

Research on Radar Target Tracking Algorithms for Complex Traffic Scenarios

Yu LIU*, Ping LI*, Lele CUI*, Zhiwen DING*, Xiuming PENG*, Xiang GAO*

* National Key Laboratory of Mechatronic Engineering and Control, School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China

67577335@qq.com, Liping85@bit.edu.cn, 3120240256@bit.edu.cn, dzw@wiahua.com, 3220240164@bit.edu.cn, bitxianggao@163.com

Abstract— To address the security risks in vehicular network communication that current technologies cannot resolve, this paper proposes a multi-key data aggregation algorithm for the Internet of Vehicles (IoV). By adopting multi-key fully homomorphic encryption over rings, the algorithm ensures that data remains encrypted during transmission and aggregation, with encryption operations conducted over rings, thereby enhancing computational efficiency and decryption accuracy. This approach protects vehicle data in the IoV from unauthorized access by institutions or attackers and ensures the security and accuracy of the aggregation process. Throughout the data processing flow, multi-layer encryption and homomorphic operations further enhance data confidentiality and processing efficiency, providing effective technical support for IoV information security. This algorithm is suitable for IoV applications with high demands for data privacy and security.

Keywords— Internet of Vehicles; Multi-key; Data Aggregation; Homomorphic Operations; Data Security.

I. INTRODUCTION

Compared to traditional vehicle communication technologies, the Internet of Vehicles (IoV) undoubtedly possesses significant advantages. It offers various new functionalities that cater to modern development needs, including remote traffic information exchange and entertainment information sharing. Companies in IoV information technology can analyze real-time road condition data obtained from vehicle sensors and predict future traffic trends. Given the limited storage space of in-vehicle systems, applying cloud storage technology enables low-cost remote data management [1]. However, due to the massive volume of data in IoV, direct uploads face bandwidth pressures. Therefore, employing data aggregation techniques can greatly reduce bandwidth consumption while retaining essential statistical information. Nonetheless, data aggregation systems face a series of serious threats, including security attacks, fraud, and privacy breaches. The core focus of this research is how to quickly, efficiently, and securely aggregate real-time vehicle data while ensuring data security.

To ensure the security and accuracy of the Internet of Vehicles (IoV), [2] propose a lightweight Differential Privacy-based Federated Learning (DP-FL) framework, termed DPF-IVN, which takes into account the impacts of heterogeneity and privacy leakage within the IoV environment. This framework adopts a "dimensionality reduction followed by

optimization" approach to handle non-lightweight data in IoV. Additionally, a secure and efficient handover authentication protocol suitable for autonomous vehicles in small industrial parks has been proposed [3]. This protocol is based on blockchain technology and the Pedersen verifiable secret sharing scheme, which ensures lightweight real-time interactions between autonomous vehicles and edge servers in a multi-server environment while rigorously protecting the security and privacy of both the vehicles and edge servers. Compared to other related protocols, the proposed scheme meets a broader range of security requirements and offers advantages in terms of computational efficiency and communication costs. Almakayeel has introduced a deep learning-based improved Transformer model for Android malware detection (DLBITM-AMD) aimed at the Internet of Vehicles (IoVs). Experimental results have demonstrated that this approach achieves higher accuracy compared to existing Android malware detection models [4]. In practical data aggregation schemes, the confidentiality of the encryption methods employed must not be overlooked. Confidentiality, integrity, and availability are essential security requirements for data aggregation in the Internet of Vehicles (IoV). The achievement of these three security requirements relies on several key technologies [5].

Fully Homomorphic Encryption (FHE) is regarded as a promising technology for addressing privacy issues in the era of machine learning due to its capability to perform computations on encrypted data without the need for decryption [6]. Homomorphic encryption offers a promising solution by allowing computations on encrypted data while preserving privacy in machine learning. Nevertheless, its integration with neural networks is hindered by high computational demands and the challenges associated with executing complex nonlinear operations. To tackle these challenges, a new method has been proposed that combines a "Self-Learnable Activation Function" (SLAF) and optimizes the structure of the linear layers in neural networks [7].

Masking techniques provide efficient encrypted computation even under hardware constraints [8]. In this approach, a blind factor is typically used to obscure data generated by connected vehicles, allowing only the aggregation of all connected vehicles to offset all blind factors and obtain the final result. Differential privacy effectively obfuscates data by adding Laplacian noise, thereby concealing the true data [9]. A novel Local Differential Privacy (LDP)

deep learning method named AMOUE has been proposed, which dynamically adjusts the noise added to 1s and 0s based on the distribution of the input data [10]. Theoretical analysis indicates that AMOUE achieves a lower expected error and variance. Experiments on real-world datasets demonstrate that AMOUE outperforms other LDP-based mechanisms in terms of deep learning classification accuracy. An innovative framework has been introduced that leverages distributed computing strategies to enhance performance [11].

The data aggregation scheme employed must consider its availability. Typically, two techniques are utilized: pre-stored blind factors and secret sharing [12]. The pre-stored blind factors are primarily managed by the final decryption party. If a connected vehicle fails to submit data, the decryption party can retrieve that user's blind factor and apply it in the decryption process to obtain the correct final aggregation result. Additionally, [13] proposed that the aggregator can reconstruct the blind factor of the faulty connected vehicle by accessing shares stored in other connected vehicles, allowing for the decryption of the final aggregation result through ciphertext reconstruction.

To address the issues outlined above, this paper proposes a multi-key data aggregation method tailored for the Internet of Vehicles (IoV). This method efficiently aggregates encrypted data while assigning different encryption keys to each vehicle, effectively addressing the information transmission security concerns of smart connected vehicles and ensuring the safety of user data. The structure of this paper is as follows: the first section introduces the background and research status of traditional vehicle communication technologies and data aggregation algorithm; the second section presents the principles of the proposed multi-key data aggregation algorithm for IoV; the third section conducts a simulation analysis of the proposed multi-key data aggregation algorithm; and the fourth section summarizes the findings based on the simulation analysis.

II. PRELIMINARY AND CONCEPT

The primary objective of this paper is to propose a multi-key data aggregation algorithm for connected vehicles, which employs a multi-key ring-based fully homomorphic encryption technique. This approach ensures that only those parties assigned specific keys are involved in the data aggregation process of the entire system. During data transmission, all information is in ciphertext form, making it extremely difficult for attackers to decipher the encryption methods and forge the ciphertext. Furthermore, since all computations are performed within the ring structure, this algorithm enables faster processing, leading to enhanced aggregation efficiency and a 100% decryption accuracy rate. As shown in Figure 1, the workflow diagram of the multi-key data aggregation algorithm for connected vehicles is presented. The following is a detailed introduction to the workflow of the data aggregation algorithm.

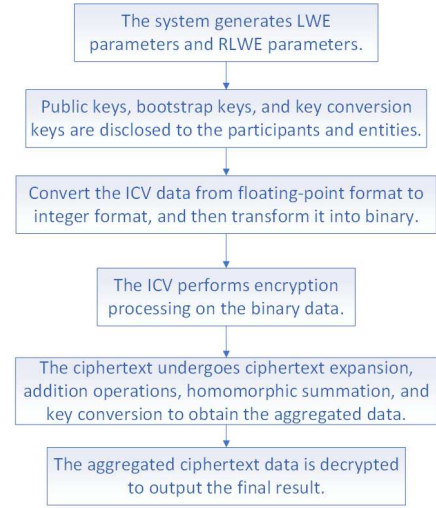


Figure 1. Flowchart of the Multi-Key Data Aggregation Algorithm

A. Set Fixed Parameters

In this algorithm, all logarithms are base 2 unless otherwise specified; vectors are denoted by lowercase letters, such as a , and matrices are denoted by uppercase letters, such as A . The symbol $\langle \cdot, \cdot \rangle$ represents the dot product of two vectors. For a real number r , $\lfloor r \rfloor$ denotes the integer closest to r , rounding up in the case of a tie. The notation $x \leftarrow D$ indicates that x is sampled according to the distribution D . For a finite set S , $U(S)$ denotes the uniform distribution over S . For a real number α , D_α represents a Gaussian distribution with variance α^2 . In this method, λ denotes the security parameter, indicating that all known effective attacks against the encryption scheme require $\Omega(2^\lambda)$ bits of computation. For a positive integer k , $[k] = \{1, 2, \dots, k\}$ represents the index set.

This process takes the initially set fixed parameters as input, calls algorithms $LWE.Setup(1^\lambda)$ and $RLWE.Setup(1^\lambda)$ to generate LWE parameter pp^{LWE} and RLWE parameter pp^{RLWE} , respectively, and outputs public parameter $pp^{MKFE} = (pp^{LWE}, pp^{RLWE})$.

$LWE.Setup(1^\lambda)$ refers to the fact that the system first generates a security parameter λ . The algorithm of this module will use the security parameter λ as input to generate the dimension n , key distribution χ , and error parameter α applied to the LWE instance generation and encryption scheme. It sets the decomposition basis B' and degree D' for the tool vector $g' = (B'^{-1}, \dots, B'^{-d'})$. Finally, the system initialization will output LWE public parameters $pp^{LWE} = (n, \chi, \alpha, B', d')$. It is particularly important to note that a key s applied to the LWE encryption scheme is randomly sampled from the distribution χ . In the subsequent aggregation process, when using key transformation, the decomposition algorithm regarding the tool vector

$g' = (B^{-1}, \dots, B^{-d'})$ involves converting element $a \in \mathbb{T}$ into a d' -dimensional vector, with coefficients in $\mathbb{Z}_{B'}$, while minimizing $|a - \langle g'^{-1}(a), g' \rangle|$. Additionally, we assume that the entire system defaults to using pp^{LWE} as part of the input when running and invoking algorithms in the *LWE* encryption scheme.

$RLWE.Setup(1^\lambda)$ refers to the entire system running operation $RLWE.Setup(1^\lambda)$ to generate parameter $pp^{RLWE} = (N, \psi, \alpha, B, d, a)$. First, set the dimension N required in the RLWE encryption scheme, which is a power of 2. Then, set the key distribution ψ at R , and choose the error parameter α . Next, set the base integer $B \geq 2$ and the decomposition level of the gadget vector at $g = (B^{-1}, \dots, B^{-d})$. After that, generate a random vector $a \leftarrow U(\mathbb{T}^d)$. Finally, the system will output an *RLWE* public parameter $pp^{RLWE} = (N, \psi, \alpha, B, d, a)$. This *RLWE*-based encryption scheme operates under the Common Reference String (CRS) model. Since public parameter pp^{RLWE} includes a common reference string $a \leftarrow U(\mathbb{T}^d)$, parameters should be chosen appropriately to ensure that the *RLWE*-type encryption problem at parameter (N, ψ, α) can achieve a security level λ of at least bits. We also assume that the mentioned encryption operations used across all systems will default to including pp^{RLWE} as part of the input.

B. Key Generation

This process will call algorithm $LWE.KeyGen()$ to generate the private keys s_i required for i participants, and will use algorithms $RLWE.KeyGen()$, $RLWE.UniEnc()$, and $LWE.KSGen()$ to generate the public key PK_i , bootstrapping key BK_i , and key-switching key KS_i . It will then return the user's private key s_i and will publicly release a triplet (composed of the public key, bootstrapping key, and key-switching key) to all participants and entities.

The described $LWE.KeyGen()$ refers to randomly sampling a key $s_i \leftarrow \mathcal{X}$ from s_i as the user's private key.

The described $RLWE.KeyGen()$ refers to randomly sampling a key $z_i \leftarrow \psi$ from ψ and setting $z_i = (1, z_i)$. A noise vector $e \leftarrow D_\alpha^d$ is randomly sampled from D_α^d , and the public key is set to $b_i = -z_i \cdot a + e \pmod{1}$. It is important to note here that we represent $z_i^* = (z_{i,0}, -z_{i,N-1}, \dots, -z_{i,1}) \in \mathbb{Z}^N$ as $z_i = z_{i,0} + z_{i,1}X + \dots + z_{i,N-1}X^{N-1}$. Specifically, we run algorithm $RLWE.KeyGen$ to generate the *RLWE* key (z_i, b_i) and set the entire system's public key to $PK_i = b_i$.

The described $RLWE.UniEnc()$ refers to encrypting each user's private key z_i in the following way, returning ciphertext $(d_{i,j}, F_{i,j}) \in \mathbb{T}^d \times \mathbb{T}^{d \times 2}$. A single encryption process outputs a ciphertext composed of three polynomial vectors of dimension d , and its size is three-quarters of the range $\mathbb{T}^{2d \times 2}$ in a typical *RGSW* ciphertext. The first vector d , together with a CRS a , forms the encryption of $s_{i,j}$ under randomness r . We can consider F as a ciphertext generated by encrypting r with key z . The output is then set as the bootstrapping key BK_i , that is $BK_i = \{(d_{i,j}, F_{i,j})\}_{j \in [n]}$.

The described $LWE.KSGen()$ refers to publicly releasing the key-switching key $KS_i = \{K_{i,j}\}_{j \in [N]} \in (\mathbb{T}^{d' \times (n+1)})^N$ from $z_i^* \in \mathbb{Z}^n$ and $s_i \in \mathbb{Z}^n$. In the i participant, for each $j \in [N]$, the elements $A_{i,j}$ and $e_{i,j}$ in the J entry are generated by random sampling $A_{i,j} \leftarrow U(\mathbb{T}^{d' \times n})$ and $e_{i,j} \leftarrow D_\beta^{d'}$, returning $e_{i,j} \leftarrow D_\beta^{d'}$, where $b_{i,j} = -A_{i,j}s_i + e_{i,j} + z_{i,j}^* \cdot g' \pmod{1}$. Simply put, the key-switching key $KS \leftarrow LWE.KSGen(z_i^*, s_i)$ can be used to convert an *LWE* ciphertext corresponding to z_i^* into another *LWE* ciphertext under key s_i for the same message.

C. Data Processing and Encryption

The main operation in this process is to input data collected from Intelligent Connected Vehicles (ICVs), typically in floating-point format, often to four decimal places. However, considering real-world temperature data collection, this scheme retains only one decimal place. Since the entire scheme requires binary plaintext, floating-point numbers are first converted into manageable integer forms, then into binary.

Converting floating-point numbers to manageable integers, then to binary, involves taking the input user electricity consumption data (floating-point), rounding to one decimal place, and multiplying by 10 to create an integer form ten times larger. This integer is then converted to a binary number with 4 to 12 bits (depending on data size). Each binary digit of the data from each participant is sequentially returned and displayed.

This algorithm process encrypts each input bit $m \in \{0,1\}$ by calling $LWE.Enc()$, returning an *LWE* ciphertext $ct = (b, a) \in \mathbb{T}^{n+1}$. After performing this operation, the *ICV* sends the ciphertext, along with the associated participant's index, to the gateway (GW).

The described $LWE.Enc()$ refers to taking the processed i binary data bit $m_{i,p} \in \{0,1\}$ of the p participant as input, randomly sampling $a_i \leftarrow U(\mathbb{T}^n)$ and noise $e_i \leftarrow D_\alpha$, and

generating ciphertext $ct_i = (b_i, a_i) \in T^{n+1}$ (which satisfies condition $b_i = -\langle a_i, s_i \rangle + \frac{1}{4}m_{i,p} + e_i \pmod{1}$).

D. Homomorphic Operations

a specific bit position of two input addends, then adds the extended ciphertexts. Next, it initializes a homomorphic accumulator with PK_i and uses BK_i to perform recursive calculations. Finally, the key-switching key KS_i is used to transform the ciphertext into a form that can be decrypted with the joint key, outputting the result. The number of additions varies depending on the bit length, and the custom bit-width adder contains several algorithms to handle this.

To implement a multi-bit signed adder, you only need to cascade multiple one-bit signed adders, which can be achieved in code by running the algorithm in a loop multiple times. A one-bit signed adder consists of a half adder and a carry generator. A half adder is constructed using two XOR gates, while a carry generator is made up of three AND gates and two OR gates. As shown in Figure 2, the schematic diagram of the half adder used in homomorphic operations is illustrated. Figure 3 shows the schematic diagram of the carry generator used in homomorphic operations. In the figure, i only indicates the i bit of the binary representation of the operands involved in the operation.

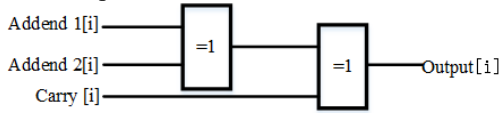


Figure 2. Schematic Diagram of the Half Adder

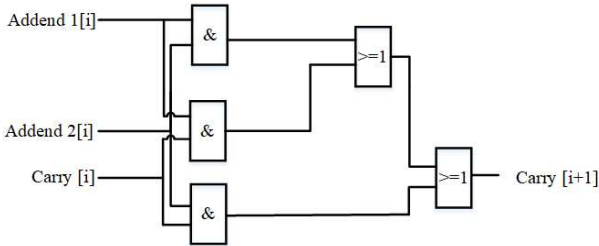


Figure 3. Schematic diagram of the carry generator

A one-bit signed adder adds the data from two participants, and each gate circuit involved in the bootstrapping algorithm will take as input the ciphertexts generated by the two participants, $\overline{ct}_1$ and $\overline{ct}_2$, a carry information cc , and a key triplet $\{(PK_i, BK_i, KS_i)\}_{i \in [k]}$, producing an output result.

Homomorphic addition data aggregation primarily includes the following steps:

- 1) Ciphertext Expansion: GW receives two ciphertexts, ciphertext $\overline{ct}_1 \in T^{k_1n+1}$ and ciphertext $\overline{ct}_2 \in T^{k_2n+1}$, sent by the connected vehicles (ICV) for the two participants (K_i denotes the distinction between the participants). The key server (KS) also sends the

corresponding key tuples $\{(PK_i, BK_i, KS_i)\}_{i \in [k]}$ for the participants to GW. GW expands ciphertexts $\overline{ct}_1$ and $\overline{ct}_2$ into $\overline{ct}'_1 \in T^{kn+1}$ and $\overline{ct}'_2 \in T^{kn+1}$.

- 2) GW calculates $\overline{ct}' = \left(\frac{1}{8} + 2 * \left(\overline{ct}'_1 + \overline{ct}'_2 \right) \right)$ based on the two expanded ciphertexts $\overline{ct}'_1$ and $\overline{ct}'_2$, which marks the step of homomorphic evaluation using an XOR gate. The final result, $\overline{ct}'$ is then transmitted to the public cloud employed by CNIC.
- 3) Homomorphic Accumulation: After the cloud

receives the data, it sets $\overline{ct}' = (b', a'_1, \dots, a'_k) \in T^{kn+1}$ and calculates $\tilde{b} = \lfloor 2N \cdot b' \rfloor$ and $\tilde{a}_i = \lfloor 2N \cdot a'_i \rfloor$. It initializes RLWE-type ciphertext

$$\tilde{c} = \left(-\frac{1}{8}h(X) \cdot X^{\tilde{b}}, 0 \right) \in T^{k+1}, \text{ where}$$

$$h = \sum_{-\frac{N}{2} < j < \frac{N}{2}} X^j = 1 + X + \dots + X^{\frac{N}{2}-1} - X^{\frac{N}{2}+1} - \dots - X^{N-1}$$

. It sets $\tilde{a}_i = (\tilde{a}_{i,j})_{j \in [n]}$ and executes

$$RLWE.CMux\left(\tilde{c}, X^{\tilde{a}_{i,j}} \cdot \tilde{c}, (d_{i,j}, F_{i,j}), \{b_l\}_{l \in [k]}\right)$$

for recursive calculations of $\tilde{c}$ for all $i \in [k]$ and $j \in [n]$.

The bootstrapping will output for the next step.

- 4) Key Conversion: The cloud employed by the Connected Vehicle Information Technology Company (CNIC) receives $\tilde{c} = (c_0, \dots, c_k) \in T^{k+1}$, setting b^* as the constant term of C_0 and a_i^* as the coefficient vector of C_i for $i \in [k]$, and computes LWE-type ciphertext $\overline{ct}^* = (b^*, a_1^*, \dots, a_k^*) \in T^{kN+1}$. It sets $\overline{KS} = \{KS_i\}_{i \in [k]}$ and executes

$$LWE.MKSwitch\left(\overline{ct}^*, \overline{KS}\right)$$

to calculate the final ciphertext $\overline{ct}''$, sending the final result to CNIC. This key conversion algorithm processes the input expanded ciphertext and the key conversion key to compute the ciphertext of the same message under joint key encryption.

E. Data Decryption and Output

This process will take the input ciphertext $\overline{ct} = (b, a_1, a_2, \dots, a_k) \in T^{kn+1}$ and a set of joint keys $(s_1, s_2, \dots, s_k)$, returning a set of bits $m \in \{0, 1\}$ that minimizes $\left| b + \sum_{i=1}^k \langle a_i, s_i \rangle - \frac{1}{4}m \right|$. Finally, the binary number will be converted to a decimal number to output the final result.

Subsequently, the correctness of the result will be verified, and the error rate and average time for the entire addition operation will be recorded. As shown in Figure 4, this is a schematic diagram of the key generation process for the multi-key data aggregation algorithm.

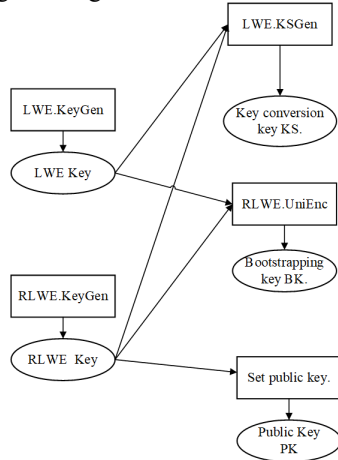


Figure 4. Schematic diagram of the carry generator

III.SIMULATION ANALYSIS

Figure 5 is a line graph showing the experimental results of the protocol disclosed in this invention using adders of different bit widths. In homomorphic operations, the accumulation step after constructing the accumulator is the most expensive part of the entire pipeline, while all other algorithms, including multi-key conversion, are asymptotically faster. As shown in Figure 5, when we execute this protocol with adders of different bit widths, we observe that as the total number of operations increases, the overall running time overhead also increases. However, this does not exhibit a linear growth trend; instead, the slope decreases as the total number of operations increases.

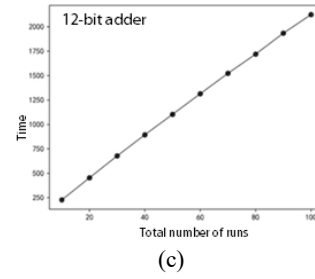
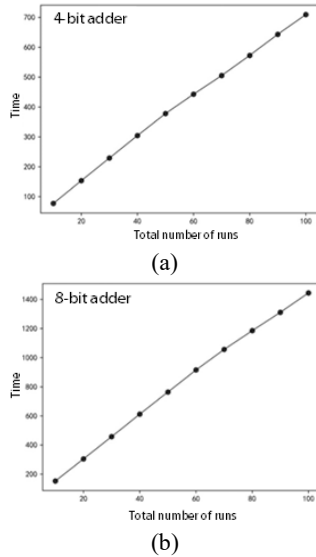


Figure 5. Simulation Results of Operations with Adders of Different Bit Widths. (a) 4-bit adder (b) 8-bit adder (c) 12-bit adder

When an adversary attempts to steal the key, the keys generated by the key conversion algorithm and the keys produced by the RLWE-related algorithms cannot be compromised under the hardness assumption of the LWE problem with parameters (n, χ, β) and the RLWE problem with parameters (N, ψ, α) . Thus, the adversary is unable to steal the key.

When an adversary attempts to launch a third-party forgery attack, the privacy protection of this method is based on the security of the multi-key scheme, and the adversary cannot break the encryption scheme presented in this paper within polynomial time. Therefore, the adversary cannot successfully carry out a third-party forgery attack.

When an adversary tries to extract data information from the encrypted data, the data is encrypted into ciphertexts using the private keys of all participants and then processed through expansion before being obtained through specific gate circuit homomorphic computations. Additionally, no single participant or entity possesses all the keys involved in the entire data aggregation scheme, and each computation provided by the gateways is executed independently during the homomorphic operation. As a result, the adversary cannot extract data information from the encrypted data.

When an adversary attempts to forge data in this method and pass the decryption verification, the key triplet (PK_i, BK_i, KS_i) required for the public cloud computing hired by KS to distribute to CNIC is related to the private keys distributed in advance to the participants. Therefore, the adversary cannot forge ciphertexts for homomorphic operations, let alone pass the decryption verification. Consequently, the adversary cannot forge data in this method and successfully pass the decryption authentication.

IV.CONCLUSION

This paper addresses the security risks associated with vehicle communication in the Internet of Vehicles (IoV) that current technologies fail to mitigate. It proposes a multi-key data aggregation algorithm specifically designed for connected vehicles, utilizing multi-key homomorphic encryption on a ring to ensure that only the participants assigned keys are involved in the data aggregation process throughout the system. During data transmission, all information is in ciphertext form, making it difficult for attackers to break the encryption and decryption methods to

forge ciphertext. Simulation analysis has verified that this approach effectively resolves the information transmission security issues of intelligent connected vehicles, thereby ensuring the security of vehicle users' information.

ACKNOWLEDGMENT

This research was supported in part by National Natural Science foundation of China (No.62203048、No.62073039、No.62472037、No.62402045).

Corresponding author: Xiang Gao.

REFERENCES

- [1] Cai J, Chen B, Wen J, et al. A joint vehicular device scheduling and uncertain resource management scheme for Federated Learning in Internet of Vehicles[J]. Information Sciences,2025,690121552-121552.
- [2] Wang H, Jiang W, Jiang Y, et al. LPF-IVN: A lightweight privacy-enhancing scheme with functional mechanism of intelligent vehicle networking[J]. Internet of Things,2024,28101400-101400.
- [3] Tian J, Shen Y, Wang Y. Blockchain-based fast handover authentication protocol for Internet of Vehicles in small industrial parks[J]. Vehicular Communications,2024,50100848-100848.
- [4] Almakayeel N. Deep learning-based improved transformer model on android malware detection and classification in internet of vehicles[J]. Scientific Reports,2024,14(1):25175-25175.
- [5] Kang W, Zhang L, Hu Z, et al. A secure and efficient data aggregation scheme for cloud-edge collaborative smart meters[J]. International Journal of Electrical Power and Energy Systems,2024,162110270-110270.
- [6] Cheng H. Recent advances of privacy-preserving machine learning based on (Fully) Homomorphic Encryption[J]. Security and Safety,2025,4.
- [7] Xiong J, Chen J, Lin J, et al. Enhancing privacy-preserving machine learning with self-learnable activation functions in fully homomorphic encryption[J]. Journal of Information Security and Applications, 2024,86103887-103887.
- [8] Ellen P, Chuna M, Stefania P. Central Yup'ik masks in the Vatican Museums; indigenous American heritage in European museums[J]. Museum Management and Curatorship,2023,38(6):620-642.
- [9] Zhao F, Fan S. Protecting Infinite Data Streams from Wearable Devices with Local Differential Privacy Techniques[J]. Information,2024,15(10):630-630.
- [10] Gao T, Fu H, Wang S, et al. AMOUE: Adaptive modified optimized unary encoding method for local differential privacy data preservation[J]. Computers and Electrical Engineering, 2024, 120(PB): 109791-109791.
- [11] Putrama M I, Martinek P. Enhancing protection in high-dimensional data: Distributed differential privacy with feature selection[J]. Information Processing and Management,2024,61(6):103870-103870.
- [12] Dan C, Yunbo T, Hao Z, et al. Incremental Factorization of Big Time Series Data with Blind Factor Approximation[J]. IEEE Transactions on Knowledge and Data Engineering,2019,33(2):1-1.
- [13] Xin X, He F, Qiu S, et al. Efficient multi-party quantum secret-sharing protocol[J]. Chinese Journal of Physics,2024,92664-674.



Yu Liu received the BS degree from National University of Defense Technology in June 2006 and the MS degree from Military Economy College of the CPCA in April 2009. He is currently working toward the Ph.D. degree with the School of Mechantronical Engineering, Beijing Institute of Technology. His research direction is Communication engineering and so on.



Ping Li received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 1995. After that, she joined the School of Mechatronical Engineering, Beijing Institute of Technology. Since 2002, she has been a Professor

with the School of Mechatronical Engineering, Beijing Institute of Technology. From 2007 to 2008, she visited the Pennsylvania State University as a Senior Visiting Scholar. Her research interests include Gaussian processes, backpropagation, jamming, radar computing, radar imaging, D2D communications, and LTE-U.



Lele Cui received the BS degree in Mechatronical Engineering from Beijing Institute of Technology in 2024, having graduated as an undergraduate student. He is now pursuing his MS degree in School of Mechatronical Engineering at the same institution. His research interests include wireless network simulation and wireless communication.



Zhiwen Ding received the BS degree and MS degree in Control Engineering from Shenyang Jianzhu University in 2006 and 2009, and now he is a Ph.D. candidate in School of Mechantronical Engineering, Beijing Institute of Technology. His research interests include UAV Automatic Control and Path Planning simulation and emulation, information security, wireless communication and so on.



Xiuming Peng received the BS degree in Mechatronical Engineering from Beijing Institute of Technology in 2024, and now he is a postgraduate in School of Mechantronical Engineering, Beijing Institute of Technology. His research interests include intelligent detection and control, information perception and confrontation.



Xiang Gao received the BS degree in Mechatronical Engineering from Beijing Institute of Technology in 2014, and the MS and Ph.D. degrees from Beijing Institute of Technology in 2016 and 2021. His research interests include network simulation and emulation, wireless communication and the next generation wireless systems such as MIMO, Cognitive radio and so on.