the entity who has the requirement to obtain large scale data (or solutions) from the workers. He posts a task with certain incentives to the CSP. A worker refers to an entity who is equipped with a mobile sensor and willing to get a reward from the requester by providing valuable sensing data. The CSP mainly acts as an intermediary to receive sensing tasks from requesters and allocate them to the suitable workers. Despite the success of crowdsensing for accomplishing complex data collection tasks with the assistance of CSP, the requesters and workers are actually exposed to the potential threats on privacy and fairness issues [1, 4, 5].

A privacy issue: as for a requester, the sensing data collected from the workers are sensitive and valuable assets that he does not want to expose them to others, even for the CSP. This is because he has to pay for a solution, and such data may leak personal private information, e.g., his current location [6]. In practice, the sensing data is collected and stored by the CSP who is responsible for their security. Furthermore, if there exists any dispute between the requester and workers, the CSP will serve as an arbiter to judge the quality by accessing the data. As a matter of fact, CSP can be regarded as a trusted third party (TTP). That is, it will not preserve the privacy of the solutions and only give them to the requester who has paid for the workers. Unfortunately, numerous causes have shown that a fully TTP does not exist in reality. For instance, the leading IT company Facebook was reported to violate the privacy protection protocols again that it exposed 533 million personal data to the public website. Apart from the policy of the law, e.g., the General Data Protection Regulation (GPDR), additional technical measures should be designed to prevent the CSP from leaking the valuable data of the participants.

An unfairness issue: as for the quality evaluation of sensing data, there exists two intrinsic attacks that have an adverse effect on the fairness between the requester and worker, i.e., *false-reporting* and *free-riding* [7]. False-reporting is caused by a malicious requester who pays for workers after he has received the solutions. He may attempt to decrease the payment for workers by reporting that their collected data is low quality, no matter the real quality. On the flip side, free-riding refers to the attack that the requester pays for workers before he receives the data. In this payment model, a malicious worker may provide useless data after he receives the payment. In considering these two attacks, it is nontrivial to realize the fairness property in mobile crowdsensing.

To achieve privacy-preserving and fair quality evaluation in vehicular crowdsensing, there are some realistic challenges: (i) Firstly, neither the requester nor the CSP knows how to evaluate the collected data using a general purpose approach. Some simple crowdsensing tasks may know the types of the answer; e.g., a task has only a Boolean answer, while most of the tasks do not have exact answers or a range to be selected. In addition, (ii) the quality of workers' sensors may be different such that they may collect invalid or obviously wrong data. Several research efforts, such [7–10], have been made to tackle the quality evaluation while preserving privacy and fairness. Some of them utilize a truth discovery method which leverages a cloud to compute the result by performing secure computation [11]. Besides, the homomorphic encryption and zero-knowledge proof are used to protect the privacy of the data while obtaining the final computation result. Nonetheless, they either rely on a central TTP, or require high efficient cryptographic primitives. If a crowdsensing task is complex during the computation of the solutions, then these methods will introduce high computation cost. Other researchers attempt to handle this by introducing the reputation mechanism [7]; however, it requires the workers and requester to keep online for a long time. Dishonest workers or requesters may receive a task for once or register another account. Thus, we argue with the following: *can we design a privacy-preserving and fairness-aware quality evaluation method for complicated tasks in mobile crowdsensing?*

To tackle the abovementioned challenges, we proposed a decentralized quality evaluation scheme, named QuaEva, based on the blockchain technology, machine learning (ML), and trusted execution environment (TEE). In particular, QuaEva is designed atop our previous work [12] which utilizes the blockchain and smart contracts to accomplish a task for crowdsensing. More precisely, facing the challenges in terms of worker quality evaluation for complicated tasks, we leverage the off-the-shelf machine learning methods to evaluate a task solution without relying on human subjective judgement. The advantage lies in that if there are a large number of crowdsensing tasks to be mediated, then it can reduce the burden of human involvement. In the meanwhile, this design can provide an efficient way to improve the accuracy of the models. Current machine learning models can distinguish many objects with high accuracy, and the accuracy will grow gradually with the evolution of machine learning technology. Recently, a business model called machine learning model market arose [13, 14], which provides paid prediction service by leveraging the trained models. Moreover, to enforce fairness among requesters and workers without relying on a trusted third party (e.g., against a single point of failure), we resort to the blockchain and smart contracts. Therefore, QuaEva can be constructed as a decentralized architecture with immutability and decentralization. Furthermore, to preserve the privacy of the solutions, we introduce the TEE into the evaluation of sensing data quality. By doing so, a sensing data or solution can be evaluated within a TEE-secured environment. In a nutshell, our specific contributions can be depicted as follows:

(1) Privacy-preserving and decentralized quality evaluation framework: we propose a privacy-preserving and decentralized quality evaluation scheme named QuaEva, in which the privacy of sensing data can be preserved and the quality evaluation can be conducted in a decentralized way based on the blockchain and smart contracts.

(2) Fair quality evaluation without subjective grounds: by leveraging machine learning and TEE to mobile crowdsensing quality evaluation, we can detect useless sensing data in an efficient and fair way, instead of resorting to a TTP who might have subjective grounds to give the final results.

(3) Implementation on real world dataset: we implement a prototype of the proposed scheme on the real world dataset and conduct several experiments, demonstrating that QuaEva can achieve secure and fair quality evaluation in mobile crowdsensing.

The remainder of the paper is organized as follows. In Section 2, we present the background of blockchain and smart contract, machine learning, and TEE. In Section 3, we present the system model, security assumptions, and threat model. Then, in Section 4, the description of our proposed framework is given. In Section 5, we give the related work with quality evaluation in crowdsensing. The experimental results are presented in Section 6. Finally, we conclude in Section 7.

2. Background

In this section, we present the basic background of the building blocks used in this paper.

2.1. Blockchain and Smart Contract. Blockchain was first introduced by Nakamoto who aims to solve the issue of double-spending in Bitcoin, and it has been utilized in many applications [15–17]. It is essentially considered as a distributed ledger (DL) which consists of consecutive blocks. Each block mainly contains a block header and several transactions which happened recently. Compared with the classic distributed database, transactions in blockchain cannot be modified or deleted once they are recorded. In particular, a secure blockchain system satisfies the basic three security properties: *chain growth, chain quality,* and *common prefix* [18]. Based on such fundamental properties, the maintainers (also called blockchain nodes) of a blockchain system have a consistent overview of the blockchain. The initial blockchain system, i.e., Bitcoin, does not support Turing-complete smart contracts and thus has significant limitations when being applied in other applications, e.g., supply chain and decentralized finance (DiFi). Therefore, several efforts have been made to enable it to be adopted in different scenarios.

In fact, the key reason that blockchain has a huge influence in various areas is the capability of supporting smart contracts. Smart contract was first proposed by Nick Szabo in the 1990s. It refers to executing a program or an agreement automatically without illegal interference. In QuaEva, by leveraging smart contracts and the decentralized blockchain, we can enforce the process of crowdsourcing to complete quality evaluation without relying on a trusted party. Further, the collected sensing data can be guaranteed with integrity in the blockchain. The main challenge of blockchain-based crowdsourcing schemes lies in the quality evaluation of sensing data with fairness. Smart contracts can be utilized to prevent a crowdsensing system to give a subjective arbitration in case of dispute.

CrowdBC, proposed by Li et.al. [12] in 2018, is a blockchain-enabled decentralized framework for crowdsourcing. Compared with previous crowdsourcing platforms, CrowdBC allows requesters and workers to achieve fair crowdsensing tasks without relying on a central party by leveraging the underlying blockchain and smart contract technologies. More precisely, a requester posts a task by the Requester-Worker Relationship Contract (RWRC) which specifies the requirements of the task. Workers can accept this task by making a deposit in the RWRC. Besides, the cryptographic primitives are utilized to protect the privacy of the solutions. CrowdBC provides a future direction for crowdsensing by combining with the trustworthy blockchain technology. However, it leaves an open problem: *how to design a solution validation function that is able to evaluate the task solution correctly while preserving the fairness in the crowdsourcing.*

2.2. Machine Learning. With the advent of the digitalization era, machine learning (ML) has been recognized as a promising paradigm for data analysis and knowledge discovery. It is a branch of artificial intelligence (AI) [19]. A large amount of data has been generated and collected nowadays, providing us plenteous resources to train the accurate ML models. In addition, a collection of machine learning algorithms, such as deep learning, reinforcement learning, and federate learning [15], have been proposed that we can use them to make a prediction more easily. As a matter of fact, a well-trained ML model can be regarded as a *Judger* who can give more accurate judgement than the crowdsensing system in various applications, e.g., vehicle identification. Correspondingly, a large amount of data is collected that we can use to train a more accurate ML model, which has favorable results for both crowdsensing and ML [20, 21]. In particular, ImageNet is an image database that has collected hundreds and thousands of labelled images. The labelling task is completed in AMT, a public crowdsourcing platform. Many companies train some ML models and provide ML services for the public, such as Microsoft Azure and Google.

2.3. Trusted Executed Environment. Trusted Execution Environment (TEE) is an isolated secure environment which is designed to protect the confidentiality and integrity of loaded code and data. It can be used to prove the correct execution of a specific program. There are several types of TEE presently, such as Intel SGX and ARM TrustZone. Take the Intel SGX as an illustration; it creates a trusted sandbox environment called enclave in which a program can run securely. Specifically, a typical TEE protocol contains three phases: initialization, install, and resume. In the initialization phase, a key pair (the public key and private key) is generated by the hardware manufacturer, who embeds a private key into the enclave that no one can obtain. The install phase mainly loads a program *pram* to the TEE and outputs a (randomness) session id for identifying the *pram*. The last phase is responsible for executing *pram* with valid inputs. After the execution, the TEE outputs an attestation to authenticate the correct execution. Recently, there have been some attacks on TEE, e.g., side channel attack and rollback attack, while we argue that existing solutions are orthogonal with our work [22, 23].

Specifically, we list the operations of an enclave that will be used in our scheme. Scheduling operations: an instance of a SGX is scheduled by a host:

(1) $i\,dx$ ←TEE.install(pram): the host creates an enclave and starts an enclave instance by providing a

software code *pram*. The enclave returns a unique identifier idx after being created successfully.

(2) TEE.resume$(i\ dx\ , inps)$: this operation is used to resume the normal execution of an enclave with an input *inps*.

3. System and Security Model

In this section, we illustrate the system model and security model of QuaEva and specify the design goals. At the beginning, we list the notations used in this paper (cf. Table 1).

3.1. System Architecture. As shown in Figure 1, there are five types of roles in QuaEva: requester, worker, blockchain node, computation node, and storage node:

(1) *Requesters*, identified by $R = \{R_1, \ldots, R_n\}$, refer to the entities who post a crowdsensing task with the description of the tasks and an amount of rewards. R specifies the requirements of the data collection task in the description and converts the requirements as executable programs that will be loaded into the computation node (i.e., the TEE-powered host).

(2) *Workers*, identified by $W = \{W_1, \ldots, W_m\}$, refer to the entities who have certain type of mobile sensing device (e.g., a vehicle) as he intends to receive a sensing task for pursuit of task rewards. The capability of W is evaluated based on the historical data when W completes tasks.

(3) *Blockchain node*, identified by B_i, refers to a maintainer of the underlying blockchain. In QuaEva, a permissionless blockchain (e.g., Ethereum) or permissioned blockchain (e.g., Hyperledger Fabric) can be used to construct the underlying blockchain. Also, R and W can take part in the maintenance of blockchain as a blockchain node.

(4) *Computation node*, identified by C, refers to an entity who is empowered with a TEE that can provide the environment for secure computation. In a permissionless setting, the centralized cryptocurrency exchange can be recognized as such node. They mainly use their secure environment to provide secure computation services by attesting the computation results to the blockchain for a reward.

(5) *Storage node*, identified by S , refers to an entity that is responsible for storing the encrypted data. We do not specify a concrete data storage; any existing distributed storage system can be used in our scheme, e.g., the InterPlanetary File System (IPFS) [24].

As depicted in Figure 1, the basic architecture of our proposed vehicular crowdsensing system mainly consists of two components: (i) task management and (ii) solution management. The task management component is responsible for task posting, task receiving, and solution submission. The solution management is in charge of solution evaluation and reward payment. Specially, requesters interact with workers based on the decentralized blockchain system. They achieve their goals with fairness based on smart contracts.

Generally, the requesters and workers are required to register in QuaEva to get their credentials (i.e., a public key and private key) before participating in the data collection of mobile crowdsensing. By doing so, every participant can be evaluated by his/her historical behaviors. Specifically, when a requester posts a task, a ML model for evaluating its collected data is deployed in a TEE primarily. The requesters and workers are assumed to place dependence on this model to evaluate the data before a task begins. The hash value of a program and function are recorded in the blockchain, which can be used for workers to check the correctness of the ML results. In QuaEva, solutions (e.g., collected data) are verified by a TEE-powered server with a ML model rather than a third party, which is to decrease the security threats from false-reporting and free-riding attacks.

Security threats: in terms of security threats, we assume both requesters and workers might behave dishonestly. More concretely, a dishonest requester may attempt to reduce his cost by denying the contributions of workers after receiving the solutions, which is known as a false-reporting attack. On the other hand, a malicious worker may try to obtain the task rewards without contributing enough time and resources, which is known as a free-riding attack. These two attacks have an impact on the fairness of crowdsensing. In addition, we assume that a computation node might behave dishonestly or be compromised. In more detail, a malicious computation node C^* can feed old version data to the TEE, enforcing the ML model to give a low quality solution (data) evaluation, which is known as rollback (replay) attack [22]. Meanwhile, a compromised computing node can collude with a requester or a worker to gain profits.

Security assumption: here, we make the security assumptions as follows: the underlying blockchain system satisfies the majority of honest assumption; that is, given a majority of blockchain nodes are honest, the blockchain system can run with *persistence* and *liveness* [18]. These two basic security properties guarantee that a transaction posted by an honest user will be confirmed and become permanent after a period of time, e.g., 6 blocks in Bitcoin. In addition, the TEE executed in the computing node is secure in our scheme. We also observe that the side channel attack is a realistic attack that could leak the private key of the enclave. However, we argue that the protection mechanisms against side channel attack have been proposed recently and are orthogonal with our work. Besides, we assume that a ML model can return accurate evaluation results.

3.2. Design Goals. We summarize the security goals of QuaEva as follows:

(1) Privacy preservation: the privacy of workers' data, i.e., their collected data or completed task solution, can be preserved without relying on any central party

Table 1: The notations of explanation.

Notation	Explanation
λ	The system security parameter
$R = \{R_1, \ldots, R_n\}$	A set of requesters
$W = \{W_1, \ldots, W_m\}$	A set of workers
$[n]$, $[m]$	The tuple of $(1, \ldots, n)$, $(1, \ldots, m)$
$\{pk_{R_1}, sk_{R_1}\}$	The public and private key of a requester R_1
$\{pk_{W_1}, sk_{W_1}\}$	The public and private key of a worker W_1
$\{mpk, msk\}$	The public and private key of an Intel SGX enclave
$M_1 \Vert M_2$	The concatenation of messages M_1 and M_2
$H_0(\cdot)$	Noncryptographic hash functions
$\sum.KGen(1^{\lambda})$	The key generation algorithm in the digital signature scheme
$\sum.\mathrm{Sign}(M, sk)$	The signing algorithm on message M using the secret key in the digital signature scheme
$\sum.\mathrm{Verify}(pk, \sigma, M)$	The verification of the signature σ using the public key and M in the digital signature scheme

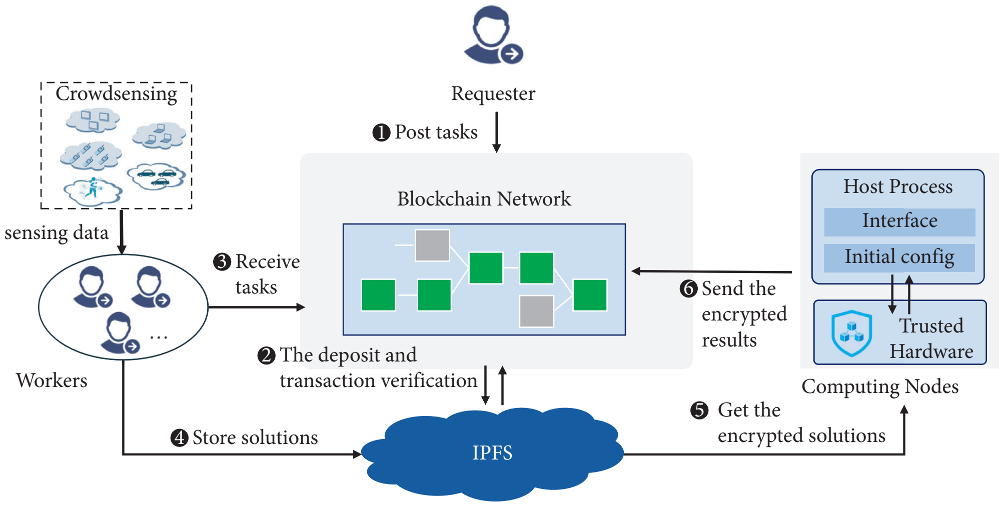


Figure 1: The system architecture of QuaEva.

(2) Fairness: our protocol can guarantee fairness by resisting the false-reporting and free-riding attacks [7]

(3) Reliable quality evaluation: the quality of the collected data during the mobile crowdsensing can be evaluated with a well-trained ML model which will give a reliable evaluation result to the workers

4. The Proposed Protocol QuaEva

4.1. The Design of QuaEva. Firstly, we present the overview design of the QuaEva protocol. To enable privacy-preserving and fairness solution evaluation, we leverage a TEE to construct an off-chain environment which can preserve the confidentiality and integrity of loaded programs. In practice, we adopt Intel's Software Guard Extensions (SGX) which has been widely used in both academic and industrial areas. The component of the computing node C contains two parts: a host process and an enclave, where the host process is mainly responsible for interacting with the blockchain and the distributed storage IPFS node, and the enclave refers to the secure environment for running the ML model.

As described in Figure 1, the proposed QuaEva protocol proceeds with four main phases: the tasking allocation phase, the task solving phase, the solution (data) evaluation phase, and the reward phase. During the first phase (steps 1–3), a requester posts a sensing task with some rewards to the RWRC contract. Meanwhile, he leverages the functionality of Intel SGX attestation to load a solution evaluation program prog to the enclave of C. Next, a set of qualified workers $\{W_1, \ldots, W_m\}$ who have registered in QuaEva (with deposits in RWRC contract) can receive this task. Specifically, these workers are required to satisfy the predefined conditions which are set in the RWRC contract; for instance, the value of reputation should be larger than a certain value. After completing the task, the workers can submit their solutions (or collected data) to the storage node S. Particularly, these data are encrypted under the public key of the enclave that can only be decrypted in the secure environment. In the meanwhile, a digest value of the solution is committed on the blockchain. Upon receiving the encrypted solutions, the enclave starts to verify them with the program prog . The evaluation result will be sent to the blockchain with a remote attestation by the computing node. The encrypted data is reencrypted in the TEE using the public key of the requester. According to the evaluation result, the reward assignment is executed automatically in the RWRC contact.

4.1.1. Smart Contracts Design. To avoid relying on a central party to conduct the quality evaluation of the solutions, QuaEva resorts to the smart contracts for ensuring the fairness and correctness of the process of crowdsensing. More concretely, the Turing-complete smart contracts enable us to depict any complex logic into contract code. Take Ethereum as an instance of the underlying blockchain system, a smart contract is converted into executable code in the EVM such that the miners can verify the correctness of the logic execution. One may think to use the smart contract to evaluate the solutions; however, due to the high on-chain transaction costs, it is unwise to put the evaluation work on-chain. Instead, we only write the core logic with smart contracts and store them in the blockchain layer, while other complex computations are put in the application layer. By doing so, we can significantly decrease the costs of on-chain transaction fees. In fact, at present, the smart contracts cannot support many complicated cryptography algorithms (e.g., Java and JavaScript); thus the off-chain solutions based on existing tools are a favorable choice in the blockchain. Different crowdsensing tasks have specific requirements; it is not easy to design on-chain quality evaluation functions for satisfying various requirements. To mitigate this challenge, we improve the smart contracts in CrowdBC. QuaEva also implements three types of smart contract: the User Register Contract (URC), User Summary Contract (USC), and Requester-Worker Relationship Contract (RWRC). Each smart contract is initialized and deployed in the blockchain for one time.

Specifically, to simplify the task posting, we devise a set of standard templates which are published by QuaEva for the tasks that have the similar logic. Each user registers with his/her addresses, profile, and pseudonym in the URC contract, where the address is corresponding to the public key of the user. In particular, in the RWRC contract, there exists a $\text{solutionEvaluate}(\cdot)$ function which is to evaluate the quality of a solution. This function can only accept inputs from the computing node C, which is achieved by verifying the signature of the transactions. Considering that C might behave dishonestly, the input is signed with the public key of the enclave. That is, the quality evaluation of a solution is realized privately in the enclave, and the result is sent to the RWRC contract with an authenticated attestation. In addition, there exists an algorithm $\text{checkWorkerQualification}(\cdot)$ that verifies the qualification of workers, e.g., checking if a worker satisfies the limited reputation value.

4.1.2. TEE-Powered Blockchain Oracle. Blockchain oracle serves as a data feed for a blockchain system. In QuaEva, the computing node C can be regarded as the blockchain oracle. More precisely, when a worker requests a reward payment after submitting the solution, he can send the request transaction to the blockchain, and the node C is notified by the "Event" mechanism in Ethereum. Then, C requests the encrypted data from the IPFS and sends it to the enclave for verification. The output of the enclave is sent back to the RWRC contract with remote attestation. Specifically, to defend against failure, we can build a distributed oracle network as in Chinklink [25], where a set of TEE-powered servers act as the blockchain oracle for feeding data.

4.2. Formal Protocol Specification. Compared with the conventional crowdsensing platform that utilizes money as a reward, QuaEva uses a cryptocurrency of the blockchain as the reward. Cryptocurrency can be obtained by mining or transacting with others. Following the assumption of [26], cryptocurrencies are fungible while they cannot be duplicated or forged. The owner who owns a private key can possess and transfer a cryptocurrency. Each party (a requester or a worker) has his own secure wallet to operate his coins. $\text{coins}(v)$ is used to denote an item that the amount is v and $\text{coin}(v)_{t_0}$ denotes the cryptocurrency v to be locked in a smart contract for t_0 times. For simplicity, we utilize a supervised learning [20] to estimate the users' data qualities without knowing a ground truth (our approach can be extended to support other machine learning algorithms), where the quality level is labelled as $Q = \{q_1, \ldots, q_k\}$ in each task $\iota \in \{1, \ldots, k\}$ and q_ι represents the best quality. In particular, inspired by CrowdBC [12], we introduce the on-chain reputation management mechanism to evaluate the behaviors of workers, where a reputation value rep is updated periodically according to the historical tasks in smart contracts. As illustrated in Figure 2, the whole process of crowdsensing consists of five phases: initialization, task posting, task receiving, solution submission, and solution evaluation. In the following, we present the protocol specification.

4.2.1. Initialization. As a first step, each party who intends to participate in a crowdsensing task is required to register in the URC contract. By doing so, each participant can be evaluated with a historical profile, e.g., the skills, experiences, and task completion degree of workers. The public keys of registered parties are published in blockchain so that anyone can check the validation of an identity. Notice that QuaEva does not require a party to register an account using a true identity, but a pseudoanonymous account, which is essentially a similar design as in Bitcoin. In particular, someone might want to register with detailed personal information to increase the possibility of task receiving; QuaEva supports this type of information to be stored in the IPFS without introducing too much on-chain cost. Besides the user registration, the computing node C also obtains a key pair $(mpk_{\text{TEE}}, msk_{\text{TEE}})$ (which is usually embedded in the TEE by the manufacturer) and publishes the public key in this phase.

4.2.2. Sensing Task Posting. In this phase, a requester R_i can post a task to call for data from qualified workers. The task defines several parameters, including $\{\text{des}, \text{coin}(\pi_R + n * v_R)_{t_d}, mpk_{\text{TEE}}, t_c, R_i^P, n, \text{rep}\}$ and $\text{solutionEvaluate}(\cdot)$, where des refers to a short description of a sensing task, pk_R refers to the public key of a requester, n is the amount of required workers, $\text{solutionEvaluate}(\cdot)$ denotes a solution evaluation function which accepts inputs from the computing node C. Specifically, each evaluation function should be loaded into the TEE using $\text{TEE.install}(\text{solutionEvaluate}(\cdot))$ by the requester previously. A proof of the remote attestation is required to check the correctness of the evaluation function. It is committed on the RWRC contract that each worker can verify. Note that the evaluation function can only accept inputs from an attested

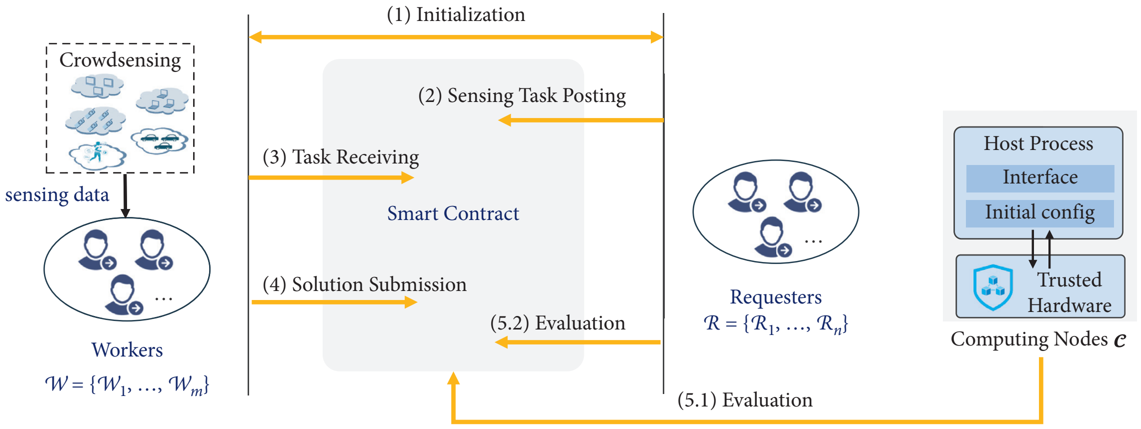


Figure 2: The process model of QuaEva.

secure processor TEE with a valid signature (using mpk_{TEE} to verify).

Specifically, the requester R_i needs to deposit a certain amount of cryptocurrency which is larger than the payment reward to achieve fairness. The remainder of the deposit will be sent back to the address of R_i if he behaves honestly. t_d refers to the time when workers are required to submit the data. t_c refers to the time when R_i is required to confirm the final evaluation results from the TEE. To prevent a low qualified worker from participating in this task, the requester can set a limited reputation rep in the RWRC contract.

4.2.3. Task Receiving. Then, a worker W_j can receive the task if he satisfies the condition of the task, e.g., a worker who drives a car and is present in the place where the requester intends to collect the data. Similar to the requester, the worker is also required to make a deposit in the RWRC contract. The deposit can be redeemed after the worker has submitted valid data in due time.

4.2.4. Solution Submission. After collecting the sensing data before t_d, the workers can submit their data to the storage node (i.e., the IPFS). In the meanwhile, a transaction with a data submission event is sent to the blockchain, and a hash value of the data is also committed in the RWRC contract. Note that an event of the RWRC contract is triggered to be sent to the C simultaneously (Ethereum solidity contract supports the Event mechanism.), which is to notify C that a sensing data has been submitted by a worker. Considering the privacy of the data, the worker encrypts the sensing data with the public key of the TEE (i.e., mpk_{TEE}) and signs it with his private key. The address of the data addr is sent to the RWRC contract so that the requester can download it.

4.2.5. Evaluation. In this phase, all submitted data are sent to the computing node C for quality evaluation. Specifically, due to the limited storage of TEE, we do not require C to evaluate these data simultaneously. More precisely, they can be split into multiple parts and evaluated separately. Inspired by proof of misbehavior in [27], as for the quality evaluation of data, it does not require all of the sensing data to be high quality, while if a part of it is evaluated as low quality, it represents that this worker provides a low quality data to this task and should not get reward from the RWRC smart contract. As mentioned before, the sensing data can be evaluated with a ML model, e.g., a capturing picture. The TEE decrypts the received data and determines the possibility of the accuracy in this task. C triggers the TEE to output an evaluation result that serves as an oracle for the blockchain. In particular, the TEE utilizes the public key of the requester to reencrypt the data after the evaluation, allowing the requester to decrypt the data by using his/her private key.

Afterwards, when the RWRC contract receives evaluation results from C, it checks if the results are signed with the TEE's public key, and it verifies the final results in solutionEvaluate$(\cdot)$ automatically. If there is no dispute between requester and worker on the solution result, the reward will be assigned to workers according to the output of the TEE. Otherwise, one of them who does not satisfy the result could post a transaction to the contract for arbitration by a third verifier. Such verifiers can only participate in the sensing task when there exists a dispute, and they can choose an authoritative party to act this third party previously.

4.3. Security Analysis

4.3.1. Fairness. In our scheme, we assume that there exists a predefined evaluation function that can automatically evaluate sensing data in the TEE. More precisely, assume that the solution is X, and then the miners can verify if solutionEvaluate$(\cdot)$ is equal to H or L according to the TEE. No one can modify the program which has been attested in the enclave. To reduce the size of on-chain data, the data can also be split into multiple parts, i.e., $X = \{X_1, X_2\}$. When the TEE outputs L result, it can upload a part of the data (e.g., X_γ) with the evaluation results to the RWRC contract. Thus, miners can verify such a part according to the output of the TEE. Notice that, during the process of task execution and result evaluation, there does not exist a trusted party to give subjective decisions. In addition, dishonest requesters or workers would be automatically punished by the RWRC contract which pays the deposit to the other party. Based on the honest-majority assumption, the underlying blockchain system is secure, and the probability that malicious parties create a fork blockchain which is in their favor is negligible. Therefore, QuaEva is able to achieve fairness by using the trusted hardware and the blockchain technology.

4.3.2. Privacy Preservation. It is straightforward that the privacy of data can be protected once a trusted hardware is adopted. More concretely, to protect the privacy of data, our protocol requires that a worker submits an encrypted task solution with the public key of the TEE. The data is reencrypted with the public key of the requester in the TEE. All of the public keys are published and attested in the task. With the security assumption of TEE, the private key cannot be retrieved by anyone, even for the manufacturer. Therefore, the data in the TEE is protected from malicious users. In addition, the encrypted data is stored in the distributed storage and can be downloaded by a unique pointer which is committed on the blockchain.

4.3.3. Nonrepudiation. It is also straightforward that nonrepudiation can be achieved with the tamper-resistant blockchain technology. Specifically, the nonrepudiation represents that R and W should be authorized to post or receive tasks in the URC contract, and they cannot refute the participation in the latter. In addition, if a worker W_m submits a low quality solution with a poor contribution, W_m cannot deny the low quality submission, because it has been committed in the RWRC contract.

5. Implementation and Evaluation

In this section, we give the evaluation of QuaEva. As for the SGX environment, to avoid the complex development of Intel SGX based on SGX SDK (Intel SGX SDK for Windows v2.13.100.2), we initialize it with SGX SDK of version 2.5. We build a TEE environment on a server (Ubuntu18.04.4LTS, Intel(R) Core(TM) i5-7500 CPU @ 3.40GHZ). Specifically, we develop the secure computation program by using *python* programming language.

To show the practicability of the proposed scheme, we implement a local Ethereum test network in our server. We evaluate the performance of our scheme by considering the whole crowdsensing process. The transaction fee is defined as the same for different transactions. Specifically, there are 1 requester and 10 workers in our experiments. We conduct 100 times for the same picture capturing task and use the Multilayer Perceptron (keras) to recognize each collected picture in the TEE. The model used in the TEE is illustrated as in Table 2, where 4-layer MP networks are specified to conduct the image recognition.

As shown in Figure 3, we analyze the performance of on-chain transactions in the test network. Specifically, we record the time consumption for each transaction involved in the execution of quality evaluation. There are 6 types of transactions involved in the proposed scheme, i.e., deposit payment, task posting, data submission, evaluation requesting, evaluation submission, and proof of evaluation by TEE. The difficulty of the local Ethereum is relatively low as that each block is generated by taking about 4.756 seconds. Each block can only store about 376 transactions. We record the time cost that starts from a transaction being sent to the network and ends at it being written on the blockchain. Note that the average confirmation time for a specific transaction is about 9.428 seconds. Namely, each transaction takes about 2 blocks of time to be finally confirmed in the local Ethereum network. The average transaction throughput can be up to 70.08 TPS (transactions per second).

Specifically, during the process of quality evaluation, we load the image recognition model (i.e., the keras model) to the TEE [20]. Each sensing data of a task is encrypted from under the public key of the TEE and can be decrypted in the enclave in a privacy-preserving way. The evaluation output of the result is sent to the contract with an authenticated attestation. Specifically, we evaluate the performance of remote attestation, enabling an output from the TEE to be authenticated by the Intel Attestation Service (IAS). Each remote attestation takes 2.73 s on average, which is a little long compared with the execution of evaluation. However, we can combine a number of outputs as a whole result and then attest it from the IAS, which can significantly decrease the time cost.

In the TEE, we test 10,000 pictures which are collected from Minst https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz. The total time of the recognition takes 12,421 ms, which takes 1.24 ms on average for recognizing a single picture in the TEE. The accuracy of the recognition can be up to 97.67%. According to our experiments, we show that, by using smart contract, TEE, and machine learning technologies, it is able to achieve fair and privacy-preserving quality evaluation for the crowdsensing task.

6. Related Work

6.1. Crowdsensing. The concept of crowdsourcing was initialized by Howe in 2005 [28]. It contains several models such as crowdsensing, crowdfunding, and microcrowdsourcing. It represents a specific model in which individuals or organizations in all over the world are connected together, in which individuals are able to contribute their skills to obtain reward from a requester. As one of promising technologies, crowdsensing has attracted much attention over the past few years. The human intelligence-based crowdsensing consists of three groups of roles: requesters, workers, and a centralized crowdsensing system. It is mainly composed of three phases: data collection, data storage, and data upload. Currently, there exist lots of crowdsensing systems and their applications grow rapidly worldwide, such as WAZE, Google Maps, and Snapchat. These applications can collect valuable information such as weather and location.

6.2. Quality Evaluation. Despite the rapid development of crowdsensing, the issue of quality evaluation of the collected data has not been settled carefully and has drawn

Table 2: The description of Multilayer Perceptron used in the experiments.

Layer (type)	Output shape	Param
Flatten (flatten)	(None, 784)	0
Dense (dense)	(None, 128)	100480
Dropout (dropout)	(None, 128)	0
Dense_1 (dense)	(None, 10)	1290

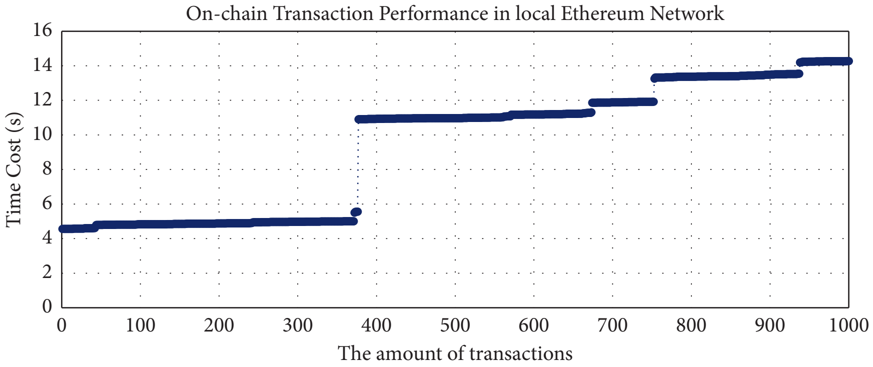


Figure 3: The time performance of a transaction to be recorded in the local Ethereum network.

much attention recently [2]. Many research works have been proposed that integrated an incentive, data aggregation, reputation management with data perturbation mechanism to achieve truthfulness and accurate aggregated results. Privacy preservation is a major research topic in crowdsensing, which is to protect the privacy of participants, e.g., location, name, and collected data. There exist three quality evaluation schemes in the crowdsensing [29, 30]. (i) The majority decision (MD): This scheme performs by aggregating all of the workers' data and the data that keep the same with the majority of participants will be the final output. (ii) The control group: This scheme generates a new validation crowdsourcing task for the collected data and selects several workers to perform the quality evaluation task. (iii) The gold standard: it proceeds by requiring workers to give a standard answer. Seldom ones of these schemes have considered the problem of worker quality evaluation and privacy preservation in a decentralized way. They mainly focus on the traditional crowdsensing architecture.

In addition, several schemes have been performed for quality control in crowdsensing, including incentive mechanism, worker selection, prior knowledge, and cheater detection. However, limitations have existed in prior schemes. Firstly, most of these methods focus on the simple tasks that can be evaluated by using the aggregation technique (AT) [31] or MD [30]. While it can address the scenario that the answers of sensing tasks have finite answers in the crowdsensing, as for the complicated skill-based tasks that do not have identical answers among the submitted result sets such as program development and diagram design, these methods cannot be applied. Secondly, requesters will generally afford some rewards for workers to get high quality data (solutions) generally. However, a main dilemma exists in the monetary incentive mechanism between the workers and the requesters. If the payment is paid before the task starts, workers may solve the task without effort, which is known as "free-riding" [7]. If the payment is paid after answers are submitted, the requester has the motivation to decrease the payment by giving an unreasonable evaluation, which is known as "false-reporting" [7]. Most current schemes are solving the dilemma based on the reputation system, but it is based on the hypothesis that workers and requesters may stay in the system for a long time, while this is not true that some dishonest users may use the crowdsensing system for one time. Lastly and importantly, the quality control approaches are executed in the crowdsensing system where workers and requesters believe that this central system will neither conduct dishonest activities nor be hit by the attackers. Unfortunately, it does not always be the case. Therefore, to accomplish the quality evaluation in a fair and privacy-preserving way, it is necessary to consider all of the security threats together.

6.3. Crowdsensing with Machine Learning. Machine learning has renovated many applications that attract considerable attention from both industrial and academic area [19, 20, 32–35]. There exist some works combining crowdsensing with machine learning. Guo et al. [34] proposed Bayesian-based predictive models that aim to accomplish the crowdsensing process within the crowdsensing architecture. Xiong et al. [35] proposed a crowdsensing method to collect large scale data to train the samples in machine learning. However, most existing methods have considered using machine learning to address the challenge of quality evaluation in the crowdsensing.

7. Conclusion

In this paper, we presented a fair-aware and privacy-preserving scheme for quality evaluation in crowdsensing. We analyze that the traditional quality evaluation functions are subjected to the weakness of human subjective intervention, single point of failure, and the complicated skill-based tasks cannot be evaluated by using conventional methods accurately. Hence, we solve the evaluation dilemma between the requester and workers with the trusted execution environment and machine learning based on our previous work [12]. Particularly, QuaEva does not rely on any third party to give judgement, and the data (i.e., the solutions) can be evaluated automatically with a committed evaluation function in the TEE. We believe that this design can provide a direction for the quality evaluation with ML and blockchain technologies.

Data Availability

The dataset analyzed during the current study are available in the Dataverse repository, https://www.kaggle.com/c/imagenet-object-localization-challenge/overview/description. These datasets were derived from the following public domain resources: https://github.com/lim60/crowdBC.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work of Zhihong Wang and Yongbiao Li was supported in part by the National Key Research and Development Plan of China under Grant 2020YFB1005600, in part by the National Natural Science Foundation of China under Grant nos. 61825203, U1736203, U2001205, 61732021, 62102166, 61902067, 62102165 and 62032025, in part by the Major Program of Guangdong Basic and Applied Research Project under Grant 2019B030302008, and in part by the National Joint Engineering Research Center for Network Security Detection and Protection Technology. The work of Dingchegn Li was supported in part by the National Natural Science Foundation of China. The work of Ming Li was supported in part by the National Natural Science Foundation of China, in part by Guangdong Basic and Applied Basic Research Foundation under Grant nos. 2020A1515111175 and 2019B1515120010, and by Guangdong Key Research and Development Plan 2020 under Grant nos. 2020B0101090002, 2020B0101090004, and 2020B0101360001.

References

[1] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 146–152, 2017.

[2] B. Zhao, S. Tang, X. Liu, and X. Zhang, "Pace: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1924–1939, 2020.

[3] X. Wang, Z. Ning, M. Zhou et al., "Privacy-preserving content dissemination for vehicular social networks: challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1314–1345, 2018.

[4] Y. Hui, Y. Huang, Z. Su et al., "BCC: blockchain-based collaborative crowdsensing in autonomous vehicular networks," *IEEE Internet of Things Journal*, vol. 1, no. 1, p. 1, 2021.

[5] D. Dang, Y. Liu, X. Zhang, and S. Huang, "A crowdsourcing worker quality evaluation algorithm on mapreduce for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 7, pp. 1879–1888, July 2016.

[6] X. Wang, Z. Liu, X. Tian, X. Gan, Y. Guan, and X. Wang, "Incentivizing crowdsensing with location-privacy preserving," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6940–6952, 2017.

[7] Y. Zhang and M. Van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proceedings of the 2012 Proceedings IEEE INFOCOM. IEEE*, pp. 2140–2148, Orlando, FL, USA, March 2012.

[8] Z. Wang, J. Hu, R. Lv et al., "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, 2018.

[9] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in iiot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2019.

[10] D. Liu, C. Huang, J. Ni, X. Lin, and X. S. Shen, "Blockchain-based smart advertising network with privacy-preserving accountability," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2118–2130, 2020.

[11] A. Yang, J. Xu, J. Weng, J. Zhou, and D. S. Wong, "Lightweight and privacy-preserving delegatable proofs of storage with data dynamics in cloud storage," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 212–225, 2021.

[12] M. Li, J. Weng, A. Yang et al., "Crowdbc: a blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 6, pp. 1251–1266, 2018.

[13] J. Weng, J. Weng, C. Cai, H. Huang, and C. Wang, "Golden grain: building a secure and decentralized model marketplace for MLaaS," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, p. 1, 2021.

[14] J. Weng, J. Weng, C. Cai, H. Huang, and C. Wang, "Fedserving: a federated prediction serving framework based on incentive mechanism," in *Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications. IEEE*, pp. 1–10, Vancouver, BC, Canada, May 2021.

[15] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2438–2455, 2019.

[16] M. Li, J. Weng, A. Yang, J.-N. Liu, and X. Lin, "Toward blockchain-based fair and anonymous ad dissemination in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11248–11259, 2019.

[17] M. Li, J. Weng, J.-N. Liu, X. Lin, and C. Obimbo, "Towards vehicular digital forensics from decentralized trust: an accountable, privacy-preserving, and secure realization," *IEEE Internet of Things Journal*, vol. 1, no. 1, p. 1, 2021.

[18] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: analysis and applications," *Advances in Cryptology - EUROCRYPT 2015*, Springer, in *Proceedings of the Annual international conference on the theory and applications of cryptographic techniques*, pp. 281–310, April 2015.

[19] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[20] X. Liu, W. Lu, W. Liu, S. Luo, Y. Liang, and M. Li, "Image deblocking detection based on a convolutional neural network," *IEEE Access*, vol. 7, pp. 26432–26439, 2019.

[21] A. Yang, J. Weng, K. Yang, C. Huang, and X. Shen, "Delegating authentication to edge: a decentralized authentication architecture for vehicular networks," *IEEE Transactions On Intelligent Transportation Systems*, vol. 1, pp. 1–15, 2020.

[22] S. Matetic, M. Ahmed, K. Kostiainen et al., "Rote: rollback protection for trusted execution," in *Proceedings of the 26th Usenix Security Symposium ({USENIX} Security 17)*, pp. 1289–1306, Vancouver, BC, Canada, April 2017.

[23] O. Oleksenko, B. Trach, R. Krahn, M. Silberstein, and C. Fetzer, "Varys: protecting SGX enclaves from practical side-channel attacks," in *Proceedings of the 2018 Usenix Annual Technical Conference ({USENIX} {ATX} 18)*, pp. 227–240, Boston, MA, USA, July 2018.

[24] J. Benet, "Ipfs-content addressed, versioned, p2p file system," 2014, https://arxiv.org/abs/1407.3561.

[25] L. Breidenbach, C. Cachin, B. Chan et al., "Chainlink 2.0: next steps in the evolution of decentralized oracle networks," 2021, https://chain.link/whitepaper.

[26] M. D. Bordo and A. T. Levin, "Central bank digital currency and the future of monetary policy," National Bureau of Economic Research, Tech. Rep., 2017.

[27] S. Dziembowski, L. Eckey, and S. Faust, "FairSwap," *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, vol. 1, pp. 967–984, Toronto Canada, October 2018.

[28] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 53, no. 10, pp. 1–4, Oct. 2006.

[29] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: challenges, solutions, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2419–2465, 2019.

[30] R. W. Ouyang, L. M. Kaplan, A. Toniolo, M. Srivastava, and T. J. Norman, "Parallel and streaming truth discovery in large-scale quantitative crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2984–2997, Oct. 2016.

[31] G. Parent, H. Meng, G. A. Levow, M. Eskenazi, and D. Suendermann, *Crowdsourcing for Speech Processing: Applications to Data Collection, Transcription and Assessment*, John Wiley & Sons, Hoboken, New Jersey, United States, 2013.

[32] R. Kohavi and F. Provost, "Machine learning," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 934–949, 2017.

[33] G. Wang, T. Wang, S. Barbara, H. Zheng, and B. Y. Zhao, "Man vs. machine: practical adversarial detection of malicious crowdsourcing workers," in *Proceedings of the 23rd USENIX Security Symposium. Usenix Security*, San Diego, CA, August 2014.

[34] B. Guo, Z. Wang, Z. Yu et al., "Mobile crowd sensing and computing," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–31, 2015.

[35] H. Xiong, Y. Huang, L. E. Barnes, and M. S. Gerber, "Sensus: a cross-platform, general-purpose system for mobile crowdsensing in human-subject studies," in *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, pp. 415–426, Heidelberg, Germany, September 2016.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1632825, 17 pages
https://doi.org/10.1155/2021/1632825

Research Article

Congestion Attack Detection in Intelligent Traffic Signal System: Combining Empirical and Analytical Methods

Yingxiao Xiang ,[1] Wenjia Niu ,[1] Endong Tong ,[1] Yike Li ,[1] Bowei Jia ,[1] Yalun Wu ,[1] Jiqiang Liu ,[1] Liang Chang ,[2] and Gang Li[3]

[1] *Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing, China*
[2] *Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China*
[3] *Australia Centre for Cyber Security Research and Innovation, Deakin University, Geelong, Australia*

Correspondence should be addressed to Wenjia Niu; niuwj@bjtu.edu.cn and Endong Tong; edtong@bjtu.edu.cn

Received 11 June 2021; Revised 31 July 2021; Accepted 11 October 2021; Published 31 October 2021

Academic Editor: Jian Weng

The intelligent traffic signal (I-SIG) system aims to perform automatic and optimal signal control based on traffic situation awareness by leveraging connected vehicle (CV) technology. However, the current signal control algorithm is highly vulnerable to CV data spoofing attacks. These vulnerabilities can be exploited to create congestion in an intersection and even trigger a cascade failure in the traffic network. To avoid this issue, timely and accurate congestion attack detection and identification are essential. This work proposes a congestion attack detection approach by combining empirical prediction and analytical verification. First, we collect a range of traffic images that correspond to specific traffic snapshots which are vulnerable to potential data spoofing attacks. Based on these traffic images, an improved generative adversarial network is trained to predict whether a forthcoming attack will cause congestion with a high probability. Meanwhile, we define a group of traffic flow features. After exploring features and conducting a thorough analysis, a TGRU (tree-regularized gated recurrent unit)-based approach is proposed to verify whether congestion occurs. When we find a possible attack that can cause congestion with high probability and subsequent traffic flows also prove congestion, we can say there is a congestion attack. Thus, we can realize timely and accurate congestion attack detection by integrating empirical prediction and analytical verification. Extensive experiments demonstrate that our approach performs well in congestion attack detection accuracy and timeliness.

1. Introduction

Connected vehicle (CV) technology [1, 2] empowers vehicles to communicate with the surrounding environment (roadside units and traffic signal control infrastructure) and is now transforming today's transportation systems. As one key component, the intelligent traffic signal (I-SIG) system [3] is responsible for performing dynamic and optimal signal control. It is based on automatic traffic situation awareness by leveraging the emerging communication infrastructure of the space-air-ground integrated network (SAGIN) [4, 5] with the advantages of coverage, flexibility, and so on. For instance, since September 2016, a series of I-SIG systems have been deployed in California, Florida, and New York by the U.S. Department of Transportation (USDOT) as a CV Pilot Program [1]. These systems are currently under testing and not yet widespread.

Unfortunately, such dramatically increased connectivity also opens a new door for cyberattacks. Recently, such I-SIG has exposed a vulnerability of the controlled optimization of phases (COP) algorithm [6, 7]. Attackers can compromise the on-board units on their vehicles and send malicious messages (such as those containing speed and location) to influence the traffic control decisions at specific times, thus causing unexpected heavy traffic congestion. Some data show that a single attack vehicle can cause a total delay 11 times greater than the total delay before the attack [8], posing a significant barrier to the development and deployment of I-SIG systems on a wide scale in the future.

Previous research [8] reveals such congestion attacks on the COP algorithm, analyzes how congestion attacks affect the COP algorithm decisions, and explains how to launch an attack using data spoofing in SAGIN. However, developers may still lack a deep understanding of such I-SIG attacks and defenses, raising some pressing concerns: (1) What is the effect of different phases where the attack vehicle is located? The different phases of the attack vehicle can cause different congestion effects. (2) What is the quantified correlation between the attack and congestion degree? The quantified correlation refers to the potential relationship between the attack and congestion degree; once identified, we can infer whether the attack occurred according to the congestion degree. (3) Are there any potential features to be utilized for revealing the above correlation? It is necessary to analyze the congestion attack mechanism firstly to solve these issues. The challenges of solving these issues include how to automatically explore multiple and multidimensional features to quantify the traffic flow characteristics under no attack and congestion attack and analyze the correlation between attack features and attack effects. Thus, demystifying the congestion attack based on the COP mechanism through quantified features and exploring new analysis methods will benefit all stakeholders for I-SIG, including transportation, SAGIN, and security specialists.

We demystify the attack and corresponding congestion from a machine learning perspective by exploring and utilizing quantified features. We deeply analyze data spoofing in SAGIN and the COP algorithm vulnerability under two different attack strategies. To explore the effect of different phases of the attack vehicle, we consider utilizing high-level image features and design a novel analysis model based on the cycle generative adversarial network (CycleGAN) [9] to reflect the relation between the attack and the congestion caused by the attack. Thus, we can predict whether a forthcoming attack will cause congestion and the congestion effect according to the traffic image at a specific moment. To explore the quantified correlation between the attack and congestion degree, we utilize traffic flow features and the TGRU classification model [10] (an explainable gated recurrent unit-based model [11] with tree regularization) to verify whether a congestion attack occurs based on all vehicles' trajectory data in an intersection. Following analysis, we also give some promising suggestions for defending I-SIG systems against a congestion attack.

We implement the I-SIG and experiment through visualized simulation in VISSIM [12]. The experiment shows the effectiveness of our approach. We find that feature-based machine learning can reflect the correlation between the attack and congestion degree well. Through the deep learning-based training, the CycleGAN-based approach output visualized results with satisfied prediction compared with real values: the MAE and RMSE of the congestion degree are near 0.02 and 0.03, respectively, and the MAE and RMSE of the congestion degree are near 0.94 and 1.14, respectively. TGRU has a 0.84 precision and 0.79 recall on predicting the spoofing attack based on 30 features. Generally, for defenses, we suggest improving the estimation of vehicle location and speed (EVLS) [7] algorithm of I-SIG if we would like to keep a limited cost, which requires fewer authentication mechanisms and SAGIN reinforcement efforts.

We summarize our contributions as follows:

(1) We perform the study to demystify the attack to I-SIG and the corresponding congestion from a machine learning perspective by exploring different kinds of features through supervised learning and unsupervised learning.

(2) For predicting the spoofing congestion attack, we automatically explore the image feature to quantify the traffic flow characteristics under no attack and congestion attack. And we propose a CycleGAN-based approach to analyze the potential relationship between the congestion attack and corresponding results two stages later based on the image feature.

(3) For verifying the spoofing congestion attack, we propose a TGRU-based approach to explore the underlying relationship between the congestion attack and traffic flow feature at the current moment based on the traffic flow features, which are firstly defined in this work.

(4) We evaluate our approach empirically from the real COP algorithm through VISSIM. We collect 4476 high-quality image samples and 3600 traffic flow data for the experiment, which enables us to demonstrate the effectiveness of our approach compared with ground truth.

2. Preliminaries

2.1. SAGIN Infrastructure of I-SIG. Figure 1 presents the basic architecture for the space-air-ground integrated network of I-SIG, in which two main segments are included: a space segment and a ground segment. The I-SIG of the CV environment is located in the network-based ground segment. There are three main components within the ground segment: on-board units (OBUs), roadside units (RSUs), and signal planning units. These refer to the devices installed in vehicles, roadside servers, and traffic lights, respectively. Both vehicle-to-vehicle (V2V) [13] communication and vehicle-to-infrastructure (V2I, e.g., roadside servers) [14] communication adopt the dedicated short-range communications (DSRC) [15] transmission protocol as 802.11p-based wireless communication; this provides a channel and enables high-speed direct communication. Every vehicle broadcasts anonymously, and surrounding vehicles receive messages. Messages containing critical information are called basic safety messages (BSMs). These contain core data elements, including vehicle size, position, speed, heading, acceleration, and brake system status. Compared with DSRC, the communication from the RSU to the signal planning unit adopts the US National Transportation Communications for Intelligent Transportation System Protocol (NTCIP) [16]. By providing two-way communication between vehicles and traffic signals, NTCIP is specially designed to achieve interpretability and interchangeability between computers and electronic traffic control equipment from different manufacturers, thus increasing use in smart city initiatives.

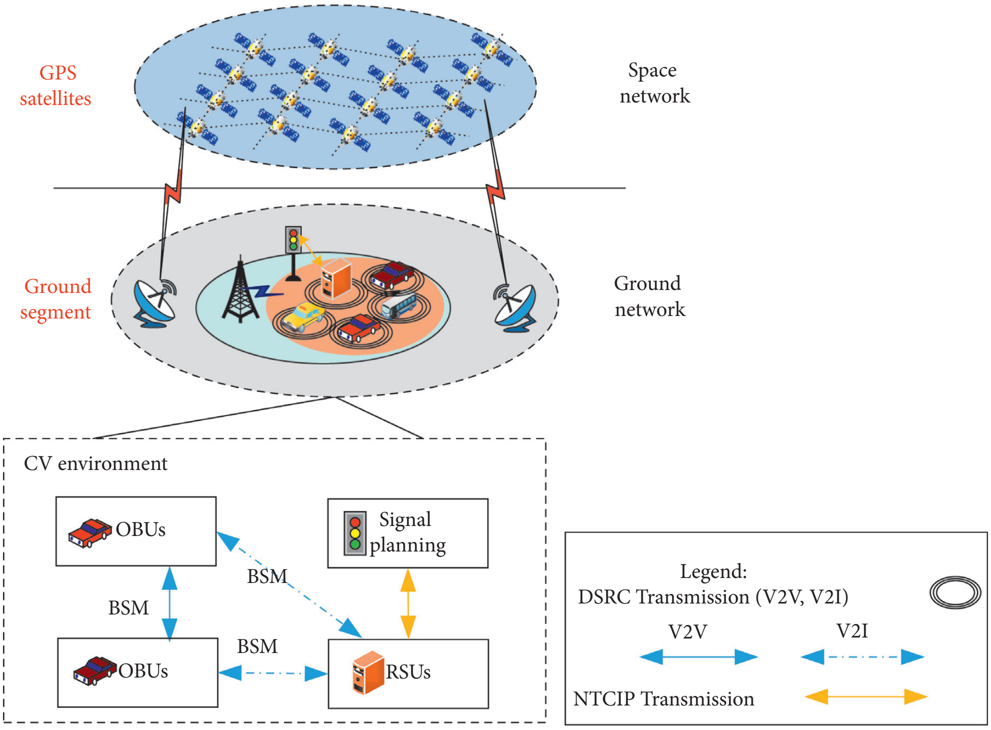


Figure 1: The architecture for space-air-ground integrated network of I-SIG.

2.2. I-SIG Data Flow. The data flow of the I-SIG system is revealed in Figure 2. Each OBU of a vehicle sends BSMs to the RSU for real-time trajectory collection. Then, the data are preprocessed to form an arrival table (Table 1) to be used as input for signal planning, which contains COP and EVLS algorithms. If the penetration rate (PR) of OBU for a vehicle is less than 95%, the arrival table will be sent to EVLS for an update. Otherwise, it will be directly sent to the COP algorithm for planning. According to the results of the COP algorithm, a downward signaling command will be transferred to the phase signal controller. After each stage of signal control, the status of the signal will be returned as feedback for continuous COP planning.

There are 8 traffic signals in I-SIG, as shown in Figure 3, called phases; odd numbers are for left-turn lanes; even numbers are for through lanes. Table 1 is the arrival table which is sent to the signal planning model. In Table 1, $T_i = i\ (0 \leq i \leq M)$ denotes the time to arrive at the stop bar from the current location. I-SIG sets $M = 130$ seconds, covering a BSM statistic of over two minutes. $N_{ij}\ (i \in [0, M],\ j \in [1, 8])$ means that in phase j, there will be N_{ij} vehicles that are going to reach the stop bar within T_i seconds. Here, the stop bar is set in front of the traffic light as it is marked in real road intersections.

The EVLS is based on Wiedemann's car-following model and is used to fill the blank monitoring area of the monitoring segment and insert vehicle data between OBU-equipped vehicles.

The key is to estimate the number of queued vehicles. Because it is assumed that a queue always begins at the stop bar, the last vehicle in the queue needs to be found to determine the queue length.

First, the historical distances to the stop bar and stop time of the last stopped connected vehicle and the second-to-the-last stopped connected vehicle in the queue are calculated; these are denoted as L_{q1}, T_{q1}, L_{q2}, and T_{q2}, respectively. The current time is T_c, and the estimated queue length is L_{es}. Assuming that the queue propagation speed v_q is constant, we have

$$v_q = \frac{L_{q1} - L_{q2}}{T_{q1} - T_{q2}} = \frac{L_{es} - L_{q1}}{T_C - T_{q1}}. \quad (1)$$

Then,

$$L_{es} = L_{q1} + v_q \left(T_C - T_{q1}\right). \quad (2)$$

If the average vehicle length is C, the number N_{0i} of vehicles in queue is then calculated as follows:

$$N_{0i} = \frac{L_{es}}{C}, \quad i \in [1, 8]. \quad (3)$$

Although such estimation provides effective support for a low PR, it also introduces a new threat of data spoofing attack to the COP algorithm.

3. Demystifying Attack on COP

3.1. Data Spoofing Threat. There are two data spoofing attack strategies proposed in I-SIG (Figure 4). The first one is a direct attack on the arrival table without considering PR; the second one is an indirect attack on EVLS when the PR is less than 95%.

The first strategy is for arrival time and phase spoofing, for both the full deployment period (PR ≥ 95%) and

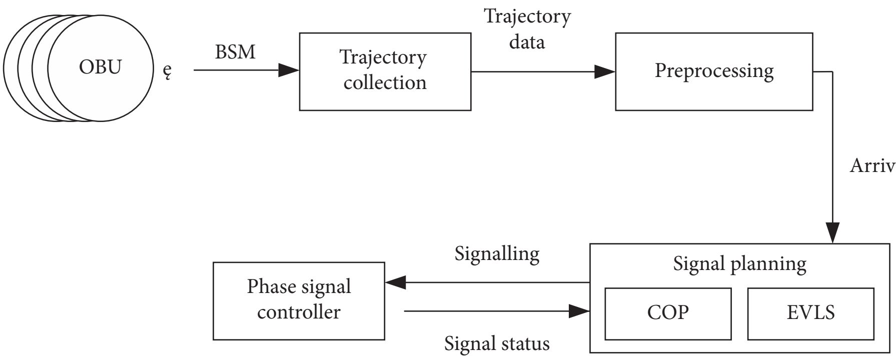


Figure 2: Data flow of the I-SIG system.

Table 1: Arrival table. Numbers 1 to 8 are phases, and T_0 to T_M are the remaining arrival time of vehicles.

Phase	1	2	...	8
T_0	N_{01}	N_{02}	...	N_{08}
T_1	N_{11}	N_{12}	...	N_{18}
T_2	N_{21}	N_{22}	...	N_{28}
...	...	...	...	...
T_M	N_{M1}	N_{M2}	...	N_{M8}

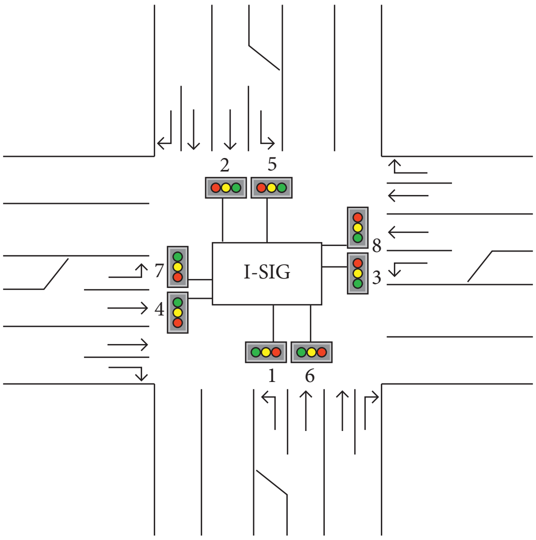


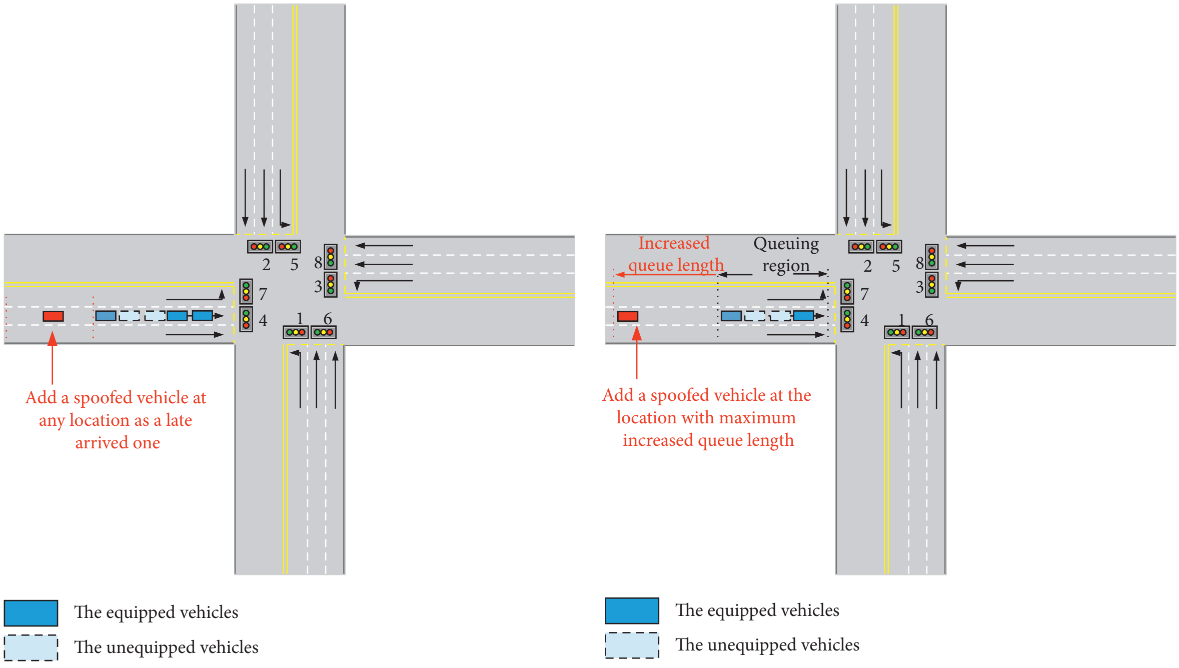
Figure 3: I-SIG signal control scenario, including 8 phases.

transition period (PR < 95%). The attacker changes the location and speed information in vehicle BSMs to alter the vehicle's arrival time and requested phase; thus, the corresponding arrival table elements in Table 1 are changed. This attack strategy can directly attack input data flow no matter what the PR is. As shown in Figure 4(a), the attacker adds a spoofed vehicle into the original vehicle queue at any location. The insertion of a spoofed vehicle makes the queue longer. Moreover, there is an increase in the duration of the green light allocated by the COP algorithm for the current phase, which delays the next start time of the green light of all phases, thus increasing the delay for vehicles to pass through the intersection.

The second strategy is for queue-length spoofing, for the transition period only. This strategy aims to extend the queue length estimated by the EVLS algorithm by changing the location and speed values in BSMs. Figure 4(b) shows that the attacker adds a stopped vehicle with the farthest distance to the stop bar. Owing to the EVLS algorithm estimating the queue length based on the location of the last stopped connected vehicle, this attack causes the estimated queue length L_{es} calculated by equation (2) to increase. Therefore, the number of vehicles in the queue N_{0i} calculated by equation (3) increases as well.

3.2. Planning-Level Congestion Analysis.

The COP algorithm is responsible for traffic signal planning; thus, it is essential for planning-level congestion analysis of I-SIG.


2
5
8
3
7
4
1
6
Add a spoofed vehicle at any location as a late arrived one
The equipped vehicles
The unequipped vehicles
Increased queue length
Queuing region
Add a spoofed vehicle at the location with maximum increased queue length

(a)

(b)

Figure 4: Two strategies of congestion data spoofing attack. PR is short for penetration rate. (a) Direct attack on arrival table without considering PR. (b) Indirect attack on EVLS when PR is less than 95%.

Through reading the published COP-related papers [6, 7] and analyzing the implementation code, we reveal a more complete and detailed COP algorithm for the first time (Algorithms 1 and 2). The authors in [6] first proposed a COP algorithm that allows optimization of various performance indices, including delay, stops, and queue lengths, for the optimal control of a single intersection. However, it did not support flexible or dual ring and phase sequences, and it is difficult to understand for most readers due to the lack of the algorithm flow. Based on the COP algorithm, the authors in [7] presented a real-time adaptive traffic control algorithm by utilizing data from connected vehicles to optimize the phase sequence. However, they did not provide the details of the algorithm. Compared with [6, 7], Algorithm 1 is the first algorithm that provides a complete and detailed flow of signal planning.

In Table 2, we list the meanings of the mathematical symbols that appear in the two algorithms.

The design of the COP algorithm uses the collaboration of two-stage planning and operation. The COP algorithm plans signals for the next-stage based on the vehicle's estimation, and such planned signal duration will be operated at the next-stage signal control time. Thus, this is a continuous alternate process in a fixed phase sequence, which means that the I-SIG system cannot change the order and duration of phases in the current stage since this is set in the previous stage. When bringing foresight of planning, such a design also opens the door to attack signal planning in order to affect next-stage operation continuously.

The spoofing of the arrival table affects the variables $A_{t,k}$ and plan$P_{r,p}$ and the later calculation of Delay$_r$ in line 19 of Algorithms 1. The change in Delay$_r$ causes the variables optV_r in line 21, opt$G_{r,0}$ in line 22, and opt$G_{r,1}$ in line 23 of Algorithms 1 to change as well. Finally, the outputs plan$P_{r,p}$, opt$G_{r,p}$, x_j^*, and v_j are changed.

3.3. *High-Level Image Feature-Based Congestion Attack Prediction.*

In this subsection, we employ an image feature-based CycleGAN to explain the relationship between the phase where the spoofed vehicle is located and the congestion image features two stages later.

As mentioned in the Data Spoofing Threat section, there are two data spoofing attack strategies, but either attack will cause congestion in a period. Different phases of spoofed vehicles lead to different congestion effects. Therefore, the image features of intersection congestion are also different. The CycleGAN model can mine the potential relationship between two different types (X and Y) of images. Through training, CycleGAN can generate the corresponding images Y according to X and generate the related images X according to Y. Therefore, we utilize the CycleGAN model to predict the congestion effects according to the phases of spoofed vehicles, which were considered the attack feature, in order to reveal the relationship between the phase of the spoofed vehicle and the caused congestion image feature.

The CycleGAN architecture is illustrated in Figure 5. One training sample is a pair of image x_i and image y_i to form (x_i, y_i), $\quad x_i \in X$, and $y_i \in Y$. x_i refers to the processed traffic image at the spoofing time, and y_i is the processed traffic congestion image two stages later. The image processing consists of three steps: (1) filter out environment background; (2) extract four images, in which each has 2 phases at one intersection; and (3) join these four images

```
     //The plan of the optimal green duration and phase sequence
     Require: $A_{t,k}$, $G_{\min}^{r,p}$, $G_{\max}^{r,p}$, $R$, T
(1)  Set j = 0, $v_j = 0$
(2)  $X_j^{\min} = \max\{G_{\min}^{0,0} + R + G_{\min}^{0,1} + R,\ G_{\min}^{1,0} + R + G_{\min}^{1,1} + R\}$
(3)  $X_j^{\max} = \min\{G_{\max}^{0,0} + R + G_{\max}^{0,1} + R,\ G_{\max}^{1,0} + R + G_{\max}^{1,1} + R\}$
(4)  $T' = T - s_{j-1} - \cdots - s_1$
(5)  for $r = 0, 1$ do
(6)    plan $P_{r,0} = 1 + j * 2 + r * 4$
(7)    plan $P_{r,1} = 2 + j * 2 + r * 4$
(8)  end for
(9)  for $s_j = 1, \ldots, T$ do
(10)   if $s_j \geq X_j^{\min}$ and $s_j \leq \min\{X_j^{\max}, T'\}$ then
(11)     $x_j = s_j$
(12)     effect $G_0 =$ effect $G_1 = s_j - 2R$
(13)     opt$V_0$ = opt$V_1$ = 99999.0
(14)       for $r = 0, 1$ do
(15)         for $i = G_{\min}^{r,0}, \ldots, G_{\max}^{r,0}$ do
(16)           $tG_{r,0} = i$
(17)           $tG_{r,1} = \text{effect}G_r - tG_{r,0}$
(18)           if $tG_{r,1} \geq G_{\min}^{r,1}$ and $tG_{r,1} \leq G_{\max}^{r,1}$ then
(19)           $\text{Delay}_r = f(r, \text{plan}P_{r,0}, \text{plan}P_{r,1}, tG_{r,0}, tG_{r,1}, x_j, A_{t,k})$
                   //Calculated by Algorithm 2
(20)           if $\text{Delay}_r < \text{opt}V_r$ then
(21)             $\text{opt}V_r = \text{Delay}_r$
(22)             $\text{opt}G_{r,0} = tG_{r,0}$
(23)             $\text{opt}G_{r,1} = tG_{r,1}$
(24)         end if
(25)       end if
(26)     end for
(27) :   end for
(28) else
(29)     $v_j = 99999$
(30)     $x_j = 0$
(31)   end if
(32) end for
(33) $x_j^* = \text{opt}G_{r,0} + R + \text{opt}G_{r,1} + R$
(34) $f_j(x_j^*) = \text{opt}V_0 + \text{opt}V_1$
(35) $v_j = f_j(x_j^*) + v_{j-1}$
     Ensure: $\text{plan}P_{r,p}, \text{opt}G_{r,p}, x_j^*, v_j$
(36) if $j < 2$ then
(37)   $j = j + 1$, go to step 2.
(38) end if
```

Algorithm 1: The COP algorithm

```
//The delay calculation of ring $r$ at stage j
// $f(r, \text{plan}P_{r,0}, \text{plan}P_{r,1}, tG_{r,0}, tG_{r,1}, x_j, A_{t,k})$
Require: $r$, $p1$, $p2$, $g1$, $g2$, $x_j$, $A_{t,k}$
$l_{0,p1} = A_{0,p1}$, $l_{0,p2} = A_{0,p2}$
for $i = 1, \ldots, x_j$ do
  $l_{i,p1} = l_{i-1,p1} - D_{i,p1} + A_{i,p1}$
  $l_{i,p2} = l_{i-1,p2} - D_{i,p2} + A_{i,p2}$
  $d_i = l_{i,p1} + l_{i,p2}$
end for
s.t.
$D_{i,p1} = \begin{cases} 1, & \text{if } i \leq g1 \text{ and } (i+1)\%2 = 0, \\ 0, & \text{if } g1 < i \leq x_j, \end{cases}$
$D_{i,p2} = \begin{cases} 1, & \text{if } (g1 + R) < i \leq (g1 + R + g2) \text{ and } (i+1)\%2 = 0, \\ 0, & \text{i } (g1 + R + g2) < i \leq x_j \text{ and } i \leq (g1 + R), \end{cases}$
$\text{Delay}_r = \sum_{i=1}^{x_j} d_i$
Ensure: $\text{Delay}_r$
```

Algorithm 2: The delay calculation algorithm.

Table 2: Mathematical symbols used in the COP algorithm.

t	Index of arrival time	k	Global phase index
$A_{t,k}$	Element of arrival table denoting the number of vehicle arrivals for phase k at time t	p	Local phase index
r	Ring index in each stage	$G_{\min}^{r,p}$	Minimum green time of phase p in ring r
$G_{\max}^{r,p}$	Maximum green time of phase p in ring r	R	Duration of yellow light and red light
T	Total number of discrete time steps in the planning horizon, in seconds	j	Index of stage
v_j	Value function given state j which represents the accumulated performance measure for the current and all previous stages	$X_j^{\min}$	Minimum possible length of stage j
$X_j^{\max}$	Maximum possible length of stage j	s_j	State variable denoting the total number of time steps allocated to stage j
Plan $P_{r,p}$	Planned phase of phase p in ring r	EffectG_r	Effective total green light time of ring r in stage j
Opt V_r	Optimal delay of ring r in stage j	Opt $G_{r,p}$	Optimal green duration of phase p in ring r
x_j	Length of stage j under the optimal solution	x_j^*	Length of stage j under the optimal solution
$f_j(x_j)$	Performance measure at stage j	$l_{i,k}$	Number of vehicle departing for phase k at time t
$D_{i,k}$	Number of vehicle departing for phase k at time t	Delay$_r$	Delay of ring r at stage j

Table 3: Feature composition schema through selecting equal features from traffic flow head and tail.

	Flow head (10 s)	Flow tail (10 s)
Macrofeatures	CR, α_{CR}, β_{CR}, ICD, α_{ICD}, $\beta_{IC\ D}$	CR, α_{CR}, β_{CR}, ICD, α_{ICD}, β_{ICD}
Microfeatures	PCD$_1$, PCD$_2$, . . ., PCD$_8$, α_{PCD_1}, α_{PCD_2}, . . ., α_{PCD_8}, β_{PCD_1}, β_{PCD_2}, . . ., β_{PCD_8}	PCD$_1$, PCD$_2$, . . ., PCD$_8$, α_{PCD_1}, α_{PCD_2}, . . ., α_{PCD_8}, β_{PCD_1}, β_{PCD_2}, . . ., β_{PCD_8}

from top to bottom to form one sample image according to the phase order of phase (4,7), phase (8,3), phase (2,5), and phase (6,1). Here, the number of phases is consistent with that shown in Figure 3, which is joined by the four parts of an intersection from top to down to form one sample image.

There are four neural networks in the CycleGAN architecture: two generative networks (G and F) and two discriminant networks (D_X and D_Y). The generator G generates a fake image $\tilde{y}$, which is similar to y given real image x, i.e., G: $X \longrightarrow Y$. Meanwhile, F generates a fake image $\tilde{x}$, which is similar to x given real image y, i.e., F: $Y \longrightarrow X$. The adversarial discriminator D_X aims to distinguish whether the input image is x and outputs probability $P(x)$. Similarly, D_Y aims to discriminate whether the input image is y and outputs probability $P(y)$.

For $x \in X$, $x \longrightarrow G(x) \longrightarrow F(G(x)) \approx x$ is a cycle, called forward cycle consistency. Similarly, for $y \in Y$, $y \longrightarrow F(y) \longrightarrow G(F(y)) \approx y$ is a cycle called backward cycle consistency. There are two kinds of losses: adversarial loss and cycle consistency loss. Adversarial loss can only guarantee that the samples generated by the generator are distributed with the real samples, but we want the images between the corresponding domains to correspond one by one. That is, X-Y-X can also be migrated back. So, forward cycle consistency and backward cycle consistency are used to make the samples generated by two generators not contradict each other.

Adversarial Loss. This refers to the difference in dataset distribution between generated images and corresponding real images. For discriminators D_{X} and D_{Y}, the closer the output value is to 1, the smaller the loss is.

The losses of G and F can be calculated as L_{G} and L_{F}, respectively, as follows:

$$L_G(G, D_Y, X, Y) = E_{y\sim \text{Pdata}(y)}[logD_Y(y)] + E_{x\sim \text{Pdata}(x)}[\log(1 - D_Y(G(x)))], \tag{4}$$

$$L_F(F, D_X, Y, X) = E_{x\sim \text{Pdata}(x)}[logD_X(x)] + E_{y\sim \text{Pdata}(y)}[\log(1 - D_Y(G(y)))]. \tag{5}$$

Cycle Consistency Loss. This prevents the learned mappings G and F from contradicting each other, making $F(G(x)) \approx x$ and $(F(y)) \approx y$. The loss of $L_{\text{cyc}}(G, F)$ is calculated by the following equation:

$$\begin{aligned} L_{\text{cyc}}(G, F) = {} & E_{x\sim \text{Pdata}(x)}[F(G(x)) - x_1] \\ & + E_{y\sim \text{Pdata}(y)}[G(F(y)) - y_1]. \end{aligned} \tag{6}$$

The total loss for CycleGAN is

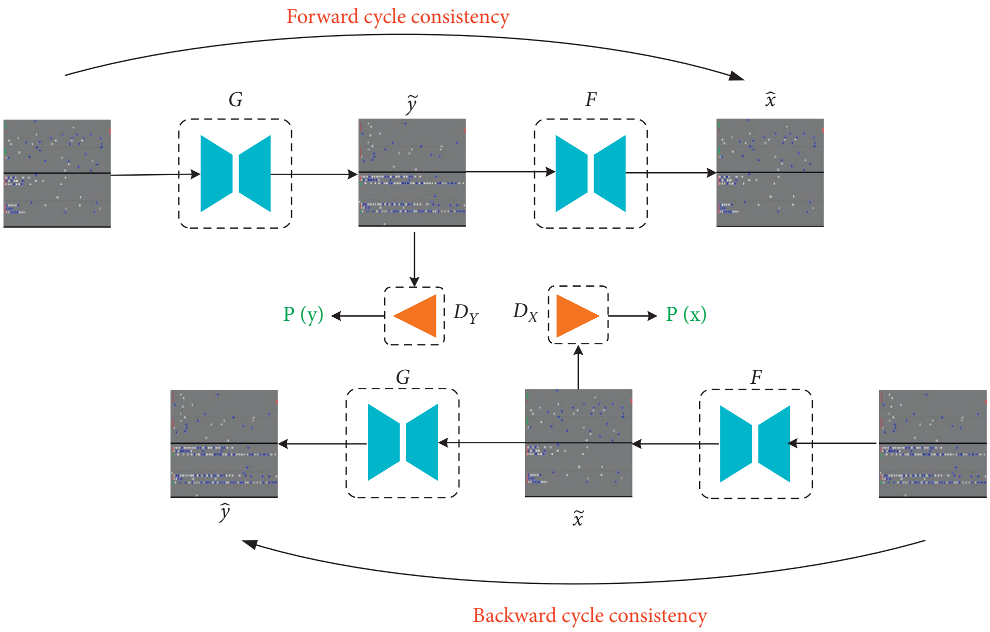


Figure 5: CycleGAN architecture.

$$L(G,F,D_X,D_Y) = L_G(G,D_Y,X,Y) + L_F(F,D_X,Y,X) + \lambda L_{\text{cyc}}(G,F), \tag{7}$$

in which λ is an important parameter. Then, the objective function of the CycleGAN is defined as follows:

$$G^*, F^* = \arg\min_{G,F}\max_{D_X,D_Y} L(G,F,D_X,D_Y). \tag{8}$$

The detailed implementation of neural networks in CycleGAN will be described in the following experiment setup.

3.4. Traffic Flow Feature-Based Congestion Attack Verification. In this subsection, we use a deep learning-based decision tree model, TGRU, to explain the relationship between the traffic flow features and the congestion attack. This relationship can then be used to verify if congestion is occurring. The TGRU model is an interpretable depth time-series model, which is very suitable for intersection traffic flow features with time characteristics. At the same time, interpretability helps to analyze better the relationship between traffic flow features and the congestion attack.

The input of the congestion prediction is the traffic image feature of the intersection, and the prediction model outputs the congestion affects two stages later according to the image feature, which indicates that whether the congestion will occur. However, after the congestion prediction, the verification model is used to verify whether the congestion attack is occurring. The verification input is the defined traffic flow feature that is calculated according to vehicles' information. When we find a possible attack that can cause congestion with high probability and subsequent traffic flows also verify congestion, then we can predict there exists a congestion attack. Thus, we can realize timely and accurate congestion attack detection by integrating empirical prediction and analytical verification.

Feature Definition Based on Traffic Flow. To measure the congestion effects caused by spoofed vehicles, we propose capacity ratio and congestion degree, as well as an attack acceleration and attack amplification ratio based on capacity ratio and congestion degree. We define features as follows:

(1) *Vehicle Capacity Ratio* (CR). $C_k^{\max}$ is the maximum vehicle capacity of each phase, and the vehicle capacity of all 8 phases is computed as $C_{\text{total}}^{\max} = \sum_{k=1}^{8} C_k^{\max}$. Then, the vehicle CR can be denoted by $CR = \sum_{k=1}^{8} N_k / C_{\text{total}}^{\max}$, where N_k is the vehicle number of the kth phase.

(2) *Congestion Degree* (CD). The number of vehicles queuing in the kth phase is denoted as Q_k. Q_{normal} is the number of vehicles during normal queuing and is a constant. Then, the CD of the kth phase can be computed by $PCD_k = \text{Q}_k/\text{Q}_{\text{normal}}$, and the global CD for an intersection is $ICD = \sum_{k=1}^{8} PCD_k$.

(3) *Attack Acceleration.* Let t_0 be the start time of the data spoofing attack. Then, the accelerations of CR, PCD_k, and ICD at time t are, respectively, calculated by $\alpha_{CR}(t) = (CR(t) - CR(t_0))/(t - t_0)$, $\alpha_{\text{PCD}}(t,k) = (PCD(t,k) - PCD(t_0,k))/(t - t_0)$, and $\alpha_{\text{ICD}}(t) = (ICD(t) - ICD(t_0))/(t - t_0)$.

(4) *Attack Amplification Ratio.* Let t_0 be the start time of the data spoofing attack. Then, the amplification ratio of CR, PCD_k, and ICD at time t is, respectively, calculated by $\beta_{CR}(t) = CR(t)/CR(t_0)$, $\beta_{\text{PCD}}(t,k) = PCD(t,k)/PCD(t_0,k)$, and $\beta_{ICD}(t) = ICD(t)/ICD(t_0)$.

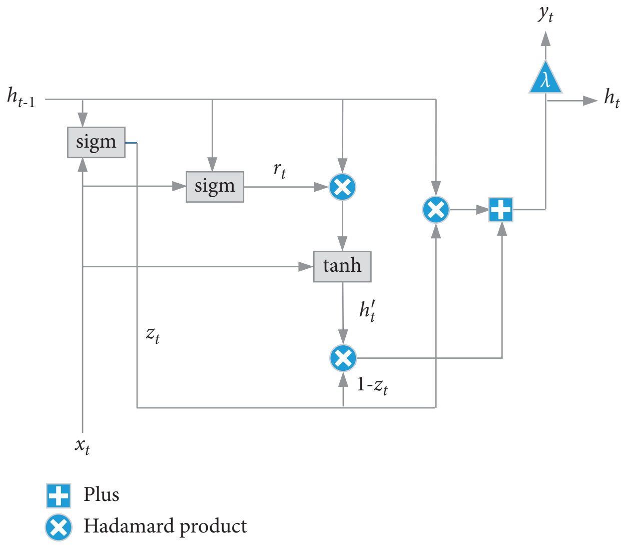


Figure 6: TGRU architecture.

Table 4: Experimental environment configuration.

Platform	Experimental environment	Environmental configuration
COP and VISSIM	Operating system	Windows 10
	CPU	AMD Ryzen5 3550H with Radeon Vega Mobile Gfx 2.10 GHz
	RAM	16G
	Software	PTV VISSIM 4.30, Visual Studio 2019
TGRU and CycleGAN	Operating system	Ubuntu 16.04.6 LTS
	CPU	Intel(R) Core(TM) i7-9700F CPU @ 3.00 GHz
	RAM	32G
	GPU	MSI GeForce RTX 2070 VENTUS
	Graphic memory	151MiB
	Framework	TensorFlow-gpu-1.14.0

Features are divided into macrofeatures and microfeatures for the sake of discussing interpretability, depending on whether they are a feature of the whole intersection or a specific phase (Table 3). Macrofeatures measure the congestion characteristics of the whole intersection, and microfeatures measure the phase of a single signal phase. Unlike the traditional traffic flow characteristics, such as traffic flow, traffic density, and speed, the traffic flow features we defined are related to attacks and are divided into the features for all single signal phases and the features of the whole intersection. For a traffic flow of 1800 seconds, we only sample the first 10 seconds of flow head and the last 10 seconds of flow tail. For flow head, we choose features of macro, micro, or both, and then we choose the same features from the flow tail. Therefore, the number of features is from 20 to 600. We use the Z-score as a standardization to adjust feature values. For values $(x_1, x_2, \ldots, x_n)$ of one feature in all samples, the new value is computed by $x' = x_i - \overline{x}/s$, in which s is the standard deviation and $\overline{x}$ is the mean value of $(x_1, x_2, \ldots, x_n)$.

TGRU Model. We try data spoofing exhaustedly using the last vehicle and collect time-sequence samples. For such data, we use TGRU, a time-series model with decision tree regularization, for interpretability. Figure 6 shows the TGRU architecture for end-to-end calculation.

There are four main calculators: sigm, tanh, plus, and Hadamard product. Sigm refers to the sigmoid function, and tanh refers to the hyperbolic tangent function. The objective function is as follows:

$$\min_{W}\left(\lambda\psi(W) + \sum_{n=1}^{N}\sum_{t=1}^{T}\text{loss}\left(y_{nt}, \tilde{y}_{nt}\left(x_n, W\right)\right)\right), \tag{9}$$

where λ $(\lambda > 0)$ is the regularization strength, W is the whole parameter space, N is the sample number, and T denotes a sampling frequency in one series. The logistic loss function is binary cross entropy.

Next, a single binary decision tree that accurately reproduces the network's thresholded binary predictions $\tilde{y}_n$ given input x_n is found. Then, the complexity of this decision tree as the output of $\Omega(W)$ is measured. The complexity is measured by the average decision path length, i.e., the average number of decision nodes that must be touched to make a prediction for an input example x_n. A regularization function $\widetilde{\Omega}(W)$ is used to map W to an estimate of the

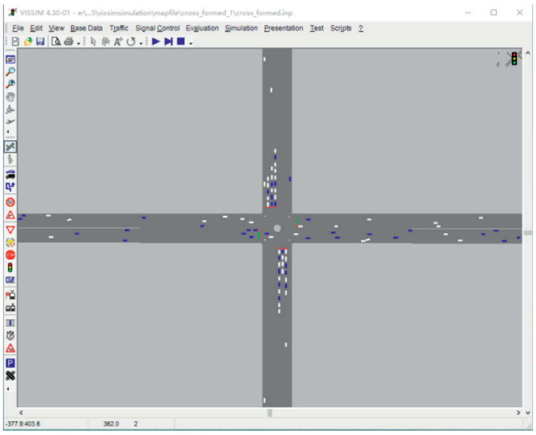


Figure 7: VISSIM simulation environment. The planned results of the COP algorithm and the real-time traffic flow are displayed in VISSIM.

Table 5: Sample datasets for TGRU and CycleGAN.

TGRU	Feature number	32
	Sample number	610
CycleGAN	Image (256 × 256 pixels) number of X	2238
	Image (256 × 256 pixels) number of Y	2238

average path length and is implemented by a multilayer perception (MLP) approximator.

Then, tree regularization is conducted, and its objective function is defined as follows:

$$\min_{\xi}\left(\sum_{i=1}^{J}\left(\Omega\left(W_j\right)-\widetilde{\Omega}\left(W_j,\xi\right)\right)^2\right)+\lambda\xi_2^2, \tag{10}$$

where J is the size of the candidate dataset of W and vector ξ denotes the parameters of this chosen MLP approximator.

4. Experiment

4.1. Setup. The platform and experimental environment configuration are shown in Table 4. We use a PC to run the COP algorithm and VISSIM for real-time traffic flow signal control and corresponding traffic simulation. We use another GPU server for both TGRU and CycleGAN training.

VISSIM, the traffic simulation platform, can capture and display the changes of traffic signal and traffic flow planned by the COP algorithm in real-time, as shown in Figure 7. Table 5 shows the sample datasets for TGRU and CycleGAN. In TGRU, we train a 3-layer MLP with 100 first-layer nodes, 100 second-layer nodes, and 10 third-layer nodes. In the CycleGAN, the generator contains encoding, transformation, and decoding. Encoding includes one 7 × 7 Convolution-InstanceNorm-ReLU layer with stride 1 and two 3 × 3 Convolution-InstanceNorm-ReLU layers with stride 2. Transformation includes 9 residual blocks for 256 × 256 images and two 3 × 3 convolutional layers with the same number of filters on both layers. Finally, decoding includes two 3 × 3 fractional strided Convolution-InstanceNorm-ReLU layers with stride 2 and one 7 × 7 Convolution-InstanceNorm-ReLU layer with stride 1. In the discriminator networks, we use 70 × 70 PatchGANs [17], and the discriminator architecture includes four 4 × 4 Convolution-InstanceNorm-Leaky-ReLU layers with stride 2. The last layer contains a convolution to produce a 1-dimensional output.

4.2. Congestion Attack Prediction and Visualized Analysis. We evaluate the performance of the CycleGAN model based on image features.

Evaluation Metric. For N samples testing, we further evaluate the CR, PCD, and ICD based on the mean absolute error (MAE) and root mean squared error (RMSE). We have MAE and RMSE of CR expressed as follows:

$$\begin{aligned}\mathrm{MAE}_{CR}&=\frac{1}{N}\sum_{i=1}^{N}\left|CR^i-\widetilde{CR^i}\right|,\\ \mathrm{RMSE}_{CR}&=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(CR^i-\widetilde{CR^i}\right)^2},\end{aligned} \tag{11}$$

where CR^i is the real value and $\widetilde{CR^i}$ is the estimated value. Similarly, we have $\mathrm{MAE}_{\mathrm{PCD}_k}$, $\mathrm{RMSE}_{\mathrm{PCD}_k}$, $\mathrm{MAE}_{\mathrm{ICD}}$, and $\mathrm{RMSE}_{\mathrm{ICD}}$.

Visualized Results and Quantitative Qnalysis. In Figure 8, the first column is the original image x, the second one is the output image $G(x)$ by CycleGAN, and the third column gives the real image y with congestion. Our approach has a satisfied generator and can predict a future result of congestion attacks to provide a visualization for better human understanding.

Table 6–8 show MAE and RMSE values under different evaluation metrics and test sets. Tables 6 and 7 show the CR

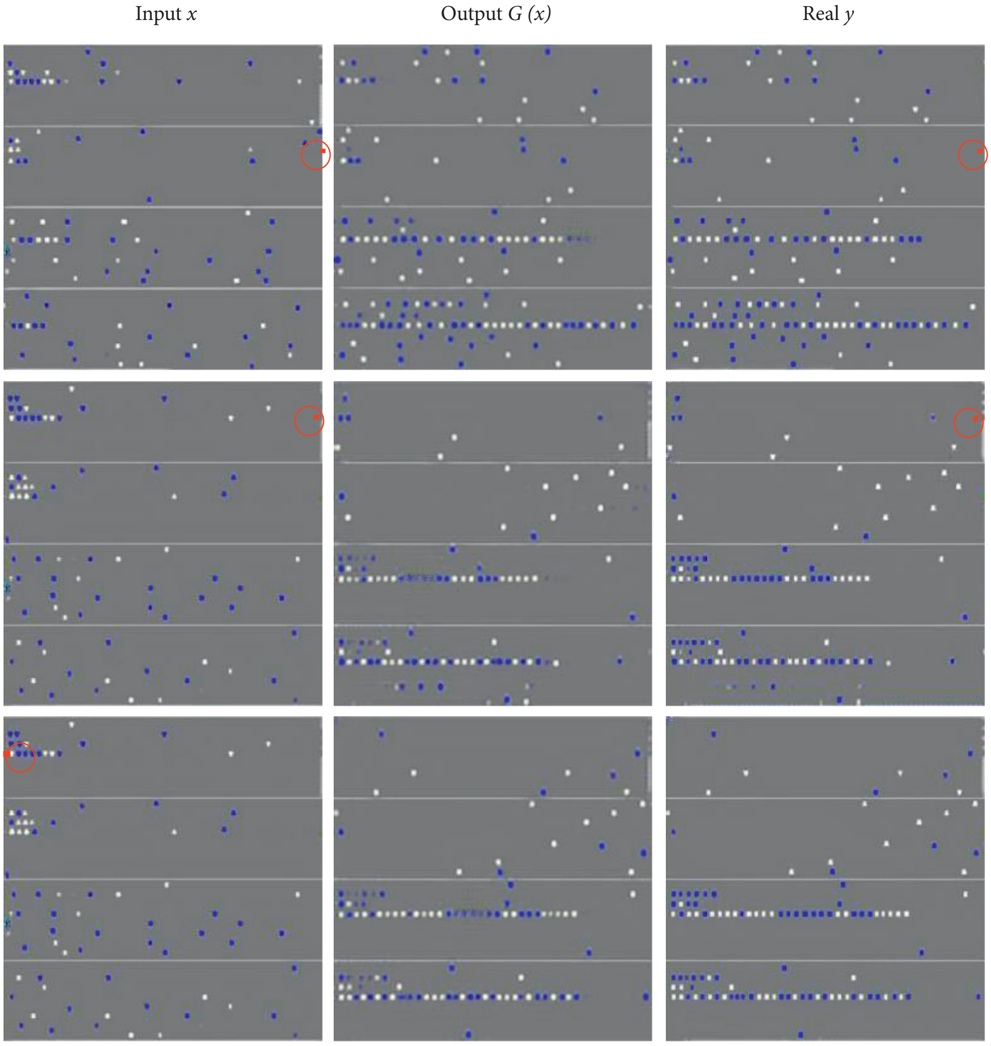


Figure 8: Visualized CycleGAN output compared with real traffic image two stages later based on three different original image inputs at the beginning of spoofing attack. The blue dots represent OBU-equipped vehicles whereas the while dots represent unequipped vehicles.

Table 6: $\mathrm{MAE}_{CR}$and RMSE_{CR} obtained on training set and with 4-fold cross validation or 10-fold cross validation.

	Training set	4-fold cross validation	10-fold cross validation
MAE_{CR}	0.0257	0.0213	**0.0205**
RMSE_{CR}	0.0310	0.0256	**0.0225**

The bold values denote the minimum values of MAE or RMSE on different training sets.

and CD measurements of one intersection and display a satisfying prediction compared with the ground truth. As shown in Table 6, in 10-fold cross validation, our CycleGAN has a pretty good performance in CR prediction. It has very small MAE and RMSE values, 0.0205 and 0.0225, respectively, which is better than that obtained with 4-fold cross validation.

For ICD (Table 7), the 4-fold cross validation results of MAE and RMSE are better than those of 10-fold cross validation, reaching 0.8100 and 0.9987, respectively. We present the detailed values of each phase for MAE and RMSE of congestion degree in Table 8. We can see that through comparing values based on the training set and cross validation, our CycleGAN-based model does not overfit by training. The best results are at $k = 3$, and we have the lowest values of $\mathrm{MAE}_{\mathrm{PCD}_k}$ and $\mathrm{RMSE}_{\mathrm{PCD}_k}$ (0.2250, 0.2050, 0.2050, 0.2617, 0.2519, and 0.2360) compared with the values of other phases. However, the errors at $k = 5$ increase a lot, which is why $\mathrm{MAE}_{\mathrm{ICD}}$ and $\mathrm{RMSE}_{\mathrm{ICD}}$ approach 1. This is because the fewer the vehicles, the better the prediction effect of the model. However, the attack vehicle is at phase 3, which has the least number of queues, while the congestion occurs at phase 5, which has the largest number of queues. Therefore, the prediction effect of phase 3 is the best of the 8 phases, so the lowest MAE and RMSE values are $k = 3$, while the prediction errors at phase 5 increased a lot.

We present bar charts for MAE and RMSE of 8-phase congestion degree in Figures 9 and 10, respectively. In

Figure 9, the best average value of MAE is based on 10-fold cross validation (with a value of 0.3844), and the worst average value is based on 4-fold cross validation (with a value of 0.4481). In Figure 10, we have similar results for RMSE; the best and the worst are 0.4471 and 0.5491, respectively. Both average values mean that the CycleGAN is robust and that we have good feature capture in our approach.

In addition, we compare the performance of the CycleGAN model with that of pix2pix [17], another GAN-based model, by quantitatively analyzing experimental results from the whole intersection and the specific phases perspective, respectively. Here, we use the experimental results under 4-fold cross validation. For the measurements of the whole intersection, as shown in Table 9, we have $\text{MAE}_{CR} = 0.0213$, $\text{RMSE}_{CR} = 0.0256$, $\text{MAE}_{\text{ICD}} = 0.8100$, and $\text{RMSE}_{\text{ICD}} = 0.9987$ for CycleGAN and $\text{MAE}_{CR} = 0.1167$, $\text{RMSE}_{CR} = 0.1297$, $\text{MAE}_{\text{ICD}} = 3.7917$, and $\text{RMSE}_{IC\ D} = 3.8500$ for pix2pix. We can see that CycleGAN has lower MAE and RMSE values than pix2pix. Therefore, the CycleGAN model has a better performance than the pix2pix model on the measurements of the whole intersection.

We further compare the model performance for the specific phases. Table 10 shows the detailed MAE and RMSE values of each phase for CycleGAN and pix2pix. There are the lowest MAE and RMSE values at $k = 3$ for both models: 0.2050 and 0.2538 for CycleGAN and 0.2519 and 0.9830 for pix2pix. For all phases, the MAE and RMSE values of CycleGAN are lower than those of pix2pix. Therefore, the CycleGAN model also has a better performance than the pix2pix model on the measurements of all phases.

To sum up, in the CycleGAN-based prediction model, we extract four-direction road images of the intersection and perform phase-based composition for generating a new sample image to quantify the traffic flow characteristics. Based on the image feature, the CycleGAN-based approach analyzes the potential relationship between the congestion attack and the corresponding congestion effect two stages later. Also, the model is used to analyze the congestion effects that different phases of the attack vehicle caused. Meanwhile, we can obtain the visualized results based on the image feature. The experimental results on the CycleGAN-based model and compared experiments with the pix2pix model demonstrated the superiority of the CycleGAN-based model.

4.3. Congestion Attack Verification. Here, we evaluate the performance of the TGRU model based on traffic flow features. We use the confusion matrix, accuracy, AUC value, precision, recall, and F1-score.

The TGRU model is trained to distinguish whether the intersection is under a spoofing attack based on traffic flow features. We collect time-series traffic flow data for 3600 seconds under both the normal state and attack state. We consider 1-second intervals as time steps. Each data vector x_{nt} has 30 features, as defined in Section 3.4. Each outcome y_{nt} is a binary label marking whether the intersection is under a spoofing attack. The sequence length is set to 20 seconds, considering that the maximum green time of each signal is 20 seconds. Hence, 360 samples are obtained in total: 180 samples of which contain 3600 traffic flow data and are used for training and the other 180 samples are used for testing.

We apply the model to the test set and calculate its AUC value, accuracy, precision, recall, and F1-score; these values are reported in Table 11. We see that for different parameter settings, the TGRU model with our defined traffic features can achieve a great prediction quality. The AUC values are all approximately 0.8, and the accuracy values are 0.79, 0.79, and 0.75 when using different parameter settings. Furthermore, the average values of precision, recall, and F1-score are satisfying, almost near 0.8.

Figure 11 shows the three ROC [18] curves of TGRU. Corresponding AUC values are shown as well. We can see that these curves are similar, and their AUC values (0.82, 0.85, and 0.78) are all around 0.8. Moreover, the TGRU model has similar performance with different parameter settings; this indicates that our defined classification features are efficient, and the different parameter settings have little effect on TGRU model's performance.

The decision tree generated by Graphviz [19] is shown in Figure 12. For 3600 traffic flows, this tree has 9 levels. From top to down, according to each feature value, the flow data can be grouped into different classes step by step. For example, when $X\ [13] \leq 0.068$, there are 32 traffic flows of 57 flows correctly predicted as the class of spoofing attack 1; this indicates the importance of the 13th dimension feature, i.e., the congestion degree of the 8th phase PCD_8, in predicting the class of spoofing attack 1.

Also, we compare the TGRU model with a time-series prediction method, seasonal autoregressive integrated moving average (SARIMA). Here, it is detected whether the congestion occurs or not based on traffic flow features. We carry out experiments for different approaches under different traffic flow feature sets. We choose the primary traffic flow data for the first feature set as the traffic flow feature FS_1. The second feature set FS_2 is shown in Table 3. According to the two approaches, we construct two feature sets based on the traffic flow data we collect. As shown in Table 12, the accuracy values of SARIMA and TGRU on the feature set FS_1 are 0.744 and 0.772, respectively, and on the feature set FS_2 are 0.784 and 0.790, respectively, which demonstrates that the TGUR model based on our defined traffic flow features is superior to others.

In conclusion, in the TGRU-based verification model, we propose some timing characteristics, including capacity ratio, congestion degree, attack acceleration, and attack amplification ratio, to measure the congestion effects based on traffic flow. Based on the defined traffic flow features, the TGRU-based model is used to analyze the underlying relationship between the congestion attack and traffic flow features at the current moment. Meanwhile, the decision tree helps better interpret the relationship between traffic flow features and the congestion attack. The experimental results on the TGRU-based model and compared experiments with

Table 7: MAE_{ICD} and RMSE_{ICD} on training set and with 4-fold cross validation or 10-fold cross validation.

	Training set	4-fold cross validation	10-fold cross validation
MAE_{ICD}	0.8350	**0.8100**	1.1800
RMSE_{ICD}	1.0500	**0.9987**	1.3609

The bold values denote the minimum values of MAE or RMSE on different training sets.

Table 8: MAE_{PCD} and RMSE_{PCD} on training set and with 4-fold cross validation or 10-fold cross validation (k denotes the kth phase).

	$\text{MAE}_{\text{PCD}_k}$			$\text{RMSE}_{\text{PCD}_k}$		
	Training set	4-fold cross validation	10-fold cross validation	Training set	4-fold cross validation	10-fold cross validation
$k = 1$	0.5650	0.4450	0.6575	0.6749	0.5497	0.7884
$k = 2$	0.3850	0.5050	0.3625	0.5074	0.6268	0.4242
$k = 3$	**0.2250**	**0.2050**	**0.2050**	**0.2617**	**0.2519**	**0.2360**
$k = 4$	0.3450	0.3200	0.2250	0.4319	0.3733	0.2639
$k = 5$	**0.8300**	**0.9200**	**0.5925**	**0.9859**	**1.1070**	**0.5720**
$k = 6$	0.3850	0.5350	0.4125	0.5035	0.6535	0.5188
$k = 7$	0.3900	0.3750	0.3600	0.4701	0.4759	0.4602
$k = 8$	0.4300	0.2800	0.2600	0.4990	0.3550	0.3131

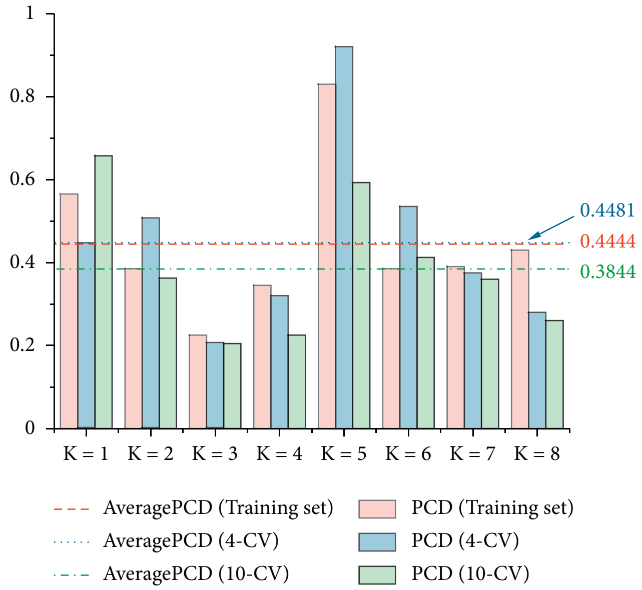


Figure 9: Bar chart of $\text{MAE}_{\text{PCD}_k}$.

the SARIMA model demonstrated the superiority of the TGRU-based approach.

5. Defense Suggestions

To proactively address the congestion attack of the I-SIG system, this section discusses how to defend against the attacks assessed above.

EVLS Improvement for COP Reinforcement. As estimated by the USDOT [20], I-SIG may take 25–30 years to reach a 95% PR for intelligent transportation systems. Thus, for I-SIG under a real low PR, I-SIG needs to adopt an EVLS algorithm to estimate non-OBU-equipped vehicles' location and speed. In the current I-SIG system design, the congestion attack on the COP algorithm utilizes a nonrobust estimation of EVLS. However, it is possible to improve EVLS and thus reinforce the COP algorithm. For single global positioning system (GPS) spoofing, we can introduce more collaboration mechanisms from the transportation field, such as the car-following model. A natural way to accomplish this is to significantly improve queue-length prediction. In the existing EVLS, this could be realized by adding a new software module that interacts with the COP algorithm. Such implementation has a low cost and brings little change to the original COP algorithm.

Another problem is the high impact of PR on security, which we have to change. In the current design, when the PR is smaller, the impact of the attack on the system is more significant because the system cannot accurately obtain the

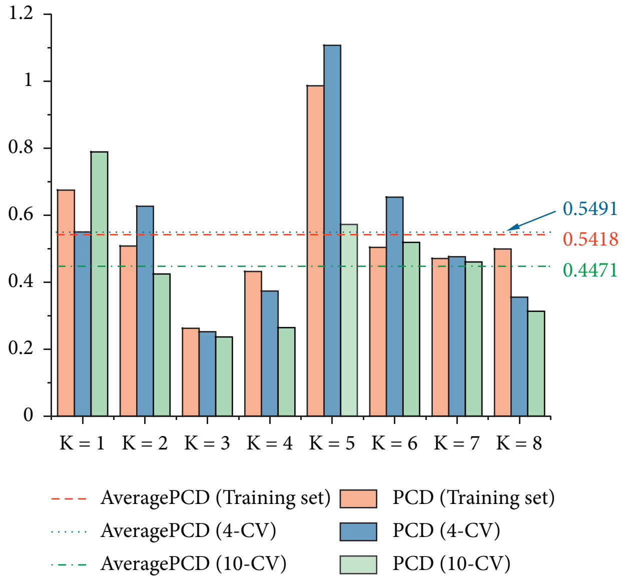


Figure 10: Bar chart of $\mathrm{RMSE}_{\mathrm{PCD}_k}$.

Table 9: MAE_{CR}, RMSE_{CR}, $\mathrm{MAE}_{\mathrm{ICD}}$, and $\mathrm{RMSE}_{\mathrm{ICD}}$ of CycleGAN and pix2pix with 4-fold cross validation.

	CycleGAN	pix2pix
MAE_{CR}	0.0213	0.1167
RMSE_{CR}	0.0256	0.1297
$\mathrm{MAE}_{\mathrm{ICD}}$	0.8100	3.7917
$\mathrm{RMSE}_{\mathrm{ICD}}$	0.9987	3.8500

Table 10: $\mathrm{MAE}_{\mathrm{PCD}}$ and $\mathrm{RMSE}_{\mathrm{PCD}}$ of CycleGAN and pix2pix with 4-fold cross validation (k denotes the kth phase).

	$\mathrm{MAE}_{\mathrm{PCD}_k}$		$\mathrm{RMSE}_{\mathrm{PCD}_k}$	
	CycleGAN	pix2pix	CycleGAN	pix2pix
$k=1$	0.4450	2.9500	0.5497	3.2383
$k=2$	0.5050	0.9850	0.6268	1.1697
$k=3$	**0.2050**	**0.2538**	**0.2519**	**0.9830**
$k=4$	0.3200	1.7550	0.3733	2.7336
$k=5$	0.9200	3.3283	1.1070	3.4750
$k=6$	0.5350	1.1250	0.6535	1.1307
$k=7$	0.3750	1.9500	0.4759	2.3499
$k=8$	0.2800	0.3342	0.3550	0.5393

The bold values denote the minimum values of MAE or RMSE when k varies from 1 to 8.

queue length with fewer data. We do not suggest providing two alternative versions of EVLS (i.e., one for high and one for low PR, respectively). Although we analyzed a car-following model in work [21], we believe that a more useful model with a collaboration mechanism should be studied; this will make the estimation of EVLS more accurate as well as COP security more robust.

Authentication and Anomaly Detection. In the current design, authentication is realized through communication between OBU and RSU. However, the attack vehicle might not be a newly joining vehicle or an unauthenticated vehicle; in fact, it can be a normal vehicle with legal authentication. Thus, although authentication reinforcement is not the solution, it can be used to aid in anomaly detection. The idea is that a vehicle cannot appear somewhere suddenly; from the beginning authentication, we should perform analysis on time-series trajectory data to discover any anomaly behavior; this requires a powerful RSU with more computing ability and storing capacity. In addition to an anomaly detection algorithm, implementation needs the support of a collaboration mechanism of multiple I-SIGs; this is a complex global design of intelligent transportation and has not been realized yet. We believe this is critical work that must be accomplished before wide I-SIG deployment.

Prevent Cold-Start Attack. Essentially, the congestion attack is a type of insider attack. Thus, it is challenging to perform anomaly detection for such an attack in a pretty

Table 11: TGRU performance in terms of AUC value, accuracy, precision, recall, and F1-score.

Iteration times	AUC	Accuracy	Classification report			
				Precision	Recall	F1-score
Iters_retrain = 25 Num_iters = 300	0.82	**0.79**	0: normal	0.71	0.98	0.82
			1: attack	0.97	0.59	0.74
			Average	**0.84**	**0.79**	0.78
Iters_retrain = 50 Num_iters = 1000	**0.85**	**0.79**	0: normal	0.72	0.95	0.82
			1: attack	0.93	0.64	0.76
			Average	0.83	**0.79**	**0.79**
Iters_retrain = 100 Num_iters = 3000	0.78	0.75	0: normal	0.69	0.90	0.78
			1: attack	0.86	0.60	0.71
			Average	0.78	0.75	0.75

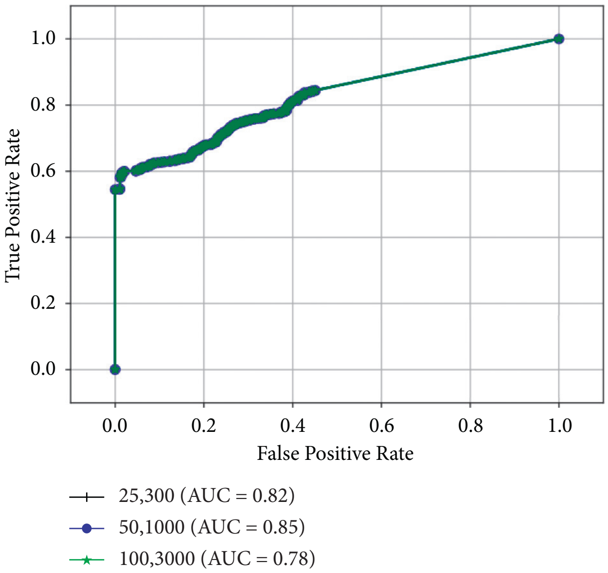


Figure 11: ROC curve of TGRU with AUC values.

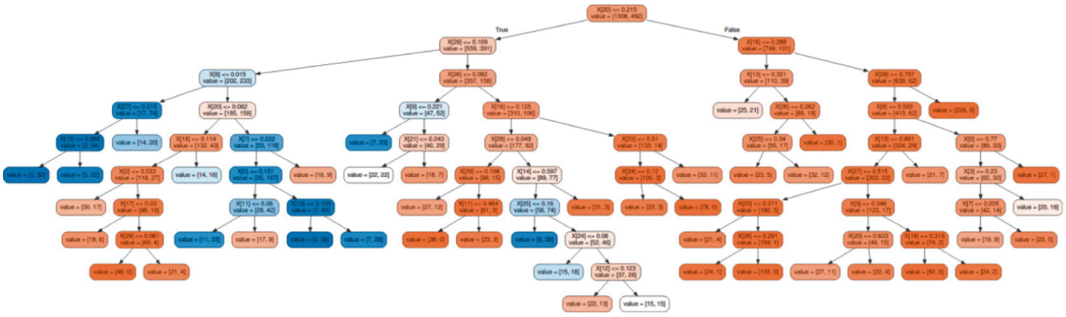

X[20] <= 0.215 value = [1308, 492]
True
False
X[29] <= 0.109 value = [559, 391]
X[16] <= 0.288 value = [749, 101]
X[8] <= 0.015 value = [202, 233]
X[26] <= 0.082 value = [357, 158]
X[13] <= 0.321 value = [110, 39]
X[29] <= 0.707 value = [639, 62]
X[27] <= 0.016 value = [17, 74]
X[20] <= 0.082 value = [185, 159]
X[9] <= 0.221 value = [47, 52]
X[16] <= 0.125 value = [310, 106]
value = [25, 21]
X[26] <= 0.262 value = [85, 18]
X[8] <= 0.593 value = [413, 62]
value = [226, 0]
X[13] <= 0.068 value = [3, 54]
value = [14, 20]
X[18] <= 0.114 value = [132, 43]
X[7] <= 0.222 value = [53, 116]
value = [7, 23]
X[21] <= 0.243 value = [40, 29]
X[28] <= 0.049 value = [177, 92]
X[23] <= 0.51 value = [133, 14]
X[25] <= 0.34 value = [55, 17]
value = [30, 1]
X[13] <= 0.801 value = [324, 29]
X[0] <= 0.77 value = [89, 33]
value = [0, 32]
value = [3, 22]
X[2] <= 0.033 value = [118, 27]
value = [14, 16]
X[0] <= 0.151 value = [35, 107]
value = [18, 9]
value = [22, 22]
value = [18, 7]
X[29] <= 0.198 value = [88, 15]
X[14] <= 0.597 value = [89, 77]
X[24] <= 0.12 value = [100, 3]
value = [33, 11]
value = [23, 5]
value = [32, 12]
X[27] <= 0.515 value = [303, 22]
value = [21, 7]
X[3] <= 0.23 value = [62, 32]
value = [27, 1]
value = [30, 17]
X[17] <= 0.03 value = [88, 10]
X[11] <= 0.06 value = [28, 42]
X[13] <= 0.106 value = [7, 65]
value = [27, 12]
X[11] <= 0.464 value = [61, 3]
X[25] <= 0.16 value = [58, 74]
value = [31, 3]
value = [22, 3]
value = [78, 0]
X[23] <= 0.211 value = [180, 5]
X[0] <= 0.346 value = [123, 17]
X[7] <= 0.205 value = [42, 14]
value = [20, 18]
value = [19, 6]
X[29] <= 0.061 value = [69, 4]
value = [11, 33]
value = [17, 9]
value = [0, 39]
value = [7, 26]
value = [38, 0]
value = [23, 3]
value = [6, 28]
X[24] <= 0.08 value = [52, 46]
value = [21, 4]
X[28] <= 0.291 value = [159, 1]
X[20] <= 0.633 value = [49, 15]
X[18] <= 0.319 value = [74, 2]
value = [19, 9]
value = [23, 5]
value = [48, 0]
value = [21, 4]
value = [15, 18]
X[12] <= 0.123 value = [37, 28]
value = [24, 1]
value = [135, 0]
value = [27, 11]
value = [22, 4]
value = [50, 0]
value = [24, 2]
value = [22, 13]
value = [15, 15]

Figure 12: Whole decision tree of 9-level depth.

Table 12: Comparison of different prediction approaches.

Feature set	FS_1	FS_2
SARIMA	0.744	0.784
TGRU	**0.772**	**0.790**

The bold values are the maximum of accuracy when the prediction approach is different.

short time only based on nearby vehicle speed and location information. This means that we cannot avoid the first spoofing data entering the arrival table of the COP algorithm. We suggest that emerging blockchain technology, especially light blockchain, should be considered to rebuild I-SIG or even the whole intelligent transportation system. After that, any data of one node have to be verified by all other nodes. This would result in nearly no chance for spoofed data to be accepted. However, the cost of rebuilding the system is obviously enormous, and more attention should be paid to the light blockchain to test the trade-off between efficiency and security. Regardless, we still believe that this is a promising future for I-SIG security defense.

6. Related Work

Data Spoofing Attacks in SAGIN. The SAGIN has a heterogeneous structure, including vehicle nodes, roadside infrastructure, mobile terminal users, drones, airships, and other stratospheric nodes, as well as high altitude satellite nodes; this brings security challenges [22], such as the various attacks of authenticity, identity, confidentiality, data integrity, and privacy [23]. As a SAGIN-based intelligent transportation system deployed in California, Florida, and New York by the USDOT, I-SIG is exposed to data spoofing attacks [8], which can cause heavy congestion. Such an attack is a position-faking attack of GPS spoofing but is different from a tunnel attack. In a tunnel attack, each vehicle of a Vehicular Ad hoc NETwork (VANET) [24, 25] is equipped with a positioning system (receiver). The attack can be achieved using a transmitter generating localization signals stronger than those generated by the real satellites [26, 27]. The victim could be waiting for a GPS signal after leaving a physical tunnel or a jammed-up area. In comparison, the position spoofing attack to I-SIG refers to an authenticated vehicle only sending the wrong position to affect the COP algorithm, which has lower attack cost and easier implementation. In such an attack, the data spoofing is just one factor, while the mechanism of the COP algorithm is the key factor. Furthermore, for the GPS spoofing attack, our work focuses on algorithm-level security analysis under a spoofing attack.

Congestion Attack Analysis. The previous work [8] reveals the existence of such congestion attacks on the COP algorithm. It analyzes how congestion attacks affect COP decisions and explains how to execute an attack using data spoofing in SAGIN. However, it lacks consideration about the potential features and the quantified correlation between the attack and congestion degree. In comparison, we demystify the attack on I-SIG and corresponding congestion from a machine learning perspective by exploring different kinds of features based on both supervised learning and unsupervised learning. In addition, as the first utilization of both traffic flow features and image features, our work can inspire all stakeholders of I-SIG, including experts of transportation, SAGIN, and security.

7. Conclusions

Toward the spoofing to connected vehicle technology and the SAGIN, a congestion attack has been revealed on the COP algorithm of I-SIG, which performs dynamic and optimal signal control based on automatic traffic situation awareness. Owing to the lack of quantified feature-level analysis, we demystify the attack on I-SIG and the corresponding congestion from both supervised learning and unsupervised learning. We propose a CycleGAN-based approach to analyze the potential relations between the congestion attack and the corresponding results two stages later. We also present a TGRU-based approach to explore the relations between the congestion attack and traffic flow features at a certain moment. In our experiment, we collect high-quality 4476 image samples and 3600 attack-oriented traffic flow data. We then evaluate our approach empirically using the COP algorithm and VISSIM, and our results show the effectiveness of our approach compared with ground truth.

This work is expected to inspire a series of follow-up studies on the security of CV-based I-SIG, but not limited to (1) more machine learning-based approaches, (2) more concrete defense implementation on SAGIN-based I-SIG, and (3) more feature fusion for attack and defense analysis.

Data Availability

All data generated or analyzed during this study are owned by all the authors and will be used to our further research. The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China under Grant nos. 61972025, 61802389, 61672092, U1811264, and 61966009, the National Key R&D Program of China under Grant nos. 2020YFB1005604 and 2020YFB2103802, and the Guangxi Key Laboratory of Trusted Software under Grant no. KX201902.

References

[1] U. S. Dot connected vehicle pilot deployment program, https://www.its.dot.gov/pilots/.

[2] Connected Vehicle Applications, https://www.its.dot.gov/pilots/cv_pilot_apps.htm.

[3] Usdot: Multimodal Intelligent Traffic Safety System (Mmitss), https://www.its.dot.gov/research_archives/dma/bundle/mmitss_plan.htm.

[4] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, “Space-air-ground integrated network: a survey,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[5] W. Zhang, L. Li, N. Zhang, T. Han, and S. Wang, “Air-ground integrated mobile edge networks: a survey,” *IEEE Access*, vol. 8, pp. 125998–126018, 2020.

[6] S. Sen and K. L. Head, “Controlled optimization of phases at an intersection,” *Transportation Science*, vol. 31, no. 1, pp. 5–17, 1997.

[7] Y. Feng, K. L. Z. Head, S. Khoshmagham, and M. Zamanipour, “A real-time adaptive signal control in a

connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 460–473, 2015.

[8] Q. A. Chen, Y. Yin, Y. Feng, Z. M. Mao, and H. X. Liu, "Exposing Congestion Attack on Emerging Connected Vehicle Based Traffic Signal Control," in *Proceedings of the Network and Distributed System Security Symposium*, pp. 39.1–39.15, San Diego, CA, USA, February 2018.

[9] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, Venice, Italy, October 2017.

[10] M. Wu, M. C. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshivelez, "Beyond sparsity: tree regularization of deep models for interpretability," in *Proceedings of the The Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 1670–1678, New Orleans, LI, USA, February 2018.

[11] K. Cho, B. Van Merrienboer, C. Gulcehre et al., "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Empirical Methods in Natural Language Processing*, pp. 1724–1734, Springer, Berlin, Germany, 2014.

[12] Ptv Vissim, http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim.

[13] A. Krok, "Us department of transportation hopes to mandate v2v communications," 2016, https://www.cnet.com/roadshow/news/us-department-of-transportation-hopes-to-mandate-v2v-communications.

[14] H. Xiong, Z. Tan, R. Zhang, and S. He, "A new dual axle drive optimization control strategy for electric vehicles using vehicle-to-infrastructure communications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2574–2582, 2020.

[15] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[16] R. K. Patel and E. J. Seymour, "The national transportation communication for its protocol (ntcip) for transportation interoperability," in *Proceedings of the Conference on Intelligent Transportation Systems*, pp. 543–548, Boston, MA, USA, November 1997.

[17] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, Honolulu, HI, USA, July 2017.

[18] X. Sun and W. Xu, "Fast implementation of DeLong's algorithm for comparing the areas under correlated receiver operating characteristic curves," *IEEE Signal Processing Letters*, vol. 21, no. 11, pp. 1389–1393, 2014.

[19] Graphviz-graph Visualization Software, https://graphviz.org/.

[20] Vehicle-infrastructure integration (vii) initiative: benefit-cost analysis, https://www.pcb.its.dot.gov/connected_vehicle.htm.

[21] X. Gao, J. Liu, Y. Li et al., "Queue length estimation based defence against data poisoning attack for traffic signal control," *IFIP Advances in Information and Communication Technology*, pp. 254–265, 2020.

[22] M. Arshad, Z. Ullah, M. Khalid et al., "Beacon trust management system and fake data detection in vehicular ad-hoc networks," *IET Intelligent Transport Systems*, vol. 13, no. 5, pp. 780–788, 2019.

[23] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz, "Anonysense: opportunistic and privacy-preserving context collection," in *Proceedings of the International Conference on Pervasive Computing*, pp. 280–297, New York, NY, USA, May 2018.

[24] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (vanets): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[25] X. Zhong, L. Li, Y. Zhang, B. Zhang, W. Zhang, and T. Yang, "Oodt: obstacle aware opportunistic data transmission for cognitive radio ad hoc networks," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3654–3666, 2020.

[26] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 760–776, 2019.

[27] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "A survey on pseudonym changing strategies for vehicular ad-hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 770–790, 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 7897527, 8 pages
https://doi.org/10.1155/2021/7897527

Research Article

Practical SM2-Based Multisignature Scheme with Applications to Vehicular Networks

Lin Hou,[1] **Wei Liu**,[1] **Lisha Yao**,[1] **Xiaojian Liang**,[1] **and Guo-Qiang Zeng**[2]

[1]*College of Information Science and Technology, Jinan University, Guangzhou 510632, China*
[2]*College of Cyber Security and the National Joint Engineering Research Center of Network Security Detection and Protection Technology, Jinan University, Guangzhou 510632, China*

Correspondence should be addressed to Wei Liu; weiliuscholar@gmail.com

Received 11 June 2021; Accepted 13 September 2021; Published 27 October 2021

Academic Editor: Rongmao Chen

In vehicular networks, the increasing value of transportation data and scale of connectivity also brings many security and privacy concerns. Peer authentication and message integrity are two vital security requirements to ensure safe transportation system. Because of the constrained resources of the units performing the cryptographic components, the proposed security-enhancing schemes should be lightweight and scalable. In this paper, we present a multisignature scheme derived from the SM2 signature which enables a group of parties to collaboratively sign a message and generate a compact joint signature at the end. Our scheme requires no preprocessing or interactions among the parties before signing, and its performance matches or surpasses known ones in terms of signing time, verification time, and signature size. Therefore, our scheme is also suitable for vehicular networks, with the goal to enhance security with small computation and storage cost.

1. Introduction

With the development of advanced information and communication-based technologies, intelligent transportation system (ITS) can provide a seamless transportation infrastructure and more functionalities for vehicles than a decade ago. Specifically, the Vehicle-to-Everything (V2X) communication technology in vehicular networks nowadays is able to support information sharing between vehicles and any other element involved in ITS [1, 2], including nearby vehicles (V2V), the infrastructure (V2I), mobile devices carried by pedestrians (V2P), and remote application servers or cloud platforms (V2N). The increasing scale of ITS ecosystem and the growing trend to integrate vehicular network deployment with other networks also bring concerns about cybersecurity for ITS since any message interception or modification by malicious units could result in fatal consequences [3, 4].

Digital signature is commonly used in vehicular networks to ensure integrity of messages exchanged among devices. However, the effectiveness of information propagation and routing, which are associated to delays and hence also have impacts on road safety, naturally depends on the computational overhead imposed by the applied security mechanisms [5]. Beyond traditional signature schemes, multisignature (MS) and aggregate signature (AS) are extended primitives considering multiuser setting to support cosigning and to reduce verification cost. The two primitives in common allow a group of signers to combine their individual signatures into a single short one. Specifically, an MS scheme [6, 7] enables a group of signers, each having a public key and a corresponding private key, to collaboratively produce a joint signature on a common message which can be publicly verified given the set of public keys of all signers. As a more general primitive, an AS scheme [8, 9] allows each of the signers to sign a different message, and all these individual signatures can still be aggregated into a single short one. As in the traditional signature scheme, the short combined signature should convince the verifier that all signers signed their designated messages.

Both MS and AS schemes have many potential uses in vehicular networks, such as in the distributed certificate authority (CA) or in V2I/V2V communications. Unfortunately, the commonly used technologies including dedicated short-range communications (DSRC) and cellular-V2X (C-V2X) mainly exploit elliptic curve-based signature schemes, e.g., ECDSA and SM2, which to the best of our knowledge has very few MS or AS extensions due to their nonlinear construction.

In this paper, we propose a candidate multisignature scheme $MS-SM2$ based on the SM2 signature algorithm and specify the applications of $MS-SM2$ for vehicular networks. SM2 is a signature algorithm standard based on the elliptic curve published by the Chinese government and has been extensively used in cryptographic devices in finance and industry. Our proposed $MS-SM2$ scheme allows dynamic joining of signers (with certified public keys) and has no burdensome assumptions on the public-key infrastructure (PKI), which makes it plausible in vehicular networks.

1.1. Our Contributions. The original contribution of this work is mainly twofold:

(i) We first present a multisignature scheme $MS-SM2$ based on the SM2 signature by designing a cosigning protocol and prove its security in plain public-key and semihonest model. No preprocessing or any proof-of-knowledge step on the signer side is required in our scheme. The experimental results also show that our protocol is relatively practical for many applications.

(ii) We then illustrate some possible applications of $MS-SM2$ in vehicular networks, especially the usage in the multiple CAs architecture to reduce the certification storage for vehicles and RSUs and in V2I communication to reduce the computational overhead for RSUs.

1.2. Related Work. A trivial way to build a multisignature from standard signatures is to concatenate all stand-alone signatures signed individually. However, the resulted multisignature is of large size and particularly of size proportional to the number of signers, which does not scale well in practice [6, 7, 10]. Therefore, a multisignature should be short, meaning its length should be (ideally) independent from the number of signers and about the same as that of an ordinary stand-alone signature. Informally, the possibility of extending standard signature schemes to multisignatures comes from the homomorphism of the involved arithmetic operations of the underlying assumptions. However, the homomorphism also brings a serious vulnerability and allows adversaries to mount *rogue key attacks*, in which the attackers without valid key pairs can set its public key as a function of those from other honest signers and finally forge multisignatures. Micali et al. [6] described the formal model for the attack and showed a way to prevent such attacks known as *knowledge of secret key* (KOSK) assumption, in which users are required to prove knowledge of their secret keys during public key registration. Bellare and Neven [7] proposed a new practical multisignature scheme based on the Schnorr signature without KOSK assumption and proved that it can avoid rogue attack in the so-called *plain public key model*. There are several following-up work on constructing 2-round Schnorr-based multisignatures, i.e., all singers only need 2 rounds of communications to produce a multisignature [11–15]. Recently, public key aggregation is introduced to a multisignature scheme by which the verifier can check the validity of a multisignature only using a short aggregate key rather than a public key list [16, 17].

2. Preliminaries

For prime number p, $\mathbb{Z}_p$ denotes the additive group of integer modulo p. We consider elliptic curve E: $y^2 = x^3 + ax + b \pmod{p}$ in $\mathbb{Z}_p$, where $a, b \in \mathbb{Z}_p$ and $4a^3 + 27b^2 \neq 0 \pmod{p}$. The set of points on E along with the infinity point $\mathcal{O}$ constitutes an additive elliptic-curve group $E(\mathbb{Z}_p)$ under points addition, denoted by $\oplus$, with $\mathcal{O}$ being the identity. Let $G(x_G, y_G) \in E(\mathbb{Z}_p)$ $(G \neq \mathcal{O})$ be the base point in $E(\mathbb{Z}_p)$ with order n. For $k \in \mathbb{Z}$, $Q(x_Q, y_Q) = [k]G$ denotes the scalar multiplication in $E(\mathbb{Z}_p)$.

Range $[x, y]$ denotes the set of integers i, $x \leq i \leq y$. Given a nonempty set S, $s^{\$} \longleftarrow$ denotes the operation of sampling an element of S uniformly at random and assigning it to s. For a randomized algorithm $\mathcal{A}$, $y \longleftarrow \mathcal{A}((x_1, \ldots, x_n); \rho)$ denotes the operation of running $\mathcal{A}$ on inputs $(x_1, \ldots, x_n)$ and random coins ρ then assigning its output to y.

2.1. Multisignature Scheme

2.1.1. Syntax. We follow the description of Bellare and Neven [7] and define a multisignature scheme as a tuple $MS = (\text{Setup}, \text{KeyGen}, \text{MSign}, \text{Vrfy})$. Note that the scheme is defined in the plain public key model, where the key generation is as same as that in any public-key cryptography and no more preprocessing protocol or key verification is required.

$\text{Setup}(1^\kappa) \longrightarrow pp$: the setup algorithm takes as input the security parameter κ and generates system parameters pp.

$\text{KeyGen}(pp) \longrightarrow (sk, pk)$: the key generation algorithm is a randomized algorithm executed by every signer on input pp to generate a key pair (sk, pk).

$\text{MSign}(pp, L, sk_i, m) \longrightarrow \sigma$: the MSign algorithm represents the signing protocol run by a group of signers who intend to collaboratively sign the same message m. Each signer i executes the protocol on input pp, a set of public keys of signers $L = \{pk_1, \ldots, pk_N\}$, private key sk_i and message m. The protocol outputs a multisignature σ.

$\text{Vrfy}(pp, L, m, \sigma) \longrightarrow 0/1$: the verification algorithm checks the validity of a multisignature σ on message m on behalf of the group of signers whose public keys are in set L and output 1 or 0 indicating the multisignature is valid or not.

2.1.2. Completeness. A multisignature scheme should satisfy the following *completeness* property, meaning that for any number n and message m, if $(pk_i, sk_i) \leftarrow \text{Key Gen}(pp)$ for $i \in \{1, \ldots, N\}$ and all signers run $\text{MSign}(pp, L, m, sk_i)$, then every signer will output the same signature σ such that $\text{Vrfy}(pp, L, m, \sigma) = 1$.

2.1.3. Security. The security of multisignature requires that it is infeasible to forge a signature involving at least one honest signer. We assume an adversary (forger) $\mathscr{F}$ that corrupts all other signers except the honest one and can choose their public keys in arbitrary ways as it likes, e.g., the rogue key attack. The unforgeability of multisignature in plain public key model is defined by the following three-phase game $\text{Exp}_{MS}^{UF-CMA}(\mathscr{F})$ between the forger $\mathscr{F}$ and a challenger.

Setup. The challenger generates system parameter $pp \leftarrow \text{Setup}(1^\kappa)$ and a challenge key pair $(pk^*, sk^*) \leftarrow \text{KeyGen}(pp)$ for the target honest signer. It returns (pp, pk^*) to $\mathscr{F}$.

Query. The forger $\mathscr{F}$ is allowed to make signature queries on any message m for any set L of signers with $pk^* \in L$. This signing oracle $\mathscr{O}(pp, \cdot, sk^*, \cdot)$ simulates the honest signer with key sk^* interacting in a signing protocol with other signers in list L. $\mathscr{F}$ can make any number of such queries concurrently.

Forge. $\mathscr{F}$ outputs a set L^* of public keys, a message m^*, and a multisignature σ^*. The forger is said to win the game if $\text{Vrfy}(pp, L^*, m^*, \sigma^*) = 1$ with $pk^* \in L^*$ and the message m^* never appeared in Query phase.

The advantage of forger $\mathscr{F}$ in breaking the multisignature scheme is defined as the probability that $\mathscr{F}$ wins the above game (over the random coins of the challenger), denoted as $\text{Adv}_{MS}^{UF-\text{CMA}}(\mathscr{F})$.

Definition 1 (UF-CMA security). A multisignature scheme is (t, q_s, N, ε)-unforgeable if it holds that $\text{Adv}_{MS}^{UF-\text{CMA}}(\mathscr{F}) \leq \varepsilon$ for every forger $\mathscr{F}$ that runs in time at most t, makes at most q_s signing queries, produces forgeries on behalf of N parties, and wins the $\text{Exp}_{MS}^{UF-CMA}(\mathscr{F})$ game with negligible probability ε. In random oracle model, we define it as $(t, q_s, q_h, N, \varepsilon)$-unforgeable where q_h denotes the maximum number of hash queries.

2.2. SM2 Signature Algorithm. The SM2 signature algorithm is initialized by taking as input a security parameter κ and outputs $pp(E(\mathbb{Z}_p), \mathscr{O}, G, n, H(\cdot))$ as public parameters, in which $H\colon \{0,1\}^* \longrightarrow \mathbb{Z}_n$ is a cryptography hash function. The SM2 signature scheme is briefly reviewed in Table 1.

2.3. General Forking Lemma. We will use the general forking lemma [7] to prove the security of our scheme, which is a useful tool by extending the forking lemma of Pointcheval and Stern [18] without mentioning concrete signatures or random oracles.

Lemma 1 (general forking lemma). *Let H be a set of size $h(\geq 2)$, and $(h_1, \ldots, h_q) \overset{\$}{\leftarrow}$. Let $\mathscr{A}$ be a randomized algorithm that on input $\{x, (h_1, \ldots, h_q)\}$ returns a pair (i, σ), where $i \in \{0, \ldots, q\}$ and σ is a side output. For some randomized input generator IG, the accepting probability of algorithm $\mathscr{A}$, denoted by acc, is defined as $\Pr[i \geq 1 || x \overset{\$}{\leftarrow}$. Consider randomized algorithm $\text{Fork}^{\mathscr{A}}$ associated with $\mathscr{A}$, taking as input x, proceeds as described in Algorithm 1. Let frk be the probability that $\Pr[b = 1 | x \overset{\$}{\leftarrow}$. Then,*

$$\text{frk} \geq \text{acc}\left(\frac{\text{acc}}{q} - \frac{1}{h}\right). \tag{1}$$

2.4. Secure Multiparty Computation. Secure multiparty computation (MPC) enables a group to jointly perform a computation without disclosing any participant's private inputs. The participants agree on a function to compute and then can use an MPC protocol to jointly compute the output of that function on their secret inputs without revealing them [19]. There are several well-studied MPC protocols such as the GMW protocol [20] and the BGW protocol [21]. Both of the two schemes are based on the secret-sharing technique and can support both Boolean circuit and arithmetic circuit.

Here, we only present the general idea of a simple addition function to show how the protocols work. The basic idea is to allow each party holding the secret shares of the inputs; therefore, each party can locally sum up their shares and get a valid sharing of the final result. We describe it in a bit more detail in Figure 1.

3. SM2-Based Multisignature Scheme: $MS - SM2$

In this section, we present a multisignature scheme based on the SM2 signature in the plain public key model. Intuitively, the original signing algorithm of SM2 involves a nonlinear combination of secret key and randomness; therefore, it is nontrivial to extend it directly to a multisignature. To cope with the problem, in the protocol, we first exploit the linear part in SM2 to produce a semiaggregated signature and then employ a simple MPC protocol for addition to finally achieve the goal. Note that we slightly modify the output of original SM2 signing algorithm in protocol where we take the inverse of s instead to be the part of signature by each party. Therefore, the multisignature in our scheme is *almost* of the same structure as the original SM2 signature and remains practical. The unforgeability of the multisignature under chosen message attack can be proved in the random oracle model using general forking lemma [7, 16].

3.1. Construction. The initialization Setup algorithm and KeyGen algorithm of the multisignature are almost the same as that in the SM2 scheme, except that there are two hash functions used in multisignature scheme, denoted as $H_0\colon E(\mathbb{Z}_p) \longrightarrow \mathbb{Z}_n, H_1\colon \{0,1\}^* \longrightarrow \mathbb{Z}_n$. We now proceed to describe the signing protocol and verification algorithm of

Table 1: SM2 signature algorithm.

Key Generation	Signing	Verification
For user j, it generates	To sign message M, j computes	To verify $(M', \sigma\prime)$ with P_j,
sk: $d_j \xleftarrow{\$}$	1. $e = H(Z_j M)$	1. If $r', s' \notin [1, n-1]$,
pk: $P_j = [d_j]G$	2. $k \xleftarrow{\$}$	Return REJECT
Z_j: public hash bits of user	3. $(x_1, y_1) = [k]G$	2. $e' = H(Z_j M')$
	4. $r = (e + x_1) \pmod{n}$;	3. $t = r' + s' \,(\text{modn})$
	If $r = 0$ or $r + k = n$, go to Step 2	If $t = 0$, return REJECT
	5. $s = (1 + d_j)^{-1}(k - r \cdot d_j)(\text{mod}n)$;	4. $(x_1', y_1') = [s_1']G + [t]P_j$
j's information including pk	If $s = 0$, go to Step 2	5. If $r_1' = (e_1' + x_1')(\text{mod}n)$,
	6. Return signature $\sigma = (r, s)$	Return REJECT
		Else
		Return ACCEPT

```
Select random coins $\rho$ for $\mathscr{A}$
$(h_1, \ldots, h_q) \xleftarrow{\$}$
$(i, \sigma) \leftarrow \mathscr{A}(x, (h_1, \ldots, h_q); \rho)$;
if $i = 0$ then
return $(0, \varepsilon, \varepsilon)$;
end
$(h_{i\prime}, \ldots, h_{q\prime}) \xleftarrow{\$}$
$(i\prime, \sigma\prime) \leftarrow \mathscr{A}(x, h_1, \ldots, h_{i-1}, h_i', \ldots, h_q'; \rho)$;
if $(i = i\prime$ and $h_i \neq h_i')$ then
return $(1, \sigma, \sigma\prime)$
else
return $(0, \varepsilon, \varepsilon\prime)$
end
```

Algorithm 1: The forking algorithm Fork$^{\mathscr{A}}$.

PARAMETERS:

N : the number of parties;
x_i : the input of party P_i;
F : the function to compute (it is addition function here);

PROTOCOL:

1. Each party P_i creates N shares of input x_i using a (N, N)-secret sharing scheme, denote each share by $p_i(j)$.
2. P_i sends each share $p_i(j)$ $(j \in [1, \mathrm{N}], j \neq i)$ toP_j.
3. P_i computes $v_i = \sum_{j=1}^{\mathrm{N}} p_j(\mathrm{i})$. That is each party adds up all shares they received.
4. P_i broadcasts v_i to all other parties.
5. Each party computes $\mathrm{v} = \sum_{\mathrm{j}=1}^{\mathrm{N}} v_j$ to get the desired output.

Figure 1: The MPC protocol for addition $\mathscr{F}_{\text{add}}$.

the $MS - SM2$ scheme. Note that we take L to be size of N for simplicity, where N is the maximum number of co-signers and $N \ll n$.

MSign(pp, L, m, sk_i): each signer i with secret key $sk_i = d_i$ and public key $pk_i = P_i$ in set L runs an interactive protocol to collaboratively sign a message m. The communication proceeds in a number of rounds, where in each round, every signer sends and receives messages to and from other signers and also performs some local computation.

(1) Choose $k_i \xleftarrow{\$}$, compute $K_i(x_{i,1}, y_{i,1}) = [k_i]G$ and $t_i = H_0(x_{i,1}, y_{i,1})$, and broadcast t_i.

(2) Upon receiving t_j from all other signers, broadcast $K_i(x_{i,1}, y_{i,1})$.

(3) Upon receiving $(x_{j,1}, y_{j,1})$ from all other signers, check the hash values and abort the protocol if for any j that $t_j \neq H_0(x_{j,1}, y_{j,1})$. Otherwise, set $e_i = H_1(Z_i \| L \| i_{i=1}^{N} K_i \| m)$, $r_i = e_i + x_{i,1} \pmod{n}$, and $\widetilde{s}_i = (k_i - r_i \cdot d_i) \pmod{n}$. Then, broadcast $\widetilde{s}_i$.

(4) Upon receiving $\widetilde{s}_j$ from all other signers, compute $\widetilde{s} = \sum_{i=1}^{N} \widetilde{s}(\text{mod}n)$ and run the protocol for $\mathscr{F}_{\text{add}}$ with input $s_i = (1 + d_i) \cdot \widetilde{s}^{-1} \pmod{n}$ to get the addition $s = \sum_{i=1}^{N} s_i(\text{mod}n)$.

At the end the interactive protocol, the algorithm outputs a multisignature $\sigma = (K, s)$, where K is the set of all points $K_i(x_{i,1}, y_{i,1})$.

$\mathrm{Vrfy}(pp, L, m, \sigma)$: given a multiset of public keys L, message m, and multisignature σ, the verifier computes $e_i = H_1(Z_i \| L \| i_{i=1}^{N} K_i \| m)$ and $r_i = e_i + x_{i,1}\,(\mathrm{mod}\,n)$, accepts the signature if $[s]\oplus_{i=1}^{N} K_i - [s]\oplus_{i=1}^{N}([r_i]P_i) = [N]G + \oplus_{i=1}^{N} P_i$, and outputs 1. Otherwise, it outputs 0.

Correctness: if $\sigma = (K, s)$ is a valid output of protocol, Vrfy algorithm always accepts and outputs 1. The equation only holds when all signers follow the protocol and use valid key pairs. Note that the integer computations are all modulo n, and we omit the notation for simplicity.

$$\begin{aligned}
[s]\overset{N}{\underset{i=1}{\oplus}} K_i - [s]\overset{N}{\underset{i=1}{\oplus}}([r_i]P_i) &= \left[s\sum_{i=1}^{N} k_i\right]G - \left[s\sum_{i=1}^{N}(r_i \cdot d_i)\right]G \\
&= \left[s \cdot \left(\sum_{i=1}^{N} k_i - \sum_{i=1}^{N}(r_i \cdot d_i)\right)\right]G \\
&= \left[\sum_{i=1}^{N} s_i \cdot \sum_{i=1}^{N} \widetilde{s}_i\right]G = \left[\sum_{i=1}^{N}(1 + d_i) \cdot \widetilde{s}^{-1} \cdot \widetilde{s}\right]G \\
&= \left[\sum_{i=1}^{N}(1 + d_i)\right]G = [N]G + \overset{N}{\underset{i=1}{\oplus}} P_i.
\end{aligned} \tag{2}$$

3.2. Security Proof. In general, we can treat the multisignature scheme as a multiparty computation protocol and prove its security in simulation-based framework for a clearer security guarantee. Unfortunately, the security of multisignature is traditionally defined in game-based framework, and on the other hand, simulation-based proof is complex in the random oracle model. Here, we follow the game-based definition of Bellare and Neven [7] and only show a proof sketch for the scheme.

The basic idea of game-based proof is to obtain from $\mathscr{F}$ two different forgeries σ and $\sigma\prime$ with the same randomness by employing the general forking lemma. As a result, we can extract the secret key from the target public key pk^*, which is usually a solution of the discrete-logarithm problem in the elliptic-curve group $E(\mathbb{Z}_p)$. For simplification, we take an equivalent verification equation into consideration, and if $\sigma = (K, s)$ and $\sigma\prime = (K, s\prime)$ satisfy

$$\begin{aligned}
\overset{N}{\underset{i=1}{\oplus}} K_i - \overset{N}{\underset{i=1}{\oplus}}([r_i]P_i) = \left[s^{-1}N\right]G + \left[s^{-1}\right]\overset{N}{\underset{i=1}{\oplus}} P_i, \\
\overset{N}{\underset{i=1}{\oplus}} K_i - \overset{N}{\underset{i=1}{\oplus}}([r_i]P_i) = \left[s^{-1}N\right]G + \left[s^{-1}\right]\overset{N}{\underset{i=1}{\oplus}} P_i,
\end{aligned} \tag{3}$$

then the secret key d^* corresponding to pk^* can be computed from the equation

$$\sum_{i=1}^{N}[(r_i' - r_i)d_i] = \left(s^{-1} - s'^{-1}\right)\left(N - \sum_{i=1}^{N} d_i\right). \tag{4}$$

However, in the process of MSign, each signer can check the value $\widetilde{s}$ before continuing to execute the protocol, which allows signers to quit cosigning immediately if there is any rogue key attack. Specifically, they can compute $[x_1', y_1'] = [\widetilde{s}]G + \sum_{i=1}^{N}([r_i]P_i)$ and check if x_1' is equal to the corresponding part in the result.

$$\begin{aligned}
[\widetilde{s}]G + \sum_{i=1}^{N}([r_i]P_i) &= \left[\sum_{i=1}^{N} \widetilde{s}_i\right]G + \left[\sum_{i=1}^{N}(r_i \cdot d_i)\right]G \\
&= \left[\sum_{i=1}^{N}(k_i - r_i \cdot d_i)\right]G + \left[\sum_{i=1}^{N}(r_i \cdot d_i)\right]G \\
&= \left[\sum_{i=1}^{N} k_i\right]G = \oplus_{i=1}^{N}\left(x_{i,1}, y_{i,1}\right).
\end{aligned} \tag{5}$$

Therefore, we can let the simulator halt if the forger successfully forged $\widetilde{s}$.

Lemma 2. *If there exists a $(t, q_s, q_h, N, \varepsilon)$-forger $\mathscr{F}'$ that can output a forgery $\widetilde{s}$, then there exists a PPT algorithm $\mathscr{A}$ which (t', ε')-solves the DL problem in $E(\mathbb{Z}_p)$.*

Proof. Note that $\widetilde{s} = \sum_{i=1}^{N} \widetilde{s}_i\,(\mathrm{mod}\,n)$ and each $\widetilde{s}_i$ has similar structure with Schnorr signature. Therefore, the proof of Lemma 2 is similar to that of the $MS-BN$ scheme. Generally, given a $(t, q_s, q_h, N, \varepsilon)$-forger $\mathscr{F}'$, we first wrap it into an algorithm $\mathscr{B}$ that can be used in the general forking lemma. We then describe an algorithm $\mathscr{A}$ that on input $pk^* = P^*$ and runs $\mathrm{Fork}^{\mathscr{B}}(pk^*)$ to output the corresponding discrete logarithm. □

Let $q = q_h + q_s$, $T_0[\cdot], T_1[\cdot]$ be the programmed hash tables for oracles H_0 and H_1, respectively, and $h_1\{h_{1,1}, \ldots, h_{1,q}\}$ be the answers of queries to H_1. Two counters ctr_1 and ctr_2 are initialized to zero. An additional array $T_2[\cdot]$ records a unique index $1 \le i \le q_h + Nq_s$ to each public key P_i occurring either as a cosigner's public key in signature queries or H_1 queries, where $T_2[P^*] = 0$. On input $pp, h_1, P^* \in E(\mathbb{Z}_p)$, $\mathscr{B}$ plays the $\mathrm{Exp}_{\mathrm{MS}}^{\mathrm{UF-CMA}}(\mathscr{F})$ game with $\mathscr{F}'$ with the target public key $pk^* = P^*$. $\mathscr{B}$ answers queries from $\mathscr{F}'$ by programming the oracles as follows:

(i) $H_0(K_i)$: if $H_0(K_i)$ is undefined, then $\mathscr{B}$ randomly assigns $T_0[K_i] \overset{\$}{\leftarrow}$ and then returns $t_i = T_0[K_i]$.

(ii) $H_1(Z_i \| L \| i_{i=1}^{N} K_i \| m)$: if $T_2[P_i]$ is undefined, then $\mathscr{B}$ increments ctr_2 and sets $T_2[P_i] = ctr_2$. Let $k = T_2[P_i]$; if $T_1[k, L \| i_{i=1}^{N} K_i \| m]$ has not yet been defined, then $\mathscr{B}$ assigns random values to all $T_1[j, L \| i_{i=1}^{N} K_i \| m]$ for $1 \le j \le q_h + Nq_s$, increases ctr_1, and assigns $T_1[0, L \| \oplus_{i=1}^{N} K_i \| m] = h_{1,\mathrm{ctr}_1}$.

(iii) $\mathscr{O}^{\mathrm{sign}}(L, m)$: if $P^* \notin L$, then $\mathscr{B}$ returns $\perp$ to the forger. Otherwise, it parses L as $\{P^*, P_2, \ldots, P_N\}$. $\mathscr{B}$ first checks whether $T_2[P_i]\,(2 \le i \le N)$ has already been defined, if not it increases ctr_2 and sets $T_2[P_i] = \mathrm{ctr}_2$. Then, it increases counter ctr_1 and sets $e_1 = h_{1,ctr_1}$. It chooses $\widetilde{s}_1 \overset{\$}{\leftarrow}$ and computes a elliptic curve point K_1 such that $K_1(x_{1,1}, y_{1,1}) = [\widetilde{s}_1]G + [r_1]P^*$, where $r_1 = h_{1,ctr_1} + x_{1,1}$. It finally sends $t_1 = H_0(K_1)$ to all cosigners. After receiving all t_j from $\mathscr{F}'$ (all other cosigners),

$\mathscr{B}$ looks up the corresponding K_j in table T_0 such that $t_j = T_0[K_j]$. If not all such values can be found, $\mathscr{B}$ randomly chooses $K_1' \overset{\$}{\leftarrow}$ and broadcasts K_1'. If there exists $K_{j\prime} \neq K_j$ such that $T_0[K_{j'}] = T_0[K_j]$, then $\mathscr{B}$ sets $\text{bad}_1 = \text{true}$ and aborts the execution of $\mathscr{F}\prime$ by outputting $(0, \perp)$. Otherwise, $\mathscr{B}$ computes $K^* = i_{i=1}^{N} K_i$ and checks whether $T_1[0, L \| K^* \| m]$ has already been defined. If the entry was taken, $\mathscr{B}$ sets $\text{bad}_2 = \text{true}$ and aborts the execution by outputting $(0, \perp)$. If not, $\mathscr{B}$ sets $T_1[0, L \| K^* \| m] = e_1$ and broadcasts K_1. Upon receiving all K_j, $\mathscr{B}$ stops the process if for any $2 \leq j \leq N$ such that $H_0(K_j) \neq t_j$. $\mathscr{B}$ then broadcasts $\widetilde{s_1}$.

Finally, if $\mathscr{F}'$ outputs a valid forgery $(K, \widetilde{s})$ on message m under the signer list L, then $\mathscr{B}$ checks $T_1[0, L \| i_{i=1}^{N} K_i \| m]$. Let J be the index that $h_{1,J} = T_1[0, L \| i_{i=1}^{N} K_i \| m]$. $\mathscr{B}$ returns $(J, (K, h_{1,J}, \widetilde{s}, L))$. The accepting probability of $\mathscr{B}$ is as follows:

$$\left.\begin{aligned} acc_{\mathscr{B}} &= \Pr\left[\mathscr{F}' \text{succeeds} \wedge \overline{\text{bad}_1} \wedge \overline{\text{bad}_2}\right] \\ &\geq \Pr[\mathscr{F}' \text{succeeds}] - \Pr\left[\overline{\text{bad}_1}\right] - \Pr\left[\overline{\text{bad}_2}\right] \\ &\geq \varepsilon - \frac{(q_h + Nq_s + 1)^2}{2n} - \frac{2q_s(q_h + Nq_s)}{n} \end{aligned}\right]. \tag{6}$$

We then construct the algorithm $\mathscr{A}$ that on input $pk^* = P^*$ and runs $\text{Fork}^{\mathscr{B}}(pk^*)$. According to the general forking lemma, it returns $(1, (K, h_{1,J}, \widetilde{s}, L), (K, h_{1,J'}, \widetilde{s'}, L))$ with probability $frk_{\mathscr{A}}$. Note that the discrete logarithm with regard to P^* can be computed through $(K, h_{1,J}, \widetilde{s}, L)$, $(K, h_{1,J'}, \widetilde{s'}, L)$. Therefore, the probability $\varepsilon\prime$ is as follows:

$$\left.\begin{aligned} \varepsilon' &\geq \text{frk}_{\mathscr{A}} \\ &\geq \text{acc}_{\mathscr{B}}\left(\frac{\text{acc}_{\mathscr{B}}}{q_h + q_s} - \frac{1}{n}\right) \\ &\geq \frac{\varepsilon^2}{q_h + q_s} - \frac{4q_s(q_h + Nq_s + 1)^2}{n(q_h + q_s)} \end{aligned}\right]. \tag{7}$$

3.3. Experimental Results. We now present the concrete experimental results based on our implementation. We implemented the $MS - SM2$ scheme in Java and ran it on an EC2 instance of type CPU 2.50 GHz with 1 GB RAM. We use the standard SM2 curve and the SM3 hash algorithm. We ran experiments from 2 to 20 parties and compare our results in two-party setting with a related protocol from Zhang et al. [22] in Table 2. Note that [22] is an SM2-based two-party distributed signing protocol, which is slightly different from multisignature in the way that parties should also cooperate in key generation. Moreover, they omit the zero-knowledge proof component in their implementation, and our demo (https://github.com/lhoou/ms-sm2) as a simulation only includes local computation and omits the communication cost in real world. As for multiuser setting, the performances of our scheme are presented in Table 3.

4. Applications to Vehicular Networks

In this section, we describe two potential applications of $MS - SM2$ to vehicular networks. We first show that it can be employed in the architecture of multiple certificate authorities to reduce the number of certificates that are required for devices in the system including on-board units (OBU) and road-side units (RSU). In addition, we also specify its possible usage in the process of V2I communications. The goal is to reduce computation and storage overhead for the units while maintaining security properties.

4.1. Multi-CA Architecture. In vehicular networks, taking C-V2X, for example, certificate authorities usually include organizations for registration, communication authorization, and pseudonym authorization. Specifically, any device that is involved in the network should first require for registration certificate from registration CA and then require for other certificates from different CAs that are needed to send and receive messages in the network.

For instance, a vehicle is required to get a certificate from the registration CA using its unique identity before joining the network. It can then require a pseudonym certificate for the anonymous V2V communication and a secure V2I communication certificate from secure communication CA using its registration certificate. The vehicle can also apply multiple registration certificates from different registration CAs. To simplify the authentication process, the distributed CAs can employ $MS - SM2$ in order to jointly generate only one certificate or one registration certificate for the vehicle at the same time, instead of generating certificates one by one.

4.2. Cooperative V2I Communication. Cooperative communication in vehicular networks has been leveraged to offer various improvements on spectral efficiency, transmission reliability, and reduced transmission delay. Vehicles can cooperate with each other either directly or through an RSU, and the vehicular node which helps the source node to transmit its data is called a helper node or relay node [23].

(i) Cooperative traffic reports: vehicles in the same traffic area, such as in an accident or in a neighborhood, can cooperatively issue a traffic report including awareness messages (CAMs), safety importance, and vehicle heading and transmit a packet to the RSU attached with a $MS - SM2$ signature. The $MS - SM2$ signature can help the RSU to check validity of the packet and also reduce the computation cost of RSU.

(ii) RSU-assisted communication: when a source RSU fails to successfully transmit a packet to the targeted destination, it forwards the packet to the next RSU along the path using the backhaul wired connection. The new RSU relays the received packet to the targeted destination. In this scenario, both the source RSU and relayed RSU can jointly sign the packet

Table 2: Comparison of performances (in milliseconds) between [22] and our scheme in two-party setting.

Scheme	Key Generation	Signing	Verification
Ref. [22]	123.44 ms	152.34 ms	4.17 ms
Ours	51.16 ms	18.04 ms	10.20 ms

Table 3: The performances of our protocol in multiparty setting.

Number of parties	Signature Length (compressed)	Signing (local computation)	Verification
$N = 5$	192 B	13.648 ms	20.916 ms
$N = 10$	352 B	10.021 ms	44.191 ms
$N = 20$	674 B	9.646 ms	49.399 ms

using $MS - SM2$ to convince the target vehicle of the message transmitted, which can also prevent any malicious RSU from sending out frauds without collusion.

5. Conclusions

In this paper, we present a candidate multisignature scheme from the SM2 signature algorithm in the plain public-key model. Compared to a list of individual signatures, the storage volume of $MS - SM2$ signature reduces nearly 50% and the computation cost is relatively low. In addition, we specify in detail some potential applications of the $MS - SM2$ scheme to vehicular networks, especially in the scenario of cooperatively secure communication, with the goal of maximizing performance and compatibility. Because of the high-speed mobility, designing more efficient protocols with fewer communication rounds for vehicular networks is still a challenging research problem.

Data Availability

The data, including algorithms and proofs, used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported in part by the National Key Research and Development Plan of China (Grant no. 2020YFB1005600), Key Area Research and Development Program of Guangdong Province (Grant nos. 2020B0101360001 and 2020B0101090004), National Natural Science Foundation of China (Grant nos. 61825203, U1736203, 61732021, and 61902067), Major Program of Guangdong Basic and Applied Research Project (2019B030302008), and Foundation for Young Innovative Talents in Ordinary Universities of Guangdong (2018KQNCX255).

References

[1] N. Xia and C.-S. Yang, "Vehicular communications: standards and challenges," in *Lecture Notes in Computer Science*, S.-L. Peng, G. Lee, R. Klette, and C.-H. Hsu, Eds., in *Proceedings of the Internet of Vehicles. Technologies and Services for Smart Cities - 4th International Conference, IOV 2017*, vol. 10689, pp. 1–12, Springer, Kanazawa, Japan, November, 2017.

[2] L. Tuyisenge, M. Ayaida, S. Tohmé, and L.-E. Afilal, "Network architectures in internet of vehicles (iov): review, protocols analysis, challenges and issues," in *Proceedings of the Internet of Vehicles. Technologies and Services Towards Smart City - 5th International Conference, IOV 2018*, A. M. J. Skulimowski, Z. Sheng, S. Khemiri-Kallel, C. Cérin, and C.-H. Hsu, Eds., vol. 11253, pp. 3–13, Springer, Paris, France, November, 2018.

[3] A. Yang, J. Weng, N. Cheng, J. Ni, X. Lin, and X. Shen, "Deqos attack: degrading quality of service in vanets and its mitigation," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4834–4845, 2019.

[4] A. Yang, J. Weng, K. Yang, C. Huang, and X. Shen, "Delegating authentication to edge: a decentralized authentication architecture for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[5] C. Pedro, A. Zúquete, S. Sargento, and M. Luís, "The impact of ECDSA in a VANET routing service: insights from real data traces," *Ad Hoc Networks*, vol. 90, 2019.

[6] S. Micali, K. Ohta, and L. Reyzin, "Accountable-subgroup multisignatures: extended abstract," in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pp. 245–254, Philadelphia, PA, USA, November, 2001.

[7] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pp. 390–399, Alexandria, VA, USA, November, 2006.

[8] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proceedings of the Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques*, E. Biham, Ed., vol. 2656, pp. 416–432, Springer, Warsaw, Poland, May, 2003.

[9] Y. Zhao, "Practical aggregate signature from general elliptic curves, and applications to blockchain," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS'19)*, pp. 529–538, New York, NY, USA, 2019.

[10] K. Itakura, "A public-key cryptosystem suitable for digital multisignature," *NEC Research and Development*, vol. 71, pp. 1–8, 1983.

[11] B. Ali, J.H. Cheon, and S. Jarecki, "Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma," in *Proceedings of the 2008 ACM Conference*

on Computer and Communications Security, CCS 2008, pp. 449–458, Alexandria, VA, USA, October, 2008.

[12] C. Ma, J. Weng, Y. Li, R. Deng, and Deng, "Efficient discrete logarithm based multi-signature scheme in the plain public key model," *Designs, Codes and Cryptography*, vol. 54, no. 2, pp. 121–133, 2010.

[13] E. Syta, I. Tamas, D. Visher et al., "Keeping authorities "honest or bust" with decentralized witness cosigning," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 526–545, SP, San Jose, CA, USA, May, 2016.

[14] E. Syta, P. Jovanovic, E. Kokoris-Kogias et al., "Scalable bias-resistant distributed randomness," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy, SP 2017*, pp. 444–460, San Jose, CA, USA, May, 2017.

[15] M. Drijvers, K. Edalatnejad, B. Ford et al., "On the security of two-round multi-signatures," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy, SP 2019*, pp. 1084–1101, IEEE, San Francisco, CA, USA, May 19-23, 2019.

[16] G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille, "Simple schnorr multi-signatures with applications to bitcoin," *IACR Cryptology ePrint Archive*, vol. 68, 2018.

[17] D. Boneh, M. Drijvers, and G. Neven, "Compact multi-signatures for smaller blockchains," *Lecture Notes in Computer Science*, in *Proceedings of the Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 435–464, Brisbane, Australia, December, 2018.

[18] D. Pointcheval and J. Stern, "Security proofs for signature schemes," *Advances in Cryptology—EUROCRYPT '96*, vol. 1070, pp. 387–398, 1996.

[19] D. Evans, V. Kolesnikov, and M. Rosulek, "A pragmatic introduction to secure multi-party computation," *Foundations Trends Privacyand Security*, vol. 2, no. 2-3, pp. 70–246, 2018.

[20] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or A completeness theorem for protocols with honest majority," in *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (STOC)*, New York, NY, USA, January 1987.

[21] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract)," in *STOC*ACM, New York, NY, USA, 1988.

[22] Y. Zhang, D. He, M. Zhang, and K.-K. R. Choo, "A provable-secure and practical two-party distributed signing protocol for SM2 signature algorithm," *Frontiers of Computer Science*, vol. 14, no. 3, p. 143803, 2020.

[23] E. Ahmed and H. Gharavi, "Cooperative vehicular networking: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 996–1014, 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9463606, 10 pages
https://doi.org/10.1155/2021/9463606

Research Article

A Certificateless Pairing-Free Authentication Scheme for Unmanned Aerial Vehicle Networks

Jingyi Li,[1] Yujue Wang,[2] Yong Ding ,[1,3] Wanqing Wu,[4] Chunhai Li,[5] and Huiyong Wang[6]

[1] *Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China*
[2] *Hangzhou Innovation Institute, Beihang University, Hangzhou, China*
[3] *Cyberspace Security Research Center, Pengcheng Laboratory, Shenzhen, China*
[4] *School of Cyber Security and Computer, Hebei University, Baoding, China*
[5] *School of Information and Communication, Guilin University of Electronic Technology, Guilin, China*
[6] *School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin, China*

Correspondence should be addressed to Yong Ding; stone_dingy@126.com

Received 11 June 2021; Revised 4 August 2021; Accepted 14 August 2021; Published 11 September 2021

Academic Editor: Anjia Yang

In unmanned aerial vehicle networks (UAVNs), unmanned aerial vehicles with restricted computing and communication capabilities can perform tasks in collaborative manner. However, communications in UAVN confront many security issues, for example, malicious entities may launch impersonate attacks. In UAVN, the command center (CMC) needs to perform mutual authentication with unmanned aerial vehicles in clusters. The aggregator (AGT) can verify the authenticity of authentication request from CMC; then, the attested authentication request is broadcasted to the reconnaissance unmanned aerial vehicle (UAV) in the same cluster. The authentication responses from UAVs can be verified and aggregated by AGT before being sent to CMC for validation. Also, existing solutions cannot resist malicious key generation center (KGC). To address these issues, this paper proposes a pairing-free authentication scheme (CLAS) for UAVNs based on the certificateless signature technology, which supports batch verification at both AGT and CMC sides so that the verification efficiency can be improved greatly. Security analysis shows that our CLAS scheme can guarantee the unforgeability for (attested) authentication request and (aggregate) responses in all phases. Performance analysis indicates that our CLAS scheme enjoys practical efficiency.

1. Introduction

Unmanned aerial vehicles in UAVN have been widely used in many civilian and military fields, for example, data collection, communication relay, and military electronic reconnaissance [1]. Unmanned aerial vehicles can be classified into three categories according to the working mode, namely, unmanned aerial vehicles under the control of a remote operator, under the supervision of a remote supervisor, and without an operator and supervisor. UAVNs can be deployed in mesh topology or multistar topology [2]. With the mesh topology, all unmanned aerial vehicles are connected to CMC directly, where all communication between unmanned aerial vehicles and CMC may cause network congestion. Although with the mesh topology, each unmanned aerial vehicle can communicate with each other, it is hard to be expanded and controlled [3]. With the multistar topology, each unmanned aerial vehicle is connected to CMC; thus, any illegal requests or responses in UAVNs can be easily detected.

However, when deployed in an open communication environment, the UAVN system confronts many security issues [4, 5]. Due to multiple connections among unmanned aerial vehicles, a malicious entity may control some unmanned aerial vehicle or launch impersonate attacks. Thus, it is important to enforce a secure and efficient authentication mechanism in UAVNs [6, 7]. Recently, Wang et al. [8] proposed an identity-based authentication scheme, which did not consider the verification mechanisms at the AGT side for validating the real sources of the authentication request from CMC and responses from UAVs. Li et al. [9] designed an identity-based aggregate authentication framework in bilinear groups, where the private keys of UAVs are generated by KGC. Thus, malicious KGC may launch attacks by sending illegal authentication request to AGTs and UAVs.

1.1. Our Contributions. To address the abovementioned issues, this paper proposes a certificateless pairing-free aggregate authentication scheme (CLAS) for UAVNs. In CLAS, KGC is responsible for generating partial private keys for all entities including CMC, AGTs, and UAVs. Each AGT acts as the cluster head of some cluster and plays the role of an intermediate between CMC and UAVs in the respective cluster. Each authentication request from CMC can be validated by AGT, which is then attested and broadcasted to UAVs in its administrative domain. A verification process can be run by each UAV so that the true source of the (forwarded) authentication request can be validated. AGT can aggregate all responses of UAVs in its administrative cluster before performing verification procedure in batch. Then, the response of AGT is further combined with the aggregated responses of UAVs, which can be validated by CMC in batch to complete the authentication process.

This paper describes a concrete CLAS construction based on the certificateless signature technology. Security analysis shows that our CLAS construction can protect malicious entity from forging the authentication request and responses of others and can resist against the malicious KGC. Performance comparison shows that our CLAS construction enjoys better computational efficiency compared with Wang et al.'s scheme [8] and Li et al.'s scheme [9].

1.2. Related Works. Taking advantages of recent advancement and development in information and communication technology, unmanned aerial vehicles have been employed to perform some special tasks in real-world applications [10]. In [11], Islam and Shin proposed a blockchain-based solution for safe healthcare, which uses the unmanned aerial vehicle (UAV) to collect health data (HD) from users. Liu et al. [1] presented a detailed survey on the opportunities and challenges of IoE supported by unmanned aerial vehicles. Jiang et al. [12] proposed a trust-based energy efficient data collection with the unmanned aerial vehicle (TEEDC-UAV) scheme, which can prolong lifetime in a trusted way. In the TEEDC-UAV scheme, an ant colony-based unmanned aerial vehicle (UAV) trajectory optimization algorithm was proposed, which constituted the most data anchor points in the working field with the shortest trajectory possible. In view of the untrusted broadcast features and wireless transmission of UAV networks, a novel privacy-preserving secure spectrum trading and sharing scheme based on blockchain technology is proposed in [13].

For the Internet of Drones (IoD) infrastructure, Cho et al. [14] proposed a framework called SENTINEL (Secure and Efficient autheNTIcation for uNmanned aErial vehicLes). Khanh et al. [15] presented a safe and effective authentication mechanism suitable for the dynamic environment of the unmanned aerial vehicle. In order to solve the information security problem of unmanned aerial vehicle ad-hoc network communication, Sun et al. [2] introduced an efficient and energy-saving distributed network architecture based on clustering stratification. Owing to the unreliable wireless channel and high-dynamic topology of Unmanned Aerial Vehicles Ad-Hoc Network (UAANET), the loss of some certain group key broadcast messages by nodes occurs frequently. Therefore, Li et al. [16] proposed a mutual-healing group key distribution scheme based on the blockchain. Yang et al. [17] investigated degradation-of-QoS attacks in vehicular ad hoc networks, where the attacker is able to relay the authentication exchanges but cannot relay the service afterwards. In [18], Gope et al. proposed a novel anonymous authentication scheme for RFID-enabled UAV applications using Physically Unclonable Functions (PUF).

Al-Riyami et al. [19] first introduced and made concrete the concept of certificateless public key cryptography (CL-PKC), a model for the use of public key cryptography which avoids the inherent escrow of identity-based cryptography. Baek et al. [20] considered a relaxation of the original model of CLPKE and proposed a new CLPKE scheme that does not depend on the bilinear pairings. In order to ensure security for interactions between these smart things, Yeh et al. [21] presented a certificateless signature scheme for smart objects in IoT-based pervasive computing environments. Jia et al. [22] made an improvement on the scheme of Yeh et al.'s certificateless signature scheme; they presented an improved scheme and demonstrated its unforgeability against super-adversaries in the random oracle model. Zhao et al. [23] presented an advanced efficient CLAS scheme with elliptic curve cryptography for the IoV environment. Furthermore, their scheme used pseudonyms in communications to prevent vehicles from revealing their identity. Shu et al. [24] presented a certificateless aggregate signature scheme for blockchain-based MCPS, which can realize the authentication of related medical staffs, medical equipment, and medical apps, ensure the integrity of medical records, and support the secure storage and sharing of medical information.

1.3. Paper Organization. The structure of this paper is organized as follows. In Section 2, we introduce the system architecture and system requirements for CLAS. A concrete CLAS construction is presented in Section 3, followed by its security and efficiency analysis in Section 4. Finally, Section 5 concludes the paper.

2. System Architecture and Requirements

This section formalizes the architecture of CLAS and summarizes its system requirements.

2.1. System Architecture. As shown in Figure 1, there are four types of entities in a CLAS system, namely, key generation center (KGC), command center (CMC), reconnaissance

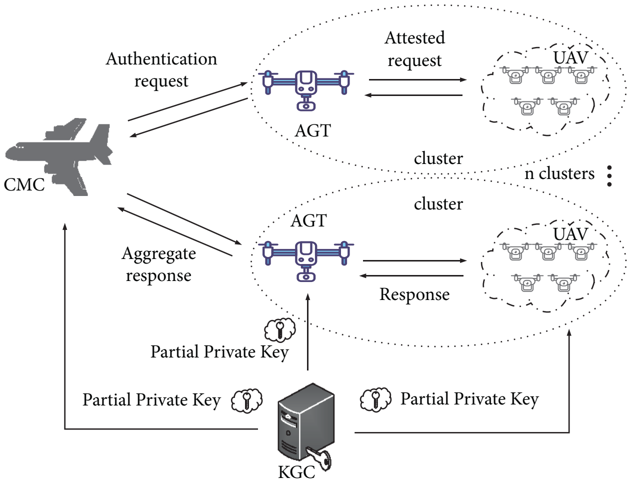


Figure 1: CLAS model for UAVNs.

unmanned aerial vehicles (UAVs), and aggregators (AGTs). KGC is assumed to be fully trusted by all the entities, which is responsible for initializing the CLAS system by generating system public parameters and producing partial private keys for all entities in UAVNs. After system initialization, CMC performs the mutual authentication process with unmanned aerial vehicles before assigning tasks. CMC initializes the authentication process so that AGT can validate, attest, and broadcast authentication request to its administrated UAVs.

As the intermediary between CMC and UAV, AGT has the computing and communication capabilities to manage its UAV cluster. UAV only has limited short-distance communication capability; thus, its communication with CMC is performed via the AGT in the cluster. Before responding to the authentication request of CMC, each UAV can verify its true source and the attested request. The responses of UAVs in the same cluster can be validated by AGT in batch. Then, the response of AGT can be further combined with that of UAVs so that the aggregated response is sent to CMC for validation.

2.2. System Requirements. Similar to [25], we define two types of adversaries for the CLAS system, namely, Type-I adversary and Type-II adversary. A Type-I adversary acts as an outsider who can replace the public keys of CMC, AGT, and UAV but cannot access the master secret key, whereas a Type-II adversary acts as the KGC that can access the master secret key but cannot replace the public keys of CMC, AGT, and UAV. A CLAS system must satisfy the following system requirements.

> Unforgeability of authentication request: in the authentication process, for the authentication request generated by CMC, it should be guaranteed that it is existentially unforgeable against Type-I adversary. That is, any entity cannot launch attacks by impersonating CMC to forge an authentication request.
>
> Unforgeability of attested request: for the attested authentication request of AGT, it should be guaranteed that it is existentially unforgeable against Type-I adversary. That is, any entity cannot launch attacks by impersonating AGT to forge an attested authentication request.
>
> Unforgeability of response: for the responses from UAVs in its administrative cluster of AGT, it should be guaranteed that it is existentially unforgeable against Type-I adversary. That is, any entity cannot launch attacks by impersonating some UAV to forge a response.
>
> Unforgeability of aggregate response: for the attested authentication request of some AGT, it should be guaranteed that it is existentially unforgeable against Type-I adversary. That is, any entity cannot launch attacks by impersonating AGT to forge an aggregate response.
>
> Resistance against malicious KGC: for the whole authentication procedure, it should be guaranteed that it is existentially unforgeable against Type-II adversary. That is, malicious KGC cannot forge a valid signature of CMC, AGT, or UAV.

A *correct* CLAS construction should satisfy the following conditions:

(1) For the partial private key sent by KGC, it can be successfully verified by respective entity including CMC, AGTs, and UAVs

(2) For the authentication request generated by CMC, it can be successfully validated by AGTs

(3) For the attested authentication request forwarded by AGT, it can be successfully validated by UAVs in the same cluster

(4) For the responses of UAVs, they can be validated by AGT in the same cluster

(5) For the aggregate response from AGT, it can be successfully validated by CMC

3. CLAS Construction

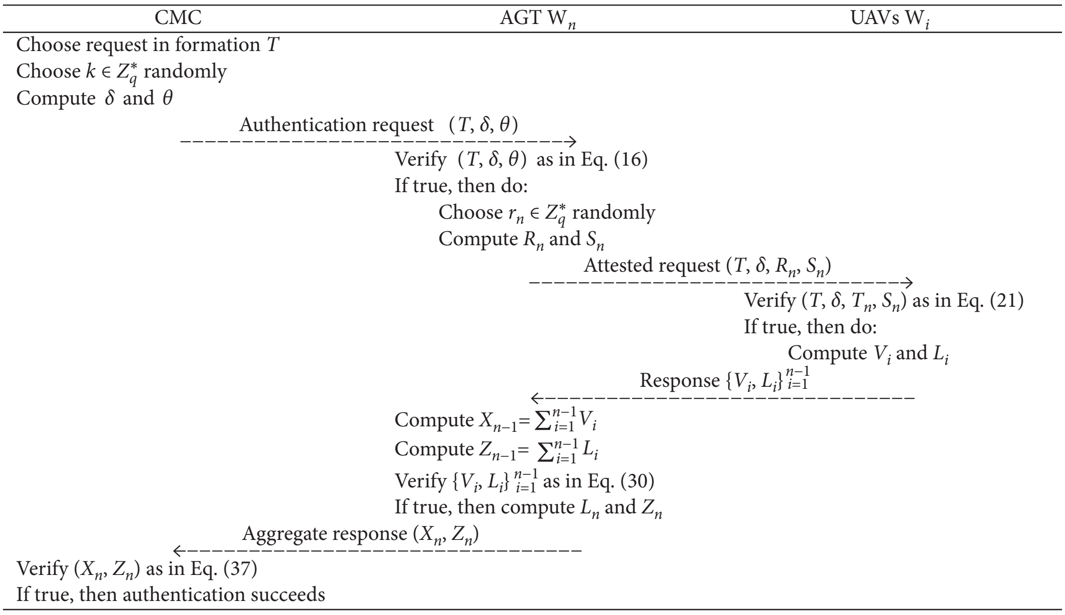
This section describes our concrete CLAS construction. The authentication process in UAVNs is shown in Figure 2.

The Discrete Logarithm Assumption in Elliptic Curve (ECDLP): let G be an elliptic curve group with prime order q. Given P and $Q \in G$, any probabilistic polynomial time algorithm ξ would have negligible probability in computing $x \in Z_q^*$ such that $Q = xP$.

3.1. System Setup. On inputting a security parameter $l \in Z^+$, KGC chooses an additive group G with prime order q on some elliptic curve, where P is a generator of G. Then, KGC chooses $b \in Z_q^*$ randomly and computes

$$B = bP. \quad (1)$$

KGC continues to choose four collision-resistant hash functions H_i: $\{0,1\}^* \longrightarrow Z_q^*$ for $i = 1, 2, 3,$ and 4. Finally, KGC publishes the system parameters



CMC | AGT W_n | UAVs W_i

Choose request in formation T
Choose $k \in Z_q^*$ randomly
Compute δ and θ

Authentication request (T, δ, θ)

Verify (T, δ, θ) as in Eq. (16)
If true, then do:
Choose $r_n \in Z_q^*$ randomly
Compute R_n and S_n

Attested request (T, δ, R_n, S_n)

Verify (T, δ, T_n, S_n) as in Eq. (21)
If true, then do:
Compute V_i and L_i

Response $\{V_i, L_i\}_{i=1}^{n-1}$

Compute $X_{n-1} = \sum_{i=1}^{n-1} V_i$
Compute $Z_{n-1} = \sum_{i=1}^{n-1} L_i$
Verify $\{V_i, L_i\}_{i=1}^{n-1}$ as in Eq. (30)
If true, then compute L_n and Z_n

Aggregate response (X_n, Z_n)

Verify (X_n, Z_n) as in Eq. (37)
If true, then authentication succeeds

Figure 2: A procedure of authentication in our CLAS construction.

params $= (q, G, P, H_1, H_2, H_3, H_4, B)$ and keeps the master secret key msk $= b$ secret.

3.2. Key Generation for CMC. KGC sets the partial private key for the control center as follows. KGC chooses a random number $e \in Z_q^*$ and computes

$$A = eP, \tag{2}$$

$$a = e + bH_1\,(\text{CMC}\|A\|B)\text{mod}q. \tag{3}$$

Then, KGC sends the partial private key (a, A) to CMC through a secure channel. CMC can validate the partial private key as follows:

$$\text{aP} \overset{?}{=} A + H_1\,(\text{CMC}\|A\|B)B. \tag{4}$$

CMC sets a secret value and generates its public key PK_c and private key SK_c as follows. CMC chooses a random number $s \in Z_q^*$ and computes

$$F = sP, \tag{5}$$

$$M = A + H_2\,(CMC\|F)F. \tag{6}$$

Then, CMC sets $PK_c = (A, M)$ and $SK_c = (a, s)$.

3.3. Key Generation for Unmanned Aerial Vehicles. Let W_i be an unmanned aerial vehicle. For the ease of representation, let W_n be an AGT and $W_1, \ldots, W_{n-1}$ be UAVs in the administration domain of W_n. KGC sets a partial private key for unmanned aerial vehicles as follows.

KGC chooses a random number $d_i \in Z_q^*$ and computes

$$Y_i = d_i P, \tag{7}$$

$$y_i = d_i + bh_{1,i}\text{mod}q, \tag{8}$$

where

$$h_{1,i} = H_1\big(W_i\|Y_i\|B\big). \tag{9}$$

Then, KGC sends the partial private key (y_i, Y_i) to W_i through a secure channel. The unmanned aerial vehicle W_i can validate the partial private key as follows:

$$y_i P \overset{?}{=} Y_i + h_{1,i}B. \tag{10}$$

The unmanned aerial vehicle W_i sets a secret value and generates its public key PK_i and private key SK_i as follows. W_i chooses a random number $c_i \in Z_q^*$ and computes

$$C_i = c_i P, \tag{11}$$

$$Q_i = Y_i + h_{2,i}C_i, \tag{12}$$

where

$$h_{2,i} = H_2\,(W_i\|C_i). \tag{13}$$

Then, the unmanned aerial vehicle W_i sets $PK_i = (Y_i, Q_i)$ and $SK_i = (y_i, c_i)$.

3.4. Authentication Request. Let $T \in \{0,1\}^*$ denote the request information chosen by CMC, which contains the timestamp. CMC randomly picks $k \in Z_q^*$ and computes

$$\delta = kP, \tag{14}$$

$$\theta = k + H_3\,(T\|\delta\|\text{CMC}\|PK_c)\,(a + H_2\,(\text{CMC}\|F)s)\text{mod}\,q. \tag{15}$$

Then, CMC sends the authentication request (T, δ, θ) to AGTs.

3.5. *Request Forwarding.*

After receiving the request (T, δ, θ) from CMC, each AGT W_n validates its authenticity by checking the following equality:

$$\theta P \overset{?}{=} \delta + H_3 (T \| \delta \| \mathrm{CMC} \| PK_c)(M + H_1 (\mathrm{CMC} \| A \| B) B). \tag{16}$$

If it holds, then AGT W_n accepts the authentication request from CMC, otherwise terminates. AGT W_n randomly chooses $r_n \in Z_q^*$ and computes

$$R_n = r_n P, \tag{17}$$

$$S_n = \theta + r_n + h_{4,n}(y_n + h_{2,n} c_n) \bmod q, \tag{18}$$

where

$$h_{4,n} = H_4 (T \| \delta \| W_n \| PK_n \| R_n), \tag{19}$$

$$h_{2,n} = H_2 (W_n \| C_n). \tag{20}$$

At last, AGT W_n broadcasts the tuple of attested authentication request (T, δ, R_n, S_n) to all UAVs $W_i\,(i = 1, 2, \ldots, n-1)$ in its administrative domain.

3.6. *UAV Response.*

Once received (T, δ, R_n, S_n) from AGT W_n, each UAV $W_i\,(i = 1, 2, \ldots, n-1)$ verifies its authenticity by checking the following equality:

$$S_n P \overset{?}{=} \delta + R_n + h_{4,n}(Q_n + h_{1,n} B) + H_3 (T \| \delta \| \mathrm{CMC} \| PK_c) \cdot (M + H_1 (\mathrm{CMC} \| A \|) B), \tag{21}$$

where

$$h_{1,n} = H_1(W_n \| Y_n \| B), \tag{22}$$

$$h_{4,n} = H_4(T \| \delta \| W_n \| PK_n \| R_n). \tag{23}$$

If it holds, then UAV W_i accepts the authentication request from CMC, otherwise terminates. W_i randomly picks $f_i \in Z_q^*$ and computes

$$V_i = f_i P, \tag{24}$$

$$L_i = f_i + \widehat{h}_{4,i}(y_i + h_{2,i} c_i) \bmod q, \tag{25}$$

where

$$\widehat{h}_{4,i} = H_4(T \| \delta \| W_i \| PK_i \| V_i), \tag{26}$$

$$h_{2,i} = H_2 (W_i \| C_i). \tag{27}$$

Then, UAV W_i sends the response tuple $\sigma_i = (V_i, L_i)$ to AGT W_n.

3.7. *AGT Aggregation.*

Upon receiving the response tuples $\{V_i, L_i\}_{i=1}^{n-1}$ from the controlled UAVs $W_i\,(i = 1, 2, \ldots, n-1)$, AGT W_n computes

$$X_{n-1} = \sum_{i=1}^{n-1} V_i, \tag{28}$$

$$Z_{n-1} = \sum_{i=1}^{n-1} L_i \bmod q. \tag{29}$$

Then, AGT W_n verifies the authenticity of the received response tuples in a batch as follows:

$$Z_{n-1} P \overset{?}{=} X_{n-1} + \left(\sum_{i=1}^{n-1} \widehat{h}_{4,i} h_{1,i} \right) B + \sum_{i=1}^{n-1} \widehat{h}_{4,i} Q_i, \tag{30}$$

where

$$\widehat{h}_{4,i} = H_4(T \| \delta \| W_i \| PK_i \| V_i), \tag{31}$$

$$h_{1,i} = H_1(W_i \| Y_i \| B). \tag{32}$$

If it holds, then all response tuples of $W_i\,(i = 1, 2, \ldots, n-1)$ are valid; otherwise, W_n validates each response tuple in individual to find the invalid one. AGT W_n continues to pick a random element $f_n \in Z_q^*$ and compute

$$X_n = X_{n-1} + f_n P, \tag{33}$$

$$Z_n = Z_{n-1} + L_n \bmod q, \tag{34}$$

where

$$\begin{aligned} L_n &= f_n + \widehat{h}_{4,n}(y_n + h_{2,n} c_n) \bmod q, \\ \widehat{h}_{4,n} &= H_4(T \| \delta \| W_n \| PK_n \| f_n P), \end{aligned} \tag{35}$$

$$h_{2,n} = H_2 (W_n \| C_n). \tag{36}$$

Then, AGT W_n sends the aggregate response (X_n, Z_n) to CMC.

3.8. *CMC Verification.*

Once received the aggregate response (X_n, Z_n) from AGT W_n, CMC validates its authenticity by checking the following equality:

$$Z_n P \overset{?}{=} X_n + \left(\sum_{i=1}^{n} \widehat{h}_{4,i} h_{1,i} \right) B + \sum_{i=1}^{n} \widehat{h}_{4,i} Q_i, \tag{37}$$

where

$$\widehat{h}_{4,i} = H_4 (T \| \delta \| W_i \| PK_i \| V_i), \tag{38}$$

$$h_{1,i} = H_1(W_i \| Y_i \| B). \tag{39}$$

If it holds, then AGT W_n and UAVs $W_i\,(i = 1, 2, \ldots, n-1)$ are all accepted as legitimate.

Theorem 1. *The proposed CLAS construction is correct.*

Proof 1. To prove the correctness of the proposed CLAS construction, it only needs to show that equalities (16), (21), (30), and (37) are satisfied.

(1) For the authentication request (T, δ, θ) generated by CMC, equality (16) satisfies as follows:

$$\begin{aligned}\theta P &= kP + H_3(T\|\delta\|\mathrm{CMC}\|K_c)(a + H_2(\mathrm{CMC}\|F)s)P \\ &= \delta + H_3(T\|\delta\|\mathrm{CMC}\|PK_c)(A + H_2(\mathrm{CMC}\|F)F \\ &\quad + H_1(\mathrm{CMC}\|A\|B)B) \\ &= \delta + H_3(T\|\delta\|\mathrm{CMC}\|PK_c)(M + H_1(\mathrm{CMC}\|A\|B)B).\end{aligned} \tag{40}$$

(2) For the attested authentication request (T, δ, R_n, S_n) from AGT W_n, equality (21) satisfies as follows:

$$\begin{aligned}S_nP &= \theta P + r_nP + h_{4,n}(y_i + h_{2,n}c_n)P \\ &= \delta + R_n + h_{4,n}(Q_n + h_{1,n}B) + H_3(T\|\delta\|\mathrm{CMC}\|PK_c)(M + H_1(\mathrm{CMC}\|A\|B)B).\end{aligned} \tag{41}$$

(3) For the response tuples $\{V_i, L_i\}_{i=1}^{n-1}$ from the controlled UAVs $W_i\,(i = 1, 2, \ldots, n-1)$, equality (30) holds as follows:

$$\begin{aligned}Z_{n-1}P &= \sum_{i=1}^{n-1} L_iP \\ &= \sum_{i=1}^{n-1}\left(f_iP + \widehat{h}_{4,i}(y_i + h_{2,i}c_i)P\right) \\ &= X_{n-1} + \sum_{i=1}^{n-1}\left(\widehat{h}_{4,i}(Y_i + h_{1,i}B + h_{2,i}C_i)\right) \\ &= X_{n-1} + \left(\sum_{i=1}^{n-1}\widehat{h}_{4,i}h_{1,i}\right)B + \sum_{i=1}^{n-1}\widehat{h}_{4,i}Q_i.\end{aligned} \tag{42}$$

(4) For the aggregate response tuple (V_n, L_n) from AGT W_n, equality (37) holds as follows:

$$\begin{aligned}Z_nP &= \sum_{i=1}^{n} L_iP \\ &= \sum_{i=1}^{n}\left(f_iP + \widehat{h}_{4,i}(y_i + h_{2,i}c_i)P\right) \\ &= X_n + \sum_{i=1}^{n}\left(\widehat{h}_{4,i}(Y_i + h_{1,i}B + h_{2,i}C_i)\right) \\ &= X_n + \left(\sum_{i=1}^{n}\widehat{h}_{4,i}h_{1,i}\right)B + \sum_{i=1}^{n}\widehat{h}_{4,i}Q_i.\end{aligned} \tag{43}$$

Thus, the proposed CLAS construction is correct.

4. System Analysis

This section analyzes the security and performance of the proposed CLAS construction.

4.1. Security Analysis

Theorem 2. *Assume that the ECDLP assumption holds in cyclic group G. The proposed CLAS construction can guarantee the unforgeability of the authentication request from CMC.*

Proof 2. In the authentication request (T, δ, θ) generated by CMC, the element θ is considered to be a certificateless signature of $T\|\delta\|\mathrm{CMC}\|PK_c$. It can be seen that θ can serve as the common signature v_i in Thumbur et al.'s scheme [26]. As proved in Theorem 1 in [26], their scheme is existentially unforgeable against Type-I adversary, which assumes that the ECDLP assumption holds in additive group G of elliptic curve points. Therefore, any attacker cannot forge a valid authentication request of CMC without knowing public key PK_c, which implies the unforgeability of the authentication request from CMC can be guaranteed.

Theorem 3. *Assume that the ECDLP assumption holds in cyclic group G. The proposed CLAS construction can guarantee the unforgeability of the attested authentication request from AGT.*

Proof 3. In the attested request (T, δ, R_n, S_n) generated by AGT, the element S_n is considered to be a certificateless signature on θ. It can be seen that S_n can serve as the common signature v_i in Thumbur et al.'s scheme [26]. As proved in Theorem 1 in [26], their scheme is existentially unforgeable against Type-I adversary, which assumes that the ECDLP assumption holds in additive group G of elliptic curve points. Therefore, any attacker cannot forge a valid attested request or response of AGT without knowing public key PK_n, which implies the unforgeability of the attested authentication request from AGT can be guaranteed.

Theorem 4. *Assume that the ECDLP assumption holds in cyclic group G. The proposed CLAS construction can guarantee the unforgeability of the responses from UAVs.*

Proof 4. For the response tuple (V_i, L_i) generated by UAV W_i, it is considered to be a certificateless signature on $T\|\delta$. It can be seen that (V_i, L_i) can serve as the common signature v_i in Thumbur et al.'s scheme [26]. As proved in Theorem 1 in [26], their scheme is existentially unforgeable against Type-I adversary, which assumes that the ECDLP assumption holds in additive group G of elliptic curve points. Therefore, any attacker cannot forge a valid authentication response of UAV without knowing public key PK_i, which implies the unforgeability of the responses from UAVs can be guaranteed.

Theorem 5. *Assume that the ECDLP assumption holds in cyclic group G. The proposed CLAS construction can guarantee the unforgeability of the aggregate response from AGT.*

Proof 5. For the aggregate response tuple (X_n, Z_n) generated by CMC, it is considered as the aggregate signature on n individual responses. It can be seen that (X_n, Z_n) can serve as the common signature v_i in Thumbur et al.'s scheme [26]. As proved in Theorem 1 in [26], their scheme is existentially unforgeable against Type-I adversary, which assumes the ECDLP assumption holds in additive group G of elliptic curve points. Therefore, any attacker cannot forge a valid aggregate response of AGT without knowing public key PK_i, which implies the unforgeability of the aggregate response from AGT can be guaranteed.

Theorem 6. *Assume that the ECDLP assumption holds in cyclic group G. The proposed CLAS construction can be resistant to malicious KGC.*

Proof 6. For the partial private key (y_i, Y_i) generated by KGC, it is considered as a Schnorr signature [27] on W_i. It can be seen that (y_i, Y_i) can serve as the common signature D_i in [26]. As proved in Theorem 2 in [26], their scheme is existentially unforgeable against Type-II adversary, which assumes that the ECDLP assumption holds in additive group G of elliptic curve points. Therefore, any malicious KGC cannot forge valid partial private key of UAVs without knowing master secret key b; thus, the authenticity of KGC can be guaranteed in producing a partial private key.

4.2. Functional Comparison. Wang et al. [8] proposed an identity-based aggregate authentication scheme for UAVNs in bilinear groups. In [8], all UAVs are able to communicate with the CMC through their respective AGTs in the cluster, to perform valid authentication. There is no mechanism for AGT to validate the authenticity of CMC before forwarding authentication request to UAVs in its administrative domain. Furthermore, when individual responses are aggregated from UAVs in the respective cluster, the AGT does not verify the authenticity of those responses.

Li et al. [9] proposed an aggregate authentication scheme, where the above two mechanisms are introduced to enhance the security of authentication in UAVNs. Note that CMC may be malicious in generating keys for UAVs, which means their scheme cannot resistant against malicious KGC. While in our CLAS construction, the partial private key for UAVs are generated by KGC. The detailed comparison on the functionalities among Wang et al.'s proposal [8], Li et al.'s proposal [9], and our CLAS construction is summarized in Table 1.

4.3. Theoretical Comparison. Let T_{SM} be the time of one scalar point multiplication and T_{BP} be one bilinear pairing operation. For the key generation procedure, Wang et al.'s scheme [8] and Li et al.'s scheme [9] require n and $2n$ scalar point multiplications for n entities, respectively. In the request verification procedure, 2 bilinear pairing operations are both required in Wang et al.'s scheme [8] and Li et al.'s scheme [9]. For the aggregate verification by AGT procedure, Li et al.'s scheme [9] requires $(n-1)$ scalar point multiplications and 3 bilinear pairing operations. In the aggregate verification by CMC procedure, compared with Li et al.'s scheme [9], our scheme requires only $(n+2)$ scalar point multiplications. More details for comparsion on computation costs are summarized in Table 2.

4.4. Experimental Performance. To evaluate the computation cost of our CLAS construction, we conduct experiments using the Java Pairing-Based Cryptography Library (JPBC, http://gas.dia.unisa.it/projects/jpbc/), on a platform with Microsoft Windows 10 operating system, Intel(R) Core(TM) i5-6500 CPU @ 3.20 GHz, and 12 GB RAM. The elliptic curve is of Type A $(y^2 = x^3 + x)$ such that q is a 160 bit prime, and the element size in group G is 512 bits.

The performance of the procedures of our CLAS construction is depicted in Figure 3, which are system setup (Setup), key generation (SUMkgen), authentication request generation (REQgen) and attestation (REQfwd), and RAV response (UAVresp). The SUMkgen stage consists of three algorithms, partial key generation for UAV (KGCkgen), key verification for UAV (UAVerify), and key generation for UAV (UAVkgen). The setup algorithm is used to initialize the CLAS system. We can see that the majority of the computation depends on B, which takes roughly 144 msec. The SUMkgen algorithm is used to generate public and private keys for UAVs, which efficiency depends on the UAVerify and the UAVkgen algorithms. Since the partial private key is generated by KGC, the time for UAVs to generate public and private keys is reduced, which is approximately 24 msec in experiments.

The REQgen algorithm can be run to generate authentication request. Its performance mainly depends on

Table 1: Functional comparison.

	Request verification	Request attestation	Aggregate verification		Resistant to malicious KGC
			By AGT	By CMC	
Our scheme	√	√	√	√	√
Li et al. [9]	√	√	√	√	×
Wang et al. [8]	√	×	×	√	×

Table 2: Theoretical comparison.

	Key generation	Request verification	Request attestation	Aggregate verification	
				By AGT	By CMC
Our scheme	$3nT_{\text{SM}}$	$3T_{\text{SM}}$	$1T_{\text{SM}}$	$(n+1)T_{\text{SM}}$	$(n+2)T_{\text{SM}}$
Li et al. [9]	$2nT_{\text{SM}}$	$2T_{\text{BP}}$	$2T_{\text{SM}}$	$(n-1)T_{\text{SM}} + 3T_{\text{BP}}$	$nT_{\text{SM}} + 3T_{\text{BP}}$
Wang et al. [8]	nT_{SM}	$2T_{\text{BP}}$	—	—	$nT_{\text{SM}} + 3T_{\text{BP}}$

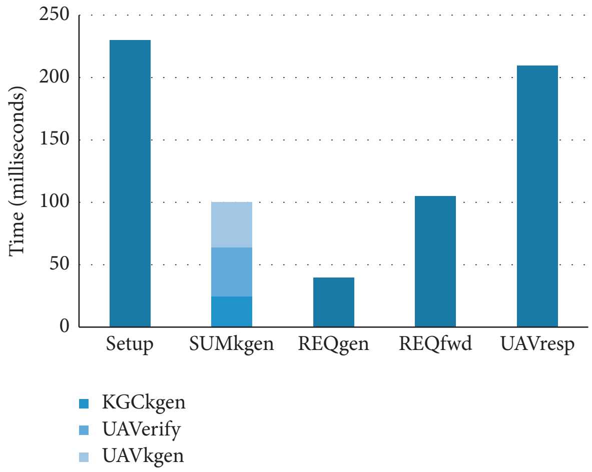


Figure 3: Performance evaluation of the setup, key generation, request generation, forwarding, and UAV response procedures.

the computation of δ, requiring one scalar point multiplication, whereas Wang et al.'s scheme [8] and Li et al.'s scheme [9] both cost two scalar point multiplications. As depicted in Figure 3, an authentication request is able to be transmitted in less than 24 msec. In the stage of REQfwd, before producing attested request, AGT verifies the authenticity of the authentication request from CMC by checking equality (16), which takes two scalar point multiplications. It requires AGT to forward the request in roughly 0.07 seconds. Before generating a response, each UAV validates the authenticity of the attested request received from its administrative AGT, requiring 5 scalar point multiplications. As a result, it takes about 0.15 seconds for each UAV to run the response procedure, while Li et al.'s scheme [9] requires more computational costs, i.e., 4 bilinear pairing operations.

In the response aggregation procedure, AGT needs to aggregate the elements $\{V_i, L_i\}$ in the received response tuples. It can be seen that prior to the batch verification of these responses, only $(n+1)$ scalar point multiplications are required in equality (30), as compared to Li et al.'scheme [9]. In the simulation, a variety of scenarios for the number of unmanned aerial vehicles are considered, that is, $n = 10, 20, \ldots, 100$, and the amount of UAVs consists of one AGT and $(n-1)$ UAVs. AGT aggregates and verifies $(n-1)$ response tuples of UAVs and further aggregates all the response tuples including its response. The experimental results are shown in Figure 4, which indicates a linear correlation between the computation time of this process and the number of unmanned aerial vehicles in a single cluster.

For the process of aggregating verification by CMC, Figure 5 shows the computation time that the CMC verifies the aggregate response from AGT for a single cluster. We also consider multiple cases where the number of unmanned aerial vehicles in a single cluster are $n = 10, 20, \ldots, 100$, respectively. As shown in equality (37), CMC is required to compute $(n+2)$ scalar point multiplications. It can be seen from Figure 5 that there is also a linear correlation between the computation time of this process and the number of unmanned aerial vehicles in a single cluster.

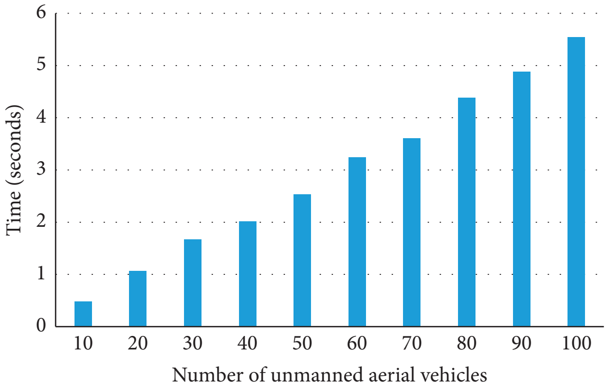


Figure 4: Performance evaluation of the AGT aggregation procedure.

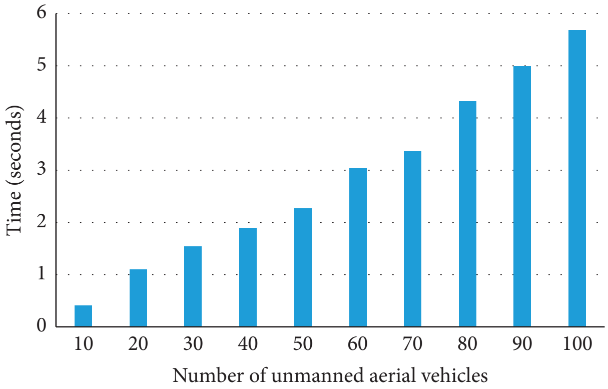


Figure 5: Performance evaluation of the CMC verification procedure.

5. Conclusion

To address the security problems in UAVNs, this paper proposed a CLAS construction without bilinear groups to realize efficient mutual authentication between control center and unmanned aerial vehicles. After the system is initialized, KGC produces the partial private key for each entity. CMC sends the authentication request to AGT; then, AGT forwards the attested request to UAVs in its adminstrative cluster. All response tuples of UAVs are validated by the cluster head AGT and then forwarded to CMC for further verificaton. Security analysis showed that our CLAS construction can not only provide unforgeability for (attested) authentication request and (aggregate) responses but also can resist malicious KGC. Experimental analysis demonstrated that the proposed CLAS construction enjoys practical performance.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This article was supported in part by the National Key R&D Program of China under project 2020YFB1006003, National Natural Science Foundation of China under projects 61772150, 61862012, and 61962012, Guangdong Key R&D Program under project 2020B0101090002, Guangxi Natural Science Foundation under grants 2018GXNSFDA281054, 2019GXNSFFA245015, and 2019GXNSFGA245004, Peng Cheng Laboratory Project of Guangdong Province under grant PCL2018KP004, and Innovation Project of Guangxi Graduate Education under grant YCSW2021176.

References

[1] Y. Liu, H.-N. Dai, Q. Wang, M. K. Shukla, and M. Imran, "Unmanned aerial vehicle for internet of everything: opportunities and challenges," *Computer Communications*, vol. 155, pp. 66–83, 2020.

[2] J. Sun, W. Wang, L. Kou et al., "A data authentication scheme for UAV ad hoc network communication," *The Journal of Supercomputing*, vol. 76, no. 6, pp. 4041–4056, 2020.

[3] M. Y. Arafat and S. Moh, "A survey on cluster-based routing protocols for unmanned aerial vehicle networks," *IEEE Access*, vol. 7, pp. 498–516, 2019.

[4] Y. Zhi, Z. Fu, X. Sun, and J. Yu, "Security and privacy issues of UAV: a survey," *Mobile Networks and Applications*, vol. 25, no. 1, pp. 95–101, 2020.

[5] R. Fotohi, E. Nazemi, and F. Shams Aliee, "An agent-based self-protective method to secure communication between UAVs in unmanned aerial vehicle networks," *Vehicular Communications*, vol. 26, 2020 https://www.sciencedirect.com/science/article/pii/S2214209620300383, Article ID 100267.

[6] R. Altawy and A. M. Youssef, "Security, privacy, and safety aspects of civilian drones: a survey," *ACM Transactions on Cyber-Physical Systems*, vol. 1, no. 2, pp. 1–25, 2016.

[7] C. Lin, D. He, N. Kumar, K.-K. R. Choo, A. Vinel, and X. Huang, "Security and privacy for the internet of drones: challenges and solutions," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 64–69, 2018.

[8] H. Wang, J. Li, C. Lai, and Z. Wang, "A provably secure aggregate authentication scheme for unmanned aerial vehicle cluster networks," *Peer-to-Peer Networking and Applications*, vol. 13, no. 1, pp. 53–63, 2020.

[9] J. Li, M. Zhao, Y. Ding, D. Y. W. Liu, Y. Wang, and H. Liang, "An aggregate authentication framework for unmanned aerial vehicle cluster network," in *Proceedings of the 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 1249–1256, Xiamen, China, December 2020.

[10] N. Mohamed, J. Al-Jaroodi, I. Jawhar, I. Ahmed, and F. Mohammed, "Unmanned aerial vehicles applications in future smart cities," *Technological Forecasting and Social Change*, vol. 153, 2020 https://www.sciencedirect.com/science/article/pii/S0040162517314968, Article ID 119293.

[11] A. Islam and S. Y. Shin, "A blockchain-based secure healthcare scheme with the assistance of unmanned aerial vehicle in internet of things," *Computers & Electrical Engineering*, vol. 84, 2020 https://www.sciencedirect.com/science/article/pii/S0045790620304821, Article ID 106627.

[12] B. Jiang, G. Huang, T. Wang, J. Gui, and X. Zhu, "Trust based energy efficient data collection with unmanned aerial vehicle in edge network," *Transactions on Emerging Telecommunications Technologies*, 2020, https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3942, Article ID e3942.

[13] J. Qiu, D. Grace, G. Ding, J. Yao, and Q. Wu, "Blockchain-based secure spectrum trading for unmanned-aerial-vehicle-assisted cellular networks: an operator's perspective," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 451–466, 2020.

[14] G. Cho, J. Cho, S. Hyun, and H. Kim, "Sentinel: a secure and efficient authentication framework for unmanned aerial vehicles," *Applied Sciences*, vol. 10, no. 9, p. 3149, 2020.

[15] T. Duy Khanh, I. Komarov, Le Duy Don, R. Iureva, and S. Chuprov, "Tra: effective authentication mechanism for swarms of unmanned aerial vehicles," in *Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1852–1858, Canberra, ACT, Australia, December 2020.

[16] X. Li, Y. Wang, P. Vijayakumar, D. He, N. Kumar, and J. Ma, "Blockchain-based mutual-healing group key distribution scheme in unmanned aerial vehicles ad-hoc network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11309–11322, 2019.

[17] A. Yang, J. Weng, N. Cheng, J. Ni, X. Lin, and X. Shen, "Deqos attack: degrading quality of service in vanets and its mitigation," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4834–4845, 2019.

[18] P. Gope, O. Millwood, and N. Saxena, "A provably secure authentication scheme for rfid-enabled UAV applications," *Computer Communications*, vol. 166, pp. 19–25, 2021.

[19] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the International Cnference on the Theory and Application of Cryptology and Information Security*, pp. 452–473, Springer, Taipei, Taiwan, December 2003.

[20] J. Baek, R. Safavi-Naini, and W. Susilo, "Certificateless public key encryption without pairing," in *Proceedings of the International Conference on Information Security*, pp. 134–148, Springer, Singapore, Asia, September 2005.

[21] K.-H. Yeh, C. Su, K.-K. Raymond Choo, and W. Chiu, "A novel certificateless signature scheme for smart objects in the internet-of-things," *Sensors*, vol. 17, no. 5, https://www.mdpi.com/1424-8220/17/5/1001, 2017.

[22] X. Jia, D. He, Q. Liu, and K.-K. R. Choo, "An efficient provably-secure certificateless signature scheme for internet-of-things deployment," *Ad Hoc Networks*, vol. 71, pp. 78–87, 2018, https://www.sciencedirect.com/science/article/pii/S1570870518300015.

[23] Y. Zhao, Y. Hou, L. Wang, S. Kumari, M. Khurram Khan, and Hu Xiong, "An efficient certificateless aggregate signature scheme for the internet of vehicles," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 5, Article ID e3708, 2020.

[24] H. Shu, P. Qi, Y. Huang, F. Chen, D. Xie, and L. Sun, "An efficient certificateless aggregate signature scheme for blockchain-based medical cyber physical systems," *Sensors*, vol. 20, no. 5, p. 1521, 2020.

[25] Y.-C. Chen and R. Tso, "A survey on security of certificateless signature schemes," *IETE Technical Review*, vol. 33, no. 2, pp. 115–121, 2016.

[26] G. Thumbur, G. S. Rao, P. V. Reddy, N. B. Gayathri, and D. V. R. K. Reddy, "Efficient pairing-free certificateless signature scheme for secure communication in resource-constrained devices," *IEEE Communications Letters*, vol. 24, no. 8, pp. 1641–1645, 2020.

[27] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5317062, 12 pages
https://doi.org/10.1155/2021/5317062

Research Article

A Reconstruction Attack Scheme on Secure Outsourced Spatial Dataset in Vehicular Ad-Hoc Networks

Qing Ren,[1] Feng Tian ,[1] Xiangyi Lu,[1] Yumeng Shen,[1] Zhenqiang Wu,[1] and Xiaolin Gui[2]

[1] *School of Computer Science, Shaanxi Normal University, Xi'an 710062, Shaanxi, China*
[2] *School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, Shaanxi, China*

Correspondence should be addressed to Feng Tian; tianfeng@snnu.edu.cn

Received 11 June 2021; Accepted 14 August 2021; Published 8 September 2021

Academic Editor: Anjia Yang

In the cloud-based vehicular ad-hoc network (VANET), massive vehicle information is stored on the cloud, and a large amount of data query, calculation, monitoring, and management are carried out at all times. The secure spatial query methods in VANET allow authorized users to convert the original spatial query to encrypted spatial query, which is called query token and will be processed in ciphertext mode by the service provider. Thus, the service provider learns which encrypted records are returned as the result of a query, which is defined as the access pattern. Since only the correct query results that match the query tokens are returned, the service provider can observe which encrypted data are accessed and returned to the client when a query is launched clearly, and it leads to the leakage of data access pattern. In this paper, a reconstruction attack scheme is proposed, which utilizes the access patterns in the secure query processes, and then it reconstructs the index of outsourced spatial data that are collected from the vehicles. The proposed scheme proves the security threats in the VANET. Extensive experiments on real-world datasets demonstrate that our attack scheme can achieve quite a high reconstruction rate.

1. Introduction

At present, the vehicular ad-hoc network (VANET) has gained a lot of attention in the field of intelligent transportation. The VANET can be used to intelligently control the traffic process, such as real-time traffic information systems to ensure traffic efficiency, and vehicle safety systems, such as rear-end collision warning systems, to improve vehicle safety. However, the powerful function of the VANET is supported by information sharing between vehicle users, which will introduce serious data security threats [1, 2]. For example, exploiting the weakness of lacking physical proximity authentication, malicious attackers may infer the location of the vehicle during a specific period of time [2]. Due to the openness and mobility of VANET, the content delivery of VANET poses serious security threats, for which some countermeasures have been proposed, such as confidentiality, integrity, and authentication [3, 4].

As the data generated by users of VANET continue to grow and beyond the processing capacity of the data owners, the data need to be outsourced and stored in the cloud server to reduce data management overhead. To ensure the security of user data on the untrustworthy server, cryptographic techniques are employed, while still allowing efficient query processing on the cloud server. However, the privacy provided by the existing secure outsourced dataset systems is poorly considered. In the searchable encryption mechanism, the search process for encrypted files is as follows: First, the authorized user will submit the query token to the service provider, who will process the query through a series of calculations and return the query result to the user in the form of ciphertext. Then, the user decrypts the query result locally. It seems very safe because the entire query process is carried out in the ciphertext state, including query submission, query process calculations, and query results feedback.

Nonetheless, this process leaks access patterns. In other words, the service provider can observe which encrypted files in the dataset are accessed and returned to the authorized users. Therefore, the honest but curious service provider clearly knows the matching relationship between encrypted files and queries.

Existing studies [5–8] have shown that attackers can use leaked access patterns to recover user privacy information. Li et al. [5] demonstrated the hidden security threats caused by leaked access patterns with an encrypted patient medical dataset stored in a third-party server. Series of examples are given sequentially to illustrate how the patient's sensitive information is gradually inferred with the leakage of access patterns. With leaked access patterns, Quan et al. [6] implemented a range injection attack on the one-dimensional and discrete dataset. Exploiting the collusion of the service provider and the secondary user, a set of selected range queries are injected into the dataset, and through the access patterns of these queries, the dataset index is completely reconstructed. The attack method proposed by Islam et al. [7] does not require collusion, which is designed for text data. Considering the collected keyword co-occurrence matrix as prior knowledge, the service provider realizes an assignment of keywords to each query. Kallaris et al. [8] reconstructed the discrete one-dimensional data and completely restored the data index stored on the server, in which neither collusion nor prior knowledge is required.

Researchers have explored various types of attack methods for different types of datasets to prove the security threats caused by the leakage of access patterns, but there is no research to prove the potential security threats caused by the leakage of spatial data access patterns.

In this paper, we propose a reconstruction attack scheme on outsourced spatial dataset using access pattern leakage in VANET systems. The threat model considered is as follows. We take the honest but curious service provider as the attacker, who will process the query correctly and honestly, but will be curious about the dataset stored on the server. Our attack aims to completely reconstruct the dataset stored on the server without any deciphering, that is, to determine the spatial index of each record on the server.

Assuming that the service provider only has a little prior knowledge of the spatial dataset and the users will issue enough one-dimensional and uniform queries to the server, our reconstruction attack will be processed as the following four steps. Firstly, the data space will be discretized in accordance with the granularity that the attacker aims to achieve. Secondly, to improve the efficiency of attack, the collected access patterns will be simplified. Thirdly, we will determine the relative order for each row/column of records. Finally, the spatial index of each record will be recovered.

The contributions of this paper are summarized as follows:

(i) A reconstruction attack against secure outsourced spatial dataset in VANET systems is proposed, proving that the security threats caused by access pattern leakage are universal.

(ii) With spatial discretization, our scheme can support the attack for optional spatial granularity. Meanwhile, utilizing the statistic of the record co-occurrence, access patterns are simplified, which guarantees the attack efficiency.

(iii) Extensive experiments on real-world datasets demonstrate that our attack scheme can achieve quite a high reconstruction rate.

2. Related Work

2.1. Location Privacy Protection of VANET. The existing location privacy protection technologies of VANET mainly include three categories. The first category is a rule-based privacy protection method [9, 10], which restricts service providers from a legal perspective and prohibits the data and user information abused through privacy protection rules, standards, and detailed specifications acting on the server side. For example, IETF's GeoPriv [9] and W3C's P3P [10] stipulate that the authorization, integrity, and privacy requirements must be met when data are used. However, the security of such methods depends on legal supervision and public opinion, and the reliability of privacy protection depends on the implementation degree of the service provider.

The second category is based on generalization and obfuscation [11–17]. Spatial concealment technology [11, 12] forms a hidden area containing k real users for each user, making it difficult for service providers to determine the real identity and accurate spatial of the user from the hidden area. But in this type of method, it is difficult to achieve a balance between privacy protection and service quality in data-sparse areas. Spatial offset and obfuscation technology [13–16] protect user's privacy by moving the real spatial in a small area or replacing the real spatial with a certain area. For example, Andrés et al. [16] achieved the geo-indistinguishability by adding controlled random noise to user's spatial, which corresponds to differential privacy. Nevertheless, this kind of method reduces the spatial accuracy [17], so the returned service data may be untrustworthy.

The third type of privacy protection method is based on cryptography [18–22], through the processing of cryptographic technology to achieve the requirements of privacy protection. The general process is as follows: Authorized client encrypts the spatial information and sends it to the service provider, who processes the corresponding query in the ciphertext and returns matching encrypted result. Finally, authorized client decrypts locally to obtain the plaintext information. After encryption processing, the data sent by the client can meet stricter privacy protection requirements, but encryption and decryption bring huge computational overhead. In addition, although the data are encrypted before outsourcing, search process is also performed in ciphertext on the server; this method still has the problem of access pattern leakage (except for the ORAM scheme [23–25]); that is, the attacker can observe the correspondence between the encrypted query and the accessed encrypted documents. Furthermore, the ORAM solution protects the access pattern by shuffling and re-encrypting after each access of data, but its huge communication overhead makes using ORAM to protect the access pattern too expensive.

Therefore, in location privacy protection technology, the privacy security threats caused by leaked access patterns need to be further studied.

2.2. Attack with Access Patterns. In recent years, researchers have made arguments for the privacy security threats caused by the leakage of access patterns from the perspective of attacks, which are mainly divided into two categories.

One is active attacks [6, 26], including injection, tampering, and forgery, by which the attacker attacks server information actively. By injecting files that added selected keywords into the dataset on the server and observing the access patterns of the injected files, Zhang et al. [26] inferred the user's query information successfully. With the collusion of the service provider and secondary users, Quan et al. [6] injected a set of selected range queries and observed access patterns to infer the user's precise information. However, because the active attacks destroy the authenticity and integrity of the information, it is easy to be detected.

The other is passive attack [5, 7, 8, 27]. The attacker can infer the user's sensitive information through the monitored access pattern without affecting normal data communication, thereby undermining the confidentiality of data transmission. Islam et al. [7] proposed that by utilizing the statistical keyword co-occurrence matrix as prior knowledge and collecting access patterns, the attacker realizes an assignment of keywords to each query that achieves maximum matching from background knowledge. Assuming that the attacker has more prior knowledge (including the number of files corresponding to each keyword), Cash et al. [27] proposed an improved passive attack method based on IKK, and utilized the access pattern leakage to recover the query keywords. Without any prior knowledge, utilizing the continuity of range query, Kellaris et al. [8] assigned index to each file on the server and completely reconstructed the dataset. This type of attack is not easy to detect, because there is no direct impact on data transmission.

In summary, existing researches have proved the security threats caused by the access pattern leakage from different angles. However, the problem of spatial data access pattern leakage has not been studied in detail. And, most of the existing attack methods are active attacks, including injection or passive attacks, requiring much prior knowledge, so the attack conditions are subject to certain restrictions.

3. System Model

3.1. Spatial Data Outsourcing System. The system model of the outsourcing dataset system is shown in Figure 1, including three entities, namely, the data owner, the server, and the authorized user.

As the data generated by users of VANET continue to grow and beyond the processing capacity of the data owners (DO), the data need to be outsourced to the cloud server. For security concerns, encryption is usually performed before outsourcing. In order to facilitate user query, the encrypted index is also generated and uploaded to the cloud server at the same time.

Authorized users (AU) have secret keys, KI and KD. Encrypted query tokens will be generated by KI and uploaded to the cloud server. Ciphertext query results obtained from the server will be decrypted by the secret key KD.

The server stores the data and their corresponding query index for the data owner, and has powerful computing capabilities as well as searchable encryption algorithm so that it supports queries under ciphertext.

As shown in Figure 1, when an authorized user generates a query Q and encrypts and sends it to the cloud server, the service provider will perform the query operation and return the matching set of records R to the authorized user.

3.2. Access Pattern Leakage. In the data outsourcing system, authorized users usually generate spatial query tokens, which are processed in ciphertext mode on the server, and finally only the matching results are accessed and returned. So, the service provider learns which encrypted records match a query, which is defined as access pattern. Since only the correct query results that match the query tokens are accessed and returned, the service provider can observe the access pattern clearly when a query is launched. So, the process leads to the leakage of data access pattern. Such leakage is typical for current VANET systems based on symmetric searchable encryption. The queries on the server are continuous, so the service provider can sniff out a large number of access patterns quietly, which provides tremendous amount of data to the attacker.

In this paper, we define the access pattern leakage $\mathbf{L}_{\mathbf{access}}$ as: the correspondence between the ciphertext query q and the matching ciphertext query result $\mathbf{R}$.

$$\mathbf{L}_{\mathbf{access}} = \{(q_1, \mathbf{R}_1), \ldots, (q_n, \mathbf{R}_\mathbf{n})\}, \tag{1}$$

$$\mathbf{R}_\mathbf{n} = \{r_j | q_n(I_j) = 1, \quad I_j \in \mathbf{I}\}. \tag{2}$$

As shown in equation (1), the leaked access pattern contains multiple queries and their matching query records sets, where q_n represents a query token, and $\mathbf{R}_\mathbf{n}$ represents the matching records set of query q_n. As shown in equation (2), the index corresponding to each record on the dataset is calculated with query token q_n, and the matching records are returned as set $\mathbf{R}_\mathbf{n}$.

3.3. Attack Model. In this section, we describe the attack model of reconstruction attack with access pattern leakage. Here, the attacker has access to the communication channel, and thus observes a set of access patterns $\mathbf{L}_{\mathbf{access}} = \{(q_1, \mathbf{R}_1), \ldots, (q_n, \mathbf{R}_\mathbf{n})\}$. Let us define a week attacker as follows:

(1) The attacker is passive. The attacker follows the predefined storage and query rules and provides users with correct query results. The attacker will not perform illegal access, injection, or tampering to the dataset, but will process information stealing and collecting. Since the attack process does not involve

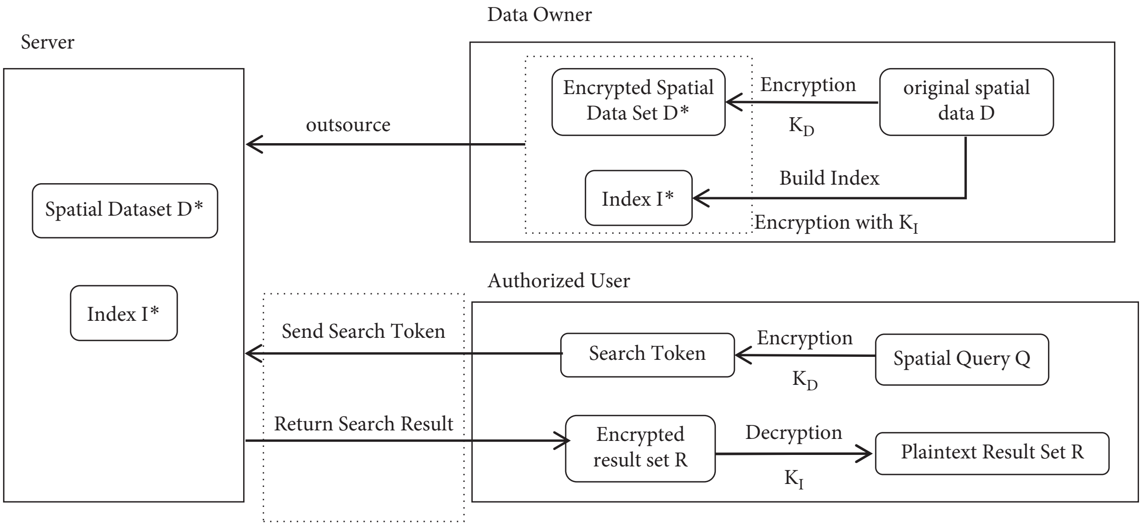


FIGURE 1: Spatial data outsourcing system.

data destruction, the legitimate users will not realize the attacker's activities at all.

(2) The attacker cannot decipher the query submitted by the authorized user through the secret key.

(3) The attacker has a very high probability of succeeding if it has a small amount of background knowledge. We assume that the attacker knows the underlying indexes for k of records in the dataset. That is, the attacker has access to the map $\mathbf{M}_{\mathbf{know}} = \left\{ (I_j, r_j) | I_j \in \mathbf{I} \,\&\, r_j \in \mathbf{R} \right\}$, where $\mathbf{R}$ includes all records stored on the server and $\mathbf{I}$ includes all Index corresponding to R. Taking that $k = |\mathbf{M}_{\mathbf{know}}|$ and $l = |\mathbf{R}|$, then $k \ll l$.

What we need to be clear is that the attacker can observe the distribution of all records stored in the cloud. Although these records are usually encrypted before being uploaded to the server and the attacker cannot obtain the plaintext information of any record, the records can be distinguished easily.

Now, we assume that the attacker knows the encrypted dataset R, the access patterns $\mathbf{L}_{\mathbf{access}}$, and the prior knowledge $\mathbf{M}_{\mathbf{know}}$, and that the user will issue enough one-dimensional and uniform queries to the server. Then, the goal of the attack is to reconstruct the index of all records such that the statistical results as seen by access patterns fit the uniform query rule. In the following sections, we will describe how the attacker uses access pattern leakage to carry out reconstruction attacks on the two-dimensional spatial outsourcing dataset system.

4. Reconstruction Attack with Access Patterns

We propose a reconstruction attack on outsourced spatial dataset that exploits access pattern leakage, that is, the correspondence between encrypted queries and matching query results, which is very common in searchable symmetric encryption.

Assume that the dataset contains n records $r_1, r_2, \ldots, r_n$, which are, respectively, pointed to by spatial indexes $I_1, I_2, \ldots, I_n$. On the spatial dataset, the index is actually the spatial coordinate of each record, that is, $I_i = \{x_i, y_i\}$, which, respectively, represent the horizontal and vertical coordinates of the record. The ultimate goal of the reconstruction attack is to restore the corresponding position coordinates for each record r_1 in the dataset. Ideally, a complete plaintext index can be established. Assuming that the service provider only has little prior knowledge of the spatial dataset and the user will issue enough one-dimensional and uniform queries to the server, the service provider can utilize the observed access patterns to determine the index of each record.

In this section, we will describe the reconstruction attack process in detail, including spatial discretization, access pattern simplification, and determination of row and column indexes.

4.1. Spatial Discretization. In this section, we will discuss the problem of attack granularity. The ultimate goal of the reconstruction attack is to recover the index of each record on the two-dimensional spatial dataset. But for different application scenarios, the attacker hopes that the granularity of the spatial obtained by the attack is different.

As shown in Figure 2, we discretize the data space into $\Delta_2^2, \Delta_4^2, \Delta_8^2$, which means that both dimensions of space is divided into 2, 4, or 8 index spaces, respectively. Under different attack granularities, the attacker recovers the index of records in Figure 2, and the obtained index coordinates are (0,1), (1,2), (2,5), respectively. Combined with the size of the dataset known to the attacker, under different attack granularities, the index spatial of the record has different degrees of privacy leakage. It is undeniable that regardless of the granularity, reconstruction attacks will expose data privacy to a certain extent.

Therefore, we first need to determine the granularity of the data index that the attacker wants to achieve before formally attacking, which determines the granularity of the

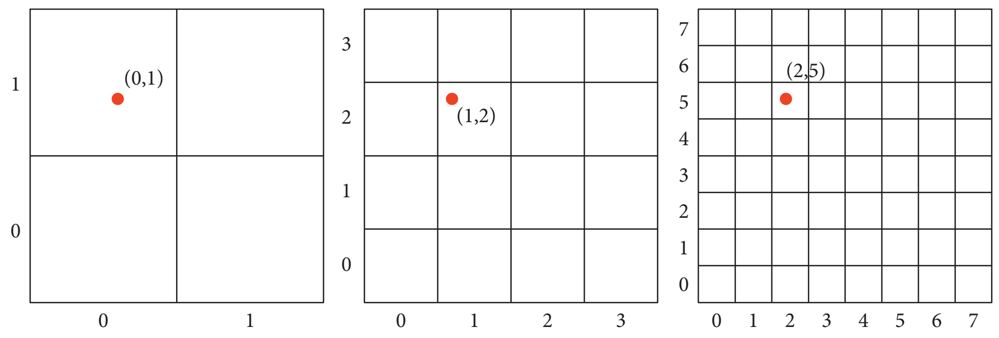


Figure 2: The record index obtained by the attack at different granularities.

space division. Discretize the data space according to the granularity to obtain a two-dimensional discrete data space. We assume that the data space is divided into *Tx* scales horizontally and T_y scales vertically, and the size of the data space is T_xT_y.

For each record in the dataset, it is pointed to by a certain position index $I_i = \{x_i, y_i\}$, where $x_i \in [0, T_x]$ and $y_i \in [0, T_y]$. The ultimate goal of our reconstruction attack is to recover the corresponding position coordinate $I_i' = \{x_i', y_i'\}$ for each record r_1 in the dataset.

4.2. Access Pattern Simplification. In section 4.1, we have introduced the process of spatial discretization. Assuming that the space has been discretized into n pieces of index space, the goal of reconstruction attack is to assign one piece of index space to each record stored on the server.

After a period of sniffing, the attacker (honest but curious service provider) collects enough access patterns to perform the attack. In order to improve the efficiency of our attack, we want to simplify these access patterns in this step. We will classify the records where the records belonging to the same piece of space are classified into one category and a representative record will be selected. By observing the access patterns and counting the co-occurrence of encrypted records, the records with co-occurrence rate of 100% are classified into the same category, and for each category, one record is taken as the representative. Then, the access pattern is simplified by replacing all records with the representative record of corresponding category in the collection of access patterns.

For example, with the attack granularity of 10×10, space is divided into 100 regions, and each region will be pointed to by the same index coordinate. Therefore, for the records that belong to the same region, we only take one record as the representive. If each region has records, we will get 100 representative records at most. Then, we simplify the observed access patterns. Because records belonging to the same category are pointed to by the same spatial index, we can simplify the collection of access patterns by keeping only representative records.

4.3. Determination of Row and Column Indexes. In this section, the row and column indexes will be determined, respectively, and we will introduce the process with 3 algorithms.

The Algorithm 1 is the main method, reconstructing the index of the dataset from the x-dimension and the y-dimension, respectively. Each dimension includes 2 steps. Firstly, the attacker utilizes the continuity of the range query to determine the relative position of records on a single dimension (Algorithm 2). Secondly, leveraging the uniformity of query and the rate of document co-occurrence in the access patterns, the attacker determines the specific index of each record (Algorithm 3). -

For the discretized two-dimensional space, we determine the x coordinate of each record line by line. Firstly, the attacker classifies the records according to the results of a single-row query. Encrypted records belonging to the same row query are grouped together. For this row of records, the attacker uses the collected access patterns set $\mathbf{L_{access1}}$ (these access patterns are generated by uniform query) to determine the relative order of these records in the row through Algorithm 2 (line 5). Next, the attacker uses Algorithm 3 to get the index of these records in the x-direction (line 6). After the line-by-line process is over, the x-coordinates of each record are saved through the Map collection, the ID of the encrypted record is used as the key, and the X-coordinate as the value (line 7–9). Then, the attacker uses the same method to determine the Y coordinate of each record in the dataset (line 13–14), and saves it through the Map collection, with the ID of the encrypted record as the key, and the Y coordinate as the value (line 15–17).

4.3.1. Determination of Relative Order. In this section, we will introduce the procession of the relative order determination for a row/column of records. As a weak attacker defined in Section 3.3, the service provider can only continuously observe the user's query process and calculate access patterns. Assume that the attacker counts a lot of uniform one-dimensional spatial queries, which is enough to perform our reconstruction attack. The two-dimensional spatial dataset reconstruction attack can be converted into multiple one-dimensional attacks and the latter can adopt Generic Attack introduced in [8].

Given that the attacker collects only one-dimensional queries, it is easy to categorize each record stored in a two-dimensional space by row or column. For each row of

```
Input:attack granularity $T_x, T_y$, dataset, access patterns
Output: $x$ index and $y$ index of all records in the dataset
Map map Index$X$
Map map Index$Y$
FOR each row
    get $\mathbf{L_{access1}}$
    ordered Record_$X$←Get ordered ($\mathbf{L_{access1}}$)
    Guess Index_$X$←Get Index ($\mathbf{L_{access1}}$, Ordered Record_$X$)
    For each record in ordered Record
        map index (record$ID$, $X$_Index)
    END FOR
END FOR
FOR each column
    get $\mathbf{L_{access2}}$
    ordered Record_$Y$←Get ordered ($\mathbf{L_{access2}}$)
    Guess Index_$Y$←Get Index ($\mathbf{L_{access2}}$, Ordered Record_$Y$)
    For each record in ordered Record
        map index (record$ID$, $Y$_Index)
    END FOR
END FOR
RETURN map Index$X$, map Index$Y$
```

Algorithm 1: Reconstruction of the spatial dataset index.

```
Input: the collection of access patterns for a row/column $\mathbf{L_{access1}}$
    $\mathbf{L_{access1}} = \{(q_1, \mathbf{R_1}), \ldots, (q_n, \mathbf{R_n})\}$
Output: the ordered set of records in the row/column Ordered Record
    Ordered Record = $(r_1, r_2, \ldots r_m)$
$m = 0, R = \{\}$
FOR each $L_i$ in $\mathbf{L_{access1}}$ DO
    IF $|R| > m$ THEN
        $m = |R_i|, R = R_i$
    END IF
END FOR
FOR each $L_i$ in $\mathbf{L_{access1}}$ DO
    IF $|R| = m - 1$ THEN
        ordered Record add $(R/R_i)$
        BREAK
    END IF
END FOR
FOR $j = 2$ to $m$ DO
    FOR each $L_i$ in $\mathbf{L_{access1}}\sqrt{a^2 + b^2}$ DO
        IF $|R_i| = |$ordered Record$| + 1$ and $R_i$Contains All (ordered Record) THEN
            ordered Record add ($R_i$/ordered Record)
            BREAK
        END IF
    END FOR
END FOR
RETURN ordered Record
```

Algorithm 2: GetOrder: Determination of the relative order for row/column records.

records, we first utilize the continuity of spatial query to determine the relative order of the records, through a process known as GetOrder, the details of which are as follows:

(1) Find the maximum set U of the row query in the access patterns

(2) Find the largest true subset S_1 of U, then the difference set between U and S_1 is the first record r_1

(3) Find the minimum superset S_1 of the set $\mathbf{R_{i-1}}$, where $\mathbf{R_{i-1}}$is formed by the confirmed records $\mathbf{R_{i-1}} = \left\{r_j \,|\, j \in [1, i-1]\right\}$. And, the difference set between S_i and $\mathbf{R_{i-1}}$ is the next record.

```
Input: The set of access patterns of uniform query for a row/column$\mathbf{L}_{\mathbf{access2}}$
  $\mathbf{L}_{\mathbf{access2}} = \{(q_1, R_1), \ldots, (q_n, R_n)\}$
  The row/column ordered record collection **Ordered Record**
  **Ordered Record** $= (r_1, r_2, \ldots r_m)$
Output: the index of the row record/the column record $I$
(1)  sum First = 0
(2)  FOR each$L_i$ in $\mathbf{L}_{\mathbf{access2}}$ DO
(3)    IF **ordered Record**[1] = in$R_i$THEN
(4)      sumFirst + +
(5)    END IF
(6)  END FOR
(7)  $I[1] = \arg\min X(\text{SumFirst}/\mathbf{L}_{\mathbf{access2}}\text{length}2x(T - x + I)/T(T + 1))$
(8)  Sum = {0, 0}
(9)  Sum[1] = Sum First
(10) FOR each $L_i$ in $\mathbf{L}_{\mathbf{access2}}$ DO
(11)   FOR $j = 2$ to $m$ DO
(12)     $R'$ = {**ordered Record**[$m$]}|$0 < m < j + 1$
(13)     IF $R'$ in $R_i$ THEN
(14)       Sum[$j$] + +
(15)     END IF
(16)   END FOR
(17) END FOR
(18) FOR $j = 2$ to $m$ DO
(19)   $I[j] = \arg\min x(\text{Sum}\,j/\mathbf{L}_{\mathbf{access2}}\text{length} - 2I[I](T - x + 1)/T(T + I))$
(20) END FOR
(21) RETURM $I$
```

Algorithm 3: GetIndex: Determination of indexes for row/column records.

To explain the process of GetOrder, let us consider that the user makes a one-dimensional range query for the single-row record distribution, as shown in Figure 3.

Then, the maximum set of query results is $\{r_7, r_{12}, r_{17}, r_{20}\}$. The largest true subsets are $\{r_{17}, r_7, r_{12}\}$ and $\{r_7, r_{12}, r_{20}\}$. Without the distribution figure, only according to the set relation, we can easily know that the first record of this row is either r_{17} or r_{20}, which is consistent with actual distribution. Assuming that the first record is r_{17}, the minimum superset of $\{r_{17}\}$ in the access pattern set is $\{r_{17}, r_7\}$. Thus, the second record is determined as r_7. The superset of the confirmed records is deduced accordingly and the third and fourth records are r_{12} and r_{20}, respectively. If we assume that the first entry is r_{20}, what we will get is the reverse order of the records. According to the prior knowledge, we can determine whether to reverse this sequence. Finally, as shown in Figure 4, we know that this row stores 4 records, and the relative order of storage is $\{r_{17}, r_7, r_{12}, r_{20}\}$, and the next step is to match each record with the correct index.

The process of GetOrder has been summarized in Algorithm 2. The input of the algorithm is the access pattern set $\mathbf{L}_{\mathbf{access1}} = \{(q_1, \mathbf{R}_1), \ldots, (q_n, \mathbf{R}_\mathbf{n})\}$ of a certain line of query observed by the attacker. First traverse all access patterns to find the most number of record sets. Assuming that the attacker samples enough queries, $\mathbf{L}_{\mathbf{access1}}$ contains the access pattern accessing all the records in the row. From this, we get the row record set R and the record number m of this row (line: 1–6). After that, we determine the first item of the row's records (line: 7–12). Utilizing the continuity of range query, we find the largest proper subset R_1 of the complete set R of the row's records in the access pattern set, and then the difference between R_1 and R is the first/last item of the row of records. We assume that it is the first item, and then judge the row record after the row order is fully determined. If it is proved to be the last item, the row record only needs to be reversed. Finally, we sort the other encrypted records in this row one by one (lines: 13–20). Utilizing the continuity of range query, we find the minimum superset of the determined record in the access pattern set, and the difference between the minimum superset and the record set in the

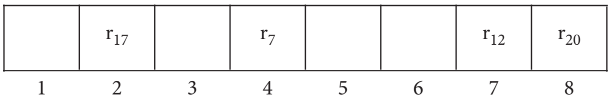


Figure 3: Records distribution of one row.

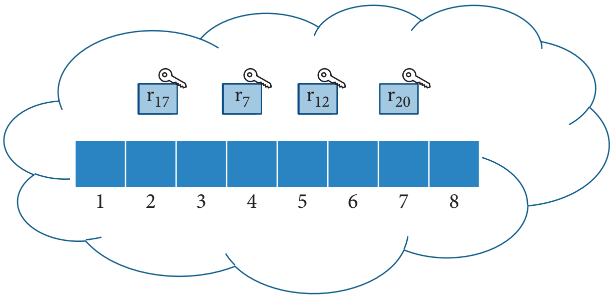


Figure 4: Relative order for the row of records.

determined order is the next record. Determine the relative order of these m records one by one and store them in the queue **orderedRecord**, which is the order/reverse order of the row of records and the algorithm finally returns.

4.3.2. Determination of Indexes. Then, we utilize the uniformity of the query to determine the one-dimensional index value of each record, denoted as GetIndex. Assume that the index space of the row is N and the index coordinate of the i-th record is Z_i. By observing the access patterns of Q uniform queries and counting the number of access patterns containing record r_1 as q_1, we present the calculation of the first record index as an optimization problem by equation (3), and the calculation of the i-th record Index as an optimization problem by equation (4), where Qi represents the number of access patterns including the previous i records.

The result of this equation satisfying the optimization problem is an assignment of index Z_i to the record that achieves a minimum distance from the uniformity of query.

$$\arg\min \sum_{Z_1} \left(\frac{Q_1}{Q} - \left(\frac{2Z_1 (N - Z_1 + 1)}{N(N+1)} \right) \right)^2, \tag{3}$$

$$\arg\min \sum_{Z_1} \left(\frac{Q_i}{Q} - \left(\frac{2Z_1 (N - Z_i + 1)}{N(N+1)} \right) \right)^2. \tag{4}$$

To explain the model described in equation (3), let us consider the following example. Assume that the index space of a row is $[1, N]$, the record r_i is the i-th record of the row, and the x coordinate of r_i is Z_i. The row is uniformly queried, according to the permutation and combination, and the number of unique queries that can be generated is $N(N+1)/2$, the number of unique queries containing the first records is $Z_1 (N - Z_1 + 1)$, and the number of unique queries containing the previous i records is $Z_1 (N - Z_1 + 1)$. Now, for the first record r_1, an attacker can calculate the probability of the r_1 appearing in the uniform query by $\alpha_1 = 2Z_1 (N - Z_1 + 1)/N(N+1)$.

For any given record r_i, $i > 1$, the attacker can calculate the probability of the previous i records $\{r_j | j \in [1, i]\}$ appearing together in the uniform query by $\alpha_1 = 2Z_1 (N - Z_1 + 1)/N(N+1)$. Therefore, by observing access patterns, the attacker can calculate the probability of the previous i records $\{r_j | j \in [1, i]\}$ appearing together in the uniform query by $\beta = Q_i/Q$, where Q represents the total number of access patterns of uniform query and Q_i represents the number of access patterns including the previous i records. Naturally, the attacker will assign coordinate Z_i to the record r_i, if the calculated probability α is close to the observed probability β from the access pattern. This closeness can be measured by the arithmetic distance function $(\alpha - \beta)^2$, where a lower value of this function is preferred over a higher value. So, the goal of the attacker will be to assign an index to records such that this distance function is minimized.

A specific example is shown as follows to explain the process of GetIndex in detail. Assuming that the attack has determined the record order as shown in Figure 4, and the access pattern collected by the service provider is the smallest set that satisfies the attack conditions as shown in Figure 5, the attacker obtains the relative order $\{r_{17}, r_7, r_{12}, r_{20}\}$ and Q access patterns, where $Q = 36$.

Since the query is uniform, assuming that the x coordinate of r_{17} is $Z_1 \in [1, N]$, theoretically, the proportion of uniform queries including the first record r_{17} is $\alpha_1 = 2Z_1 (N - Z_1 + 1)/N(N+1)$. Observing sampled access patterns, according to the statistics, the proportion of access patterns including the first record r_{17} is q_1/Q (i.e.14/36). Then, the difference between the actual value and the theoretical value is $q_1/Q - 2Z_1 (N - Z_1 + 1)/N(N+1)$. We could find Z_1 that minimizes the absolute value of the difference and infer the value of Z_1 should be 2, so the x-coordinate of r_{17} is 2.

Determine the number of q_2 queries with $\mathbf{S}_2 = \{r_{17}, r_7\}$ included in the access patterns of the uniform query, where S_2 is the union of the records including records' determined position and the record of the position to be determined in this step. Then, in this example, q_2 is 10 and the proportion is q_2/Q. Since the query is uniform, assuming that the x coordinate of the second record r_7 is $\mathrm{Z}_2 \in [1, \mathrm{N}]$, theoretically, the proportion of uniform queries including $\mathbf{S}_2 = \{\mathrm{r}_{17}, \mathrm{r}_7\}$ is $2Z_1 (N - Z_2 + 1)/N(N+1)$, then the difference between the actual value and the theoretical value is $q_1/Q - 2Z_1 (N - Z_1 + 1)/N(N+1)$, find Z_2 minimizes the absolute value of the difference and get Z_2 is 4, so the x coordinate of r_7 is 4. And, we can adopt the same measures for other records for their index values, and get the x coordinate of r_{12} and r_{20} as 7 and 8, respectively.

At this point, the attacker knows that there are four records $r_{17}, r_7, r_{12}, r_{20}$ in this row of the data space, and their x-coordinates are 2, 4, 7, and 8, respectively (Figure 6).

The process of GetIndex has been summarized in Algorithm 3. The input of the algorithm is the access pattern set $\mathbf{L}_{\mathbf{access2}} = \{(q_1, R_1), \ldots, (q_n, R_n)\}$ of a uniform single-row query observed by the attacker, and the relative order queue of the row **orderedRecord** determined by algorithm GetOrder.

First, we count the number of access patterns including the first record and store in sumFirst (line: 1–6). Second, we determine the index number of the first record (line: 7). Utilizing the continuity of range query, assuming that a uniform query set is generated for the row of records, and the index number in the x-direction of the first record is x, theoretically, the probability that the query result contains the first record is $2x(T - x + 1)/T(T+1)$, where T is the data space size of the row. Then, traverse the access pattern set $\mathbf{L}_{\mathbf{access2}}$, and according to statistics, the ratio of the access pattern including the first record is sumFirst/$\mathbf{L}_{\mathbf{access2}}$.length. Find the x value that minimizes the difference between the theoretical value and the actual statistical value, which is the index number of the first item.

Finally, we determine the index of other records one by one (lines: 8–20). Utilizing the continuity of range query, assuming that a uniform query set is generated for the row of records, and the index number in the x-direction of the record **orderedRecord** $[j]$ is x, theoretically, the probability that the query result contains the first j records is

1.	{ }	11.	$\{r_{17},r_7\}$	21.	$\{r_7,r_{12},r_{20}\}$	21.	{ }
2.	$\{r_{17}\}$	12.	$\{r_{17},r_7\}$	22.	$\{r_7\}$	22.	$\{r_{12}\}$
3.	$\{r_{17}\}$	13.	$\{r_{17},r_7\}$	23.	$\{r_7\}$	23.	$\{r_{12},r_{20}\}$
4.	$\{r_{17},r_7\}$	14.	$\{r_{17},r_7,r_{12}\}$	24.	$\{r_7\}$	24.	$\{r_{12}\}$
5.	$\{r_{17},r_7\}$	15.	$\{r_{17},r_7,r_{12},r_{20}\}$	25.	$\{r_7,r_{12}\}$	25.	$\{r_{12},r_{20}\}$
6.	$\{r_{17},r_7\}$	16.	{ }	26.	$\{r_7,r_{12},r_{20}\}$	26.	$\{r_{20}\}$
7.	$\{r_{17},r_7,r_{12}\}$	17.	$\{r_7\}$	27.	{ }		
8.	$\{r_{17},r_7,r_{12},r_{20}\}$	18.	$\{r_7\}$	28.	{ }		
9.	$\{r_{17}\}$	19.	$\{r_7\}$	29.	$\{r_{12}\}$		
10.	$\{r_{17}\}$	20.	$\{r_7,r_{12}\}$	30.	$\{r_{12},r_{20}\}$		

Figure 5: Access pattern samples of uniform query.

	r_{17}		r_7			r_{12}	r_{20}
1	2	3	4	5	6	7	8

Figure 6: Index for the row of records.

$2x(T-x+1)/T(T+1)$, where i_1 is the index number of the first item and T is the data space size of the row. Then, we determine the statistical value of the probability. Traverse the access pattern set $\mathbf{L_{access2}}$, and count the number of the access patterns including the first j records (**orderedRecord** [1], ... ,**orderedRecord** [j]) as sum[j] (lines: 8–17). Therefore, the ratio of each record is sum [j]/$\mathbf{L_{access2}}$.length. Find the x value that minimizes the difference between the theoretical value and the actual statistical value, which is the index number of the **orderedRecord** [j] (lines: 18–20). At last, the index set I corresponding to the records in orderedRecord will be returned by the algorithm.

5. Experimental Results

The experiments are conducted on a laptop with limited resources (Intel Core i5 2.5 GHz CPU and 8 GB RAM).

We simulate three entities in this experiment, namely, DO, AU, and server. As shown in Figure 7, the AU stores a dataset to the server through the DO, and selects the spatial attribute as the data index. DO encrypts each record before storing it on the server. Then, the AU asks for a series of range queries on the spatial index, and DO retrieves the required encrypted records from the server, decrypts, and sends them to the user. In addition, we simulated a sniffer in Java on the server to observe data packets between the server and the DO for access pattern statistics. Finally, utilizing the observed access patterns, we performed the reconstruction attacks on the server side.

To evaluate the performance of our attack, we leverage a real-world spatial dataset, the distribution of North America Post Office including 175811 tuples [28]. We preprocess the raw dataset and normalize it to $[0, 1]^2$ before applying it to our experiment. The detailed distribution of this test dataset is shown in Figure 8. We encrypted and uploaded these tuples to the server and took spatial coordinates as their index.

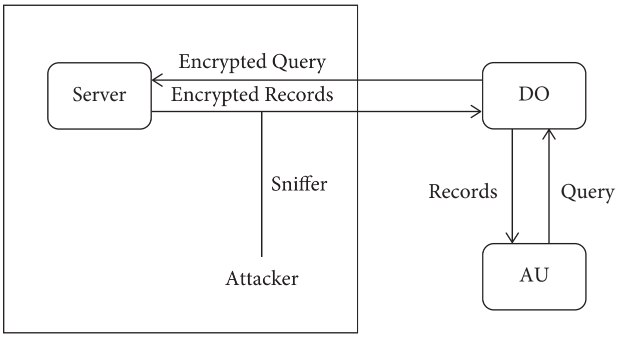


Figure 7: System implementation.

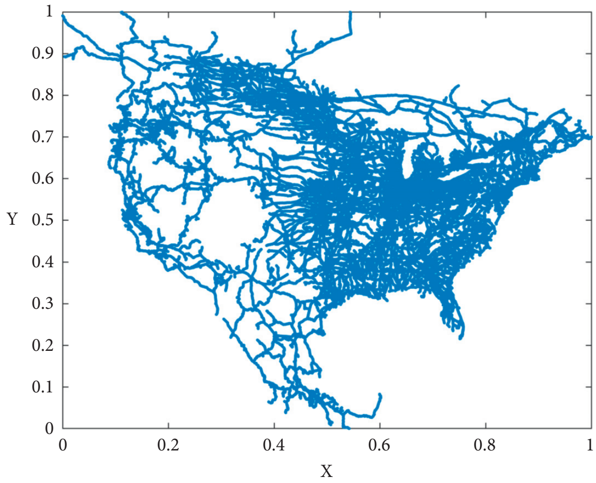


Figure 8: The distribution of test dataset.

According to the granularity that the attacker wants to achieve, we first discretize the dataset with different granularities. As shown in Table 1, column Attack Granularity describes the granularity of space discretization, and space is discretized into $10 \times 10 \cdots\cdots 100 \times 100$, while Index Number depicts the number of indexes in the different discretization conditions. And, Records Per Index represents the average number of records per position Index. Under different granularities, we will recover the index of each record through our attack.

Table 1: Space discretization.

Attack granularity	Index number	Records per index
10×10	100	1758.11
20×20	400	439.53
30×30	900	195.35
40×40	1600	109.88
50×50	2500	70.32
60×60	3600	48.84
70×70	4900	35.88
80×80	6400	27.47
90×90	8100	21.70
100×100	10000	17.58

After that, we gathered enough access patterns in order to run our attacks. The AU generates uniform range queries and issues to the DO. For each query, the DO retrieves encrypted matching records, decrypts them, and sends them back to the AU.

Due to the size of the query range, the network speed, and the number of users, the time of access patterns collection for our dataset will be very long. Therefore, in this experiment, we used a single user to simulate the process of query retrieval, and generate the minimum number of queries required for experimental conditions (Number of corresponding access patterns shown in Table 2). However, in the actual application scenario, when different users issue queries, the sniffer will sniff the access patterns from different users at the same time, and the collection speed will multiply.

Then, we preprocess the observed access patterns. By counting the co-occurrence of encrypted records in the access patterns, the records with co-occurrence rate of 100% are classified into categories, and for each category, one record is taken as the representative. The column Represent Number collects the number of representative records in the dataset under different attack granularities. For example, with the attack granularity of 10×10, the space is divided into 100 regions, each of which is pointed to by an index coordinate. By observing the access patterns, we classified 175,811 records, resulting in 68 representative records of 68 classes. Therefore, there are 32 regions with no record distribution under this attack granularity.

Finally, we preprocess the observed access patterns. Because records belonging to the same category are pointed to by the same position index, we can simplify the collection of access patterns by keeping only representative records.

Our reconstruction attack was performed on the preprocessed set of access patterns; Figure 9 summarizes our attack results. As shown in Figure 9, with the increase of attack granularity, our attack reconstruction rate showed a slight trend of decline, but it does not change much and the overall effect of the attack is good. Assuming that the data space is divided into $T_x \times T_y$, the higher granularity indicates that we divide the space more finely and that $T_x \times T_y$ is larger. As the granularity increases, the size of the index set increases, and it becomes more difficult for the attacker to assign indexes to records, so the reconstruction rate tends to decline.

The result curve looks a little wobbly because there are two main reasons that may affect the reconstruction rate of attack.

One is when some rows/columns do not conform to the prior knowledge, which will result in the reconstructed order being reversed during the process of GetOrder (Algorithm 2). In our reconstruction attack, we first determine the relative order of each row of records utilizing statistics on leaked access patterns. However, due to the symmetry of row query, we cannot determine whether the order we recover is in the positive or reverse order. If this is uncertain, the reconstruction rate of our attack will be very low. Therefore, we utilize prior knowledge $\mathbf{M}_{\mathbf{know}}$ to help us choose the correct option between the positive and reverse orders. If $\mathbf{M}_{\mathbf{know}}$ has no prior knowledge about any record of this row, the reconstruction rate of this row will be affected.

The other is when the first record in a row is in the second half of this row ($I_1 > T/2$), which will cause a recovery error during the process of GetIndex (Algorithm 3), and further lead to the error of other records in the same row. In the process of GetIndex, we determine the indexes for the records of each row/column. Similarly, due to the uniformity of row queries, when calculating the index of the first term of each row with equation (3), the optimal solution will be two indexes A and B, which are symmetric in the row. Assuming that A < B, then A is in the first half of the index, and B is in the second half. For dense datasets, the first record is naturally placed in the first half, so A is usually taken as the index of the first record. Therefore, when the first record in a row is in the second half of this row ($I_1 > T/2$), the reconstruction rate of this row will be affected.

Our attack includes four steps and Table 3 summarizes the running time of each step in our attack using access patterns.

(1) getRepresent: Count the co-occurrence probability of the encrypted records by the access patterns. Classify all records through co-occurrence probability and get a representative record for each category;

(2) Simplify access patterns: Simplify the access patterns, and keep only representative records;

(3) getRowIndex: Process simplified column access patterns and reconstruct the row index of records through the algorithm of GetOrder and GetIndex;

(4) getColIndex: Process simplified row access patterns and reconstruct the column index of records through the algorithm of GetOrder and GetIndex.

Table 2: The number of access patterns and represent records.

Attack granularity	Access pattern number	Represent record number
10 × 10	550 × 2	68
20 × 20	4200 × 2	224
30 × 30	13950 × 2	454
40 × 40	32800 × 2	756
50 × 50	63750 × 2	1105
60 × 60	109800 × 2	1488
70 × 70	173950 × 2	1931
80 × 80	259200 × 2	2433
90 × 90	368550 × 2	2927
100 × 100	505000 × 2	3474

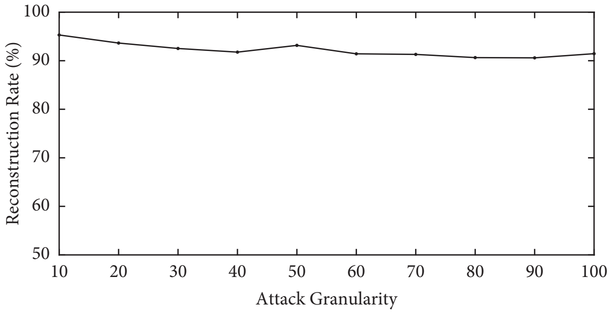


Figure 9: Attack reconstruction rate under different attack granularities.

Table 3: Attack time per step.

Attack granularity	GetRepresent (ms)	Simplification (ms)	GetRowIndex (ms)	GetColIndex (ms)
10 × 10	5	4	3	3
20 × 20	20	21	18	28
30 × 30	69	63	79	99
40 × 40	128	131	172	217
50 × 50	321	342	424	484
60 × 60	788	740	842	996
70 × 70	1708	1653	1379	1613
80 × 80	3358	3300	2535	2839
90 × 90	6086	6210	4336	4685
100 × 100	11229	10445	6987	7317

As shown in Table 3, GetRepresent represents the average time of finding representative records among 175811 records, while Simplification represents the average time of simplification for access patterns. GetRowIndex represents the average time to reconstruct the row index using these access patterns, while GetColIndex represents the average time to reconstruct the column index.

Table 3 shows that the time consumed for each step is up to only 11 seconds. Taking the most fine-grained granularity in this experiment as an example, when the space is discretized into 100 × 100 regions, we achieve that on average only 18 records are pointed to by the same position index, and the reconstruction rate of the record index reaches 91.45%. Under such fine-grained granularity and reconstruction rate, our attack time only needs a second level.

6. Conclusion

In this paper, a reconstruction attack on secure outsourced spatial dataset is proposed, proving that access pattern leakage will lead to security threats in VANET. With spatial discretization, our scheme can support the attack for optional spatial granularity. Meanwhile, utilizing the statistic of the record co-occurrence, access patterns are simplified, which improves the attack efficiency. Using the continuity and uniformity of the range query, the attacker determines the index for each record in the dataset. Extensive experiments on a real-world dataset demonstrate that our attack scheme can achieve a reconstruction rate of more than 90 percent even at a relatively fine-grained granularity. In future work, we will investigate the defense mechanism to address these security threats.

Data Availability

The parameters and datasets used to support the findings of this study are included within the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported in part by National Natural Science Foundation of China under Grant nos. 61602290, 61902229, 61672334, 61802242, 61802241, and 61762058, Fundamental Research Funds for the Central Universities under Grant nos. GK202103090 and GK202103084, Natural Science Basic Research Plan in Shaanxi Province of China under Grant nos. 2017JQ6038 and No. 2020JM-288, and Foundation of Guizhou Provincial Key Laboratory of Public Big Data under Grant no. 2018BDKFJJ004.

References

[1] R. Kaur, T. P. Singh, and V. Khajuria, "Security issues in vehicular ad-hoc network(VANET)," in *Proceedings of the 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 884–889, Thirunelveli, India, May 2018.

[2] A. Yang, J. Weng, N. Cheng, J. Ni, X. Lin, and X. Shen, "DeQoS attack: degrading quality of service in VANETs and its mitigation," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4834–4845, 2019.

[3] M. Li, "Security in VANETs," *Vehicular Communications*, vol. 1, no. 2, 2014.

[4] A. Yang, J. Weng, K. Yang, C. Huang, and X. Shen, "Delegating authentication to edge: a decentralized authentication architecture for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[5] S.-P. Li and M.-H. Wong, "Privacy-Preserving queries over outsourced data with access pattern protection," in *Proceedings of the 2014 IEEE International Conference on Data Mining Workshop*, pp. 581–588, Shenzhen, China, December 2014.

[6] H. Quan, H. Liu, B. Wang, M. Li, and Y. Zhang, "Randex: mitigating range injection attacks on searchable encryption," in *Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS)*, pp. 133–141, Washington DC, DC, USA, June 2019.

[7] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: ramifification, attack and mitigation," in *Proceedings of the NDSS'12*, San Diego, CA, USA, February 2012.

[8] G. Kellaris, K. George, K. Nissim, and O. N. Adam, *Generic Attacks on Secure Outsourced Databases*, Association for Computing Machinery, New York, NY, USA, 2016.

[9] IETF, "Geographic spatial/privacy (geopriv) [EB/OL]," 2004, http://datatracker.ietf.org/wg/geopriv/charter/.

[10] World Wide Web Consortium (W3C), *Platform for Privacy Preferences (P3P) Project*, World Wide Web Consortium (W3C), Cambridge, MA, USA, 2000.

[11] K. Lim, K. M. Tuladhar, X. Wang, and M. Liu, "A scalable and secure key distribution scheme for group signature based authentication in VANET," in *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) IEEE*, New York, NY, USA, October 2017.

[12] N. Guo, L. Ma, and T. Gao, "Independent mix zone for location privacy in vehicular networks," *IEEE Access*, vol. 6, pp. 16842–16850, 2018.

[13] M. L. Yiu, C. S. Jensen, J. Møller, and H. Lu, "Design and analysis of a ranking approach to private location-based services," *ACM Transactions on Database Systems*, vol. 36, no. 2, pp. 1–42, 2011.

[14] C. Ardagna, M. Cremonini, S. De Capitani Di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13–27, 2011.

[15] C. Zhou, C. Ma, and S. Yang, "Research of LBS privacy preserving based on sensitive location diversity," *Journal on Communications*, vol. 36, no. 4, pp. 14–25, 2015.

[16] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, *Geo-Indistinguishability: Differential Privacy for Spatial-Based Systems*, Association for Computing Machinery, New York, NY, USA, 2013.

[17] W. Ni, X. Chen, and Z. Ma, "Location privacy preserving k nearest neighbor query method on road network in presence of user's preference," *Chinese Journal of Computers*, vol. 38, no. 4, pp. 884–896, 2015.

[18] R. Paulet, M. G. Kaosar, X. Xun Yi, and E. Bertino, "Privacy-Preserving and content-protecting location based queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1200–1210, 2014.

[19] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008*, ACM, Vancouver, Canada, June 2008.

[20] X. Yi, R. Paulet, E. Bertino, and V. Varadharajan, "Practical approximate k nearest neighbor queries with location and query privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1546–1559, 2016.

[21] D. Dawn Xiaoding Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pp. 44–55, Berkeley, CA, US, May 2000.

[22] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[23] M. Naveed, *The Fallacy of Composition of Oblivious RAM and Searchable Encryption*, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 2015.

[24] S. Garg, P. Mohassel, and C. Papamanthou, *Tworam: Effificient Oblivious Ram in Two Rounds With Applications to Searchable Encryption*, Springer, New York, NY, USA, 2016.

[25] B. Fuller, M. Varia, A. Yerukhimovich et al., "SoK: cryptographically protected database search," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 172–191, San Jose, CA, USA, May 2017.

[26] Y. Zhang, J. Katz, and C. Papamanthou, "All your queries are belong to us: the power of file-injection attacks on searchable encryption," in *Proceedings of the USENIX Security*, pp. 707–720, Austin, TX, USA, August 2016.

[27] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," in *Proceedings of the CCS'15*, Denver, CO, USA, October 2015.

[28] "Real datasets for spatial databases: road networks and points of interest," 2005, http://www.cs.utah.edu/lifeifei/SpatialDataset.htm.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9967829, 13 pages
https://doi.org/10.1155/2021/9967829

Research Article

Research on Manhattan Distance Based Trust Management in Vehicular Ad Hoc Network

Xiaodong Zhang,[1,2] **Ru Li**,[1,2] **Wenhan Hou**,[1,2] **and Jinshan Shi**[1,2]

[1]*Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing, Hohhot 010021, China*
[2]*College of Computer Science, Inner Mongolia University, Hohhot 010021, China*

Correspondence should be addressed to Ru Li; csliru@imu.edu.cn

Received 1 April 2021; Revised 31 May 2021; Accepted 23 June 2021; Published 6 July 2021

Academic Editor: Rongxing Lu

In recent years, Vehicular Ad Hoc Network (VANET) has developed significantly. Coordination between vehicles can enhance driving safety and improve traffic efficiency. Due to the high dynamic characteristic of VANET, security has become one of the challenging problems. Trust of the message is a key element of security in VANET. This paper proposes a Manhattan Distance Based Trust Management model (MDBTM) in VANET environment which solves the problem in existing trust management research that considers the distance between the sending vehicle and event location. In this model, the Manhattan distance and the number of building obstacles are calculated by considering the movement relationship between the sending vehicle and event location. The Dijkstra algorithm is used to predict the path with the maximum probability, when the vehicle is driving toward the event location. The message scores are then calculated based on the Manhattan distance and the number of building obstacles. Finally, the scores are fused to determine whether to trust the message. The experimental results show that the proposed method has better performance than similar methods in terms of correct decision probability under different proportions of malicious vehicles, different numbers of vehicles, and different reference ranges.

1. Introduction

With the development of wireless communication technology and the automotive industry, the Vehicular Ad Hoc Network (VANET) has made significant development, which enhances driving safety and traffic efficiency. Intelligent traffic management has been realized through the communication collaboration of Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Vehicle to Pedestrians (V2P), Vehicle to Cloud (V2C), and so on. The application scenarios in VANET mainly include safety application scenario and nonsafety application scenario [1]. These applications are based on the exchange of messages between entities. However, security is one of the main issues in VANET, and how to ensure the security of these messages has become an important issue in this filed. While mechanisms based on certificates [2, 3], signatures [4], and Public Key Infrastructure (PKI) [5] already exist to address the issue of message security, they can only solve the problem of transmitted message not being tampered maliciously and ensure that the message comes from an authorized vehicle; they cannot resolve the authenticity of the messages themselves (i.e., the trust of the message). For example, malicious vehicle can broadcast information that claims that the road is not congested, but that traffic accident or congestion has actually occurred. Such malicious behaviour may seriously jeopardize traffic safety or efficiency. The trust of message is therefore a key element of security [6]. How to effectively evaluate the trust of the messages sent by vehicles has become an important issue. In other words, trust management of the messages sent by vehicle is very important.

At present, many researches focus on trust management in the VANET environment, mainly including three types: entity-centric trust management [7–9], data-centric trust management [10–15], and combined trust management [16, 17].

Many researches [7, 10, 11, 13, 14] consider the distance between the sending vehicle and event location, suggesting that such distance can indirectly reflect trust of message. The farther away from the event, the lower trust value of message. However, in these researches, the calculation of the distance is not discussed in detail. As a matter of fact, the traditional Euclidean distance cannot reflect the actual distance when vehicles are on city roads. In addition, on city roads, there may be building obstacles from the sending vehicle to event location. The line of sight between the sending vehicle and event location is affected by the existence of building obstacles. Whether building obstacles exist or not, this can result in entirely different trust. However, the existing trust management model does not take into account the existence of building obstacles.

Manhattan distance is the city block distance, that is, from one point to another on the actual road. Manhattan distance can reflect the actual distance between the sending vehicle and event location. At the same time, on the path of the actual distance, the number of building obstacles can also be determined. Therefore, this paper proposes a Manhattan Distance Based Trust Management model (MDBTM) in VANET. In this model, the receiving vehicle first calculates the Manhattan distance and the number of building obstacles on Manhattan distance path, then calculates score based on the Manhattan distance and the number of building obstacles for each message about a certain event, and finally fuses all the scores to calculate its trust value to determine whether it trusts the received message.

The contributions of this paper mainly include the following:

(1) Considering that the vehicle is on the road and the Euclidean distance cannot reflect the actual driving distance of vehicle, a method of calculating the distance between the sending vehicle and event location using Manhattan distance is proposed.

(2) This paper proposes a trust management model that takes into account both the Manhattan distance and the number of building obstacles.

(3) The experimental results show that the proposed method has better performance than similar methods in terms of correct decision probability under different proportions of malicious vehicles, different numbers of vehicles, and different reference ranges.

The rest of the paper is organized as follows: Section 2 introduces current research on trust management in VANET and analyzes the existing problems. Section 3 introduces the system model and the problem formation. Section 4 introduces the MDBTM scheme. Section 5 verifies the effectiveness of the proposed scheme by experimental simulation. Section 6 summarizes full paper and proposes future work.

2. Related Works

At present, many researches focus on trust management in the VANET environment, mainly including three types: entity-centric trust management, data-centric trust management, and combined trust management.

In entity-centric trust management research, trust level of the entity mainly is studied and the trust value of message is judged indirectly. Minhas et al. [7] proposed a trust model that took into account the trustworthiness of the agents of other vehicles. This model considers location closeness, time closeness, experience-based trust, and role-based trust when aggregating messages. Marmol et al. [8] proposed an infrastructure-based trust and reputation model. This model considers recommendation value given by other vehicles and RSUs and trust value of vehicle at the last moment in calculating the trust value of message. Haddadou et al. [9] proposed a distributed trust management method which used the job market signaling model to motivate more cooperation among selfish nodes.

In data-centric trust management research, the focus is on the consistent judgment of received messages. Raya et al. [10] proposed a data-centric trust framework. The framework first calculates the trust levels of a report on the same event by default trustworthiness, event- or task-specific trustworthiness, dynamic trustworthiness factors, location, and time and then combines those trust levels to decide whether the reported event has occurred. Wu et al. [11] proposed an RSU-Aided scheme for data-centric trust establishment in VANETs. In this scheme, RSU calculates the observation factor of the received reports according to confidence (one of the factors that affect confidence is the distance from the sending vehicle to event location) and weight and then integrates the observation factor and feedback factor through the ant colony optimization algorithm to recalculate the trust level of each evidence. Gurung et al. [12] proposed an information-oriented trust model which considered three factors: content similarity, route similarity, and content conflict. Shaikh et al. [13] proposed a distributed intrusion-aware trust model for vehicular ad hoc networks that worked in three phases. The first phase calculates the confidence value of each message based on location closeness, time closeness, location verification, and time verification, and the second phase calculates trust value based on confidence of each message. A decision is taken in the third phase. Yang et al. [14] proposed a distributed trust management scheme based on the blockchain. First, the credibility of the message is calculated by the distance between the sending vehicle and event location, and the credibility of all messages is fused through Bayesian inference to generate a message rating. The message rating is aggregated to calculate trust value offset, and finally offset value is stored in the blockchain. Chen et al. [15] proposed a topology-based secure message transmission method, which modeled the actual transmission path of a message in network to determine the probability of the correct message decision.

In combined trust management research, the focus is on the trust level of the entity and the consistent judgment of received messages at the same time. Chen et al. [16] proposed a beacon-based trust management system which considered entity trust and data trust at the same time. This system constructs entity trust from beacon messages and calculates data trust by cross-checking the plausibility of event messages and beacon messages. Li et al. [17] proposed an attack-resistant trust management scheme that could detect and cope with malicious attacks and evaluate the trust of data and mobile nodes in VANET.

In short, current researches of trust management mainly focus on trust level of the entity and the consistency of the message content. At present, in the researches of distance considerations shown in Table 1, there is the problem of no detailed discussion on the method of calculating distance. In this paper, a method of calculating distance is proposed to solve the above problems. This method takes into account the vehicle in the city road environment and the situation where buildings block the line of sight, which makes up for the inadequacy of existing work.

3. System Model and Problem Formation

In this section, this paper first introduces the system model including network model, data propagation model, and attack model. Then it briefly describes the problem to be solved in this paper.

3.1. Network Model and Data Propagation Model. This system operates in the city road environment. Vehicles on the road have the function of communicating via VANET. Vehicles in the network can send messages on their own initiative, for either entertainment-related or security-related ones. This paper considers security-related messages. The content of a specific report is called event $e_i\,(i = 1, 2, \ldots, \mathrm{Enum})$, where i is used to distinguish between event types, and Enum is the number of events. For example, "whether or not a traffic accident occurred at X location" is an event, with two situations occurring and not occurring for each event, expressed in terms of 1 and 0, respectively.

The vehicle receiving a message will decide whether to respond to the message, for example, by changing the driver path based on what is reported in the message. However, due to the existence of malicious vehicles, the vehicle will receive false messages and be required to manage the trust of message. The roads in the city are very complicated. There are many vehicles on the roads. Messages sent by vehicles away from the event location have no referential meaning and increase the amount of computation during trust management. Therefore, this paper considers a reference range R. The reference range R is a circular area centered on the event location and only the messages sent by vehicles within this range are considered when calculating the trust value of message. The specific network model diagram is shown in Figure 1.

When vehicles report safety-related messages, there is no need to consider which is the destination vehicle. Therefore, this paper considers the way of broadcasting to transmit the messages. In addition to the content of the event, the transmitted message also requires the transmission of vehicle identification and Global Positioning System (GPS) information. The information transmitted belongs to the vehicles' privacy data. In order to protect their private data, the data are encrypted during transmission, and other vehicles must be authorized to access them. The specific methods of privacy preserving are not the focus of this paper. Please refer to [18–21] for details. Because propagation speed of message is much faster than moving speed of vehicle, it ignores the time it takes to propagate messages from a vehicle to other vehicles. The process is considered to be a static network [22]. Therefore, when a vehicle receives a message, it can be assumed that the message is at the current moment. In other words, there is no need to consider how the delay in message propagation causes the state of the event to change.

3.2. Attack Model. Vehicles on the road include normal vehicles and malicious vehicles. Normal vehicles will send true message about an event. However, malicious vehicles will send false message about an event.

In the VANET environment, the malicious vehicles can generate three types of threats including attacks addressing secure communications, attacks addressing safety applications, and attacks addressing infotainment applications. Different types of threats target different services, including authenticity, confidentiality, privacy, availability, integrity, and nonrepudiation [23]. This paper mainly solves the problem that the malicious vehicle launches betrayal attack aiming at authenticity; i.e., vehicle deliberately sends false messages to affect the traffic safety.

The vehicle sending the message is called the source vehicle, and the vehicle receiving the message is called the destination vehicle. Due to the high dynamic characteristics of VANET, the source and destination vehicles may not be able to communicate directly, and relayed vehicles may be required for forwarding messages. Therefore, vehicles that affect the credibility of the destination vehicles' judgment include source vehicle and relay vehicle. In other words, malicious vehicles may exist in both source vehicles and relay vehicles. When the source vehicle is a malicious vehicle, a false message will be sent. When the relay vehicle is a malicious vehicle, it will tamper with the content of the received message before forwarding it, thus resulting in a false message. This paper mainly studies the effect of the distance on the trust value of message and assumes that the system has adopted the methods of certificate and signature to ensure the relay vehicle cannot tamper with the message. Therefore, this paper mainly studies the situation where the source vehicle is a malicious vehicle.

Table 1: Comparison of researches considering distance.

Approach	Trust metric	Architecture	Advantage	Disadvantage
Minhas et al. [7]	✓Time closeness ✓Location closeness (distance) ✓Experience ✓Role	Centralized	Easy to find malicious vehicles.	No discussion of the calculation method of distance.
Raya et al. [10]	✓Time ✓Distance ✓Node type ✓Event type	Distributed	Easy to find false messages.	No discussion of the calculation method of distance.
Wu et al. [11]	✓Distance ✓Number of sensors ✓Node type	Centralized	Easy to find false messages.	No discussion of the calculation method of distance.
Shaikh et al. [13]	✓Location closeness (distance) ✓Time closeness ✓Location verification ✓Time verification ✓Number of senders	Distributed	Easy to implement in VANETs.	No discussion of the calculation method of distance.
Yang et al. [14]	✓Distance	Distributed	Provide security trust management method using blockchain.	No discussion of the calculation method of distance.

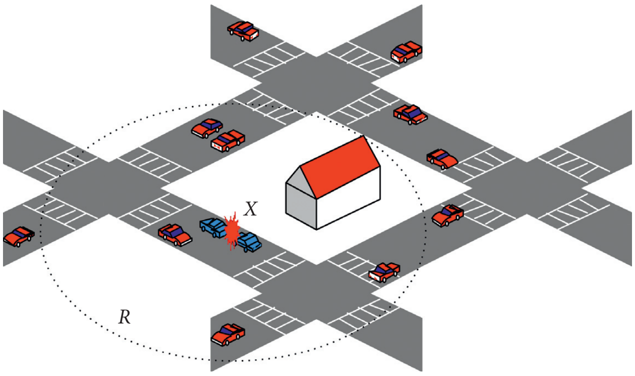


Figure 1: The diagram of network model.

3.3. Problem Formation. Vehicles on the road will send safety-related messages. When the destination vehicle receives the message m_0, it needs to determine whether it is trusted. Assume that the message is about a certain event $e\prime, e\prime \in e_i\,(i = 1, 2, \ldots, \text{Enum})$, where Enum represents the number of the event types. If a judgment is made immediately upon receipt of a message, the trust value of message cannot be judged because no message is referenced. Therefore, it requires a waiting time T and then uses the messages received in the time period T about event e' as a reference message set $M'\{m_1, m_2, \ldots, m_{\text{Num}}\}$ to determine whether the message is trusted. The Num is the number of messages received, which can be calculated in equation (1):

$$\text{Num} = \text{Fre} \times V\text{num} \times T, \quad (1)$$

where Fre represents the frequency at which messages are sent by the vehicle, Vnum represents the number of vehicles in the reference range R, and T represents the waiting time. However, if the vehicle sends messages very frequently, it may receive multiple messages about event e' from the same vehicle within the T time. Therefore, it is necessary to remove duplicate messages from the reference set M' and then use the rest of the messages as the final reference message set $M\{m_1, m_2, \ldots, m_N\}$, in which N is the number of messages from different vehicles within a reference range R about event e'.

If the report of event e' in the reference set M is consistent with that of the message m_0, the trust value of message can be directly judged. However, because of the existence of malicious vehicles, they can send false messages about certain events. When other vehicles receive messages about event e', they receive conflicting messages and cannot directly determine the trust value of message m_0.

The Manhattan distance and the number of building obstacles can indirectly reflect the trust value of message. The vehicles are driving on the road, so the actual road needs to be modeled first. The actual road is a road network composed of nodes and road sections. Therefore, this paper uses the graph in the data structure to model the actual road. In the graph, nodes are represented by the vertices, and the road segments between two nodes are represented by the edges of graph. The node is an intersection on a city road. Its basic attributes include the node identifier, node longitude, and node latitude. The road segment is a road between two nodes. Its basic attributes include the road identifier, starting node, end node, road length, whether it can go straight, whether it can turn right, or whether it can turn left. The attributes of the forward and reverse road segments are not necessarily the same between the two nodes, so the weighted directed graph $G = (V, E)$ is used to model the actual road. The weight value is a specific attribute value of the road segment. In this paper, the road identifier is selected as the weight to easily correspond to the road segment attributes.

Through the above method, the actual road can be modeled, and the Manhattan distance and the number of building obstacles can be calculated by combining with the vehicle's motion state. The research goal of this paper is to calculate the message score $S_i^{e'}(i = 0, 1, \ldots, N)$ by the Manhattan distance and the number of building obstacles between the vehicle sending message m_i about event e' and the event location. Then all the scores are fused to calculate the trust value of message m_0 about event e' denoted by $\text{Trust}(e' \longrightarrow m_0)$. If $\text{Trust}(e' \longrightarrow m_0) > 0$ then the message m_0 can be trusted; otherwise the message cannot be trusted. $\text{Trust}(e' \longrightarrow m_0)$ is formally defined by

$$\text{Trust}\left(e' \longrightarrow m_0\right) = \text{Fuse}\left(S_0^{e'}, S_1^{e'}, \ldots, S_N^{e'}\right). \tag{2}$$

4. Proposed MDBTM

The proposed MDBTM scheme is discussed in detail in this section. First, according to the event e' reported by the received message m_0, the reference message set $M\{m_1, m_2, \ldots, m_N\}$ is obtained, and the Manhattan distance and the number of building obstacles of vehicles that sent these messages including message m_0 and messages in M are calculated. Then scores of all these messages are calculated by the Manhattan distance and the number of building obstacles. Finally, all these scores are fused to calculate the trust value of event e' reported by the message m_0. That is, the trust value of message m_0.

4.1. The Calculation of Manhattan Distance. For a town street that is regularly laid out in the direction of south and north, east and west, the Manhattan distance is the distance from north to south plus the distance from east to west. However, the actual road is not the same. The attributes of the nodes are different, and the road cannot go straight, turn left, or turn right at any time. Therefore, it is necessary to calculate the Manhattan distance in combination with the actual road. In addition, the movement relationship between the sending vehicle and event location is different, which will lead to different Manhattan distance. Therefore, when calculating the Manhattan distance, it also needs to consider the movement relationship.

There are three types of movement relationship: driving away from the event location, not passing the event location, driving toward the event location.

Driving away from the event location: If the vehicle passes the event location based on the historical trajectory information of that vehicle, the movement relationship is driving away from the event location. The Manhattan distance can be obtained from the historical trajectory information of the vehicle. The historical trajectory information can be obtained from RSU and is also privacy data of vehicle. In order to protect it, the data are encrypted during transmission, and other vehicles must be authorized to access them from RSU.

Not passing the event location: If the vehicle does not pass through the event location based on the historical trajectory information of the vehicle and the vehicle's movement direction is far away from the event location, the movement relationship is not passing the event location. In this case, we believe that the vehicle will not pass the event location or the probability is small, so the Manhattan distance is infinite.

Driving toward the event location: If the vehicle does not pass the event location based on the historical trajectory information of the vehicle and the vehicle's movement direction is close to the event location, the movement relationship is driving toward the event location. In this case, the vehicle may or may not pass the event location. Therefore, it is necessary to predict whether the vehicle will pass the event location based on the GPS information of sending vehicle and the actual road.

The Manhattan mobility model is a model that simulates the movement of vehicles on city roads. In this model, when the vehicle reaches the intersection, it will go straight with a probability of 0.5 and turn left or right with a probability of 0.25 [24]. If the vehicle is not allowed to go straight, turn left, or turn right at the intersection, the corresponding selection probability will be divided equally to other options. For example, if an intersection is not allowed to turn left, then it will go straight with a probability of 0.625 and turn right with a probability of 0.375 when the vehicle arrives at the intersection. It can be seen that this model can describe the movement of vehicles at the intersection on city roads. Therefore, this paper uses this model and the actual road to predict the probability of the vehicle passing the event location. There may be multiple paths from the vehicle to event location. This paper selects the path of maximum probability to calculate the Manhattan distance.

In summary, the flow chart for calculating the Manhattan distance between the sending vehicle and event location is shown in Figure 2.

4.1.1. The Vehicle's Movement Direction. The vehicle's movement direction includes close to the event location and far away from the event location. The location of the vehicle can be obtained by the GPS information on it. Assume that the sending vehicle is located in $A(lng_1, lat_1)$ at the previous time t' and that vehicle is in $\text{B}(lng_2, lat_2)$ at the current time t and the event occurred in $\text{C}(lng_3, lat_3)$. So, the movement direction vector of the vehicle is $\overrightarrow{AB}(lng_2 - lng_1, lat_2 - lat_1)$, and the vector from its current position to event location is $\overrightarrow{BC}(lng_3 - lng_2, lat_3 - lat_2)$. Define the angle between vector $\overrightarrow{AB}$ and vector $\overrightarrow{BC}$ as θ. If $0^\circ \leq \theta < 90^\circ$, i.e., $\cos\theta > 0$, the vehicle's movement direction is close to the event location. If $90^\circ \leq \theta \leq 180^\circ$, i.e., $\cos\theta \leq 0$, the vehicle's movement direction is away from the event location. The $\cos\theta$ is calculated as

$$\cos\theta = \frac{\overrightarrow{AB} \cdot \overrightarrow{BC}}{|\overrightarrow{AB}| \cdot |\overrightarrow{BC}|} = \frac{(\ln g_2 - \ln g_1) \cdot (\ln g_3 - \ln g_2) + (lat_2 - lat_1) \cdot (lat_3 - lat_2)}{\sqrt{(\ln g_2 - \ln g_1)^2 + (lat_2 - lat_1)^2} \cdot \sqrt{(\ln g_3 - \ln g_2)^2 + (lat_3 - lat_2)^2}}. \tag{3}$$

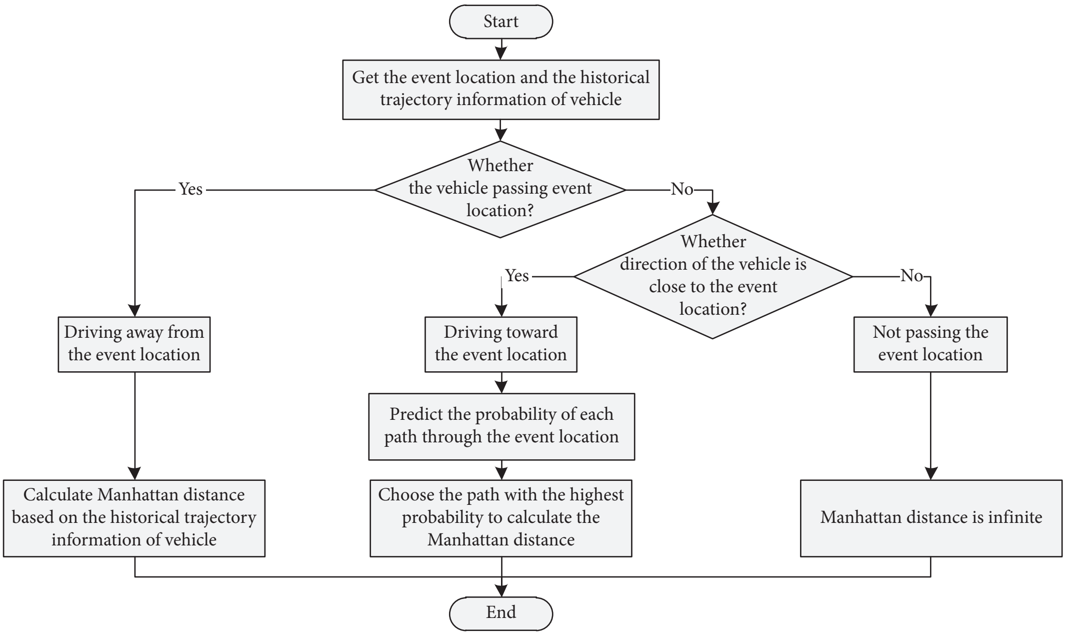

Start
Get the event location and the historical trajectory information of vehicle
Whether the vehicle passing event location?
Yes
No
Whether direction of the vehicle is close to the event location?
Yes
No
Driving away from the event location
Driving toward the event location
Not passing the event location
Predict the probability of each path through the event location
Calculate Manhattan distance based on the historical trajectory information of vehicle
Choose the path with the highest probability to calculate the Manhattan distance
Manhattan distance is infinite
End

Figure 2: Flow chart for calculating the Manhattan distance.

As shown in Figure 3, when a vehicle moves from position A to position B, the angle between movement direction vector of the vehicle and the vector from its current position to event location is less than 90°, so the vehicle's movement direction is close to the event location. When a vehicle moves from position A′ to position B′, the angle between movement direction vector of the vehicle and the vector from its current position to event location is greater than or equal to 90°, so the vehicle's movement direction is away from the event location.

4.1.2. The Prediction of the Path with Maximum Probability. There may be multiple paths for vehicle from the current location to event location. Based on the Manhattan mobility model and the actual road, this paper predicts the path with the maximum probability of the vehicle passing the event location.

Firstly, a weighted directed graph $G' = (V', E')$ based on the Manhattan mobility model and the actual road is established to record all the paths of sending vehicle from the current location to the event location and the transition probability at intersection. In weighted directed graph G', the vertex is the road segments in the actual road model, and the edge indicates the transition from one road segment to another road segment, and the transition direction is used as the direction of the edge. Whether the road segments can be transitioned (i.e., whether there is an edge between the two vertices) is determined by the three attributes of the road segment in the actual road model (whether it can go straight, whether it can turn right, or whether it can turn left). Combining these three attributes with the transition probability of vehicle at the intersection specified by the Manhattan mobility model, we can determine the transition probability of the vehicle at the intersection which is used as weight of the edge in graph G'. The sum of the probability of transition to other nodes is 1 in graph G', as shown in the following equation:

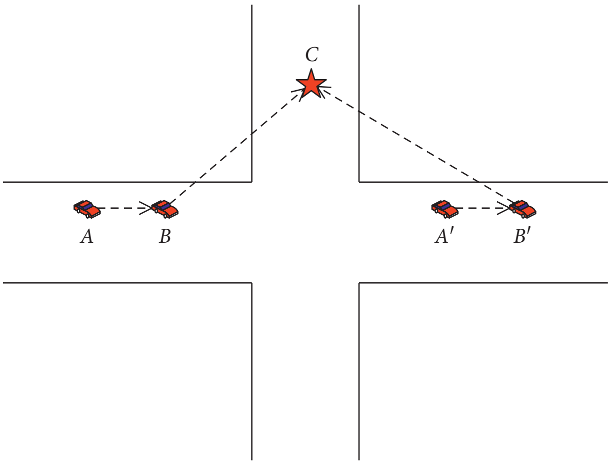


Figure 3: Diagram of the relationship between vehicle movement direction and event location.

$$\sum_{j=1}^{n} W\left(\langle V_i, V_j \rangle\right) = 1, \tag{4}$$

where n is the number of nodes that the node V_i can transfer to other nodes, V_j is the other nodes to which the node V_i can transfer, and $W(\langle V_i, V_j \rangle)$ represents the weight of the edge $\langle V_i, V_j \rangle$.

According to the Manhattan mobility model combined with actual road, a weighted directed graph can be constructed as shown in Figure 4. Vertex A is the road segment where the sending vehicle is located, and Vertex M is the road segment where the event location is located. There are three paths from Vertex A to Vertex M, namely, ACEIM, AFGIM, and AFJLM.

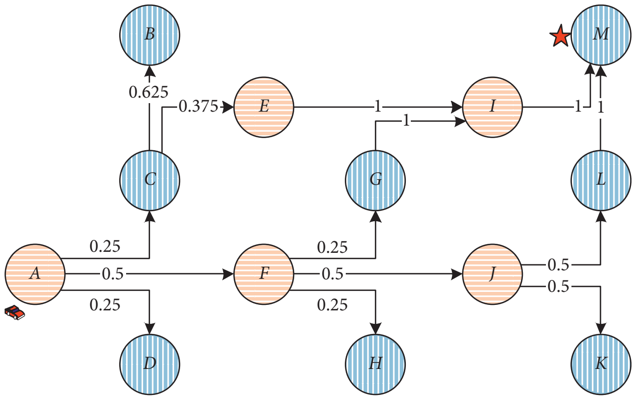


Figure 4: Weighted directed graph constructed.

The vehicle V_i that sends a message m_i may have Pnum paths to the event location. The j-th path path_i^j can be expressed as $\text{path}_i^j = \left\{ V_j^1 V_j^2 \ldots V_j^{Vn_j} \right\} (j = 1, 2, \ldots, P\text{num})$, where Vn_j represents the number of vertices contained in the j-th path. The probability $\Pr(\text{path}_i^j)$ that the vehicle moves on path path_i^j is defined as

$$\Pr\left(\text{path}_i^j\right) = \prod_{k=1}^{Vn_j - 1} W\left(\langle V_j^k, V_j^{k+1} \rangle\right), \qquad (5)$$

where Vn_j represents the number of vertices contained in the j-th path, and $W(\langle V_j^k, V_j^{k+1} \rangle)$ represents the weight of the edge $\langle V_j^k, V_j^{k+1} \rangle$.

Calculating the path with the maximum probability is equal to finding path by minimizing inverse probability. Therefore, the method for calculating the path with the maximum probability is given in the following equation

$$\max_{j=1,\ldots,Pnum}\left(\Pr\left(\text{path}_i^j\right)\right) = \max_{j=1,\ldots,Pnum}\left(\prod_{k=1}^{Vn_j-1} W\left(\langle V_j^k, V_j^{k+1}\rangle\right)\right) = \min_{j=1,\ldots,Pnum}\left(\frac{1}{\prod_{k=1}^{Vn_j-1} W\left(\langle V_j^k, V_j^{k+1}\rangle\right)}\right) = \min_{j=1,\ldots,Pnum}\left(\prod_{k=1}^{Vn_j-1} \frac{1}{W\left(\langle V_j^k, V_j^{k+1}\rangle\right)}\right). \qquad (6)$$

The Dijkstra algorithm is used to calculate the shortest path from one vertex to the other vertices of the weighted graph. Since calculating the path with the maximum transition probability is equal to finding path by minimizing reciprocal of transition probability, the Dijkstra algorithm can be used to calculate the path with the maximum probability. The method of using the Dijkstra algorithm to obtain the shortest path is to add the weights of each path and select the path with the minimum result. However, when selecting the path with the maximum transition probability, we need to multiply the reciprocal of weight (i.e., the reciprocal of transition probability) and choose the path with the minimum result. Therefore, when using Dijkstra algorithm, it is necessary to change the addition of weights to multiplication. Algorithm 1 introduces the steps of calculating the path with the maximum probability in detail.

By using Algorithm 1, the path with the maximum transition probability denoted by $\text{path}_i^{\max} = \left\{ V_{\max}^1 V_{\max}^2 \ldots V_{\max}^{Vn_{\max}} \right\}$ can be obtained. The Manhattan distance as expressed by $\text{man}Dis_i$ can be obtained from $\text{path}_i^{\max}$ and the actual road model. However, because the $\text{path}_i^{\max}$ is a prediction, and the vehicle may not move along the $\text{path}_i^{\max}$, the probability of $\text{path}_i^{\max}$ needs to be considered. The method of calculating the final Manhattan distance as expressed by $Man_i^{e'}$ is given in the following equation:

$$\text{Man}_i^{e'} = \frac{\text{man}Dis_i}{\Pr\left(\text{path}_i^{\max}\right)}, \quad (i = 1, 2, \ldots, N), \qquad (7)$$

where $\Pr(\text{path}_i^{\max})$ represents the probability of the vehicle moving along the path $\text{path}_i^{\max}$.

4.2. The Calculation of the Number of Building Obstacles.

Due to the existence of building obstacles, the line of sight of vehicle will be affected, which will affect the trust value of message. In a city road environment, building obstacles generally occur at intersection. Vehicle cannot obtain the conditions (traffic accident information) of another road segment to which the vehicle turns left or right from the current road segment. This paper takes the intersection where the vehicle turns left or right as the

```
INPUT:
The storage matrix $\text{cost}[n][n]$ of the weighted directed graph $G'$ and the vertex set $V$, where $n$ represents the number of vertices in
  the graph.
OUTPUT:
    The path with the maximum probability ($\text{path}[n]$)
(1)   Initialize the shortest path length array dist, let $\text{dist}[j] = \cos t[0][j]$, where $j = 0, 1 \ldots, n-1$;
(2)   Initialize the path array with the maximum probability, let $\text{path}[j] = 0$, where $j = 0, 1 \ldots, n-1$;
(3)   Set $U = \{V_1\}$, vertex $V_1$ is the road segment where the vehicle sending message is located;
(4)   Select the vertex $k$ with the shortest path from the set $V - U$, ($k = \min\{\text{dist}[j]\}, j \in V - U$);
(5)   Add vertex $k$ to set U, let $U = U \cup \{k\}$;
(6)   For (each $j \in V - U$)
(7)     $IF(\text{dist}[j] > \text{dist}[k] \times 1/\cos t[k][j])$;
(8)       let $\text{dist}[j] = \text{dist}[k] \times 1/\cos t[k][j]$;
(9)       let $\text{path}[j] = k$;
(10)    End If
(11)    End For
(12)  If ($V \neq U$)
(13)  Go to step 4;
(14)  End If
```

Algorithm 1: Calculating the path algorithm with the maximum probability.

turning point. The number of turning points between the sending vehicle and event location is that of building obstacles. In calculating the number of building obstacles as expressed by $Obs_i^{e'}$, three kinds of movement relationships between the sending vehicle and event location are also considered.

4.2.1. Driving away from the Event Location. When the vehicle drives away from the event location, this means that the vehicle passes through the event location. Since there are no building obstacles when the vehicle passes through the event location, the number of building obstacles is set at 0 ($Obs_i^{e'} = 0$).

4.2.2. Not Passing the Event Location. When the vehicle does not pass the event location, the Manhattan distance is infinite, and there is no path between the sending vehicle and event location, so the number of building obstacles is also infinite.

4.2.3. Driving toward the Event Location. When the vehicle drives toward the event location, the number of building obstacles is the number of turning points with the maximum transition probability on the path $\text{path}_i^{\max} = \{V_{\max}^1 V_{\max}^2 \cdots V_{\max}^{Vn_{\max}}\}$. For each edge $\langle V_{\max}^j, V_{\max}^{j+1} \rangle$ $(j = 1, 2, \ldots, Vn_{\max} - 1)$, if the vehicle turns left or right on the actual road, the number of building obstacles increases by 1 ($Obs_i^{e'} = Obs_i^{e'} + 1$). The final $Obs_i^{e'}$ is the number of building obstacles between the sending vehicle and event location.

4.3. The Calculation of Message Scores. The score of the message can be calculated by the Manhattan distance and the number of building obstacles. However, the value of the Manhattan distance and the number of buildings obstacles are of different orders of magnitude. Therefore, before calculating the score, the value needs to be normalized first. The normalization method is given in the following equation:

$$\text{Man}_i^{e'} = \frac{\text{Man}_i^{e'} - \min\left(\text{Man}^{e'}\right)}{\max\left(\text{Man}^{e'}\right) - \min\left(\text{Man}^{e'}\right)}, \tag{8}$$

$$Obs_i^{e'} = \frac{Obs_i^{e'} - \min\left(Obs^{e'}\right)}{\max\left(Obs^{e'}\right) - \min\left(Obs^{e'}\right)}, \tag{9}$$

where $\max(\text{Man}^{e'})$ and $\min(\text{Man}^{e'})$ are the maximum and minimum Manhattan distances between all sending vehicles about event e', respectively, and $\max(Obs^{e'})$ and $\min(Obs^{e'})$ are the maximum and minimum number of building obstacles between all sending vehicles about event $e\prime$, respectively.

After the value is normalized, the score $S_i^{e'}$ for the message m_i about event $e\prime$ can be calculated using the following equation:

$$S_i^{e'} = \alpha \cdot e^{-\rho \text{Man}_i^{e'}} + \beta \cdot e^{-\sigma Obs_i^{e'}}, \tag{10}$$

where α, β, ρ, and σ are the four preset parameters. ρ and σ set the rate of exponential function change and control the influence of the Manhattan distance and the number of building obstacles on the message score. α and β control the influence ratio of the Manhattan distance and the number of building obstacles, where $\alpha + \beta = 1$. When $\text{Man}_i^{e'}$ and $Obs_i^{e'}$ are infinite, let $S_i^{e'} = 0$.

4.4. The Fusion of Message Scores. After obtaining the scores $S_0^{e'}, S_1^{e'}, \ldots, S_N^{e'}$ of all messages about event e', it is needed to fuse these scores together to finally determine the trust value of message. There are many methods of data fusion, including majority voting [25], weighted voting [26, 27], Bayesian inference [28], and Dempster-Shafer theory [29]. This paper mainly studies the influence of distance on the trust value of message and takes the score generated by distance as the weight of each message. Therefore, the weighted voting method is chosen for score fusion. The calculation method of the trust value of message m_0 about $e\prime$ is given in

$$\mathrm{Trust}\left(e\prime \longrightarrow m_0\right) = \mathrm{Fuse}\left(S_0^{e'}, S_1^{e'}, \ldots, S_N^{e'}\right) = \left\{ \sum_{i=1}^{N} d_i \cdot S_i^{e'}, \quad \text{if}\,(d_0 = 1), -\sum_{i=1}^{N} d_i \cdot S_i^{e'}, \quad \text{if}\,(d_0 = -1), \right. \tag{11}$$

where the value of d_i is +1 or −1. If the message m_i describes the occurrence of event e' as 1, then $d_i = 1$; otherwise $d_i = -1$. If $\mathrm{Trust}\,(e' m_0)$ is greater than 0, the message m_0 is trusted; otherwise the message m_0 is not trusted.

When event e' actually occurs, $\sum_{i=1}^{N} d_i \cdot R_i^{e'} > 0$. At this time, if the vehicle sending the message m_0 is a normal vehicle and sends a correct message, then $d_0 = 1$, and $\mathrm{Trust}\,(e' \longrightarrow m_0) = \sum_{i=1}^{N} d_i \cdot R_i^{e'} > 0$, so the conclusion is that the message m_0 is trusted; otherwise, if the vehicle sending the message m_0 is a malicious vehicle and sends a false message, then $d_0 = 1$, and $\mathrm{Trust}\,(ee' \quad m_0) = -\sum_{i=1}^{N} d_i \cdot R_i^{e'} < 0$, so the conclusion is that the message m_0 is not trusted. When event e' does not occur, $\sum_{i=1}^{N} d_i \cdot R_i^{e'} < 0$. At this time, the vehicle sending the message m_0 is a normal vehicle and sends a correct message, then $d_0 = -1$, and $\mathrm{Trust}\,(e' \longrightarrow m_0) = -\sum_{i=1}^{N} d_i \cdot R_i^{e'} > 0$; the conclusion is that the message m_0 is trusted; otherwise, if the vehicle sending message m_0 is a malicious vehicle and sends a false message, then $d_0 = 1$, and $\mathrm{Trust}\,(e' \longrightarrow m_0) = \sum_{i=1}^{N} d_i \cdot R_i^{e'} < 0$; the conclusion is that the message m_0 is not trusted. It can be seen that equation (11) can correctly determine whether message m_0 is trusted.

5. Simulation and Discussion

This section mainly performs experimental simulations to verify the effectiveness of the proposed MDBTM scheme. The tools used in the experimental simulations include the traffic flow simulation tool VanetMobiSim [30] (version 1.1) and the network simulation tool OPNET [31] (version 14.5).

5.1. The Experimental Setup

5.1.1. The Experimental Environment. The method proposed in this paper is based on the city road environment. First, it is necessary to use the VanetMobiSim tool to model city roads. This experiment uses the VanetMobiSim tool to generate a city road simulation area of 3200 m * 3200 m. There are 25 intersections, 40 road segments. Each road segment is 800 meters. The movement trajectories of the vehicles are generated by VanetMobiSim through the simulation area and then imported into the OPNET simulation environment for mobile nodes. The movement trajectories generated by VanetMobiSim cannot be used directly in OPNET and need to be converted to the format used by OPNET.

In the OPNET simulation environment, vehicles communicate with neighboring vehicles using a logarithmic normal connection model [32]. Through C−V2X technology [33], the communication range of the vehicle can reach 450 meters. Based on the 450-metre range of communications, it can be seen that at least 72 vehicles are required to communicate with each other via multihop. Therefore, the number of vehicles selected in this experiment is more than 72.

In the course of the experiment, the randomly selected road segment from the scene is chosen as the event location, and vehicles on the road periodically send messages about the event. Normal vehicles send the correct messages, while malicious ones send false messages.

5.1.2. The Experimental Parameters. The parameters used in the experiment are shown in Table 2.

5.1.3. The Performance Metric. For trust management, it is important to correctly judge the authenticity of a message. Therefore, in order to verify the performance of the method proposed in this paper, the correct decision probability of a message expressed by Psucc is used as the performance metric, and its definition is given in

$$P\mathrm{succ} = \frac{\mathrm{Num}_{\mathrm{succ}}}{\mathrm{Num}_{\mathrm{total}}} \times 100\%, \tag{12}$$

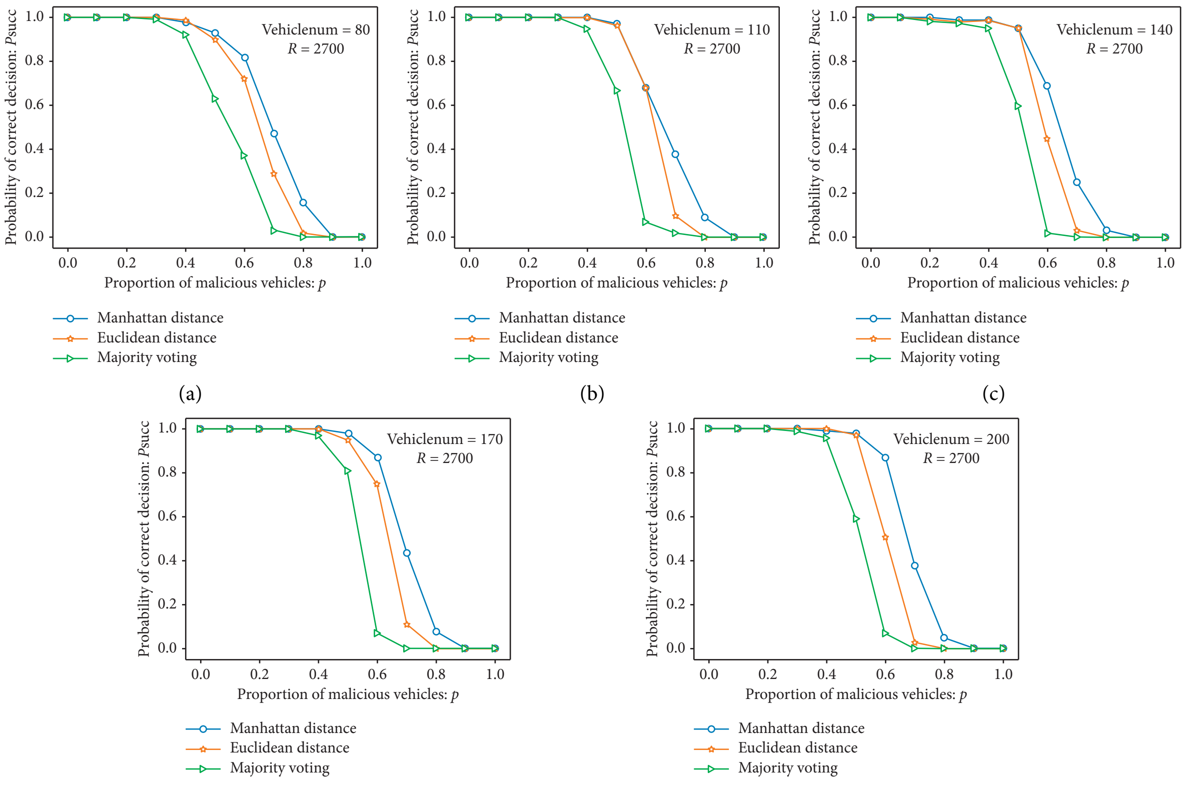
where $\mathrm{Num}_{\mathrm{succ}}$ represents the number of successful decisions, and $\mathrm{Num}_{\mathrm{total}}$ represents the total number of decisions.

5.2. The Experimental Analysis. When analyzing the influence of the proportion of malicious vehicles and the influence of the reference range R, this paper compares the proposed MDBTM method (labeled with Manhattan Distance) with the method based on Euclidean distance (labeled with Euclidean Distance) and the majority voting [25] method (labeled with Majority Voting). The method based on Euclidean distance uses the formula $R_i^{e'} = b + e^{-\gamma \cdot d}$ proposed by Yang et al. [14] to calculate the message scores and uses the method of (11) to fuse message scores. During the experiment, the value of b is 0, the value of γ is 1, and the d is the Manhattan distance between the sending vehicle and event location. Moreover, the data in this experiment are averaged after multiple experiments.

5.2.1. The Influence of the Proportion of Malicious Vehicles. As shown in Figure 5, the abscissa represents the proportion of malicious vehicles from 0.0 to 1.0, and the ordinate represents the correct decision probability. Figures 5(a)–5(e)

Table 2: Simulation parameters.

Parameters	Values
Traffic flow model	IDM_LC model
The number of vehicles	80, 110, 140, 170, 200
The speed	10–60 km/h
Simulation time	1800 s
Safety distance	100 m
The proportion of malicious vehicles	0.0–1.0
Communication range	450 m
α	0.5
β	0.5
ρ	2
σ	5



(a)

(b)

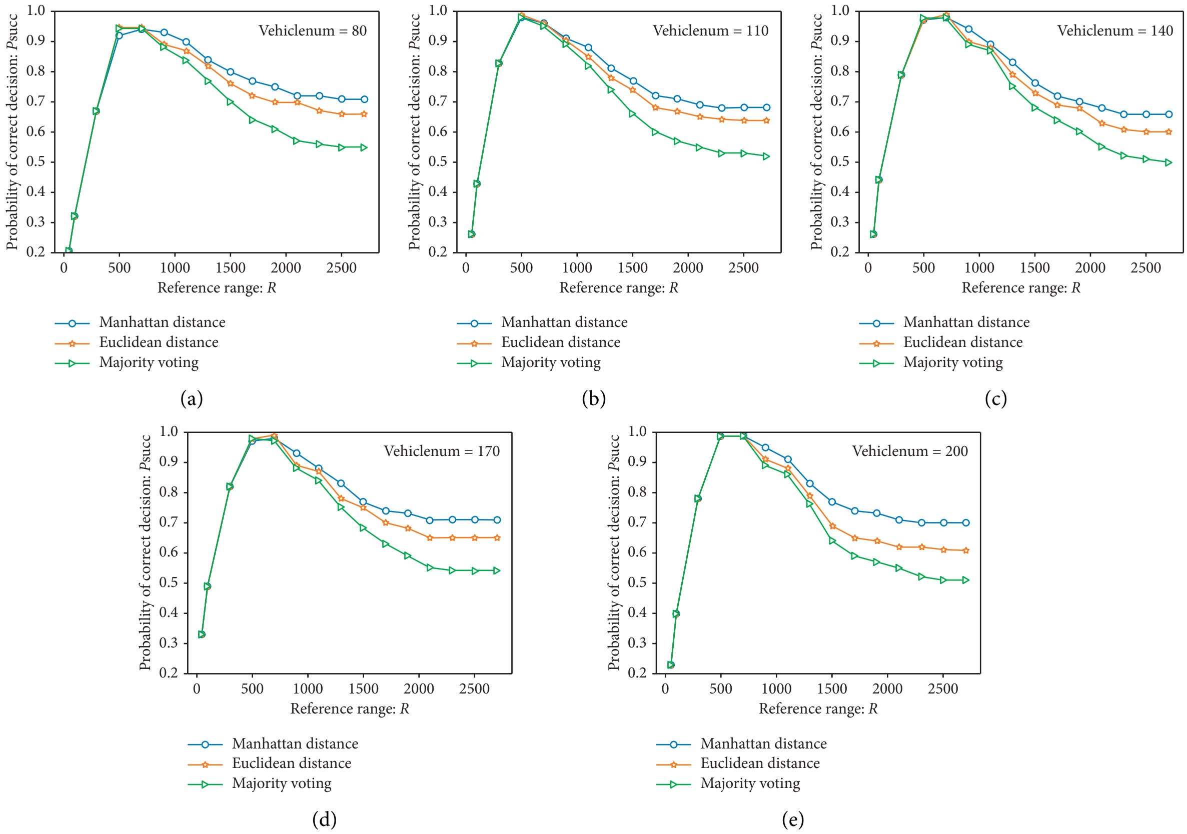
(c)

(d)

(e)

Figure 5: The comparison of correct decision probability under different proportions of malicious vehicles. (a) 80 vehicles. (b) 110 vehicles. (c) 140 vehicles. (d) 170 vehicles. (e) 200 vehicles.

represent the influence of the proportion of different malicious vehicles on the correct decision probability where the number of vehicles is, respectively, 80, 110, 140, 170, and 200, and the reference range R is 2700 meters. It can be seen from Figure 5 that when the proportion of malicious vehicles is less than 0.3, the correct decision probability for each method is close to 1 in the scene of different vehicle numbers. As the number of malicious vehicles increases, the correct decision probability for each method begins to decline when the proportion of malicious vehicles is greater than 0.4. However, the correct decision probability of the method proposed in this paper is higher than the other two methods. And for the other two methods, when the proportion of malicious vehicles is 0.8, the correct decision probability is close to 0. But the MDBTM method starts to approach 0 when the proportion of malicious vehicles is 0.9. It can be seen that the MDBTM method shows a better correct decision probability than the other two methods under different proportions of malicious vehicles and different numbers of vehicles. This shows that considering the Manhattan distance that the vehicle moves along the actual road and the obstruction of the line of sight by building



(a)

(b)

(c)

(d)

(e)

Figure 6: The comparison of correct decision probability under different reference ranges. (a) 80 vehicles. (b) 110 vehicles. (c) 140 vehicles. (d) 170 vehicles. (e) 200 vehicles.

obstacles can improve the robustness of the system against malicious vehicle attacks.

5.2.2. The Influence of the Reference Range R. Figure 6 shows the influence of different reference ranges on the correct decision probability. The abscissa represents the size of the reference range R (from 50 meters to 2500 meters), and the ordinate represents the correct decision probability (this probability is the average value under different proportions of malicious vehicles). Figures 6(a)–6(e), respectively, represent the influence of different reference ranges on the correct decision probability in the scenarios where the number of vehicles is 80, 110, 140, 170, and 200. It can be seen from Figure 6 that no matter the method proposed in this paper or the method based on Euclidean distance and majority voting, the correct decision probability is very low when the reference range R is too small in the scene of different vehicle numbers. This is because there are fewer messages for reference. As the reference range R increases, the number of reference messages increases, and the correct decision probability gradually rises. However, when the reference range R is too large, the number of malicious vehicles within the reference range R also increases which results in a decrease in the correct decision probability.

It can be seen from Figure 6 that there is a threshold. Whether it is greater than or less than the threshold, the correct decision probability is less than that of this threshold. When the number of vehicles is 80, 110, 140, 170, and 200, the threshold is 700 meters, which is close to the actual road length of 800 meters. This is because the number of building obstacles on the same road segment is 0, and vehicles are relatively close to the event location, thus leading to a higher correct decision probability. This is consistent with the theory of this paper. The design of this paper takes into account the Manhattan distance and the number of building obstacles at the intersection. On the same road segment, no building obstacles are blocking the line of sight, and the event location is relatively close to vehicles, so the correct decision probability is also high. As you can see, too large or too small reference range R will affect the correct decision probability. When the reference range R is close to the length of the actual road segment, the correct decision probability is higher.

It can also be seen from Figure 6 that with the same number of vehicles when the reference range R is less than the threshold 700 meters, the correct decision probability for each method is basically the same. This is because when the reference range R is small, the message available for reference is relatively small and the distance has little influence on the correct decision probability. When the reference range R is greater than the threshold 700 meters, because this paper considers the Manhattan distance and the number of building obstacles at the intersection, the method proposed in this paper has better performance than other methods in terms of the correct decision probability.

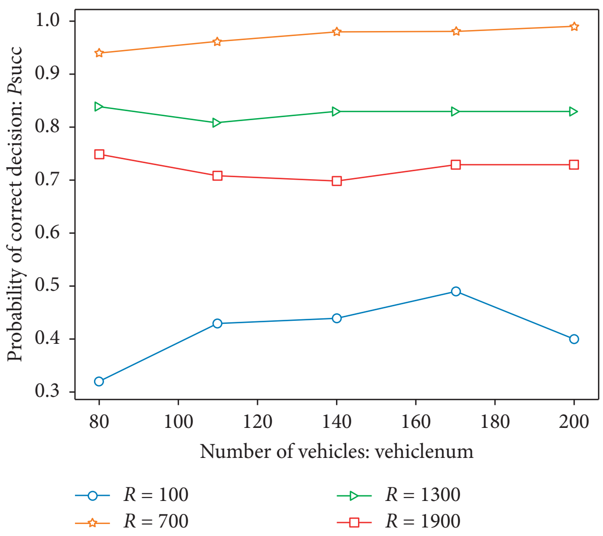


Figure 7: The comparison of correct decision probability under different number of vehicles.

5.2.3. The Influence of the Number of Vehicles in the Network. Figure 7 shows the influence of the number of vehicles (i.e., vehicle density) on the correct decision probability. The abscissa represents the different numbers of vehicles (80, 110, 140, 170, and 200), and the ordinate represents the correct decision probability (this probability is the average value under different proportions of malicious vehicles). As can be seen from Figure 7, when the reference range R is 100 meters, the correct decision probability varies greatly in the scene of different vehicle numbers because of too few messages available for reference. When the reference range R is 700 meters, 1300 meters, and 1900 meters, the correct decision probability varies very little. It can be seen that the number of vehicles in the network will not affect the correct decision probability of the proposed method when the reference range R is appropriate.

6. Conclusions

In this paper, a MDBTM model for calculating the distance in VANET is proposed, which solves the problem of no detailed discussion about the way of calculating the distance. In this model, the Manhattan distance and the number of building obstacles are calculated by considering the movement relationship between the sending vehicle and event location. The experimental results show that the method proposed in this paper shows better performance in terms of the correct decision probability than similar methods in the case of different proportions of malicious vehicles, different numbers of vehicles, and different reference ranges. It is also found that the correct decision probability is higher when the reference range R is set close to the length of the actual road segment, and the number of different vehicles in the network will not affect the correct decision probability.

In future work, we will consider the combination of this method and blockchain technology to store the score information in the blockchain, which can ensure the data's security (nontampering, traceability) and further improve the security of trust management in the VANET environment.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China under Grant 61862046; the Inner Mongolia Autonomous Region Science and Technology Achievements Transformation Project under Grant CGZH2018124; the Graduate Research Innovation Foundation Project of Inner Mongolia University under Grant 10000–15010109; and the Science and Technology Program of Inner Mongolia Autonomous Region under Grant 2019GG376.

References

[1] E. C. Eze, S. Zhang, and E. Liu, "Vehicular ad hoc networks (VANETs): current state, challenges, potentials and way forward," in *Proceedings of the 2014 20th Conference on Automation and Computing*, pp. 176–181, Cranfield, UK, September 2014.

[2] K. P. Laberteaux, J. J. Haas, and Y. C. Hu, "Security certicate revocation list distribution for VANET," in *Proceedings of the Fifth International Workshop on Vehicular Ad Hoc Networks*, pp. 88-89, San Francisco, CA, USA, September 2008.

[3] S. Dietzel, R. V. D. Heijden, H. Decke et al., "A flexible, subjective logic-based framework for misbehavior detection in V2V networks," in *Proceedings of the 15th International Symposium on A World of Wireless, Mobile and Multimedia Networks*, pp. 1–6, WoWMoM), Sydney, Australia, June 2014.

[4] J. Guo, J. P. Baugh, and S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," in *Proceedings of 2007 Mobile Networking for Vehicular Environments*, pp. 103–108, Anchorage, AK, USA, May 2007.

[5] A. Wasef, R. Lu, X. Lin, and X. Shen, "Complementing public key infrastructure to secure vehicular ad hoc networks," *IEEE Wireless Communications Security and Privacy in Emerging Wireless Networks*, vol. 17, no. 5, pp. 22–28, 2010.

[6] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[7] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen, "A multifaceted approach to modeling agent trust for effective communication in the application of mobile ad hoc vehicular networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 3, pp. 407–420, 2011.

[8] F. G. Mármol and G. M. Perez, "TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks,"

Journal of Network and Computer Applications, vol. 35, no. 3, pp. 934–941, 2012.

[9] N. Haddadou, A. Rachedi, and Y. Ghamri-Doudane, "A job market signaling scheme for incentive and trust management in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3657–3674, 2015.

[10] M. Raya, P. Papadimitratos, V. D. Gligor et al., "On data-centric trust establishment in ephemeral ad hoc networks," in *Proceedings of the IEEE INFOCOM 2008-the 27th Conference on Computer Communications*, pp. 1238–1246, Phoenix, AZ, USA, April 2008.

[11] A. Wu, J. Ma, and S. Zhang, "RATE: a RSU-aided scheme for data-centric trust establishment in VANETs," in *Proceedings of the 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–6, WiCom), Wuhan, China, September 2011.

[12] S. Gurung, D. Lin, A. Squicciarini, and E. Bertino, "Information-oriented trustworthiness evaluation in vehicular ad-hoc networks," *Network and System Security, Lecture Notes in Computer Science*, vol. 7873, pp. 94–108.

[13] R. A. Shaikh and A. S. Alzahrani, "Intrusion-aware trust model for vehicular ad hoc networks," *Security and Communication Networks*, vol. 7, no. 11, pp. 1652–1669, 2013.

[14] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2019.

[15] J. Chen, G. Mao, C. Li, and D. Zhang, "A topological approach to secure message dissemination in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 135–148, 2020.

[16] Y.-M. Chen and Y.-C. Wei, "A beacon-based trust management system for enhancing user centric location privacy in VANETs," *Journal of Communications and Networks*, vol. 15, no. 2, pp. 153–163, 2013.

[17] W. Li and H. Song, "ART: an attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 960–969, 2016.

[18] C. Lai, R. Lu, D. Zheng, and X. Shen, "Security and privacy challenges in 5G-enabled vehicular networks," *IEEE Network*, vol. 34, no. 2, pp. 37–45, 2020.

[19] J. Zhang, H. Zhong, J. Cui, M. Tian, Y. Xu, and L. Liu, "Edge computing-based privacy-preserving authentication framework and protocol for 5G-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7940–7954, 2020.

[20] J. Cui, L. Wei, H. Zhong, J. Zhang, Y. Xu, and L. Liu, "Edge computing in VANETs-an efficient and privacy-preserving cooperative downloading scheme," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1191–1204, 2020.

[21] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PA. CRT.: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 722–735, 2021.

[22] J. Chen and G. Mao, "On the security of warning message dissemination in vehicular Ad hoc networks," *Journal of Communications and Information Networks*, vol. 2, no. 2, pp. 46–58, 2017.

[23] C. A. Kerrache, C. T. Calafate, J.-C. Cano, N. Lagraa, and P. Manzoni, "Trust management for vehicular networks: an adversary-oriented overview," *IEEE Access*, vol. 4, pp. 9293–9307, 2016.

[24] F. Bai F, N. Sadagopan, and A. Helmy, "IMPORTANT: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *Proceedings of the IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 825–835, IEEE Cat. No.03CH37428), San Francisco, CA, USA, April 2003.

[25] B. Ostermaier, F. Dotzer, and M. Strassberger, "Enhancing the security of local danger warnings in VANETs-a simulative analysis of voting schemes," in *Proceedings of the Second International Conference. on Availability, Reliability and Security*, pp. 422–431, ARES'07), Vienna, Austria, April 2007.

[26] Z. Huang, S. Ruj, M. A. Cavenaghi, M. Stojmenovic, and A. Nayak, "A social network approach to trust management in VANETs," *Peer-to-Peer Networking and Applications*, vol. 7, no. 3, pp. 229–242, 2014.

[27] Y. Zhu, *Multisensor Decision and Estimation Fusion*, Kluwer Academic Publishers, Amsterdam, Netherlands, 2002.

[28] J. P. Huelsenbeck and F. Ronquist, "MRBAYES: Bayesian inference of phylogenetic trees," *Bioinformatics*, vol. 17, no. 8, pp. 754-755, 2001.

[29] J. Dezert, P. Wang, and A. Tchamova, "On the validity of dempster-shafer theory," in *Proceedings of the International Conference on Information Fusion*, pp. 655–660, Singapore, July 2012.

[30] H. Jérme, M. Fiore, F. Filali, and C. Bonnet, "Vehicular mobility simulation with VanetMobiSim," *Simulation*, vol. 87, no. 4, pp. 275–300, 2011.

[31] M. Chen, *OPNET Network Simulation*, Tsinghua University Press, Beijing, China, 2004.

[32] G. Mao, *Connectivity of Communication Networks*, Springer International Publishing AG, New York, NY, USA, 2017.

[33] Y. Li, *5G and Internet of Vehicles: Internet of Vehicles Technology and Intelligent Connected Vehicles Based on Mobile Communication*, Publishing House of Electronics Industry, Beijing, China, 2019.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9969605, 20 pages
https://doi.org/10.1155/2021/9969605

Research Article

Energy-Efficient Relay-Based Void Hole Prevention and Repair in Clustered Multi-AUV Underwater Wireless Sensor Network

Amir Chaaf,[1] Mohammed Saleh Ali Muthanna,[2] Ammar Muthanna,[3,4] Soha Alhelaly,[5] Ibrahim A. Elgendy,[6] Abdullah M. Iliyasu,[7,8,9] and Ahmed A. Abd El-Latif[10]

[1]*School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China*
[2]*School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China*
[3]*Department of Telecommunication Networks and Data Transmission, The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, 193232 Saint Petersburg, Russia*
[4]*Department of Applied Probability and Informatics, Peoples' Friendship University of Russia (RUDN University), 6 Miklukho-Maklaya St, Moscow 117198, Russia*
[5]*College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia*
[6]*School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China*
[7]*Electrical Engineering Department, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia*
[8]*School of Computing, Tokyo Institute of Technology, Yokohama 226-8502, Japan*
[9]*School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China*
[10]*Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin El-Koom, Egypt*

Correspondence should be addressed to Ahmed A. Abd El-Latif; a.rahiem@gmail.com

Received 16 March 2021; Accepted 17 June 2021; Published 1 July 2021

Academic Editor: Leo Y. Zhang

Underwater wireless sensor networks (UWSNs) enable various oceanic applications which require effective packet transmission. In this case, sparse node distribution, imbalance in terms of overall energy consumption between the different sensor nodes, dynamic network topology, and inappropriate selection of relay nodes cause void holes. Addressing this problem, we present a relay-based void hole prevention and repair (ReVOHPR) protocol by multiple autonomous underwater vehicles (AUVs) for UWSN. ReVOHPR is a global solution that implements different phases of operations that act mutually in order to efficiently reduce and identify void holes and trap relay nodes to avoid it. ReVOHPR adopts the following operations as ocean depth (levels)-based equal cluster formation, dynamic sleep scheduling, virtual graph-based routing, and relay-assisted void hole repair. For energy-efficient cluster forming, entropy-based eligibility ranking (E2R) is presented, which elects stable cluster heads (CHs). Then, dynamic sleep scheduling is implemented by the dynamic kernel Kalman filter (DK2F) algorithm in which sleep and active modes are based on the node's current status. Intercluster routing is performed by maximum matching nodes that are selected by dual criteria, and also the data are transmitted to AUV. Finally, void holes are detected and repaired by the bicriteria mayfly optimization (BiCMO) algorithm. The BiCMO focuses on reducing the number of holes and data packet loss and maximizes the quality of service (QoS) and energy efficiency of the network. This protocol is timely dealing with node failures in packet transmission via multihop routing. Simulation is implemented by the NS3 (AquaSim module) simulator that evaluates the performance in the network according to the following metrics: average energy consumption, delay, packet delivery rate, and throughput. The simulation results of the proposed REVOHPR protocol comparing to the previous protocols allowed to conclude that the REVOHPR has considerable advantages. Due to the development of a new protocol with a set of phases for data transmission, energy consumption minimization, and void hole avoidance and mitigation in UWSN, the number of active nodes rate increases with the improvement in overall QoS.

1. Introduction

Underwater wireless sensor network (UWSN) has many applications over the ocean environment. In UWSN, energy efficiency is the major constraint since the nodes are resource constraint [1–3]. This represents one of the main reasons that leads to the appearance of void holes, reducing the performance of the network. To achieve energy efficiency, various approaches were presented in UWSN. Here, the data transmission is carried over multiple hops between a number of sensor nodes through a selected route to reach the autonomous unmanned vehicles (AUVs), and then the final surface sink node and further collision-free medium access (MAC) protocols were presented. However, routing is also the best way to improve energy efficiency [4]. A cluster-based mobile data gathering is used to improve energy efficiency in the large-scale network [5]. The basic cluster concept is considered in this work to form initial clusters [6, 7]. This cluster formation is performed in nonoptima manner which is inefficient [8]. However, cluster head (CH) is performed in a random manner which makes this work ineffectual [9]. In addition, processing the distributed clustering algorithm needs a large amount of control packet exchange which consumes lots of energy. Autonomous unmanned vehicles (AUVs) are specially designed for data gathering in the underwater environment [10–12]. An AUV-assisted energy-efficient clustering UWSN mechanism faces many serious issues as follows [13, 14]:

(i) Energy consumption in existing research works is high, which leads to a large number of holes in the network.

(ii) Network clusters with unequal size introduce energy imbalance in certain regions, leads to a large number of holes.

(iii) Optimal sleep scheduling is necessary in order to reduce the energy consumption of the nodes and avoid holes.

(iv) Route selection considers only limited metrics, which leads to large packet loss and energy consumption which induces trap nodes.

In AUV-assisted UWSN, the predefined path determination is the critical issue which increases the distance to the nodes, the energy consumption, and delay in data transmission [15, 16]. On the other hand, the unnecessary sensing of the sensor nodes increases energy consumption. These are only limited factors since the forwarder selection mechanism must consider more criteria. Furthermore, route selection based on single metric is ineffective in underwater scenarios [17, 18]. Traditional routing algorithms follow ocean depth-based routing. This leads to high packet loss due to the void hole issue. Void hole avoidance and recovery is an emerging part of UWSN. Furthermore, it occurs frequently in the sparse node distribution with a limited amount of energy. In addition, various important issues remain untouched in UWSN for reducing energy consumption and avoiding energy hole creation [19, 20]. Table 1 describes the abbreviations that we have used throughout the paper:

Table 1: List of abbreviations.

Abbreviation	Expansion
UWSN	Underwater wireless sensor network
AUVs	Autonomous unmanned vehicles
ReVOHPR	Relay-based void hole prevention and repair protocol
DK2F	Dynamic kernel Kalman filter
BiCMO	Bicriteria mayfly optimization
EEDG	Energy-efficient data gathering
E2R	Eligibility ranking
CH	Cluster head
LECA	Level-based equal clustering algorithm
AEC	Energy-efficient clustering
MFO	Moth flame optimization
CMDG	Cluster-based mobile data gathering

(i) There is no unified protocol for reliable and energy-efficient data transmission for a specific type of UWSN.

(ii) Existing protocols focus on one aspect for energy consumption, i.e., clustering, routing, or void hole repair. Hence, energy consumption may occur by other aspects of the issue.

(iii) Current protocols used a single AUV for data collection, which increases the end-to-end delay of each sensor, and thus, energy consumption rate is increased [21, 22].

1.1. Motivation. Figure 1 illustrates the void hole problem in UWSN. In UWSN, the presence of routing void holes leads to higher packet loss which makes the data unreliable. The main cause for routing voids is the higher energy consumption of the sensor nodes in the network, i.e., nodes which lose energy makes the hole. In this context, there are two major research problems arise [23–25]:

(i) Most of the works have concentrated on energy-efficient route selection without deploying AUV in the network. In this case, the energy consumption and delay for data transmission is high. Although these works select optimal route, it fails to transmit the data in a timely manner since the hole mitigation process generally transmits the data in longest path or backward path.

(ii) In some works, AUVs are deployed to mitigate the problem of void holes. However, there is an issue in predicting the trajectory of the AUVs since the travel length is high. As the trajectory detection methods use the energy level alone for optimal positioning to collect the sensor's data.

(iii) The network is managed with unequal clusters which imbalances the load among clusters. Thus, some of the cluster heads suffer form higher energy consumption while some cluster heads suffer from lower energy consumption. In general, all underwater sensor nodes are continuously sensing the

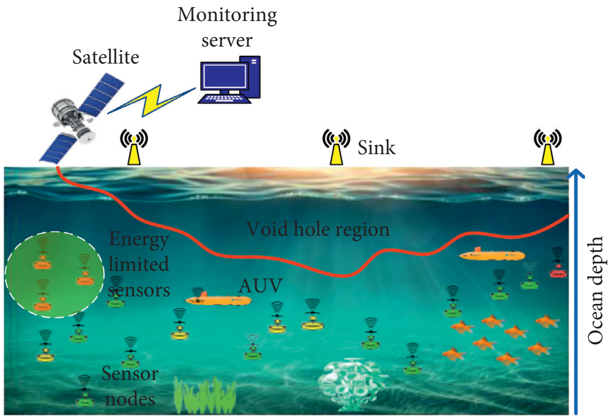


Figure 1: Void hole problem in UWSN.

environment, which consumes a lot of energy and introduces void holes. In the absence of an optimal sleep scheduling mechanism, the energy consumption will be high and the holes are unavoidable.

This paper mainly focuses on void hole prevention and mitigation in underwater wireless sensor networks (UWSNs). For that, we intend to combine intercluster routing performance and a relay-assisted void mitigation mechanism initiated by an AUV according to an optimal trajectory and a positioning of a suitable relay node for data transmission continuity. We also prevent the holes by minimizing overall energy consumption in the network by introducing multiple mechanism clustering, sleep scheduling. This research work is motivated from the problems presented in the existing research works.

1.2. Contributions. In this paper, we deeply tackle all the issues of reducing the energy consumption in UWSN. In particular, we presented the following contributions to address the energy consumption and QoS issues:

(i) The level-based equal clustering algorithm (LECA) is presented that utilizes distance and load criteria. In each level, equal clusters are formed to avoid energy consumption. Entropy-based eligibility ranking (E2R) protocol is presented for CH selection. Optimal CH is selected in two levels based on energy, centrality, and success rate which increase the lifetime of CH.

(ii) To reduce energy consumption, dynamic sleep scheduling is presented by the dynamic kernel-based Kalman filter (DK2F) which is proposed. This considers residual energy, buffer value, and coverage rate to make the decision on node status (sleep/active).

(iii) Optimal route is selected in multiple hops by considering multiple factors. A virtual graph-based routing is presented, which uses maximum matching theory for optimum selection of next hops. Optimal positioning of relay nodes by the AUV repairs the voids in the network. Proposed maximum matching approach chooses optimal criteria according to the position of AUV.

(iv) The optimal repair position initiated by the AUV by repositioning relay nodes is computed by the mayfly optimization algorithm which is proposed and works upon multiple criteria to find the optimal trajectory of AUV and the repositioning of relay nodes. The mayfly algorithm works well in terms of convergence rate and convergence time.

(v) Overall network follows level-based clustering and optimal routing, which minimizes the energy consumption that prevents the holes. Sleep scheduling prevents the nodes from dead, which further prevents the hole.

1.3. Paper Layout. The remaining part of the paper is organized as follows: Section 2 presents the literature review in the area of clustering, routing, sleep scheduling, and void hole detection and repair in UWSN. Section 3 focuses on the problems that existed previously in void hole avoidance. Section 4 describes the research methodology, pseudocode, and algorithms in details. Section 5 illustrates the performance of network simulation for the proposed and previous protocols. Section 6 concludes the paper and presented the future works.

2. Lietarture Review

An energy-efficient data gathering (EEDG) scheme was proposed in [26] for the underwater wireless sensor network. The data transmission is performed in a multihop manner. At first, the energy consumption is balanced by grouping the nodes into smaller groups. Furthermore, the forwarder nodes are selected to gather the data from the subset nodes. Here, the communication is carried out in a one-hop manner. Furthermore, a medium access control (MAC) protocol is utilized to improve collision rate and packet loss.

A fault resilient routing for the underwater wireless sensor network was presented in [27] for underwater data transmission. The fault-tolerant routing follows the moth flame optimization (MFO) algorithm. The data transmission is carried through AUVs to base stations. Here, the AUVs act as cluster heads that are responsible to collect data from the sensor nodes. The use of AUVs avoids reclustering and overloading problems. To overcome the path disjoint issues, additional mobile nodes are deployed in the network. In this work, multiple AUVs are deployed to support data forwarding in the underwater network.

A cluster-based mobile data gathering (CMDG) scheme was studied in [28] for the large-scale underwater sensor network. At first, the cluster formation and CH selection problem is formulated as an optimization problem. In this work, the AUV tour planning scheme is presented to handle the sensor mobility. In order to achieve an energy-latency tradeoff, the travel length is shortened for AUV movement. A centralized clustering algorithm is proposed to form initial clusters. Then, the distributed clustering algorithm is proposed to maintain the formed clusters. This work has two drawbacks as follows: (1) CH selection is inefficient since it considers only a minimum number of parameters and (2) the distributed clustering algorithm exchanges a large number of control packets, which is ineffective, and the consumption of higher energy and delay on sensor nodes due to data transmission.

AUV-assisted energy-efficient clustering (AEC) mechanism was presented in [29]. The proposed AEC mechanism introduces wake-up sleep cycle for the underwater sensor network. The overall mechanism includes cluster formation, cluster head nomination, and sleep wake-up scheduling. To form clusters, virtual sectoring approach is presented. In each virtual sector, the cluster is formed and a CH is selected. The CH is selected based on the distance with the cluster centroid point. Then, the path of AUV is a predefined path. The predefined path of AUV is inefficient since it consumes large amount of energy and increases delay. The CH selection was poor due to the estimation of only energy.

Author proposes a cluster-based sleep scheduling mechanism in UWSN [30]. The overall network is considered as the 3D underwater sensor network. A 3D partition unit is considered with a basic cluster structure. All sensor nodes are in the temporary control of clusters. In each cluster, sleep-awake scheduling is enabled based on the remaining energy level. The major goal of this work is to achieve minimum energy consumption and guarantee maximum sensing coverage in the network. Cluster formation in performed in a nonoptimal manner which is inefficient.

In [31], the authors propose a two-stage routing protocol. The main purpose of this protocol is to enable communication between not only connected nodes but also for nonconnected or partially connected nodes so that the packet delivery rate will be improved. To delay the death of nodes, an energy threshold method and rerouting scheme is proposed. Involvement of energy threshold and rerouting processes improve the connectivity of the network, which prevents the holes, and preserve resource constraints, forwarding loops. However, unnecessary sensing of the sensor nodes increases the energy consumption.

An energy-balanced efficient and reliable routing (EBER2) protocol for UWSNs was presented in [32]. Energy balancing among neighbors and reliability are achieved in EBERR protocol. The EBERR protocol considers residual energy and potential forwarding nodes (PFNs) of the forwarder node. The transmission range is divided into power levels, and the forwarders can adjust the transmission range adaptively. In order to suppress the duplicate packets, depth of the nodes is compared. Forwarder selection is inefficient since it considers only limited metrics, and also the sensor nodes nearer to the sink will drain with larger energy due to the transmission of sensed data from the nodes in the network.

A distance vector-based opportunistic routing (DVOR) protocol was proposed to address the problem of void regions in the underwater sensor network [33]. The main idea behind this protocol is to use the depth information for route selection. The DVOR uses a query-based mechanism to enable the distance vectors for underwater sensor networks. From the distance vectors, each node records the smallest hop count information towards sink node. Based on this hop count information, routing is performed. The void hole is avoided by selecting a route with small distance. When the network is sparse, then the hop count information will have a large distance which increases the energy consumption. Optimal route selection based on single metric is insufficient to cope with UWSN.

In [34], sink mobility, i.e., AUV, and courier nodes are deployed in network for data collection, aggregation, and transmission. The entire network is divided into four sectors. Both courier nodes and AUV are movable with random trajectory. In this protocol, routing is fixed and sink mobility is dynamic. Comparison is made between several existing protocols and the proposed protocol. However, the mobility of mobile sink and courier nodes increases energy consumption and decrease the network lifetime. This is a linear type of network, it does not suit for complex ocean depth scenarios, and also realistic applications are not adopted with this protocol. In [35], the authors proposed a new data collection protocol for QoS provisioning. For that, the bioinspired routing algorithm is proposed which facilitates the natural features of the genetic algorithm. Clusters are formed which provides a highly stable and different size of clusters for traffic load balancing. The proposed routing algorithm predicts high stable links as a forwarding node. This work eliminates the data transmission by the upward and downward transmission. The main drawback of this protocol is that it is not aware of node mobility and packet delivery. In [36], fuzzy clustering is presented, which designs the fitness function for selecting the CH according to the distance between the nodes. For cluster formation, the fuzzy algorithm is used, whereas CHs are selected by the PSO algorithm. The overall network topology is arranged in a hierarchical structure, and the comparative analysis is made between the proposed hybrid (fuzzy and PSO) algorithm with LEACH and traditional PSO algorithms. This hybrid protocol has several drawbacks:

(i) Hybrid fuzzy and PSO algorithms consume more energy by underwater sensors since the computation of both algorithms is very high. Due to limited battery issue of underwater sensors, this protocol is not suited.

(ii) The overall work partially reduces the energy level, which does not suit risky oceanic applications, and also, it does not increase the lifespan of the UWSN.

In [37], the void hole alleviation issue is addressed using enhanced geographic and opportunistic routing protocol in the harsh underwater WSN. There are three problems, such as void hole occurrence, higher energy consumption, and low packet delivery rate that are addressed in this paper. Furthermore, the network scalability issue is addressed in this paper. The performance of the proposed protocol is compared with the geographic and opportunistic routing with topology control protocol based on depth adjustment as well as transmission-adjusted neighbor node approaching distinct algorithms with energy-efficient mate.

Table 2 shows a comparison summary of various existing works [28–33] that have studied the main elements of energy consumption minimization and void hole avoidance and mitigation in UWSN that address the critical issues related to clustering, node sleep scheduling, routing, and hole mitigation.

3. Problem Statement

The purpose of this paper is to investigate the problem of void hole and repair in multi-AUV-enabled UWSN. This section summarizes the important issues in current works.

Authors in [38] focus on void hole detection and mitigation in an underwater sensor network. For that, two routing schemes are proposed. The first routing scheme, called energy-aware scalable reliable and void hole mitigation routing (ESRVR), intends to avoid holes during route selection. In that, the two-hop neighbor information is collected before initiation of route selection. As this scheme considers two hops, the void hole is avoided in the route selection itself. The second scheme, namely, cooperative ESRVR (Co-ESRVR), focuses on mitigating void holes through backward transmission. The major drawbacks of this paper are follows:

(i) Two-hop neighbor information alone is insufficient to avoid holes since the data from the nodes that are deployed in deep level need to transmit through multiple hops. Thus, there is a need for gathering multihop information which is not efficient.

(ii) In the sparse network environment, it is hard to gather two-hop neighbor information, since it is not sure that always two-hop nodes will be presented.

(iii) In backward routing energy consumption and delay is high. That is Co-ESRVR scheme also introduces multiple holes that need to be mitigated.

(iv) When the network is sparse, or there is a limited possibility for backward transmission, then the ESRVR and Co-ESRVR are not feasible.

In [39], the authors propose a game-theoretic approach for energy-efficient routing in the 3D underwater WSN. However, game theory approaches have several drawbacks, which are as follows:

(i) The game played by the nodes is noncooperative. Thus, the strategy played by the players is unknown to other nodes. It leads to the same route is selected by multiple source nodes, which introduces a large number of collisions. Due to collisions, packet retransmission count is high. This leads to large energy consumption.

(ii) The game theory approach has high complexity. In addition, all players in the forwarding region are considered (other than neighbors) which makes the algorithm more complex.

Hence, high complexity in game theory approach requires more energy by sensors and also takes higher processing time. In [40], the authors propose an AUV-assisted data gathering approach to minimize energy consumption in Smart Ocean. For that, an AUV-assisted data gathering scheme based on the clustering and matrix completion (ACMC) method for UWSN is proposed. The drawbacks in this work are the follows:

(i) The K-means algorithm forms clusters based on distance value. The formed unequal clusters will have imbalanced load among clusters. Presence of an imbalanced load leads to energy consumption in some regions.

(ii) CH and secondary CH are selected in each cluster. Here, the cluster center criteria are considered for CH selection while secondary CH is selected in a random manner. Thus, CH selection is inefficient as the selected CH may lose the energy, which becomes void holes.

(iii) The greedy algorithm-based AUV trajectory only considers the trajectory length. This means that nearby position is selected as AUV moving position. It does not make decisions based on the energy level of the nodes. If the nodes with low energy are located far away from the current AUV position, then there will be higher energy consumption. The greedy algorithm has higher time consumption and complexity.

In [41], the authors proposed an AUV-assisted void prediction and repair mechanism in the underwater sensor network. The repair position is calculated by the particle swarm optimization (PSO) algorithm. In this paper, PSO-based void prediction causes more issues that are listed as follows:

(i) Repair position for AUV is computed by PSO, which traps the solutions into local optima. It leads to the nonoptimal positioning of AUV.

(ii) Here, the single AUV collects the repair requests from multiple sensor nodes. And the rules are predefined. When the number of holes in the network is large, then the AUV suffers in decision-making. The AUV could not handle a large amount of requests from the sensor nodes. Thus, this work is unable to mitigate the void holes effectively.

(iii) This work only mitigates the void but unable to prevent the voids due to a lack of optimal clustering and route selection procedures. The main reason for holes is high energy consumption. In this work, energy consumption is high in data collection, data transmission, and sensing.

Based on the shortcomings and issues cited, we aimed to design a global solution by introducing an efficient clustering method on the sensor nodes, an optimal cluster head selection, and also an efficient sleep scheduling method to avoid the continuous sensing of the sensors to reduce the overall energy consumption which also decreases the creation of void holes accordingly, a routing method is also used to gather and transmit the sensors data between the clusters

Table 2: Summarize the contribution of existing works and comparison between them.

Existing work	Clustering	Node sleep scheduling	Routing	Hole mitigation	Key contributions and limitations
[28]	✓	✓	×	×	(i) Propose a clustering scheme and a CH selection to gather the sensor data from each cluster and deploy an AUV with a tour planning scheme to collect data from CHs. (ii) Finding a tradeoff between consumption of energy and data gathering delay. Limitations: CH selection is inefficient since it considers only a minimum number of parameters, and distributed algorithm exchanges large number of control packets which is ineffective.
[29]	✓	✓	×	×	(i) Introduces a wake-sleep cycle for the sensors to reduce their energy consumption. (ii) A virtual sectoring approach is presented, cluster is formed, and CH is selected to reduce energy consumption; the CH data will be gathered by a predefined path of the AUV. Limitations: the predefined path of AUV is inefficient since it consumes large amount of energy by the CHs to transmit their data and increases the delay.
[30]	✓	✓	×	×	(i) The major goal of this work is to achieve minimum consumption of energy and guarantee maximum sensing coverage in the network. Limitations: cluster formation is performed in a nonoptimal manner which is inefficient.
[31]	×	×	✓	✓	(i) The main goal of this protocol is to enable communications between not only connected nodes but also for nonconnected or partially connected nodes so that the packet delivery rate will be improved. (ii) To delay the death of nodes, an energy threshold method and a rerouting scheme are proposed. Limitations: unnecessary sensing of the sensor nodes increases the energy consumption.
[32]	×	×	✓	×	(i) The main goal is to achieve the energy balancing among neighbor nodes and reliability. Limitations: during the data transmission process, forwarder selection is inefficient since it considers only limited metrics.
[33]	×	×	✓	✓	(i) The main ideas behind this protocol are to use the depth information for route selection. (ii) The void hole is avoided by selecting a route with small distance. Limitations: when the route is sparse, then the hop count information will have a large distance, which increases the energy consumption. Optimal route selection based on a single metric is insufficient to cope with underwater sensor network.

efficiently, and introducing the use of multi-AUV aims to detect and repair the creation of the hole according to the current trajectory position of the AUV to mobile relays that are used as a replacing part of the failed node and an intermediate receiver of the data from the cluster heads to the AUV, which will be selected and repositioned by the AUV. By combining all the presented solutions, the decrease of the overall energy consumption, reducing, detecting, and repairing of the void holes will be ensured.

4. System Model

4.1. System Overview. In this work, we present an energy-efficient relay-assisted 3D-UWSN model that absorbs the surrounding by collecting data and transmitting the information in which void hole prevention and mitigation procedure is focused. The overall 3D-UWSN is constructed as x_i, y_i, z_i coordinates, and this 3D network is divided into multiple levels as L_1, L_2, . . ., L_{N} based on the depth of the ocean covering shallow water and deepwater areas. Each level LN is composed of n number of sensors $L_N = (n_1, n_2, \ldots, n_N)$.The network model comprises underwater sensor nodes, sink node, mobile relays, and multiple AUVs. We construct the network based on multiple levels. In each level, an AUV is deployed to gather data from the underwater sensor nodes, and AUVN collects data from L_N level. All the nodes have the same initial energy E_{Ini} and the sensing range R_{S}. Sensors can be transmitted into two types as topology information and event. The size of the event and topology information packages is M. The network connectivity rate is defined as the ratio of S_C which is computed by $N(S)_C$ (number of sensors that communicates with the sink node) via single hop, which means that the sensors can achieve their sensed data directly to the surface sink that is present in their coverage range and a multihop communication which means that the sensors transmit their collected data through other sensor nodes by constituting a route to achieve the data to the final sink and it is computed as follows:

$$S_C = \frac{N(S)_C}{n}. \tag{1}$$

When the network connectivity is 1, the network can obtain the full network connectivity, and all sensors can communicate with the surface sink with either one-hop or multihop communication.

The overall process comprises four major phases that are explained in the following sections. Figure 2 describes the overall network model.

4.2. Level-Based Equal Cluster Formation. The overall network is segregated into multiple equal clusters based on the depth of the underwater environment. We propose a new level-based equal clustering algorithm (LECA). Generally, the nodes presented in underwater-based sensor networks are considered resource-constrained (i.e., battery-powered nodes). The node cannot participate in the network if the level of energy for that node is drained. Therefore, the selected node should have sufficient energy for transmitting the data. Thus, E2R considers the node's residual energy as another metric. Thereby, the sensor with the more residual energy has the large possibility to be CH.

4.2.1. Energy Consumption Model. Significant efforts have been made to address the UWSN's energy consumption, in which all nodes in the network are energy constraint (i.e., the energy sources of the nodes drop by usage). In addition, the large consumption of energy leads to early dead which decreases the lifetime of the network. The consumption of energy for the node n_i can be calculated as

$$E(n_i) = E_{\text{idle}}(n_i) + \sum_{v \in V} \sum_{p \in P(v)} w(p) \times A(v) \times E(n_i, p). \tag{2}$$

Assuming the path $p \in P(v)$, then the path weight $w(p)$ is expressed as

$$\sum_{p \in P(v)} w(p) = 1, \tag{3}$$

where $A(v)$ denotes the average amount of consumed energy through the node n_i for a time unit regarding the transmission of data, $E(n_i, p)$ denotes the amount of consumed energy only either in reception pr transmission, and E_{idle} denotes the average amount of consumed energy at the idle state by the node n_i per unit time. Regarding the estimated consumption energy for the node, the node's lifetime is expressed as

$$\text{LT}(n_i) = \frac{E_{\text{ini}}}{E(n_i)}, \tag{4}$$

where E_{ini} denotes the sensor node's initial energy, which is initially fed into the node for its network's participation. The proposed LECA divides the network environment into multiple levels as $L_1, L_2, \ldots, L_n$. In each level, clusters are formed based on the load level. Then, each cluster performs entropy-based eligibility ranking (E2R)-based CH selection protocol to select optimal CH. Here, Tsallis entropy is utilized to formulate an optimal energy threshold. Upon threshold value, candidate nodes are filtered by E2R protocol. The proposed E2R protocol uses residual energy level $\mathfrak{R}_\xi$, centrality factor $\mathbf{C_f}$, and success rate criteria $\mathbf{SR_c}$.

Algorithm 1 explains the procedure of E2R algorithm-based cluster formation and CH selection. Involvement of the E2R algorithm improves the data aggregation process as well as network QoS performance. Furthermore, involvement of E2R-based data aggregation also helps to minimize the risk of instability in the network. Generalized Tsallis entropy is computed by

$$T_Q(P_1 \ldots P_W) = \frac{1}{1-q}\left(\sum_{i=1}^{w} p_i^q - 1\right), \tag{5}$$

where q is the logarithmic function. For all sensors in the network, $T_Q(P_1 \ldots P_W)$ is computed for the number of given input parameters. Using T_Q, the weight value for CH selection is implemented as

$$T_Q = (S_1.C_1)w_1 \times (S_2.C_2)w_2 \times (S_3.C_3)w_3, \tag{6}$$

where w_1, w_2, and w_3 are the weight values for C_1, C_2, and C_3, respectively.

The CH is selected based on

$$\text{CH}_{\text{prob}} = \max\left(C_{f\,\text{prob}} \mathfrak{R}_{\xi\,\text{max}}, \text{SR}_{c\,\text{max}}\right). \tag{7}$$

After CH selection, clusters are formed based on sensor node's cost function as

$$\text{CM}(i, j) = \frac{\text{SR}_{c(ij)}}{\text{SR}_{c(\text{max})}} * \frac{E_{\text{cur}}(i)}{E_{\text{cur}}(j)} * \frac{C(ij)}{C_{\text{ave}}}, \tag{8}$$

where E_{cur} (i) represents the current energy of node i and E_{cur} (j) represent the current energy of CH j. Each CH in the cluster aggregates the sensor data and forwards them to the AUV. The proposed E2R protocol has $O(n)$ complexity which is due to the message transmission overhead where n is the number of sensors. CH announces, join request, and join response messages are exchanged within the cluster, which introduces the complexity in cluster.

4.3. Dynamic Sleep Scheduling. In each cluster, a dynamic sleep scheduling procedure is established in order to reduce the energy consumption of the sensor nodes by avoiding their continuous sensing of the environment which causes an unnecessary energy consumption that leads to void holes. By taking into consideration different factors on the nodes, the decision on the nodes status (sleep, awake, and idle) can be taken effectively. For that, we propose the dynamic kernel Kalman filter (DK2F) algorithm. In particular, Cauchy kernel function is used for nonlinear cases. The DK2F algorithm takes residual energy level $\mathfrak{R}_\xi$, buffer factor ς_f, and coverage rate χ_r to make decision on the node status. The considered statuses are sleep, active, and transmit. The DK2F is executed by each node, and the report is sent to the CH. Then, the CH activates the mode for each node in the cluster. The procedure for DK2F is described as follows.

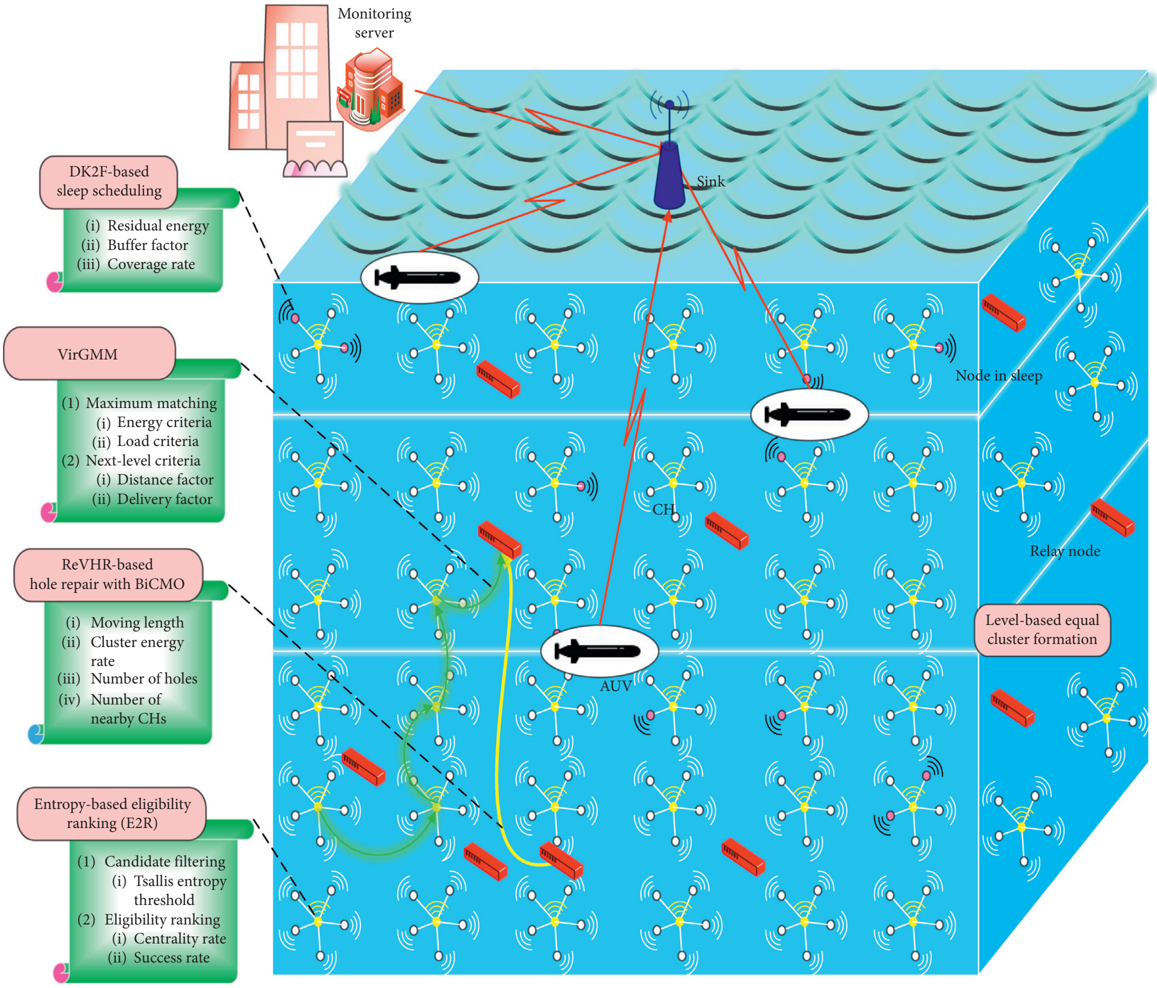


Figure 2: Network model.

```
  Input: $\mathbf{S_i} = \{\mathbf{S_1}, \mathbf{S_2}, \mathbf{S_3}, \ldots, \mathbf{S_N}\}$,
  Output: Number of Clusters (Cluster Formation and CH Selection)
  Start
Step 1: $\mathbf{S_i} \longrightarrow$ REQ
 Step 2: if $\mathbf{S_i} \longrightarrow \mathfrak{R}_\xi$, $\mathbf{C_f}$, $\mathbf{SR_c}$//higher energy, central value and success rate)
        Then
        $\mathbf{S_i} \longrightarrow \mathbf{CH_i}$
        If ($\mathbf{CH_i}$ found)
            $\mathbf{CM_i} \longrightarrow \mathbf{CH_i}$ (based on REP)
        End if
    End if
  Step 3: if $\mathbf{CH_i} \longrightarrow$ low $\mathfrak{R}_\xi$, $\mathbf{C_f}$, $\mathbf{SR_c}$
        Then elect another CH
         $\mathbf{CH_i} \longrightarrow$ nearer $\mathbf{CH_i}$
    End if
  Step 4: $\mathbf{CH_i} \longrightarrow$ SN
  End
```

Algorithm 1: E2R protocol.

Firstly, kernel function is initiated which is defined by

$$\delta_{\aleph}(e) = \frac{1}{1 + e^2/\aleph}, \tag{9}$$

where e is the exponential term between two different variables, $\aleph$ is the dynamic kernel bandwidth, and δ is the range between 0 and ∞. DK2F gives the optimum solution for both linear and nonlinear cases of the model in dynamic nature. The mathematical formulation of DK2F is described as follows:

$$\begin{aligned} X_T &= f_T X_T + A_T, \\ Y_T &= h_T X_T + B_T, \end{aligned} \tag{10}$$

where X_T is the state vector and Y_T is the observation measurements at time T. f_T and h_T represent the state transition matrix and the observation matrix, respectively. A_T and B_T denote the noise values in observation and Gaussian noise, respectively. In the probabilistic model, it is represented as

$$\begin{aligned} \rho(Y_T|X_T) &= N(Y_T|hX_T, a), \\ \rho(X_T|X_{T-1}) &= N(Y_T|fX_T, b). \end{aligned} \tag{11}$$

Figure 3 illustrates the dynamic sleep scheduling model. The assumption and prediction of the underwater sensor node is illustrated as follows (Algorithm 2):

(i) Every node in underwater environment follows only three kinds of states as active, sleep, or transmit. In active state, sensors are working and listening to the surrounding events and process the computations. Besides, it can also possible to switching to the idle state.

(ii) All sensors can be possible to act as a relaying state for packet transmission to the near AUV. For that, each node maintains next hop nodes in the neighbors list.

(iii) The duration of the active status is exponentially distributed with mean $1/S$. In the active state, the sensors will sense packets, relay packets, and process packets. When all nodes are under sleep state, the CH cannot aggregate or transmit any packets. In this case, entire cluster putted in OFF state. When at least one node in an active state, then CH turns into ON state again. In this case, CH can transmit or receive packets from cluster members. The actual and predicted result for the DK2F is indicated in Figure 4.

```
Input: Total Cluster Members $i$
Output: Scheduled Mode
(1) Begin
(2) Initialize $i = \{s_1, s_2, s_3, \ldots\}$ /*sensors in a cluster*/
(3) For each $i$
(4) Find $\mathfrak{R}_{\xi}$, $\varsigma_f$, and $\chi_r$
(5) List RE $\varsigma_f$, and $\chi_r$ for each $k$
(6) Find Dynamic Kernel Values
(7) if $(\mathfrak{R}_{\xi} < \mathfrak{R}_{\xi Th} \&\& \varsigma_f < \varsigma_{f(Th)} \&\& \chi_r < \varsigma_{f(Th)})$
    /*comparison with threshold*/
      {
        assign sleep mode
    else
        assign active mode
      }
    end if
(8) End for
(9) End
```

4.4. Virtual Graph-Based Routing. Intercluster routing is performed to improve energy efficiency and delivery rate. We present a novel Virtual Graph-enabled Maximum Matching (VirGMM) algorithm. In first step, the virtual graph is constructed for the CHs. Then, maximum matching nodes are selected based on dual-criteria as energy criteria {residual energy level and expected energy consumption} and load criteria {current load level and expected load level}. Among maximum matching nodes, optimal forwarder is selected upon optimum criteria. The criteria are selected upon the following rules:

> {If AUV is presented in Same Level, then Choose [Delivery Rate Criteria];
>
> Else, Choose [Distance Criteria]}

In this way, data are transmitted to AUV through optimal forwarders. Figure 5 describes the virtual graph-based routing.

To minimize the sensor energy usage in data transmission, virtual graph with maximum matching theory is applied in which nearest next hop is selected for fast data transmission without any packet loss. Sensors in active state sense the event about the environment and then send the sensed report to the sink node through AUV and next hops.

In this algorithm, bicriteria is used such as $\mathfrak{R}_{\xi}$ and $E\mathfrak{R}_{\xi}$ for first criteria and C_L and EC_L are considered as the second criteria to find the perfect match. For each node, the connectivity to become maximum match (MM) is derived from the Bayesian theory as follows:

$$C\left(S_i\left(\mathfrak{R}_{\xi}\right)|\mathrm{MM}\right) = \frac{C\left(\mathrm{MM}|S_i\left(\mathfrak{R}_{\xi}\right)\right)C\left(S_i\left(\mathfrak{R}_{\xi}\right)\right)}{C(\mathrm{MM})}, \tag{12}$$

$$C\left(S_i\left(E\mathfrak{R}_{\xi}\right)|\mathrm{MM}\right) = \frac{C\left(\mathrm{MM}|S_i\left(E\mathfrak{R}_{\xi}\right)\right)C\left(S_i\left(E\mathfrak{R}_{\xi}\right)\right)}{C(\mathrm{MM})}, \tag{13}$$

$$C(S_i(C_L)|\mathrm{MM}) = C(\mathrm{MM}|S_iC_L)C\left(\frac{S_i(C_L)}{C(\mathrm{MM}),}\right. \tag{14}$$

$$C(S_i(\mathrm{EC}_L)|\mathrm{MM}) = C(\mathrm{MM}|S_i\mathrm{EC}_L)C\left(\frac{S_i(\mathrm{EC}_L)}{C(\mathrm{MM}),}\right. \tag{15}$$

where equation (12) computes the probability of a node S_i to become a next hop based on its $\mathfrak{R}_{\xi}$. Similarly, equations

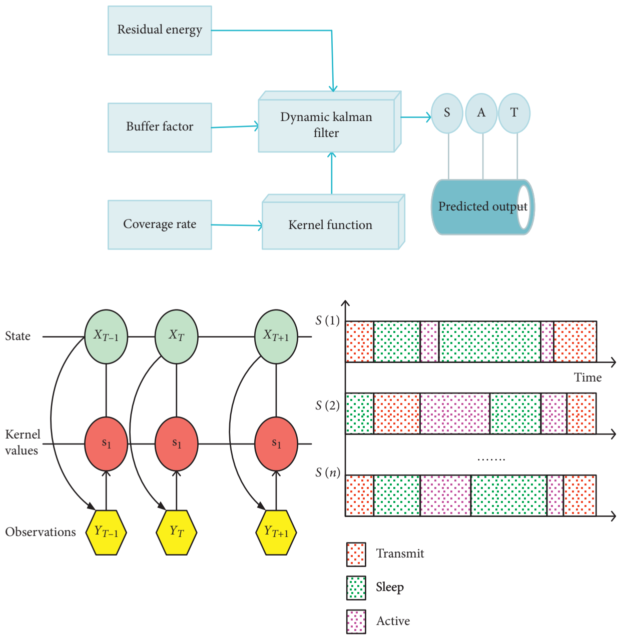


Figure 3: Dynamic sleep scheduling.

```
Input: Total Cluster Members $i$
Output: Scheduled Mode
Begin
Initialize $i = \{s_1, s_2, s_3, \ldots.\}$ /*sensors in a cluster*/
For each $i$
Find $\Re_\xi$, $\varsigma_f$, and $\chi_r$
List RE $\varsigma_f$, and $\chi_r$ for each $k$
Find Dynamic Kernel Values
if $(\Re_\xi < \Re_\xi Th \&\& \varsigma_f < \varsigma_f(Th) \&\& \chi_r < \varsigma_f(Th))$
/*comparison with threshold*/
  {
    assign sleep mode
  else
    assign active mode
    }
  end if
End for
End
```

Algorithm 2: Member-balanced scheduling.

(13)–(15) compute the probability based on current load and expected load, respectively. Here, the maximum matched values are mapped between 0 and 1 range. The source CH found the MM for all available next hops and sorted the best set of matches. The set of possible matches by matching theory is illustrated in Table 3.

4.5. Relay-Assisted Void Hole Repair. During the route selection process when transmitting the data through different sensor nodes, a VOID hole may appear. Once the void hole is detected, then the report is generated and sent to the AUV. Mainly each CH in the cluster aggregates the sensor data and forwards them to the AUV through an optimal selected route, and in case a hole is detected during the routing process, the AUV then takes the optimal decision on the *void hole repairing by selecting an optimal repair position by the relay nodes.* In this, we introduced the novel relay-assisted void hole repair mechanism (ReVHR). On receiving void requests, the AUV selects an optimal relay node to repair the hole by replacing the failed node in the route. The relay node selection is carried based on AUV trajectory distance factor from the detected hole. Then, optimal position of the relay is determined by the bicriteria mayfly optimization (BiCMO) algorithm. The BiCMO considers the following objective functions:

$$\mathrm{OF} = \left\{\mathrm{Min}\left[m_l\right] \&\& \, \mathrm{Min}\left[n_{(h)}\right]\right\}, \tag{16}$$

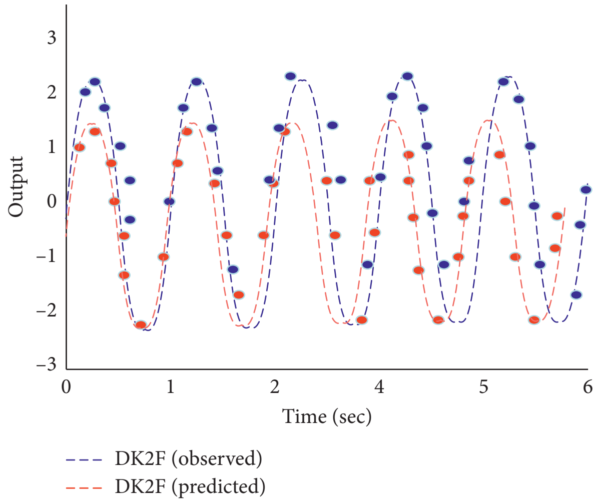


Figure 4: Performance plot for DK2F.

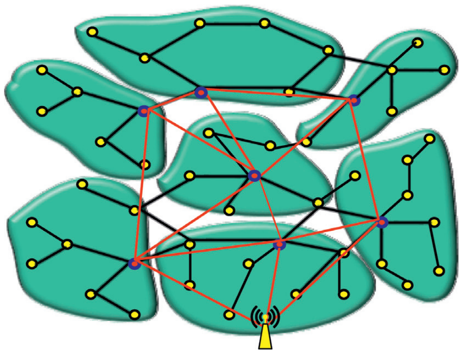


Figure 5: Virtual graph-based routing.

Table 3: Set of rules for forwarder selection.

$\mathfrak{R}_\xi$	$E\mathfrak{R}_\xi$	C_L	EC_L	Status of node
<0.5	<0.5	<0.5	<0.5	Partially match
<0.5	<0.5	>0.5	>0.5	No match
<0.5	>0.5	<0.5	<0.5	No match
<0.5	>0.5	>0.5	>0.5	No match
>0.5	**<0.5**	**<0.5**	**<0.5**	**Highly match**
>0.5	<0.5	>0.5	>0.5	Partially match
>0.5	>0.5	<0.5	<0.5	Partially match
>0.5	>0.5	>0.5	>0.5	No match

where m_l is the moving length and $n_{(h)}$ is the number of holes. The moving distance, number of holes, number of CHs, and energy level of the region are considered for repositioning the void holes. Once the relay node is repositioned, the source CH transmits the data to AUV through the relay node. In this way, the void hole is repaired and the data are transmitted to AUV without loss.

Algorithm 3 describes the pseudocode for mayfly-based replay selection. The fitness value is computed for each node by biobjectives. The computational complexity for this algorithm is $O(N)$, where N denotes the number of underwater sensors.

5. Simulation Results

In this section, the simulation results are presented for the proposed REVOHPR protocol is evaluated in terms of energy consumption, packet delivery ratio, and throughput for effective data transmission and hole detection and repair mechanisms. In addition, the proposed ReVHR protocol is compared to similar UWSN protocols such as ESRVR [38],

```
Input: n nodes' number
Output: Void Repair
Procedure:
Initialize population
For all sensors
Estimate the fitness
Find the total fitness ( ) = (1), (2), . . .
Compute the node probability
Identification of Relay Node
    If (Probability = = High)
        Assign Node → VHR//VHR–Void Hole Repair
        Alert broad-casted to next relay nodes
    Else
        Move to next node
    End if
End for
End
```

Algorithm 3: Mayfly-based relay node selection.

ACMC [40], and PSO [41]. The detailed description of the simulation environment and comparative study is specified as follows.

5.1. Simulation Setup. The simulation of the proposed vs. existing protocols for data transmission in UWSN is implemented using NS3.27. In NS3, AquaSim as shown in Figure 6 is one of the significant modules for underwater sensor environment simulation, and besides other modules, it supports to create the network model. The simulation parameters used in the proposed REVOHPR model are illustrated in Table 4. The simulation parameters are not constrained by any limit. The simulation is performed using the Ubuntu 14.04 LTS operating system with a 32 bit processor. Compared to the other simulators, NS3 is a more flexible tool to simulate clustering, sleep scheduling, intercluster routing, and void hole detection and repair mechanism. The procedure for simulation is depicted in Figure 7.

As discussed above, the simulation result is indicated in Figure 8. The sensor data are collected by the AUV and transmitted to the onshore sink for further processing.

Figure 9 shows the flowchart of the proposed ReVOHPR protocol. This step evaluates the performance of the protocol. At the end of the simulation, graphical plots are drawn by the simulation result. Our protocol is dynamic and supported for diverse nature (applications and dynamic range of simulation).

5.2. Application Scenario: Sea Life Monitoring. The proposed protocol is tested for sea life monitoring. In this case, the sensors used are suitable for underwater animal monitoring that is deployed in the ocean and record real-time events from the environment. Two types of underwater sensors are used for sea life monitoring, i.e., physical sensors and chemical sensors. Pressure, oxygen, and temperature are the physical sensors, whereas salinity, turbidity, pH, nitrate, chlorophyll, and dissolved oxygen are the chemical sensors.

The representation of the sea life monitoring is depicted in Figure 10. The type of sensors and their purpose is described in Table 5. The specification of each sensor is illustrated in Table 6. Ocean climate is changed over a long period which is hazardous to marine life. For example, abnormal sea temperatures affect the life of sea animals.

5.3. Comparative Study. In this section, we illustrate the performance analysis of the proposed and previous protocols in terms of various QoS and energy consumption metrics as energy consumption, delay, throughput, and packet delivery ratio (PDR). The previous protocols for data transmission and void hole repair can be follows: ESRVR [38], ACMC [40], and PSO [41].

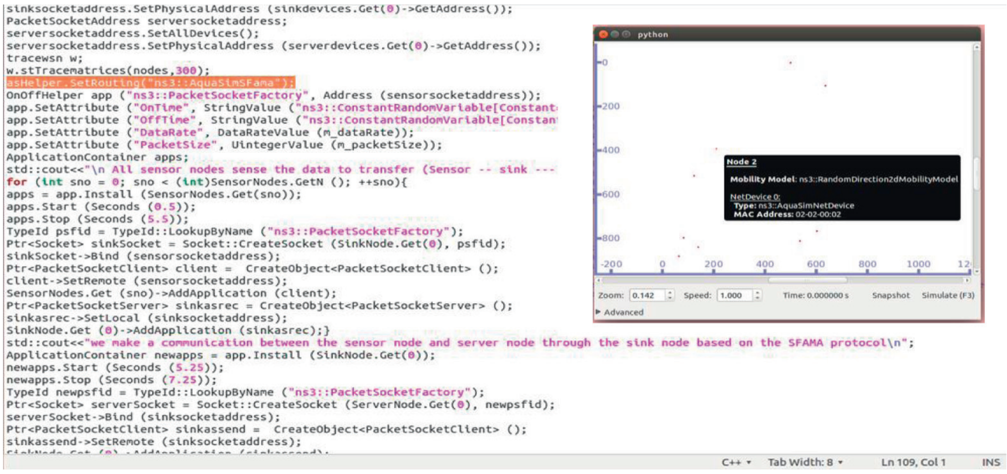
5.3.1. Energy Consumption. Energy is a significant metric in underwater sensor communications. In the underwater sensor network, acoustic signals are transmitted in a cylindrical way. Higher energy consumption must be avoided since it represents the worst performance of the network. Packet transmission loss between two nodes is the major reason for higher energy consumption. This transmission loss $t(l)$ is computed by follows:

$$t(l) = 10 \log \frac{R_1}{R_2}, \tag{17}$$

where R_1 and R_2 are the source and destination nodes of the transmission. Average energy consumption is the sum of energy consumed by all nodes in the network during idle, packet transmission, reception, and sensing:

$$E_{(i)} = \sum_{i=1}^{N} E(\mathrm{Tx}_i) + \sum_{i=1}^{N} E(\mathrm{Rx}_i) + \sum_{i=1}^{N} E(\mathrm{Se}_i) + \sum_{i=1}^{N} E(I_i), \tag{18}$$

where N represents the total number of nodes, Tx_i is the sum of energy for transmission, Rx_i is the sum of energy for



```
sinksocketaddress.SetPhysicalAddress (sinkdevices.Get(0)->GetAddress());
PacketSocketAddress serversocketaddress;
serversocketaddress.SetAllDevices();
serversocketaddress.SetPhysicalAddress (serverdevices.Get(0)->GetAddress());
tracewsn w;
w.stTracematrices(nodes,300);
asHelper.SetRouting("ns3::AquaSimSFama");
OnOffHelper app ("ns3::PacketSocketFactory", Address (sensorsocketaddress));
app.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant
app.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constan
app.SetAttribute ("DataRate", DataRateValue (m_dataRate));
app.SetAttribute ("PacketSize", UintegerValue (m_packetSize));
ApplicationContainer apps;
std::cout<<"\n All sensor nodes sense the data to transfer (Sensor -- sink ---
for (int sno = 0; sno < (int)SensorNodes.GetN (); ++sno){
apps = app.Install (SensorNodes.Get(sno));
apps.Start (Seconds (0.5));
apps.Stop (Seconds (5.5));
TypeId psfid = TypeId::LookupByName ("ns3::PacketSocketFactory");
Ptr<Socket> sinkSocket = Socket::CreateSocket (SinkNode.Get(0), psfid);
sinkSocket->Bind (sensorsocketaddress);
Ptr<PacketSocketClient> client =  CreateObject<PacketSocketClient> ();
client->SetRemote (sensorsocketaddress);
SensorNodes.Get (sno)->AddApplication (client);
Ptr<PacketSocketServer> sinkasrec = CreateObject<PacketSocketServer> ();
sinkasrec->SetLocal (sinksocketaddress);
SinkNode.Get (0)->AddApplication (sinkasrec);}
std::cout<<"we make a communication between the sensor node and server node through the sink node based on the SFAMA protocol\n";
ApplicationContainer newapps = app.Install (SinkNode.Get(0));
newapps.Start (Seconds (5.25));
newapps.Stop (Seconds (7.25));
TypeId newpsfid = TypeId::LookupByName ("ns3::PacketSocketFactory");
Ptr<Socket> serverSocket = Socket::CreateSocket (ServerNode.Get(0), newpsfid);
serverSocket->Bind (sinksocketaddress);
Ptr<PacketSocketClient> sinkassend =  CreateObject<PacketSocketClient> ();
sinkassend->SetRemote (sinksocketaddress);
SinkNode.Get (0)->AddApplication (sinkassend);
```

Node 2
Mobility Model: ns3::RandomDirection2dMobilityModel
NetDevice 0:
Type: ns3::AquaSimNetDevice
MAC Address: 02-02-00:02

Zoom: 0.142 Speed: 1.000 Time: 0.000000 s

Figure 6: AquaSim module in NS3 for acoustic communication.

Table 4: NS3 simulation parameters.

Parameter		Specification
Network parameters	Simulation area	1000 × 1000 × 1000
	# of underwater sensors	100
	# of relay nodes	10
	# of AUVs	3
	# of sink	1
	Number of clusters	5–7
	Simulation time	300 seconds
	Modules used	AquaSim, antenna, config store, CSMA, LTE, AODV, mesh, mobility, DSR, flow monitor, and internet
Underwater sensor parameters	Packet size	512 kB
	# of packets	20–200
	Packet time interval	100 milli seconds
	Data rate	10–20 Mbps
	Initial energy per sensor	70 J
	Transmission range	300 m

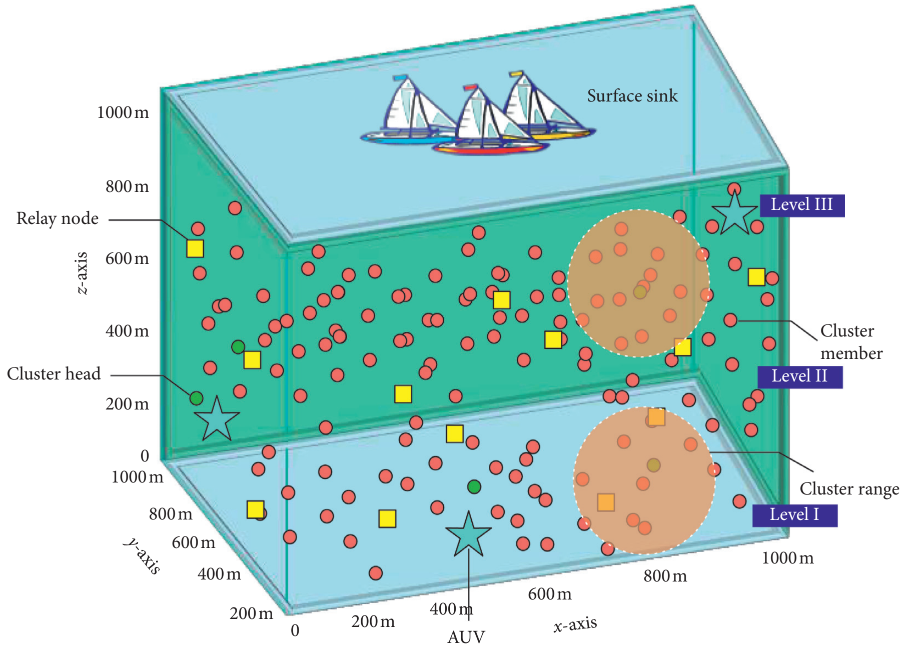


Figure 7: Simulation setup for REVOHPR protocol.

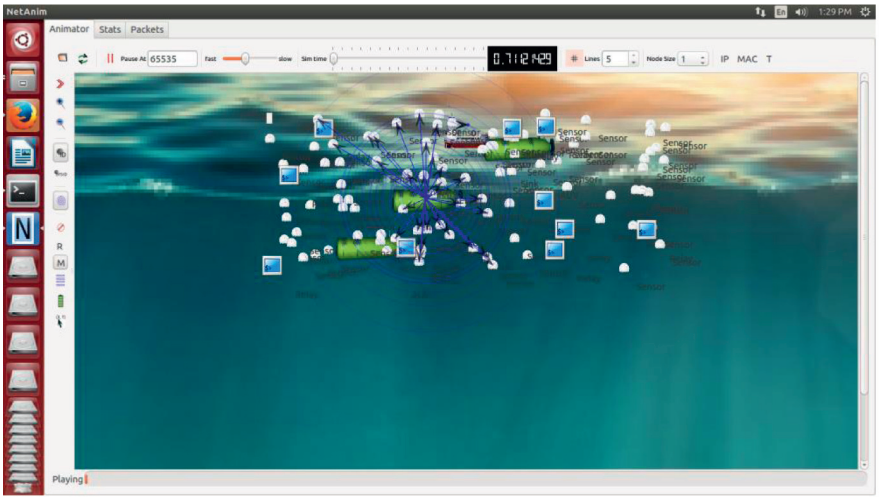


Figure 8: Simulation running in NS3.

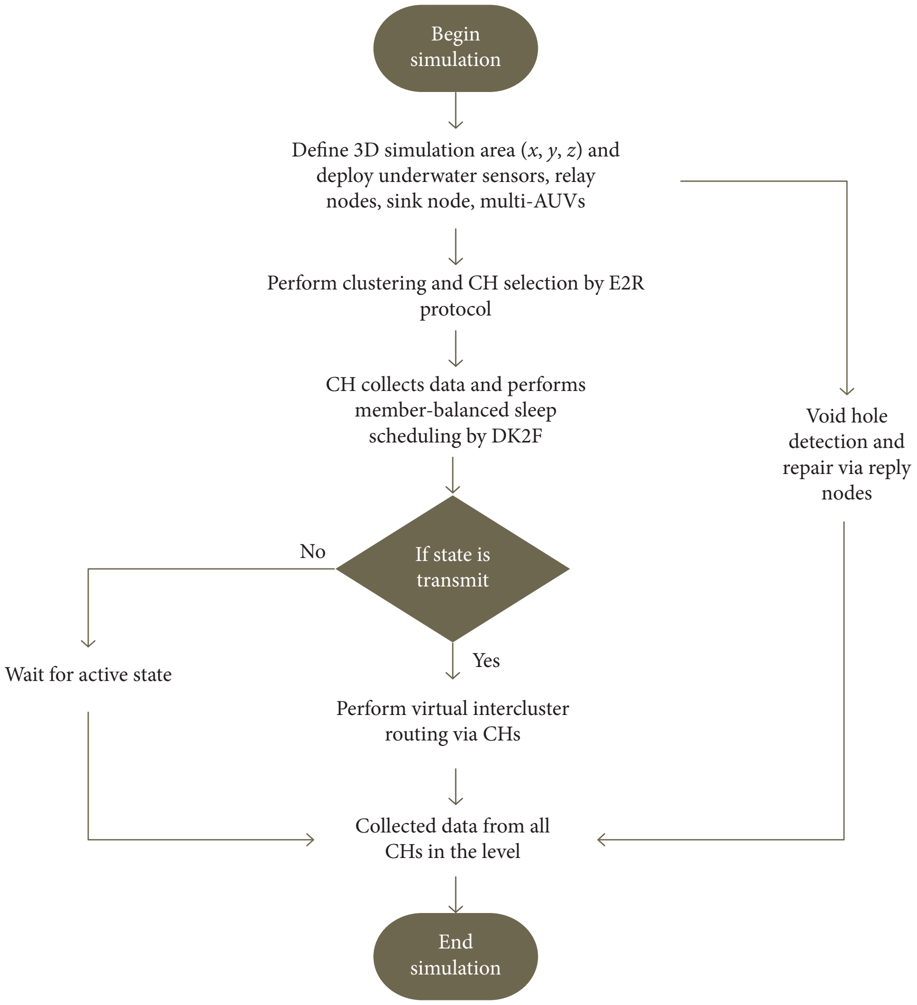

Begin simulation
Define 3D simulation area (x, y, z) and deploy underwater sensors, relay nodes, sink node, multi-AUVs
Perform clustering and CH selection by E2R protocol
CH collects data and performs member-balanced sleep scheduling by DK2F
If state is transmit
No
Wait for active state
Yes
Perform virtual intercluster routing via CHs
Void hole detection and repair via reply nodes
Collected data from all CHs in the level
End simulation

Figure 9: Flowchart for REVOHPR protocol.

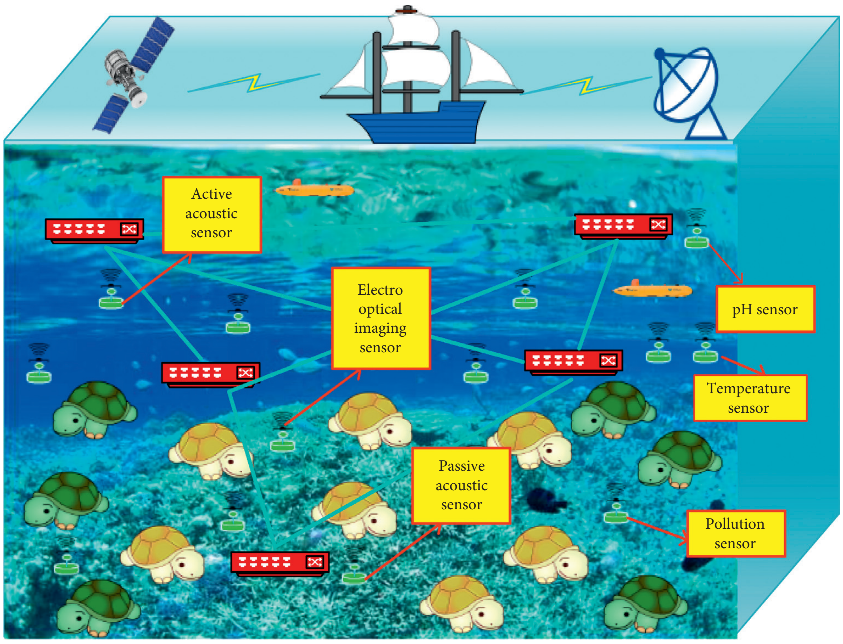


Figure 10: Sea life monitoring in UWSN.

Table 5: Sea life sensors.

Animal type	Animal status	Suitable sea sensors
Dolphins Whales	Below sea surface	Active acoustic monitoring sensors
Seals Turtles	Frequently near or at sea surface	Electro optical imaging sensors
Fish Porpoises	Frequent distinct vocalization	Passive acoustic monitoring sensors

Table 6: Underwater sea life monitoring sensors.

Sensors	Observed pattern	Range	Accuracy	Power supply	Unit
SBE16plus V2	Temperature	−5–+30°C	±0.0055°C	9–30 V	°C
GT301	Pressure	0–60	<±0.5% of FRO	24 V	Bar
SBE16plus V2	Conductivity	0–9	±0.0005	9–30 V	S/m
OBS–3+	Turbidity	Mud: 5000–10,000 mg/L	0.5 NTU	15 V	NTU
PS 2102	pH	0–14 pH	± 0.1	N/A	pH
YSI 6025	Chlorophyll	0–400 μg/L	0.1 μg/L	6 V	μg/L
ISUS V3	Nitrate	0.007–28 mg/L	± 0.028 mg/L	6–18 V	mg/L
SBE 63	Dissolved oxygen	120%	0.1	6–24 V	mg/L

reception, Se_i is the sum of energy consumed in sensing, and I_i is the node at idle state.

Figure 11 represents the sum of energy consumption rate of ReVOHPR against the previous protocols as ACMC, ESRVR, and PSO. Mathematical computations and number of iterations for cluster formation, CH selection, and routing consider more energy consumption. These processes were used to reduce the residual energy. The effective selection of CH and routing by the virtual graph algorithm reduce overhead in packet transmission. Furthermore, sleep scheduling idea is used which saves energy and improves the network lifetime. The graphical plots show that the amount of energy consumption increases with respect to the number of nodes. However, network density is the primary factor that affects the network performance in terms of energy consumption and QoS. Figure 11 shows the performance of average energy consumption with respect to the number of packets per second. Data transmission is performed in a multihop fashion. When the sensors in the network communicate to the surface sink directly by single-hop fashion,

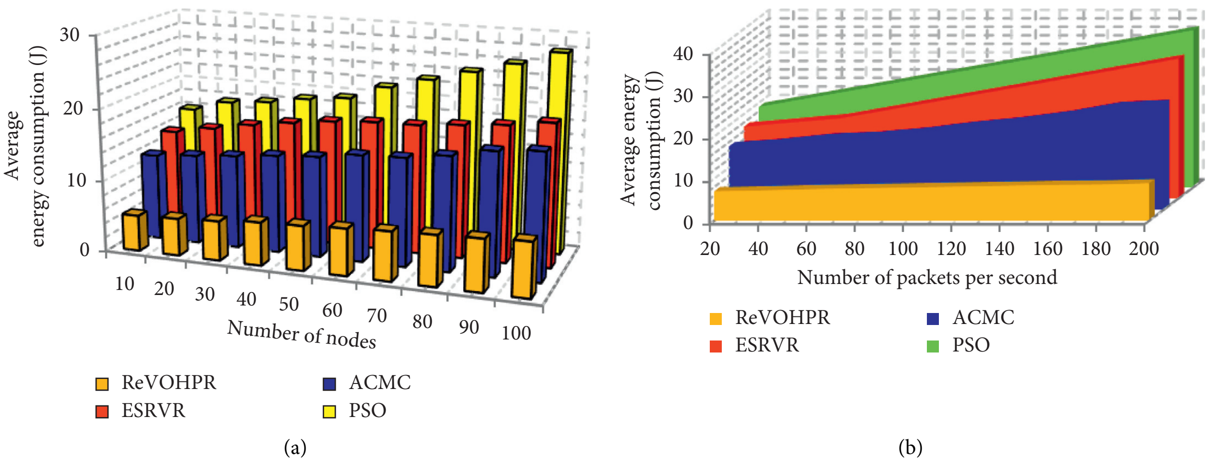


Figure 11: Impact of energy consumption.

then the energy consumption rate is higher. In addition, single-hop and multihop network performance is not similar in energy consumption rate. Distance between the source to the sink is higher or lesser in single hop whereas multihop routing often has a smaller number of relay nodes. In PSO and ESRVR, packets are transmitted in single hop to the surface sink. Compared to ACMC, ESRVR, and PSO, the proposed ReVOHPR consumes minimum energy, i.e., 37%, 42%, and 67%, respectively.

5.3.2. Delay. It is the amount of time required for data transmission from source (underwater sensor) $P_{(S)}$ to the destination node (surface) $P_{(D)}$. In other words, delay is computed from the generation to the destination reception. The successful packets are counted up in delay computation. This is computed as follows:

$$D = P_{(S)} - P_{(D)}. \tag{19}$$

Figure 12 shows the performance of delay with respect to the number of nodes. The proposed ReVOHPR forwards packets by CH, and aggregated packets forward to next hop using the multihop routing algorithm. Ocean depth is the major element to consider in both clustering and routing. When the depth of ocean is higher, then the packet collection time is higher. The delay is higher in PSO when it processes with packets without processing the void hole detection and prevention. The void management in the proposed ReVOHPR protocol improves the network data transmission and improves the node's presence. Thus, the delay in packet transmission is eliminated. The PSO-based algorithm is not sufficient for data transmission. Since, it has a low convergence rate in the iterations.

The total delay required for data transmission from source to the destination with respect to the number of packets is very higher for previous protocols. It is illustrated in Figure 12. It is computed by several factors as propagation delay, transmission delay, number of hop counts, and distance between two nodes. ReVOHPR utilizes the relay-based void hole prevention and repair. This efficiently helps to identify the presence of void and handles it precisely. The increased frequency of void occurrence increases delay when the number of nodes increases for the lengthy route. The computational time required in high traffic congestion is exponential, and it does not suit for event-based data transmission.

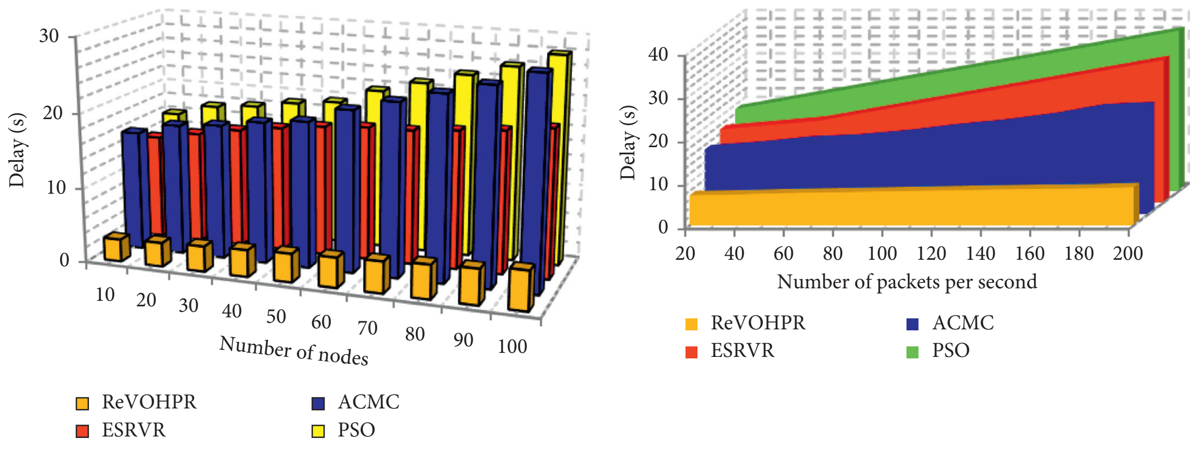
5.3.3. Throughput. It is the positive metric that defines the amount of packet transmitted in a time. When measuring the maximum throughput in packet transmission, then the communication link or network access is reliable. It is computed as follows:

$$\text{Throughput}\left(\frac{\text{bits}}{\text{sec}}\right) = \frac{\text{Sum }(N(\text{Sp}) * \text{APS})}{T(t)}, \tag{20}$$

where $N(\text{Sp})$ is the number of successful packets, APS is the average packet size, and $T(t)$ is the total time spend for packet transmission. The simulation results in Figure 13 show that the performance of the proposed ReVOHPR is higher than the previous protocols. It is analyzed by both number of nodes and the number of packets per second.

Due to the less traffic congestion and immediate route identification from CH to the AUV, throughput for transmitted packets is high. Unfortunately, existing works have obtained high communication overhead and high traffic congestion. When network size expands, then the performance of throughput is low. Hence, ACMC is not able to handle the high volume of traffic. ESRVR is suited only when the network has limited number of nodes and processing with limited number of communications. In addition, it uses two-hop information for routing packets. This is insufficient in achieving higher throughput. A large number of void holes decrease network performance. In this case, frequent void hole mitigation is important that degrades network throughput.

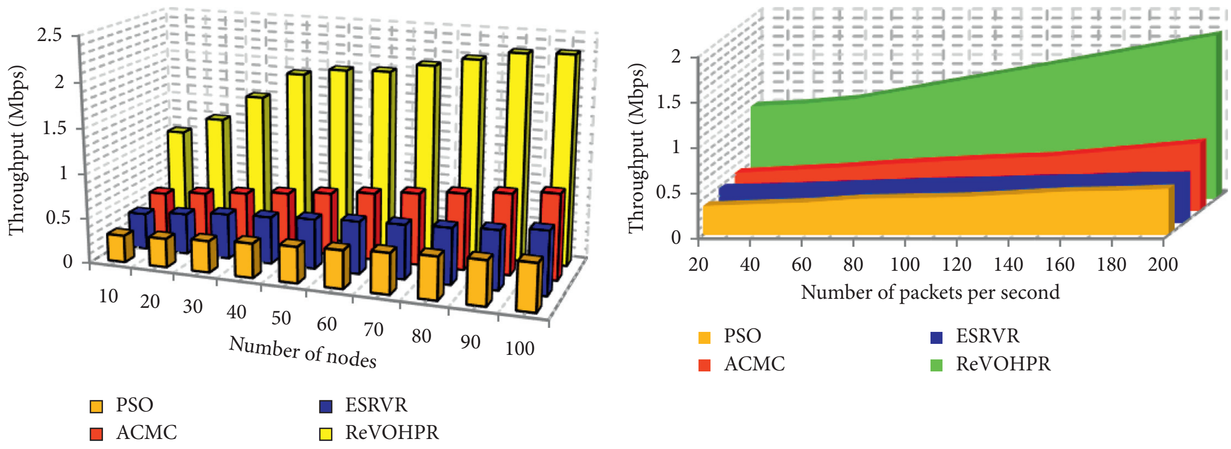
Figure 13 shows the performance throughput with respect to the number of packets processed per second. Relay-based void hole detection and mitigation addresses and avoids the multivoid hole prediction. This problem improves energy efficiency and QoS-based metrics such as throughput,



(a)

(b)

Figure 12: Impact of delay.



(a)

(b)

Figure 13: Impact of throughput.

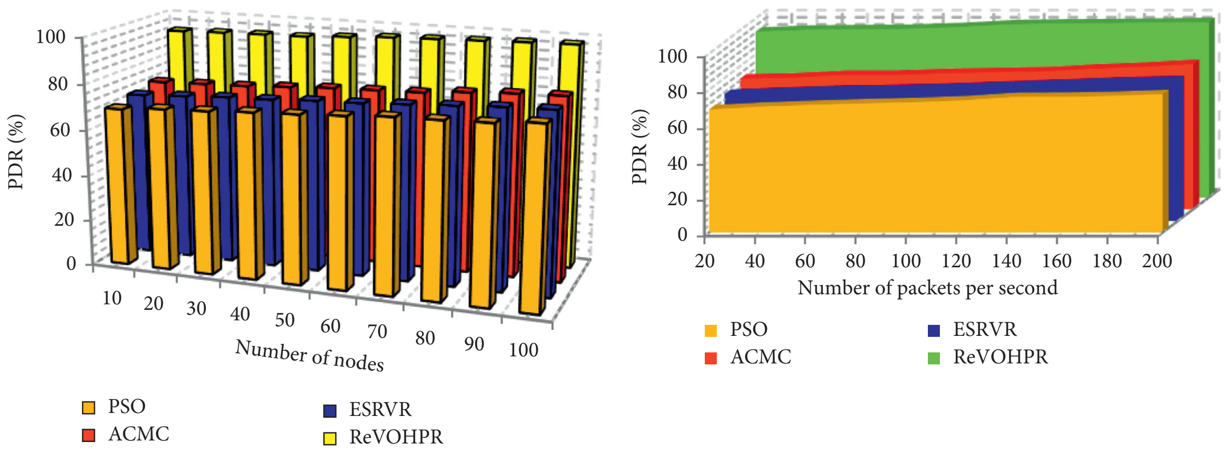
delay, and PDR. In ACMC, clusters are formed using the K-means algorithm in which centroid selection must be optimum and cluster size must be known. Data transfer time and mode are frequently changed and adapted in the proposed protocol. Hence, we obtained higher throughput than the previous protocols. We have used the end-to end approach to improve the network throughput for a longer period.

5.3.4. PDR. PDR is the packet delivery rate metric analyzed for every node in the network in data transmission time. It is defined as the sum of packets successfully received at the destination node from the source node. It is computed as follows:

$$\mathrm{PDR} = \frac{N(P(R))}{N(P(G))}, \tag{21}$$

where $N(P(R))$ is the number of packets received at the source node and $N(P(G))$ is the number of packets generated at the source node.

Figure 14 shows the simulation results for the PDR with respect to the number of nodes and number of packets processed per second. The plot of PDR in ReVOHPR increases when the network density increases. Optimum relay selection for routing packets from source CH to the destination CH improves the PDR, and also a number of void holes are detected and mitigated in the proposed protocol. The existing protocol, i.e., ACMC, uses simple void handling procedure that failed since single metric is considered for void hole detection. On behalf of void hole mitigation, packets transmitted to the next hop are guaranteed, and also it ensures the packet delivery. When void hole is determined, then relay node is near to use as a replacing part, and in contrast, a previous protocol such as PSO and ESRVR does not fit for robust data transmission. In PSO, two-level CHs are elected which increases overhead in data transmission. Furthermore, in PSO, void hole prediction consumes more processing and hence packet losses are very high whereas ESRVR uses two hops for data transmission. The selection for two hops is not reliable in this work. Hence, PDR is very lower.



(a)

(b)

Figure 14: Impact of PDR.

5.4. Results and Discussion. Based on the simulation results of the proposed REVOHPR protocol with the previous protocols, it is concluded that the REVOHPR offers various benefits. Due to the development of a new protocol for data transmission and void hole mitigation in UWSN, the number of active nodes rate increases, where the dead node count is reduced. The minimization of the overall energy consumption and the improvement of the overall QoS in the proposed REVOHPR is achieved due to the following set of processes implemented.

(i) Cluster formation and optimum CH selection by E2R protocol that improves these processes and avoids frequent cluster formation by considering the centrality factor and success rate of each node. By this method 37% of energy consumption is reduced.

(ii) Void hole repair algorithm addresses the packet dropping issues and also eliminates the packet retransmission.

(iii) Virtual graph construction process reduces the complexity, which increases the lifetime of network than the previous protocols.

6. Conclusion

In this paper, the void hole problem is addressed for energy consumption reduction. For this purpose, ReVOHPR protocol is proposed which deals with the four processes, the level-based clustering in which E2R protocol is presented for stable CH selection. Then, dynamic sleep scheduling mechanism is considering to improve the lifetime of a network which is dynamic by implementing the DK2F algorithm. The virtual graph-based routing algorithm is presented for data transmission in which virtual route is established between the source CH and the destination. To avoid data transmission delay, multiple AUVs deployed to gather data packets. Finally, relay-assisted void hole detection and repair is presented which eliminates the multiple void hole problems for a longer period. Our simulation results show that the proposed ReVOHPR protocol exceeds the performance than baseline protocols as ESRVR, ACMC, and PSO in terms of energy consumption, packet delivery ratio, throughput, and delay.

In the future, we planned to focus on the security aspect of data transmission to avoid threats in UWSN. In this case, various attacks in UWSN are detected and mitigated to further reduce the energy consumption and improve the QoS [42–44].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the RUDN University Strategic Academic Leadership Program and by the Prince Sattam Bin Abdulaziz University, Saudi Arabia, via the Deanship for Scientific Research funding for the Advanced Computational Intelligence and Intelligent Systems Engineering (ACIISE) Research Group, project number 2020/01/12173.

References

[1] G. Yang, L. Dai, G. Si, S. Wang, and S. Wang, *Challenges and Security Issues in Underwater Wireless Sensor Networks*, IIKI, Beijing, China, 2018.

[2] N. Kanthimathi and Dejey, “Void handling using geo-opportunistic routing in underwater wireless sensor networks,” *Computers & Electrical Engineering*, vol. 64, pp. 365–379, 2017.

[3] R. W. L. Coutinho, A. Boukerche, L. F. M. Vieira, and A. A. F. Loureiro, “Performance modeling and analysis of void-handling methodologies in underwater wireless sensor networks,” *Computer Networks*, vol. 126, pp. 1–14, 2017.

[4] M. Ahmed, M. Salleh, and M. I. Channa, "Routing protocols based on protocol operations for underwater wireless sensor network: a survey," *Egyptian Informatics Journal*, vol. 19, no. 1, pp. 57–62, 2018.

[5] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, "A cluster-based mobile data-gathering scheme for underwater sensor networks," in *Proceedings of the 2018 International Symposium on Networks, Computers and Communications (ISNCC)*, June 2018.

[6] J. Ma, S. Shi, X. Gu, and F. Wang, "Heuristic mobile data gathering for wireless sensor networks via trajectory control," *International Journal of Distributed Sensor Networks*, vol. 16, pp. 1–12, 2020.

[7] Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato, and A. A. A. El-Latif, "A secure federated learning framework for 5G networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 24–31, 2020.

[8] V. Artem, M. Al-Sveiti, I. A. Elgendy, A. S. Kovtunenko, and A. Muthanna, "Detection and recognition of moving biological objects for autonomous vehicles using intelligent edge computing/LoRaWAN mesh system, lecture notes in computer science," *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, Springer, Cham, Switzerland, pp. 3–15, 2020.

[9] R. Shakila and B. Paramasivan, "Performance analysis of submarine detection in underwater wireless sensor networks for naval application," *Microprocessors and Microsystems*, vol. 13, Article ID 103293, 2020.

[10] D. Wang, J. Liu, and D. Yao, "An energy-efficient distributed adaptive cooperative routing based on reinforcement learning in wireless multimedia sensor networks," *Computer Networks*, vol. 178, no. 4, Article ID 107313, 2020.

[11] M. Jouhari, K. Ibrahimi, H. Tembine, and J. Ben-Othman, "Underwater wireless sensor networks: a survey on enabling technologies, localization protocols, and internet of underwater things," *IEEE Access*, vol. 7, pp. 96879–96899, 2019.

[12] M. Khayyat, A. Alshahrani, S. Alharbi, I. Elgendy, A. Paramonov, and A. Koucheryavy, "Multilevel service-provisioning-based autonomous vehicle applications," *Sustainability*, vol. 12, no. 6, p. 2497, 2020.

[13] S. Basagni, V. D. Valerio, P. Gjanci, and C. Petrioli, "MARLIN-Q: multi-modal communications for reliable and low-latency underwater data delivery," *Ad Hoc Networks*, vol. 82, pp. 134–145, 2018.

[14] N. Ismat, R. Qureshi, R. N. Enam, S. Noor, and M. Tahir, "Cluster estimation in terrestrial and underwater sensor networks," *Wireless Personal Communications*, vol. 116, no. 2, pp. 1443–1462, 2020.

[15] M. Zhang and W. Cai, "Energy-efficient depth based probabilistic routing within 2-hop neighborhood for underwater sensor networks," *IEEE Sensors Letters*, vol. 4, no. 6, pp. 1–4, 2020.

[16] M. Ahmed, M. Salleh, and M. I. Channa, "Routing protocols for underwater wireless sensor networks based on data forwarding: a review," *Telecommunication Systems*, vol. 65, no. 1, pp. 139–153, 2016.

[17] Z. Wang, G. Han, H. Qin, S. Zhang, and Y. Sui, "An energy-aware and void-avoidable routing protocol for underwater sensor networks," *IEEE Access*, vol. 6, pp. 7792–7801, 2018.

[18] M. Ahmed, M. Salleh, and M. I. Channa, "CBE2R: clustered-based energy efficient routing protocol for underwater wireless sensor network," *International Journal of Electronics*, vol. 105, no. 11, pp. 1916–1930, 2018.

[19] M. Chen and D. Zhu, "Data collection from underwater acoustic sensor networks based on optimization algorithms," *Computing*, vol. 102, no. 1, pp. 83–104, 2019.

[20] A. Jamshidi, "Efficient cooperative ARQ protocols based on relay selection in underwater acoustic communication sensor networks," *Wireless Networks*, vol. 25, no. 8, pp. 4815–4827, 2018.

[21] N. Javaid, F. Ahmed, Z. Wadud, N. Alrajeh, M. Alabed, and M. Ilahi, "Two hop adaptive vector based quality forwarding for void hole avoidance in underwater WSNs," *Sensors*, vol. 17, 2017.

[22] N. Javaid, O. Karim, A. Sher, M. Imran, A. Yasar, and M. Guizani, "Q-Learning for energy balancing and avoiding the void hole routing protocol in underwater sensor networks," in *Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 702–706, Limassol, Cyprus, June 2018.

[23] G. Latif, N. Javaid, A. Sher, M. Khan, T. Hameed, and W. Abbas, "An efficient routing algorithm for void hole avoidance in underwater wireless sensor networks," in *Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pp. 305–310, Krakow, Poland, May 2018.

[24] A. Sher, A. Khan, N. Javaid, S. H. Ahmed, M. Y. Aalsalem, and W. Khan, "Void hole avoidance for reliable data delivery in IoT enabled underwater wireless sensor networks," *Sensors*, vol. 18, 2018.

[25] A. Signori, F. Campagnaro, F. Steinmetz, B.-C. Renner, and M. Zorzi, "Data gathering from a multimodal dense underwater acoustic sensor network deployed in shallow fresh water scenarios," *Journal of Sensor and Actuator Networks*, vol. 8, no. 4, p. 55, 2019.

[26] F. Banaeizadeh and A. Toroghi Haghighat, "An energy-efficient data gathering scheme in underwater wireless sensor networks using a mobile sink," *International Journal of Information Technology*, vol. 12, no. 2, pp. 513–522, 2020.

[27] S. Kumari, P. K. Mishra, and V. Anand, "Fault resilient routing based on moth flame optimization scheme for underwater wireless sensor networks," *Wireless Networks*, vol. 26, no. 2, pp. 1417–1431, 2020.

[28] S. M. Ghoreyshi, A. Shahrabi, T. Boutaleb, and M. Khalily, "Mobile data gathering with hop-constrained clustering in underwater sensor networks," *IEEE Access*, vol. 7, pp. 21118–21132, 2019.

[29] M. T. R. Khan, S. H. Ahmed, and D. Kim, "AUV-aided energy-efficient clustering in the internet of underwater things," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 4, pp. 1132–1141, 2019.

[30] W. Zhang, J. Wang, G. Han, X. Zhang, and Y. Feng, "A cluster sleep-wake scheduling algorithm based on 3D topology control in underwater sensor networks," *Sensors*, vol. 19, 2019.

[31] T. Islam and S.-H. Park, "A two-stage routing protocol for partitioned underwater wireless sensor networks," *Symmetry*, vol. 12, no. 5, p. 783, 2020.

[32] Z. Wadud, M. Ismail, A. B. Qazi et al., "An energy balanced efficient and reliable routing protocol for underwater wireless sensor networks," *IEEE Access*, vol. 7, pp. 175980–175999, 2019.

[33] Q. Guan, F. Ji, Y. Liu, H. Yu, and W. Chen, "Distance-vector-based opportunistic routing for underwater acoustic sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3831–3839, 2019.

[34] M. Akbar, N. Javaid, A. Khan, M. Imran, M. Shoaib, and A. Vasilakos, "Efficient data gathering in 3D linear

underwater wireless sensor networks using sink mobility," *Sensors*, vol. 16, no. 3, p. 404, 2016.

[35] M. Faheem, M. A. Ngadi, and V. C. Gungor, "Energy efficient multi-objective evolutionary routing scheme for reliable data gathering in internet of underwater acoustic sensor networks," *Ad Hoc Networks*, vol. 93, Article ID 101912, 2019.

[36] V. Krishnaswamy and S. Manvi, "Fuzzy and PSO based clustering scheme in underwater acoustic sensor networks using energy and distance parameters," *Wireless Personal Communications*, vol. 108, no. 3, pp. 1529–1546, 2019.

[37] M. Awais, I. Ali, T. A. Alghamdi et al., "Towards void hole alleviation: enhanced GEographic and opportunistic routing protocols in harsh underwater WSNs," *IEEE Access*, vol. 8, pp. 96592–96605, 2020.

[38] A. Khan, K. Aurangzeb, E.-U.-H. Qazi, and A. Ur Rahman, "Energy-aware scalable reliable and void-hole mitigation routing for sparsely deployed underwater acoustic networks," *Applied Sciences*, vol. 10, no. 1, p. 177, 2019.

[39] Q. Wang, J. Li, Q. Qi, P. Zhou, and D. O. Wu, "A game theoretic routing protocol for 3D underwater acoustic sensor networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9846–9857, 2020.

[40] M. Huang, K. Zhang, Z. Zeng, T. Wang, and Y. Liu, "An AUV-assisted data gathering scheme based on clustering and matrix completion for Smart ocean," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9904–9918, 2020.

[41] Z. Jin, Q. Zhao, and Y. Luo, "Routing void prediction and repairing in AUV-assisted underwater acoustic sensor networks," *IEEE Access*, vol. 8, pp. 54200–54212, 2020.

[42] K. Abbas, L. A. A. Tawalbeh, A. Rafiq et al., "Convergence of blockchain and IoT for secure transportation systems in smart cities," *Security and Communication Networks*, vol. 2021, Article ID 5597679, 13 pages, 2021.

[43] G. N. Nguyen, N. H. L. Viet, M. Elhoseny, K. Shankar, B. B. Gupta, and A. A. A. El-Latif, "Secure blockchain enabled cyber-physical systems in healthcare using deep belief network with ResNet model," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 150–160, 2021.

[44] W. Z. Zhang, I. A. Elgendy, M. Hammad et al., "Secure and optimized load balancing for multi-tier IoT and edge-cloud computing systems," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8119–8132, 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5580089, 16 pages
https://doi.org/10.1155/2021/5580089

Research Article

Towards a Smart Privacy-Preserving Incentive Mechanism for Vehicular Crowd Sensing

Lingling Wang, Zhongda Cao, Peng Zhou, and Xueqin Zhao

School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, China

Correspondence should be addressed to Lingling Wang; wanglingling@qust.edu.cn

Received 3 March 2021; Accepted 30 April 2021; Published 17 May 2021

Academic Editor: Anjia Yang

Vehicular crowd sensing is a promising approach to address the problem of traffic data collection by leveraging the power of vehicles. In various applications of vehicular crowd sensing, there exist two burning issues. First, privacy can be easily compromised when a vehicle is performing a crowd sensing task. Second, vehicles have no incentive to submit high-quality data due to the lack of fairness, which means that everyone gets the same paid, regardless of the quality of the submitted data. To address these issues, we propose a smart privacy-preserving incentive mechanism (SPPIM) for vehicular crowd sensing. Specifically, we first propose a new SPPIM model for the scenario of vehicular crowd sensing via smart contract on the blockchain. Then, we design a privacy-preserving incentive mechanism based on budget-limited reverse auction. Anonymous authentication based on zero-knowledge proof is utilized to ensure the privacy preservation of vehicles. To ensure fairness, the reward payments of winning vehicles are determined by not only the bids of vehicles but also their reputation and the data quality. Then, any rewarded vehicle can get the fair payment; on the contrary, malicious vehicles or task initiators will be punished. Finally, SPPIM is implemented by using smart contracts written via Solidity on a local Ethereum blockchain network. Both security analysis and experimental results show that the proposed SPPIM achieves privacy preservation and fair incentives at acceptable execution costs.

1. Introduction

As the population of cities starts to grow, the number of cars begins to increase, which has caused congestion problems on the roadways and the parking lots [1]. It is not only an inconvenience for commuters but can also cause billions of dollars in lost time and wasted fuel. Smart transportation is a solution to make real-time control decisions for traffic efficiency and security, where large amounts of traffic information are needed [2]. Nowadays, vehicles have more powerful sensoring, storing, and computing capabilities, and they are capable of collecting and sharing data. As for data acquisition, the ubiquity of crowd sensing has enabled the emergence of vehicular crowd sensing (VCS), which leverages the power of vehicles to collect massive traffic data [3]. As shown in Figure 1, when there is an emergent traffic event (e.g., rear-end accident or traffic jam) on the roadway, the vehicles around the location of the event can submit the real-time traffic data to the nearby road side units (RSUs), i.e., vehicles perform the crowd sensing task distributed by the transportation administration (TA) via RSUs. However, due to the resource consumption, fairness, and privacy leakage problems, vehicles may be reluctant to participate in crowd sensing tasks without an effective and fair incentive mechanism and privacy protection solutions.

Some privacy-preserving incentive mechanisms (PPIMs) have been proposed for protecting vehicles' privacy in VCS. However, these schemes either rely on a central platform [4] or lack of considering the fairness of the incentive mechanism [5], leading to collusion attack [6], potential privacy disclosure, or inadequate incentive. As the most popular distributed technology, blockchain has enabled incentive mechanism in VCS for secured authentication and collusion attack resistance. To be specific, smart contracts running on the blockchain take the place of the centralized platform to run the incentive mechanisms, which handles all interactions and overcomes the challenges of centralized execution, e.g., collusions between TA and RSUs, RSUs and

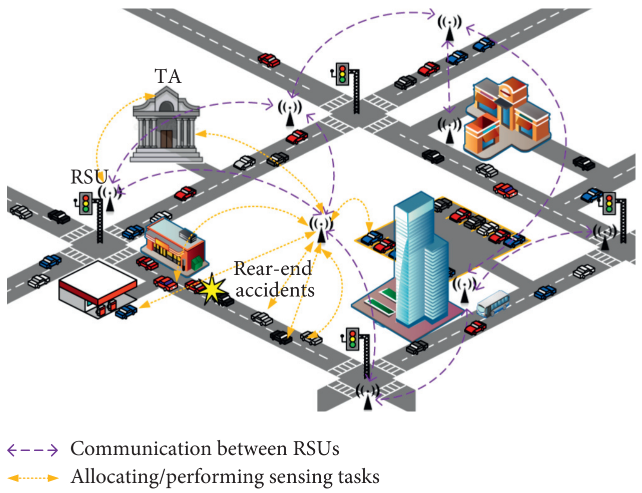

TA
RSU
Rear-end accidents
Communication between RSUs
Allocating/performing sensing tasks

Figure 1: A vehicular crowd sensing scenario of an emergent traffic condition.

vehicles. Although a few blockchain-based PPIMs [4, 5, 7] have been proposed, they either need a trusted third party to assist the privacy protection [4, 7] or lack the fairness of the payments [5].

To address the privacy issue, a common method is to take advantage of the anonymous mechanism, i.e., each vehicle has multiple pseudonyms or anonymous credentials which can be anonymously authenticated to protect the vehicle's privacy. However, there exists limitation to perform complex operations on a blockchain, e.g., in Ethereum, and the gas requirements of an operation cannot exceed the block gas limit. Hence, it is challengeable to design "light" operations of anonymous authentication on the blockchain to fulfill the privacy protection of the vehicles in VCS.

To address the fairness issue, most auction-based incentive mechanisms [4, 8–10] encourage vehicles to take part in crowd sensing tasks, where they submit bids to the central platform to compete for a task. The platform selects winning users to perform tasks and get paid. Nevertheless, it is unfair to determine the winners and the payment only depending on the bids and without considering the reputation of the vehicles and the submitted data quality. Hence, it cannot motivate people to submit high-quality sensory data.

In this paper, to address these challenges, we propose a smart privacy-preserving incentive mechanism (SPPIM) to stimulate the vehicles to submit high-quality sensory data and get fair payment with privacy protection. We focus on the vehicles' privacy protection without a trusted platform and aim to design a fair incentive mechanism which results in a rational payment according to the past and present performance of the vehicle. Specifically, the main contributions of this paper are as follows:

(i) We design a smart privacy-preserving incentive mechanism model and give an effective SPPIM based on budget-limited reverse auction via smart contract, which can ensure fairness of the payments for vehicles and data quality assurance for the task initiator.

(ii) SPPIM preserves the vehicles' privacy by using anonymous credentials without any trusted party. Meanwhile, bids preservation is achieved by using Pedersen commitment [11] from the vehicles to the task initiator. Anyone who obtains a committed bid, except the task initiator, is unable to get information about the bid's value.

(iii) We make a theoretical security and privacy analysis of the proposed SPPIM and evaluate the performance of the incentive mechanism by computing the utility of the vehicle and the task initiator. Furthermore, we implement the proposed SPPIM on the Ethereum testnet to verify its feasibility and provide a comprehensive analysis of the performance.

2. Problem Statement

In this section, we formalize the system model of vehicular crowd sensing, the smart PPIM model, and the threat model and also identify our design goals.

2.1. System Model. The system model mainly consists of the following four entities: block chain network, fog servers, task initiator, and vehicles as shown in Figure 2.

(i) Blockchain network has a decentralized and public ledger, which is shared with the legitimate miners and vehicles, and serves for SPPIM in vehicular crowd sensing. Smart contracts are designed to define and execute contracts, consisting of functions and data. Without a trusted platform, SPPIM is executed in a verifiable manner via smart contracts. New blocks, with all transactions of the incentive mechanism for vehicular crowd sensing task, will be audited and finally added to the block chain.

(ii) Fog servers are honest but are curious and connect with vehicles via wireless links. We assume that fog servers, acting as miners, have powerful computing and storage capabilities, and they act as the consensus nodes to maintain the blockchain network. Each fog server stores the whole ledger, which enables the validation of the blocks and transactions. Fog servers are also in charge of verifying the registration of the task initiator and vehicles and take control of the data quality.

(iii) Task initiator publishes the sensing task and pays the reward to winning vehicles via smart contracts. The task initiator communicates with the fog servers via the smart contracts, which are deployed on the fog servers. In our scenario, transportation administration (TA) takes the role of the task initiator.

(iv) Vehicles assume that there are N vehicles, denoted by $V = (V_1, V_2, \ldots, V_N)$, competing for a sensing task, and each vehicle V_j will submit a bid b_j, the current reputation R_j, and the sensory data D_j. Then, vehicles can get some rewards according to their reputation and the submitted data quality.

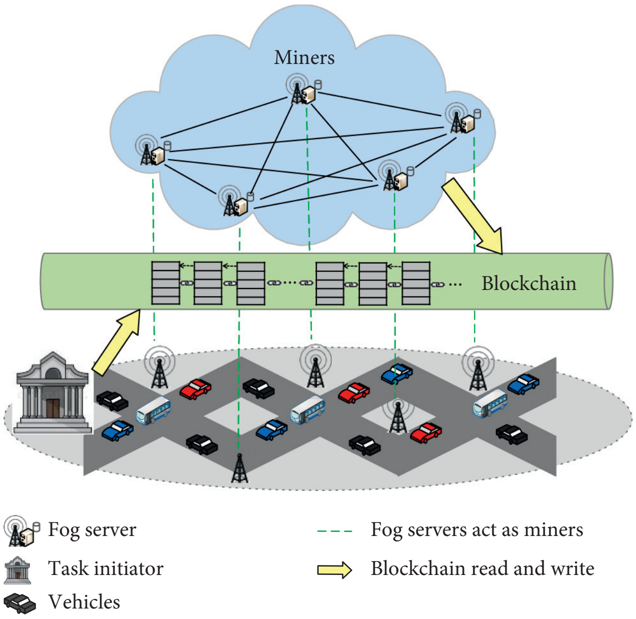


Figure 2: A system model of vehicular crowd sensing.

2.2. SPPIM Model. A decentralized PPIM system can be obtained by our SPPIM model as shown in Figure 3. We leverage the smart contract to replace the centralized platform. Our SPPIM model, based on budget-limited reverse auction [9], consists of a task initiator TA and N vehicles, i.e., $V = (V_1, V_2, \ldots, V_N)$. The task initiator wants to gather some data, such as emergent traffic conditions, and then publish a crowd sensing task. Vehicles are willing to collect this type of data and bid for the task. In this model, the task initiator acts as a buyer and vehicles act as sellers. All vehicles and the task initiator enter the auction process for reward payments and sensory data acquisition. The workflow of the proposed SPPIM model is as follows.

(i) TA deploys a sensing task via smart contract on the block chain.

(ii) Vehicles prove their legitimate identities to fog servers anonymously.

(iii) Legitimate vehicles provide their bids to fog servers, which run the deployed smart contracts to determine the winner set B_w of the auction.

(iv) Winning vehicles submit the required data, and fog servers calculate the reward payment P. Vehicles whose data quality meets the requirements get paid according to the data quality.

(v) TA gets the collected information from the fog server.

(vi) Fog servers verify all the transactions and build new blocks periodically.

We present the key notations used in this article in Table 1.

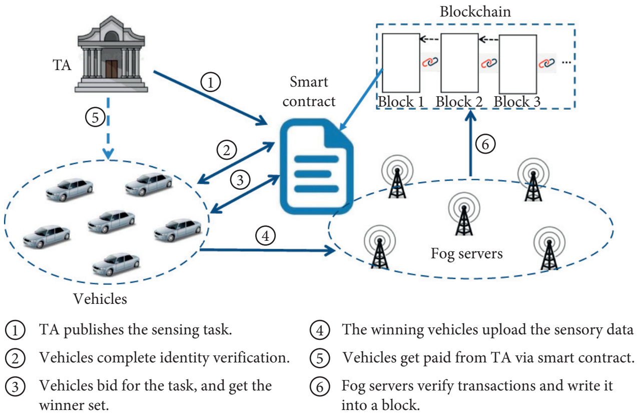

Blockchain
TA
Smart contract
Block 1
Block 2
Block 3
①
②
③
④
⑤
⑥
Vehicles
Fog servers
① TA publishes the sensing task.
② Vehicles complete identity verification.
③ Vehicles bid for the task, and get the winner set.
④ The winning vehicles upload the sensory data
⑤ Vehicles get paid from TA via smart contract.
⑥ Fog servers verify transactions and write it into a block.

Figure 3: The smart PPIM model of vehicular crowd sensing.

2.3. Threat Model. We assume that fog servers follow the protocols but are also curious about vehicles' privacy. Task initiators and vehicles are not trusted because they can launch attacks out of self-interests.

(i) Fog servers can launch passive attacks, and they are interested in the identity of task initiators and vehicles from the submitted messages and transactions. And they may be compromised or colluded with some vehicles or task initiator leading to privacy disclosure.

(ii) A task initiator may publish a sensing task without a reward guarantee and prematurely abort a task, and it may also try to obtain the private information of vehicles by accessing the block chain.

(iii) Vehicles are also curious about other vehicles' identities and bids. A dishonest vehicle may forget its reputation and try to get private information of other vehicles. A misbehaved vehicle may steal sensory data or collude with other vehicles to get extra rewards.

(iv) External adversary can eavesdrop the transmitting messages to violate vehicles' privacy. And it may impersonate a legitimate vehicle to perform a crowd sensing task and even trick a task initiator into paying for a reward.

2.4. Design Goal. Our goal is to design a privacy-preserving incentive mechanism with enhanced fairness for vehicular crowd sensing. Specifically, the proposed SPPIM will achieve the following requirements:

(i) *User Authentication*. No adversary can impersonate a legitimated vehicle. Any participant, including the task initiator and the vehicles, should be authenticated in an anonymous way.

(ii) *Identity Privacy*. The task initiator and the vehicles' privacy can be protected. Anyone including the fog

Table 1: Key notations.

Notation	Definition
λ	The security parameter
q	A large prime whose length is λ
Z_q	An additive group of order q
G, G_T	Two cyclic groups of the same prime order q
g, g_1	Generators of G
μ	A daily verification key
$e(.,.)$	A nondegradable bilinear mapping
$H(.)$	A collision-resistant hash function
Ω	A bloom filter for fast authentication factors
$(s, Y_F = g^s)$	The fog's private key and public key
X, Y, Z	$X = g^x, Y = g^y, Z = g^z$ for $x, y, z \in Z_q$
Cred	An anonymous credential of a vehicle
V	Legitimate vehicles consist of $V_1, V_2, \ldots, V_n$
$B_{\max}$	TA's budget
$\langle b_i, R_i, \mathrm{loc}_i \rangle$	The bid, reputation, and the location of vehicle V_i
C	Pederson commitment
D	The data structure of the sensory data
$\omega_i(c), \omega_i(t)$	The weight of the accuracy of the data and the submission time
q_i	The data quality of vehicle V_i
$m\prime$	The amount of vehicles who submit high-quality data
p_i	The reward for vehicle V_i in the reward payment P

server cannot identify vehicles' real identities when a task initiator publishes a task or a vehicle performs a sensing task.

(iii) *Bid Privacy*. All vehicles cannot know the bids submitted by others before committing to their own bids. This can help prevent the vehicles' collusion.

(iv) *Financial Fairness*. Vehicles get paid depending on their bids, the past, and current performance, i.e., reputation and the data quality. Meanwhile, vehicles or TA may attempt to deviate from the contract or prematurely abort, which will affect the SPPIM. The aborting parties will be financially penalized.

(v) *Collusion Attack Resistance*. If the TA or a fog server is compromised or colluded with some vehicles or task initiator, the SPPIM can still work well.

3. Preliminaries

We take advantage of the following cryptographic building blocks and technologies to construct our SPPIM.

3.1. Cryptographic Building Blocks. Bilinear Pairing [12]. Let G_1, G_2, and G_T be three cyclic groups of the same prime order q. A function e: $G_1 \times G_2 \longrightarrow G_T$ is a bilinear map if the following properties hold:

(i) Bilinearity: $e(u^a, v^b) = e(u, v)^{ab}$, for all $u \in G_1$, $v \in G_2$, and $a, b \in Z_q$

(ii) Nondegeneracy: $e(g_1, g_2) \neq 1$, where g_1 and g_2 are generators of G_1 and G_2, respectively

(iii) Computability: there exists an algorithm which can compute $e(u, v)$ efficiently for all $u \in G_1$ and $v \in G_2$

Zero-Knowledge Proof [13]. A zero-knowledge proof is a two-party (i.e., a prover and a verifier) protocol which allows a prover to convince the verifier that something is true without revealing any information. Specifically, a prover convinces a verifier of knowledge of values $(a_1, \ldots, a_n)$ that satisfy the predicate P denoted by

$$\mathrm{ZkPoK}\{(a_1, \ldots, a_n) | P(a_1, \ldots, a_n)\}, \tag{1}$$

ZkPoK can be used as an effective way to design a secure public-key cryptosystem. In this paper, we use zero-knowledge proofs to generate the anonymous credentials of vehicles and to complete the anonymous authentication.

3.2. Reverse Auction. The auction usually acts as an effective way to allocate goods or services to bidders who give the highest bidder [14]. An auction becomes a reverse auction when swapping the roles of the buyers and the sellers. The reverse auction model was first applied in a participatory perception system in Lee and Hoh [15] and has been widely used as a design model for incentive mechanisms in mobile crowd sensing [16–18]. Similarly, the reverse auction is a good solution for monetary incentives in vehicular crowd sensing, which encourages vehicles to sell their data.

In this paper, we use reverse auction with budget constraints [9] to model our incentive scenario. The vehicles act as sellers/bidders and will be selected to collect data. And the TA acts as a buyer, who purchases data provided by the vehicles with a limited budget.

3.3. Blockchain and Smart Contracts. Blockchain [19] is a distributed and public ledger which maintains an ever-growing list of digital transactions, which can be verified and audited by any users. Since blockchain provides a secure method for online transactions among anonymous participants, it is inherently consistent with our requirements, i.e., without a trusted third party. Recently, smart contract [20] has been adopted to allow users to define and execute contracts on the block chain. A smart contract is a computer code running on top of a blockchain containing a set of rules under which the parties to that smart contract agree to interact with each other. If and when the predefined rules are met, the instruction of the agreement is automatically enforced. The rules of transactions in our SPPIM can be enforced with smart contracts, which avoid the collusion attack [21] between the TA and the fog server, and then vehicles' equity is guaranteed.

4. The Proposed Smart Privacy-Preserving Incentive Mechanism

In this section, we first describe the overview of our SPPIM and then specify the detailed mechanism and the corresponding smart contracts.

4.1. Overview of the Proposed Incentive Mechanism. Our proposed SPPIM consists of anonymous authentication mechanism, privacy-preserving winner selection algorithm, and fairness-enhanced reward payment scheme.

(i) *Anonymous Authentication Mechanism.* The fog servers generate the system parameters and set the private key and public key. Vehicles get their anonymous credentials with the help of fog servers via the zero-knowledge proofs of knowledge. A fast authentication factor corresponding to each legitimate vehicle is stored in a bloom filter [22], which is kept on the blockchain for quick anonymous authentication. After passing the authentication, the vehicle can compete for a sensing task.

(ii) *Privacy-Preserving Winner Selection Mechanism.* When a vehicle competes for a sensing task, it first anonymously authenticates itself and bids for it. All legitimate vehicles and TA enter the reverse auction process for the sensing task. Since the location of the same abnormal traffic condition should not be very different, vehicles with excessive location deviation will be filtered. The winner selection mechanism also takes advantage of Pedersen commitment [23] to maintain bid's privacy. Vehicles first submit their commitments to the sealed bids on the smart contracts and then reveal the commitments secretly to the fog server, who runs the winner selection algorithm to determine the winner set of the task depending on the bids, reputations, and their precise locations.

(iii) *Fairness-Enhanced Reward Payment Scheme.* To enhance the fairness of payment, the payment profile should be generated according to the current and previous performance of vehicles. This can motivate the vehicles to actively take part in crowd sensing tasks and to provide high-quality data. In time-sensitive VCS scenarios, the untimely information is useless, so vehicles are required to submit their reports timely. Hence, the data quality is quantified by two factors: the data accuracy and the submission time. The payment profile is generated by the submitted bids and the quantified data quality of the vehicles. The task initiator TA pays and gets the balance, and the vehicles get rewards according to the payment profile in an anonymous way via smart contracts. Finally, fog servers verify and write the transactions into the block.

4.2. Detailed Mechanism

4.2.1. Anonymous Authentication Mechanism. The proposed anonymous authentication mechanism consists of ① system setup; ② anonymous certificate generation; and ③ anonymous authentication described as follows:

① System setup (offline):

(i) The fog server runs setup to obtain public parameters para $= \{G, G_T, q, g, g_1, g_T, e, H, X, Y, Z, \mu, Y_F\}$. (G, G_T) is a bilinear map group of a prime order $q > 2^{\lambda}$, where λ is the security parameter. $e(.,.)$ is the bilinear map satisfying $e\colon G \times G \longrightarrow G_T$. g and g_1 are generators of G, and $e(g, g)$ is defined as g_T. $H\colon Z_q \longrightarrow Z_q$ is a collision-resistant hash function. The fog server F selects $s \in Z_q$ randomly as its private key, and the public key is computed as $Y_F = g^s$. F also selects $x, y, z, \mu \in Z_q$ to compute $X = g^x$, $Y = g^y$, and $Z = g^z$. μ is a period verification key. F initializes an empty set Ω using bloom filter. Ω is reset periodically by the fog server because anonymous credential is only valid for a certain period.

(ii) Note that a vehicle cannot apply for more than one anonymous credential within one hour for the sake of security. We use a Boolean tag T to mark the state of the vehicle, and $T = 1$ represents that the vehicle has applied for an anonymous certificate at some point. T will be updated to be $T = 0$ once in a while, e.g., one hour or later.

② Anonymous certificate generation:

(i) Assume all vehicles are welcomed to compete for the sensing task. They need to subscribe for an anonymous credential when they want to perform the task. Once the vehicle requests an anonymous credential, the fog server will set the state tag $T = 1$. Then, the vehicle selects $(k, h) \in Z_q^2$ randomly, calculates $\Delta = Y^k Z^h$, and sends $(\Delta, H(k))$ to the local fog server.

(ii) The fog checks whether $H(k)$ does exist in Ω or not. If it does, the vehicle will be guided back to the above step. Otherwise, the fog server stores $H(k)$ into Ω. Here, we call $H(k)$ as the fast authentication factor. The fog server verifies the identity of the vehicle by the zero-knowledge proof of knowledge:

$$\text{ZkPoK}\left\{\{k, h\}\colon\ \Delta = Y^k Z^h\right\}. \tag{2}$$

(iii) The fog returns "failure" if the proof is unsuccessful. Otherwise, the fog sends (W, ν) to the vehicle, where $\nu \in Z_q$ and $W = (X\Delta)^{(1/\nu+s+\mu)}$.

(iv) The vehicle checks whether the equation $e(W, Y_F g^{\nu+\mu}) \overset{?}{=} e(X\Delta, g)$ holds. It returns "failure" if the equation does not hold. Otherwise, the vehicle's anonymous credential cred = (W, ν, k, h) is stored.

③ Anonymous authentication:

If a vehicle competes for a sensing task, it first authenticates itself by offering $H(k)$ to the fog servers. The fog runs the fast authentication algorithm to obtain TF. If $TF = 0$, which means $H(k)$ does not exist in Ω, the vehicle will be rejected as an illegal participant. Otherwise, the vehicle proves itself to the fog server in the zero-knowledge proof of knowledge:

$$\text{ZkPoK}\left\{(W, \nu, k, h)\colon\ W^{\nu+s+\mu} = XY^k Z^h\right\}. \tag{3}$$

If the proof is successful, the vehicle will be maintained as a legitimate candidate vehicle for bidding (Algorithm 1).

4.2.2. Privacy-Preserving Winner Selection Mechanism. The goal of the proposed winner selection mechanism is to select the winning vehicles with privacy preservation. Three factors, the submitted bid's value, the vehicle's location, and the reputation, have been combined to determine the winning vehicles. Reverse auction with TA's budget constraints is adopted to model our mechanism. All vehicles and the TA enter the auction process for crowd sensing task. Each vehicle acts as a bidder and submits a bid commitment. The winning vehicles are determined by the winner selection algorithm as shown in Algorithm 2.

(i) The vehicle chooses a random $\gamma \in Z_q$, computes the commitment of a bid $b \in Z_q$ as $C = g^b g_1^{\gamma}$ (see function Commit in section 4.3.), and then sends C to the local fog server.

(ii) Then, the vehicle reveals the values of b and γ (see function Decrypt and Reveal in section C) to open the commitment C. Each vehicle V_i sends the outcome ciphertext cipher$_i$ of encrypting (b_i, γ_i) by the public key of the local fog server Y_F.

(iii) The fog servers verify the correctness of the opening commitments to ensure that only the valid commitments store on the SPPIM contract.

Note that the ciphertext of b and γ is stored on the SPPIM contract rather than being sent directly to the fog server.

Assume the fog server F receives n bids $\langle C_i, R_i, \text{loc}_i\rangle$, $i = 1, \ldots, n$ from n legitimate vehicles $V = (V_1, V_2, \ldots, V_n)$, where C_i represents the commitment of the bid b_i submitted by V_i, R_i refers to its current reputation, and $\text{loc}_i = (l_1^i, l_2^i)$ is the location using longitude and latitude, respectively. F first computes the central position (l_1^0, l_2^0) of n locations and calculates the Euclidean distance ρ^i of (l_1^0, l_2^0) and $(l_1^i, l_2^i)_{i=1}^n$. Then, F checks whether $\rho^i < 100$ m holds. If $\rho^i > 100$ m, the accuracy of the data is not up to the standard and then the vehicle V_i will be rejected. Otherwise, the reward payment of the vehicle V_i will be computed depending on the submitted data quality.

The winner selection algorithm, depicted in Algorithm 2, is given by taking the bid set $B = \langle C_i, R_i, \text{loc}_i\rangle$, $i = 1, \ldots, n$, the ciphertext cipher$_i$, TA's budget $B_{\max}$, and the highest bid price b_0 as inputs. The output of the algorithm is the winner set B_w.

4.2.3. Fairness-Enhanced Reward Payment Scheme. We propose a fairness-enhanced reward payment scheme, where payment profile is generated depending on the data quality and the reputation of the vehicletbl2alg3.

① *Data Quality Measurement.* To measure the data quality submitted by the vehicle, the data structure of the sensory data is defined as $D =$ (task, cause, proof, time). The sensory data uploaded by the winning vehicles are stored in the form as Table 2.

(i) Task is represented by the task number to distinguish different crowd sensing tasks;

(ii) Cause refers to the cause of abnormal traffic conditions. For instance, "000″ means there is an accident at the location (l_1, l_2); "001" means there is a traffic jam at the location (l_3, l_2).

(iii) Proof is the evidence the vehicle can upload to prove the cause. How to identify the evidence is out of the scope of this paper.

(iv) Time is the current time of submitting the sensory data.

Assume the fog server F receives m sensory data $\{D_1, D_2, \ldots, D_m\}$ from m winning vehicles for the same task task, where $D_i = (\text{task}, c_i, \text{proof}_i, t_i)$. The data quality is quantified by the submission time t_i and the data accuracy, which is determined by hamming distance $d(.,.)$ of the *causes*.

Given cause c_i computes $m-1$ hamming distance $d(c_i, c_j)$ for all $j \neq i$ to measure the similarity of the abnormal traffic conditions. A weight $\omega_i(c)$ is assigned to measure the accuracy of the data as shown in Algorithm 3. A weight $\omega_i(t)$ is assigned to measure the submission time of vehicle V_i. The earlier the upload is, the greater weight the vehicle will gain. Finally, the data quality of V_i is quantified by

$$q_i = \theta\omega_i(c) + (1-\theta)\omega_i(t), \tag{4}$$

where θ denotes the importance of the data accuracy.

② *Payment Profile Generation.* The payment profile is generated by the data quality of the vehicle. For m vehicles, the sum of the submitted bids is $\sum_{i=1}^m b_i$, which satisfies $\sum_{i=1}^m b_i \leq B_{\max}$. Finally, the payment for the vehicle V_i is given as follows:

```
Input: $H(k)$, $\Omega$
Output: TF = $\{1, 0\}$
Check whether $H(k)$ exists in $\Omega$;
if $H(k) \notin \Omega$ then
TF = $\{0\}$;
else TF = $\{1\}$;
end if
return TF
```

Algorithm 1: Fast authentication algorithm.

```
Input: $B = \langle C_i, R_i, \mathrm{loc}_i \rangle$, cipher$_i$, $B_{\max}$ and $b_0$;
Output: The winner set $B_w$.
$B_w = \varnothing, B\ D = 0$;
$R_0 = (1/n) \sum_{k=1}^{n} R_k$;
for $(i = 1; i++; i \leq n \&\& \mathrm{BD} \leq B_{\max})$do
  $\rho^i = \sqrt{(l_1^i - l_1^0)^2 + (l_2^i - l_2^0)^2}$;
  if $\rho^i < 100\mathrm{m}$ then
    invoke Decrypt(cipher$_i$);
    invoke Reveal($C_i$);
    obtain $\langle b_i, R_i \rangle$;
    if $(R_i \geq R_0)$ then
      $B_w = B_w \cup \{b_i | b_i \leq b_0\}$;
      BD = BD + $b_i$;
    end if
  end if
end for
return $B_w$
```

Algorithm 2: Privacy-preserving winner selection algorithm.

Table 2: The storage format of the sensory data.

Task	Cause	Proof	Time
No. 3	000	*.jpg	9 : 00am
No. 3	001	*.mp4	9 : 01am
No. 3	000	*.jpg	9 : 03am
⋮	⋮	⋮	⋮

$$p_i = q_i \sum_{i=1}^{m} b_i. \quad (5)$$

This mechanism guarantees that as long as the vehicle provides higher quality data, they will get more rewards.

4.3. Smart PPIM Contract. In this section, our SPPIM is implemented via smart contract. After the SPPIM contracts are created, vehicles can take part in the crowd sensing task. The contract accepts the submitted messages from the TA and vehicles and executes the proposed algorithms automatically. Figure 4 illustrates the process of a SPPIM contract, involving all interactions among the TA, the vehicles, and the smart contract.

(i) TA and the vehicles first register on the fog servers. After the registration, vehicles get their anonymous credentials via zero-knowledge proof. TA can launch a crowd sensing task.

(ii) A new SPPIM contract is deployed on the blockchain. TA initiates a crowd sensing task.

(iii) Vehicles access the blockchain for new tasks and authenticate themselves by providing their anonymous credentials and the zero-knowledge proofs.

(iv) After vehicles pass the authentication step, they submit their sealed bids.

(v) The smart contract verifies the validity of the sealed bids and then executes the winner selection algorithm to determine winning vehicles.

```
Input: Sensory data $\{D_1, D_2, \ldots, D_m\}$ from $m$ winning vehicles, of which the reputations are $\{R_1, R_2, \ldots, R_m\}$;
Output: Payment profile $P = \{p_i\}_{i=1}^{m}$.
for $(i = 1; i++; i \le m)$ do
    counter$_i$ = 0;
    for $(j = 1; j++; j \le m)$ do
        if $d(c_i, c_j) = 0$ then
            counter$_i$ ++;
        end if
    end for
    if counter$_i \ge \lceil (m/2) \rceil$ then
        $\omega_i(c) = (1/m)(1 + (R_i/\sum_{\alpha=1}^{m'} R_\alpha))$;
    else
        $\omega_i(c) = (1/m)(1 - (R_i/\sum_{\alpha=1}^{m-m'} R_\alpha))$;
    end if
    Sort the submission time $t$ in a nondescending order and obtain $t_1 \le t_2 \le \ldots \le t_n$;
    $\omega_i(t)$ is assigned to measure $t_i$;
    $q_i = \theta\omega_i(c) + (1-\theta)\omega_i(t)$;
    $p_i = q_i \sum_{i=1}^{m} b_i$;
end for return $P$
```

Algorithm 3: Fairness-enhanced reward payment algorithm.

Table 3: A breakdown of gas costs for different functions of SPPIM contract when 10 of 20 vehicles are rewarded.

Function	Gas units	Gas cost (USD)
Create (.)	3774689	28.14
Authen (.)	4068500	30.39
Reveal (.)	1555410	11.63
WinnerSel (.)	1315028	9.75
Finalize (.)	688789	4.87

(vi) The winning vehicles submit the crowd sensing data to the fog servers.

(vii) The smart contract determines the payment profile by executing the reward payment algorithm.

(viii) The smart contract returns the balance of the TA, and vehicles get their rewards according to the payment profile.

Figure 5 provides the functions of the SPPIM contract in a detailed overview.

The Init (.) function defines all the parameters about the registration. Fog server calls the Init (.) function to get the public parameters para and generate anonymous credentials together with vehicles. After Init (.), fast authentication factors of legitimate vehicles are maintained in the bloom filter Ω.

The Create (.) function is used to deploy a new SPPIM contract on the blockchain. If TA wants to start a task, it calls Create (.) function with the parameters such as t_1, t_2, t_3, t_4, t_5, t_6 which define the time intervals for the six phases: the budget of a task TA.budget, the highest bidding price b_0, the legitimate vehicles set V, the list of bids B, the list of winning vehicles B_w, and the reward payment profile P. TA is required to pay at least TA.budget to the contract in order to prevent a malicious task initiator from initializing fake tasks and then withdrawing illegally. The highest bidding price b_0 is used to prevent malicious vehicles to submit an excessive bidding price. After the contract is deployed, it can be accessed by legitimate vehicles.

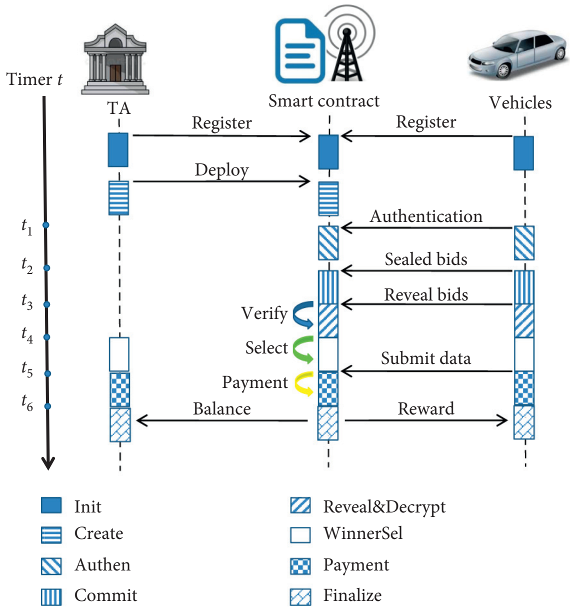


Figure 4: The process of the SPPIM contract.

The Authen (.) function authenticates all vehicles, which compete for the task, via the fast authentication algorithm and the zero-knowledge proof. Once the vehicle passes the authentication, it can submit bid.

The Commit(.) function seals the bids to protect them from being observed by other vehicles before the bidding interval ends. Pedersen commitment scheme is used to commit a bid. Each vehicle submits a bid commitment along with the location and the reputation of the vehicle.

The Reveal(.) function is triggered by the vehicles to reveal their bids. After that the contract can execute the winner selection algorithm. The inputs of Reveal(.) function are the cipher of the bids encrypted by the public key of the fog server Y_F. To avoid repudiation attack, the ciphertext is stored on the SPPIM contract rather than being sent directly to the fog server.

The Decrypt(.) function decrypts the ciphertext of the bids submitted by the vehicles.

The WinnerSel(.) function orders the bids after all bids are revealed to determine the winning vehicles. It takes as inputs the bids, the reputation, the location of the vehicles, the TA.budget, and the highest bidding price b_0. The result of the function is the winning vehicles set B_w.

The Payment(.) function computes the reward payment of each vehicle depending on the data quality, bids, and their reputation so that the potential vehicles can get reward payment.

The Finalize(.) function returns the balance of the TA and pays incentives to vehicles after the payment profile is determined.

5. Privacy and Security Analysis

This section proves that our proposed SPPIM achieves user authentication, identity privacy, bid privacy, financial fairness, and collusion attack resistance.

5.1. User Authentication. In our SPPIM, all vehicles need to authenticate themselves before performing any task. We use anonymous credentials to authenticate vehicles. The unforgeability of vehicle's identity is enabled by the security of the anonymous credentials generation. An adversary can authenticate himself by forging a verified credential and then showing it to the fog server, since the anonymous credential is generated through the zero-knowledge proof, of which the security is guaranteed by the CL signature scheme [24]. So, the proposed scheme satisfies the property of user authentication as long as the credentials are not forgeable.

Furthermore, a malicious vehicle may create multiple online identities to rig the mechanism. To prevent this attack, each vehicle should and must have only one valid anonymous credential when performing one crowd sensing task. Hence, in our SPPIM, we require that a vehicle can only apply for one anonymous credential within one hour. When the vehicle requests an anonymous credential, the fog server will set the state tag T to be 1. If a vehicle requests another anonymous credential within one hour, the fog can check the recorded state tag of the vehicle to refuse its request.

```
Init        $para = \{\}, \Omega = \{\text{factors}\}$
Create      upon receiving from TA ($t_1, t_2, t_3, t_4, t_5, t_6$, TA.budget, $b_0$):
            Set state: = INIT, Vehicles: = {}
            Set Winning vehicles $B_w$: ={ }, Payment Profile P: ={ }
            Assert $t < t_1 < t_2 < t_3 < t_4 < t_5 < t_6$
            Assert ledger[TA] ≥ TA.budget
            Set budget: = TA.budget, highestBid = $b_0$
Authen      upon receiving from vehicle V (cred, zk-proof):
            Assert $t_1 < t < t_2$
            Set V→Vehicles
Commit      upon receiving from a vehicle V (bid):
            Assert $t_2 < t < t_3$
            Assert V.commit = com (bid)
Reveal      upon receiving from avehicle V (ciphertext):
            Assert $t_3 < t < t_4$
            Assert $V \in$ Vehicles
            Set Vehicles[V].ciphertext : = ciphertext
Decrypt     upon receiving from a vehicle V(ciphertext):
            Assert $t_3 < t < t_4$
            Set V.bid = decrpy (ciphertext)
WinnerSel   upon receiving from vehicle V (B, ciphertext):
            Assert $t_4 < t < t_5$
            Algorithm 2 (B, ciphertext, TA.budget, $b_0$)→$B_w$
Payment     upon receiving $B_w$
            Assert $t_5 < t < t_6$
            Algorithm 3($B_w$, Data)→P
Finalize    upon receiving P
            Assert $t > t_6$
            Set ledger[$B_w$]: = P
```

Figure 5: Functions of the SPPIM contract.

5.2. Identity Privacy. We make sure the identity privacy of our SPPIM by proving the pseudonymity and unlinkability of vehicles.

First, each vehicle has different anonymous credentials cred = (W, v, k, h) corresponding to different tasks in our SPPIM. The anonymous credential can be verified by the smart contract as the valid anonymous credential. Hence, the vehicle's pseudonymity depends on the security of the zero-knowledge proof [24], of which the security proofs are relatively straightforward.

As for the unlinkability, the fog server cannot link vehicle's identity and the vehicle's anonymous credential during vehicle registration, and the fog cannot link the vehicle's different anonymous credentials. This property also depends on the zero-knowledge proof protocols. When a vehicle is applying for an anonymous credential, the fog server does not know the values of (k, h). Meanwhile, the anonymous credential cred = (W, v, k, h) can still be acknowledged as a valid BBS signature [25].

5.3. Bid Privacy. Our SPPIM protects the bid privacy by using Chaum–Pedersen noninteractive ZKP [23]. Vehicles send commitments rather than the actual bid. When the commitments need to be opened, each vehicle sends the ciphertext of (b, γ) using the public key of the fog server to the function Reveal(.). The ciphertext will be stored on the SPPIM contract rather than being sent directly to the fog server. And we also require that the fog server should verify the correctness of the commitments opening once they are submitted. This requirement can prevent the malicious fog server from denying a correct opening of a commitment. Given a semihonest fog server, all committed bids maintain privacy from other vehicles. This ensures bid privacy.

5.4. Financial Fairness. On the one hand, in the phase of committing bids, once the bid interval is closed (after t_3 in Figure 4), vehicles cannot change their commitments. This property can help guarantee the financial fairness from preventing some vehicles' cheating, which violates the fairness.

On the other hand, the payment profile is determined by the data quality, which depends on the data accuracy and the submission time detailed as the expression $q_i = \theta\omega_i(c) + (1-\theta)\omega_i(t)$ in Algorithm 3. In our reward payment algorithm, the weight of the data accuracy $\omega_i(c)$ is calculated according to the accuracy of the submitted data measured by the Hamming distances and the vehicles' reputation. The weight of the submission time $\omega_i(t)$ is given according to the corresponding speed of each vehicle. Vehicles with good performance can get higher reward, and their reputation ranking can also be increased. It is fair, and it can also stimulate honest vehicles to submit high-quality data in time.

In addition, vehicles or TA may try to deviate from the SPPIM and aborts early to affect SPPIM execution. The aborting task initiator will be financially penalized by forfeiting its budget money deposited on the ledger, while aborting vehicles will be punished by lowering the reputation rating.

5.5. Collusion Attack Resistance. In our SPPIM, we consider the collusions among vehicles and between the fog server and the vehicles. In our winner selection algorithm, the highest bid price b_0 is limited to prevent the malicious vehicles' collusion to bid for an over large bid price. Suppose there are some colluded vehicles which submit very high bids with the purpose of getting high reward. Under this circumstance, the sum of their bids must be larger than the budget of the TA, which is not allowed in our budget-limited reverse auction model. So, the fixed highest bid price b_0 can successfully prevent this attack.

If a fog server is compromised or even colludes with some TA or some vehicles, the SPPIM can run fine. Since vehicles generate part of their credentials by themselves, private information $(k, h) \in Z_q^2$ is also kept secret by vehicles. Fog server cannot divulge identity information about any other vehicle to some colluded vehicles. Once the smart contract is deployed, it cannot be changed. The rules of the proposed SPPIM are executed faithfully via smart contracts, which can avoid the collusion between the fog server and the vehicles and between the fog server and the TA.

6. Performance Evaluation

We conduct extensive experiments to evaluate the performance of the proposed SPPIM with multiple vehicles and a task initiator TA, including the computational and storage costs of authentication, the utility of the vehicle and the TA, and the gas cost of each function on the SPPIM contract.

6.1. Authentication Performance. The process of generating an anonymous credential is of the smart contract. We make a simulation related to the acquisition of anonymous credentials. We use JAVA pairing-based cryptography library to implement the cryptographic algorithms in our simulation. The number of total vehicles requesting for anonymous credentials N_{vehicles} is set as $\{100, 200, 300, 400, 500\}$, and the number of authenticated vehicles N_{authen} is set as $\{10, 20, 30, 40, 50\}$. In each set of experiments with different number of vehicles, we took an average result of 100 times round.

When the vehicle requests for an anonymous credential, the execution time is around 38 ms and 45 ms at the vehicle side and the fog side. Figure 6(a) shows that, as the number of requesting vehicles increases, the time spent on each vehicle and the fog server almost maintains the same. When a vehicle competes for a task, the execution time of anonymous authentication is 18 ms and 25 ms at the vehicle side and the fog side. Figure 6(b) shows that the total time of the authentication is 213 ms, 451 ms, 659 ms, 1091 ms, and 2162 ms assuming $N_{\text{authen}} = \{10, 20, 30, 40, 50\}$ at the fog side.

Figure 7(a) indicates that our SPPIM requires at most 112 byte bandwidth per authentication. Only the fast authentication factor $H(k)$ and the credential cred $= (W, v, k, h)$ need to be transmitted to the smart contract deployed on the fog server. As the number of authenticated vehicles increases to 50, the bandwidth requirement is less than 6 kb, which is feasible.

As for the storage cost, the fog server needs to maintain a list of fast authentication factors of legitimate vehicles and its private key at the fog side. Since the number of the legitimate vehicles is large, we use bloom filter Ω to help diminish the storage overheads, which depends on the size of the bloom filter. The vehicle only stores the anonymous credential cred $= (W, v, k, h)$. Figure 7(b) shows that the storage cost is very small at both the vehicle and fog server side.

6.2. SPPIM Performance. From the reward payment algorithm in Algorithm 3, we get that the utility of vehicle V_i is p_i and the utility of TA is

$$u_{\text{TA}} = B_{\max} - \sum_{i=1}^{m} p_i. \tag{6}$$

The proposed SPPIM is effective because it brings profits to the task initiator and the honest vehicles. In our experiments, we study several factors that affect the utility of the vehicle and the TA, including the number of rewarded vehicles, the budget of the TA, and the data quality. The number of the rewarded

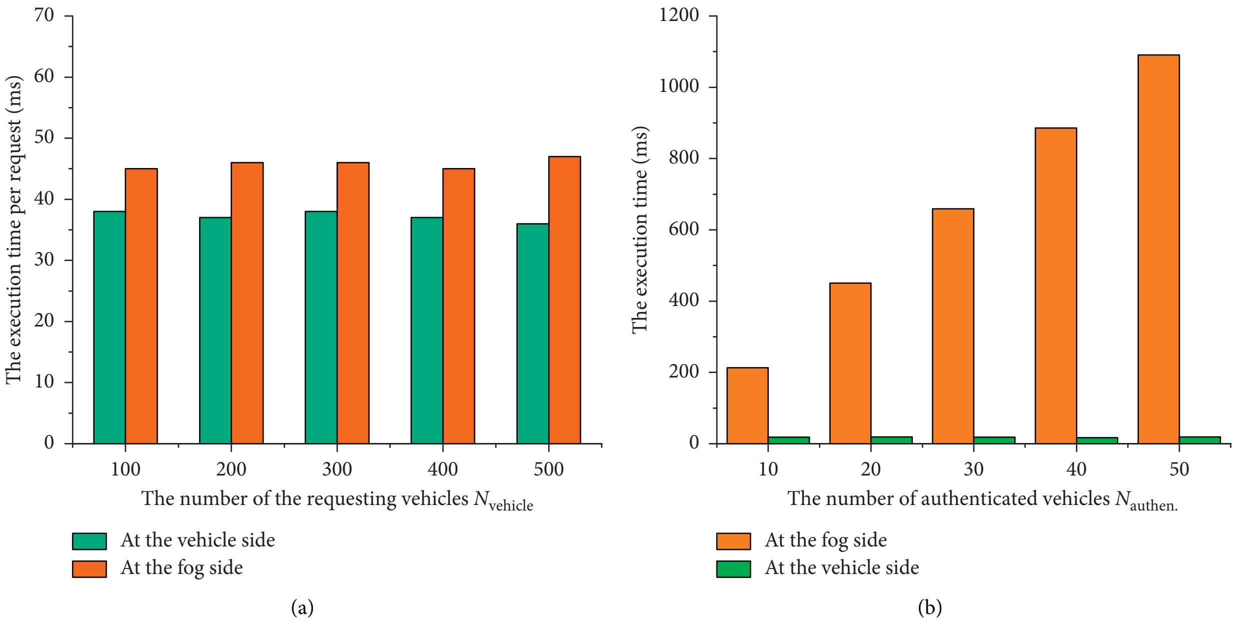


FIGURE 6: Computational costs for the vehicles and fog server: (a) computational costs of generating credential; (b) computational costs of generating authentication.

vehicles ranges from 10 to 50, and the submitted data are different in two ways: data accuracy and the submission time. As shown in Figures 8(a) and 8(b), the utility of the vehicle is nearly independent of the number of the rewarded vehicles N_{reward}, and the vehicle can get a higher utility when the budget of TA becomes bigger. Since the highest bidding price b_0 increases as the budget increases, vehicles can bid a higher price if b_0 is bigger. The utility p_i increases roughly linearly with the total bidding price of vehicles.

Figure 8(c) shows that the utility of the vehicle increases as the reputation of the vehicle increases on the premise that the data accuracy satisfies the requirement, given a fixed budget TA.budget = 200. If the submission time t_i of the vehicle is shorter (e.g., t_3 in Figure 8(c)), it can get more reward.

The utility of the TA is almost not influenced by the budget of the TA and the number of the rewarded vehicles as shown in Figure 8(d). The approximation number of the rewarded vehicles can be determined by the expression TA.budget/b_0. Once N_{reward} is fixed, the utility of the TA is determined by the payment P of the vehicles, and the average of each payment p_i is around b_0.

6.3. *SPPIM Contract Cost.*

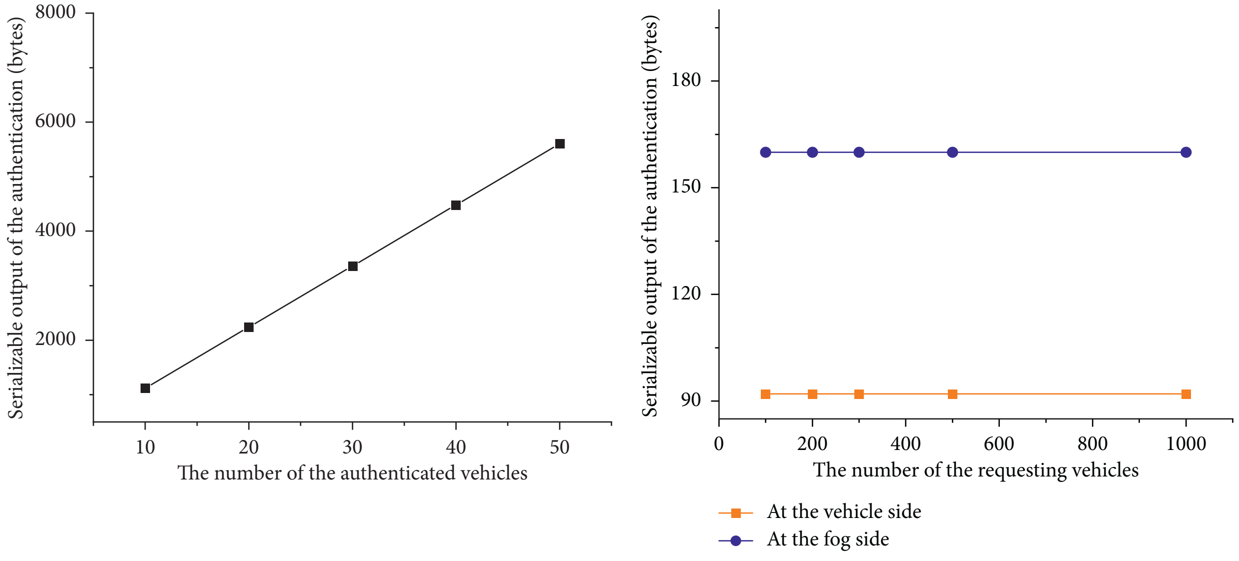
We implement the SPPIM contract in Solidity 0.4.18 [26] and test it on the Ethereum network. We run the experiments on a HP Pavilion Notebook with a 2.3 GHz Intel i5-6300HQ CPU and 8 GB RAM. To be specific, we create a local private Ethereum blockchain to test our SPPIM using the Geth client version 1.7.3 [27]. To realize the cryptographic algorithms on the SPPIM contract, we use Ethereum Improvement Proposals, EIP-196 [28], to fulfill elliptic curve point addition and scalar multiplication operations efficiently in the algorithm. Barreto–Naehrig E: $y^2 = x^3 + 3$ over F_q [29] is adopted in EIP-196.

Table 3 shows the consumed gas cost for different functions in SPPIM tested on the private Ethereum network where there are 20 vehicles competing for a task, and the number of the rewarded vehicles is 10. Table 3 gives the gas cost consumed by each function and the converted monetary value in US dollar. As of October 19, 2020, the ether exchange rate is 1 ether = 375.27\$ [30] and the gas price is approximately 20 Gwei = 20×10^{-9} ether. We find that the financial cost of running the SPPIM contract on the Ethereum network is within reasonable bounds. The Create (.) function, to deploy SPPIM contract on the blockchain, and the Authen (.), to verify zero-knowledge proof, cost more than other functions. However, the Create (.) function executes only once to create and deploy the SPPIM contract on the Ethereum network, so it is a one-time cost and requires no more cost for its maintenance.

The cost of the Authen (.) can be seen as the price of privacy protection, which increases linearly with the number of authenticated vehicles N_{authen}, as shown in Figure 9. Note that the execution of "heavy" functions in Ethereum is impossible due to the block gas limit. In Figure 9, the gas cost is over 8m when the number of authenticated vehicles is larger than 40. When the block gas limit is 8m, the maximum value of N_{authen} should be less than 40.

Figure 10 shows the gas cost of the TA and a vehicle when the number of authenticated vehicles N_{authen} and rewarded vehicles N_{reward} varies. TA's cost increases linearly with N_{reward} as shown in Figure 10(a), while the vehicle's cost keeps constant as shown in Figure 10(b).

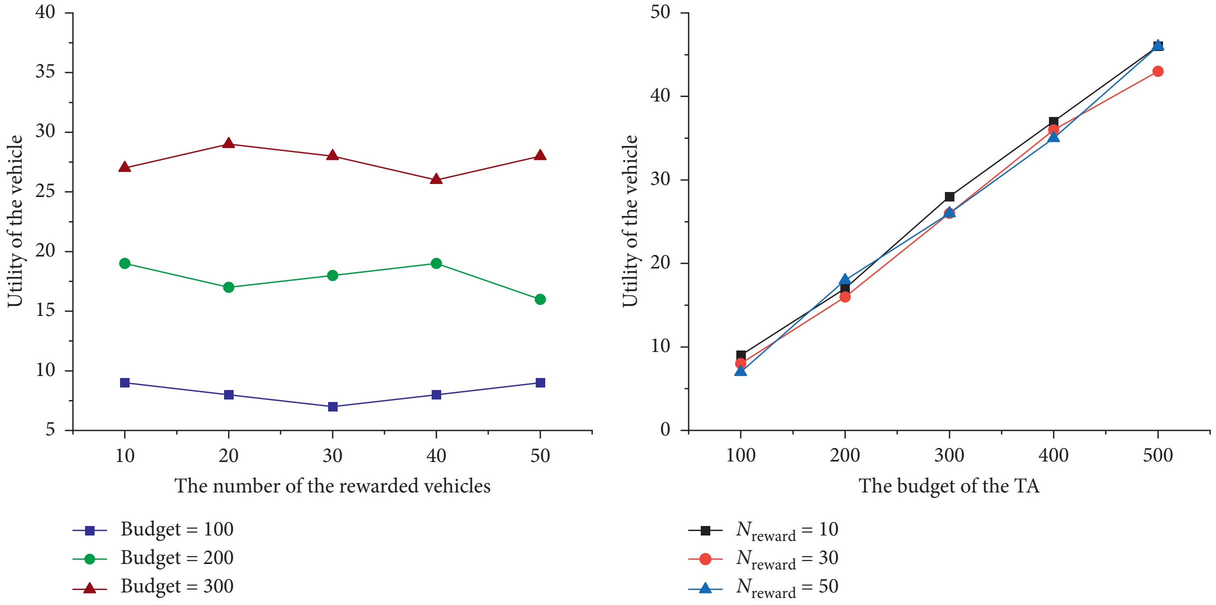
Figure 11 shows that the gas cost of the TA depends on the number of rewarded vehicles N_{reward}. Given a fixed N_{authen}, the gas cost of each function increases linearly with N_{reward} except for Create (.) function.



(a)

(b)

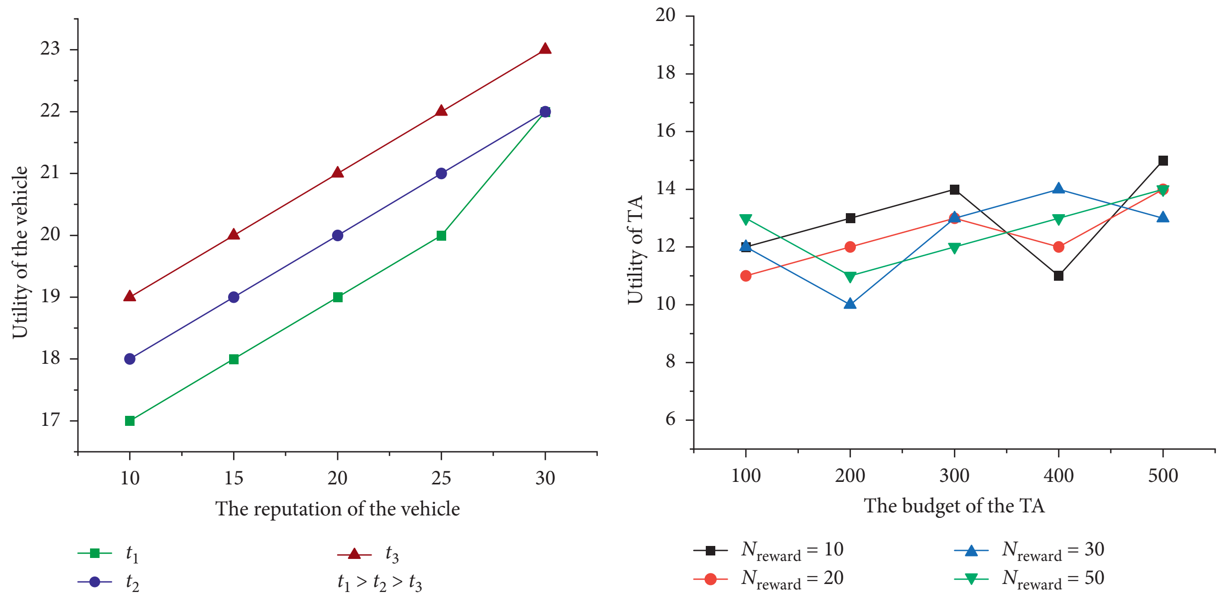
Figure 7: (a) Communication overheads between fog and vehicles; (b) storage costs at fog side and vehicle side.



(a)

(b)

Figure 8: Continued.



(c)

(d)

Figure 8: The utility analysis of the vehicle and the TA.

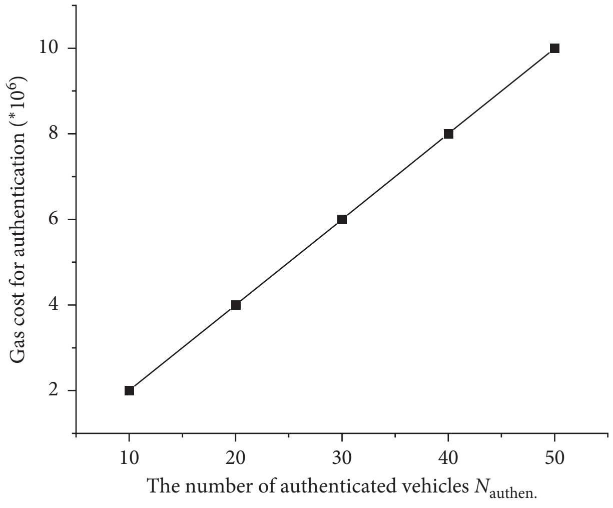


Figure 9: Gas cost for the authentication of vehicles.

7. Related Work

In this section, some related works are divided into three categories: (1) incentive mechanisms for vehicular crowd sensing; (2) blockchain-based works in vehicular network; and (3) privacy preservation for incentive mechanism in vehicular crowd sensing.

7.1. Incentive Mechanisms for Vehicular Crowd Sensing. The incentive mechanisms in vehicular crowd sensing mainly include monetary incentives and nonmonetary incentives, which stimulate vehicles via some forms of compensation, such as reputation [31], credits [32], and virtual coins [7]. Correspondingly, monetary incentive mechanism motivates vehicles to take part in tasks by financial incentives, which have stronger motivational effects and are easy to accomplish together with other incentives. Recently, Yin et al. [33] considered the scheduling problem of emergent tasks in the Internet of Vehicles and proposed a bidding mechanism to encourage vehicles to perform tasks. The winner vehicles can get some monetary reward after finishing the task. Li et al. [7] proposed an incentive announcement network, where users manage their reputation points which are earned or spent as incentives. Guo et al. [8] presented a dynamic incentive mechanism for mobile crowd

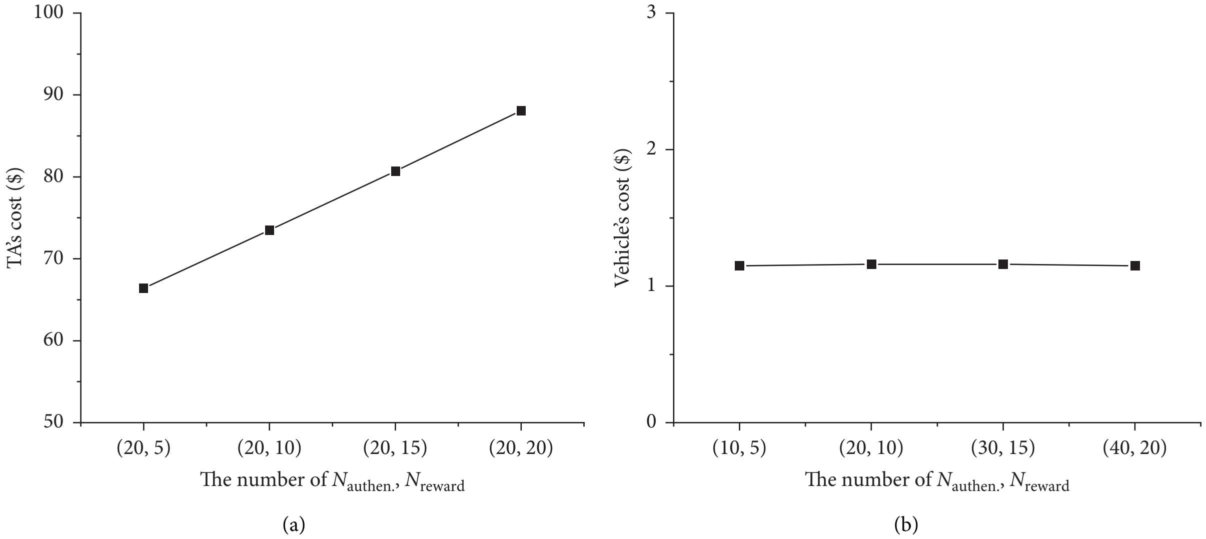


FIGURE 10: The average cost for the TA and a vehicle based on $(N_{\text{authen}}, N_{\text{reward}})$.

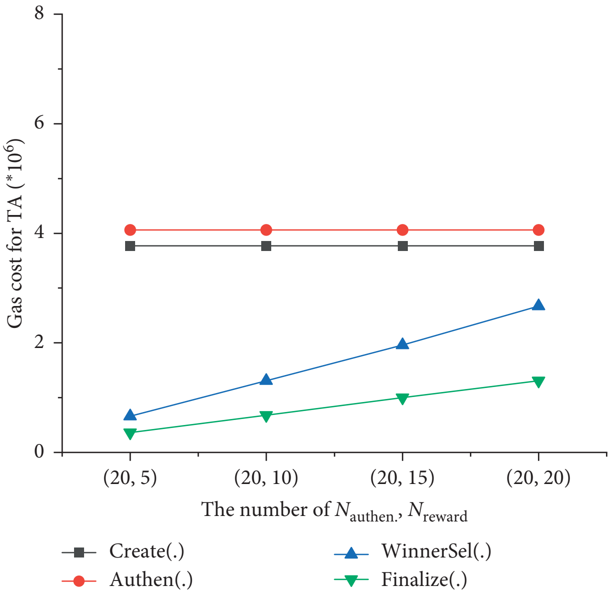


FIGURE 11: The gas cost for the TA based on $(N_{\text{authen}}, N_{\text{reward}})$.

sensing, and they pointed out that the quality of the sensing data is often neglected in the existing monetary-based incentive studies. Zhang et al. [9] presented an auction-based incentive mechanism in crowdsourcing systems, in which two allocation algorithms were given to guarantee the truthfulness and budget feasibility. In general, monetary incentives often increase user participation enthusiasm and enhance high-quality data collection habits [34]. However, most proposed monetary incentive mechanisms are centralized, which will cause privacy leakage and the single point of failure problem.

7.2. Some Works Based on Blockchain in Vehicular Network.

Nowadays, there have been several works [5, 7, 35, 36] related to blockchain technology in vehicular networks. Dorri et al. [35] proposed a blockchain-based privacy-preserving communication scheme for smart vehicles. Sharma et al. [36] gave a blockchain-based transport management system in the smart city. Li et al. [7] proposed an incentive announcement network based on blockchain for smart vehicles. Li et al. [5] constructed an anonymous advertising scheme in vehicular networks. Vehicles can send transactions to the blockchain to get a predefined reward, which does not consider the data quality problem of Ad dissemination.

7.3. Privacy Preservation for Incentive Mechanism in VCS.

In order to protect the privacy of vehicles, some privacy-preserving incentive mechanisms in VCS have been proposed. Lai et al. [32] took advantages of symmetric encryption to protect personal profiles and the designated verifier signature to preserve transaction privacy in highway VANETs. Wang et al. [37] gave a node cooperation privacy protection method based on k-anonymity technology. Ten or more nodes form a k-anonymous group and submit signcrypted group data. Miners verify the legality of group data by the group blind signature algorithm that could resist against user's privacy leakage. Similarly, Wang et al. [10] utilized differential privacy technology to obfuscate bids in mobile crowd sensing. Li et al. [7] proposed a privacy-preserving incentive mechanism in VANETs. Vehicles protect their privacy by acquiring other vehicles' encrypted signatures to construct a threshold ring signature. However, there is a trusted third party who needs to generate keys and can actually trace vehicles' privacy, which is different from our SPPIM design. Lai et al. [4] utilized a blockchain-based payment system to guarantee the fairness of payments. The partially blind signature was applied to realize pseudonym management, which is designed to protect the privacy of users. However, pseudonyms are assigned by a third-party authority, which is avoided in our design. The scheme proposed by Lai et al. is simulated in MATLAB, not fulfilled on the blockchain, while our SPPIM is accomplished by the smart contract on the blockchain.

Different from existing works, we propose a hybrid solution SPPIM to address the privacy preservation and fairness problem in incentive mechanisms of VCS. Our SPPIM not only utilizes smart contracts to replace the centralized platform but also addresses the privacy-preserving problem of the vehicles and the fairness problem of the incentive mechanisms.

8. Conclusion

In this paper, we propose an effective smart privacy-preserving incentive mechanism via smart contract on the blockchain, which can ensure privacy preservation and fairness for vehicles and data quality assurance for the task initiator. Our SPPIM preserves the privacy of vehicles by utilizing zero-knowledge proof-based anonymous credentials without any trusted third party. Meanwhile, fairness-enhanced reward payments are determined by the committed bids, the reputations, and the submitted data quality of the winning vehicles. We verify the performance and the feasibility of the proposed SPPIM by implementing it on the Ethereum testnet. In the future work, we will design the optimized algorithms to enrich our current design, which can reduce the execution cost of the contract.

Data Availability

The data and VS code used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant 61802217 and in part by the Natural Science Foundation of Shandong Province under grant ZR2020MF061.

References

[1] L. Wang, X. Lin, E. Zima, and C. Ma, "Towards airbnb-like privacy-enhanced private parking spot sharing based on blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2411–2423, 2020.

[2] IoT Solutions for Smart Cities and Smart Transportation (2020), https://www.telit.com/industries-solutions/smart-cities-smart-transportation/".

[3] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 146–152, 2017.

[4] C. Lai, M. Zhang, J. Cao et al., "SPIR: a secure and privacy-preserving incentive scheme for reliable real-time map updates," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 416–428, 2020.

[5] M. Li, J. Weng, A. Yang et al., "Toward blockchain-based fair and anonymous Ad dissemination in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 99, pp. 11248–11259, 2019.

[6] J. Wang, M. Li, and Y. He, "A blockchain based privacy-preserving incentive mechanism in crowdsensing applications," *IEEE Access*, vol. 6, p. 1, 2018.

[7] L. Li, J. Liu, L. Cheng et al., "CreditCoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.

[8] B. Guo, H. Chen, Z. Yu, W. Nan, X. Xie et al., "TaskMe: toward a dynamic and quality-enhanced incentive mechanism for mobile crowd sensing," *International Journal of Human-Computer Studies*, vol. 102, pp. 14–26, 2017.

[9] Q. Zhang, Y. Wen, X. Tian et al., "Incentivize crowd labeling under budget constraint," in *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2812–2820, IEEE, Hong Kong, China, May 2015.

[10] Z. Wang, J. Li, J. Hu et al., "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *Proceedings of the 2019 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2053–2061, IEEE, Paris, France, May 2019.

[11] T. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology—CRYPTO '91*Springer, Berlin, Germany, 1991.

[12] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3113–3121, 2008.

[13] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *Journal of Cryptology*, vol. 1, no. 2, pp. 77–94, 1988.

[14] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.

[15] J. Lee and B. Hoh, "Sell your experiences: a market mechanism based incentive for participatory sensing," in *Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 60–68, IEEE, Mannheim, Germany, April 2010.

[16] M. Xiao, K. Ma, A. Liu et al., “SRA: Secure reverse auction for task assignment in spatial crowdsourcing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 4, pp. 728–796, 2019.

[17] Y. Wei, Y. Zhu, H. Zhu et al., “Truthful online double auctions for dynamic mobile crowdsourcing,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pp. 2074–2082, Hong Kong, China, May 2015.

[18] X. Zhang, G. Xue, R. Yu et al., “Truthful incentive mechanisms for crowdsourcing,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pp. 2830–2838, Hong Kong, China, May 2015.

[19] S. Nakamoto: Bitcoin: “A peer-to-peer electronic cash system,” https://bitcoin.org/bitcoin.pdf.

[20] N. Szabo, “Formalizing and securing relationships on public networks,” *First Monday*, vol. 2, p. 9, 1997.

[21] S. Wu, Y. Chen, Q. Wang, M. Li, C. Wang, and X. Luo, “CReam: a smart contract enabled collusion-resistant e-auction,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1687–1701, 2019.

[22] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors,” *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[23] T. P. P.D. Chaum, “Wallet databases with observers,” in *Advances in Cryptology—CRYPTO’ 92*, pp. 89–105, Springer, Berlin, Germany, 1992.

[24] J. Camenisch and A. Lysyanskaya, “Signature schemes and anonymous credentials from bilinear maps,” *Advances in Cryptology—CRYPTO 2004*, Springer, Berlin, Germany, pp. 56–72, 2004.

[25] D. Boneh, X. Boyen, and H. Shacham, “Short group signatures,” *Advances in Cryptology—CRYPTO 2004*, Springer, Berlin, Germany, pp. 41–55, 2004.

[26] Solidity 0.4.18, available: “https://solidity.readthedocs.io/en/v0.4.18/.

[27] Geth 1.7.2, https://geth.ethereum.org/downloads/.

[28] R. Christian: EIP-196: “Precompiled Contracts for Addition and Scalar Multiplication on the Elliptic Curve Alt Bn128,” Ethereum Improvement Proposals, No. 196, 2017. https://eips.ethereum.org/EIPS/eip-196.

[29] P. Barreto and M. Naehrig, “Pairing-friendly elliptic curves of prime order,” in *Proceedings of SAC 2005*, pp. 319–331, Springer, Berlin, Germany, 2006.

[30] (2020) Ethereum Price. https://ethereumprice.org/.

[31] X. Wang, J. Zhang, X. Tian et al., “Crowdsensing-based consensus incident report for road traffic acquisition,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2536–2547, 2017.

[32] C. Lai, K. Zhang, N. Cheng et al., “SIRC: A secure incentive scheme for reliable cooperative downloading in highway VANETs,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1559–1574, 2016.

[33] B. Yin, Y. Wu, T. Hu et al., “An efficient collaboration and incentive mechanism for Internet of vehicles (IoV) with secured information exchange based on blockchains,” *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1582–1593, 2019.

[34] S. Reddy, D. Estrin, M. Hansen et al., “Examining micropayments for participatory sensing data collections,” in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, pp. 33–36, ACM, New York, NY, USA, 2010.

[35] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, “Blockchain: a distributed solution to automotive security and privacy,” *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.

[36] P. K. Sharma, S. Y. Moon, and J. H. Park, “Block-vn: a distributed blockchain based vehicular network architecture in smart city,” *Journal of Information Processing Systems*, vol. 13, no. 1, p. 84, 2017.

[37] J. Wang, M. Li, Y. He, H. Li, K. Xiao, and C. Wang, “A blockchain based privacy-preserving incentive mechanism in crowdsensing applications,” *IEEE Access*, vol. 6, pp. 17545–17556, 2018.