Experimental Evaluation of Post-Quantum Homomorphic Encryption for Privacy-Preserving V2X Communication

Abdullah Al Mamun, Ph.D.*
Glenn Department of Civil Engineering
Clemson University, Clemson, South Carolina, 29634
Email: abdullm@clemson.edu

Kyle Yates
School of Mathematical and Statistical Sciences
Clemson University, Clemson, South Carolina, 29634
Email: kjyates@clemson.edu

Antsa Rakotondrafara
School of Mathematical and Statistical Sciences
Clemson University, Clemson, South Carolina, 29634
Email: arakoto@clemson.edu

Mashrur Chowdhury, Ph.D.
Professor
Glenn Department of Civil Engineering
Clemson University, Clemson, South Carolina, 29634
Director
National Center for Transportation Cybersecurity and Resiliency (TraCR)
Email: mac@clemson.edu

Ryann Cartor, Ph.D.
Assistant Professor
School of Mathematical and Statistical Sciences
Clemson University, Clemson, South Carolina, 29634
Email: rcartor@clemson.edu

Shuhong Gao, Ph.D.
Professor
School of Mathematical and Statistical Sciences
Clemson University, Clemson, South Carolina, 29634
Email: sgao@clemson.edu

*Corresponding Author

ABSTRACT

Intelligent Transportation Systems (ITS) fundamentally rely on vehicle-generated data for applications such as congestion monitoring and route optimization, making the preservation of user privacy a critical challenge. Homomorphic Encryption (HE) offers a promising solution by enabling computation on encrypted data without revealing underlying content. This study presents the first real-world experimental evaluation of three post-quantum secure HE schemes, i.e., Brakerski-Fan-Vercauteren (BFV), Brakerski-Gentry-Vaikuntanathan (BGV), and Cheon-Kim-Kim-Song (CKKS), for vehicular communication scenarios. Two representative privacy-preserving use cases are considered: encrypted vehicle counting and average speed aggregation. Experiments are conducted over both Wi-Fi and Ethernet to assess performance under wireless and wired vehicle-to-everything (V2X) settings. Results show that BFV and BGV are suitable for latency-tolerant applications such as intersection monitoring and regional traffic analysis, with total end-to-end latencies under 10 seconds. While CKKS experiences higher overhead, it remains viable for periodic encrypted aggregation of numerical data. The experimental results demonstrate that HE can be feasibly deployed in ITS environments under 128-bit post-quantum security, provided that scheme-specific latency constraints are considered. This reinforces its potential to serve as a foundational tool for secure and privacy-preserving V2X data processing.

INTRODUCTION

Intelligent Transportation Systems (ITS) integrate advanced information and communication technologies into transportation infrastructure and vehicles to improve traffic efficiency, enhance road safety, and reduce congestion. These systems rely on continuous data exchange between vehicles, roadside units (RSUs), transportation infrastructure, and cloud services, enabling applications such as collision avoidance, queue monitoring, speed advisory systems, and route optimization (*1*). However, the wireless broadcast nature of vehicle-to-everything (V2X) communication poses serious privacy and security risks. Sensitive information such as vehicle locations, speed profiles, and trip histories may be intercepted by eavesdroppers or misused by untrusted service providers (*2*).

Traditional cryptographic solutions, such as AES (*3*), protect data at rest and in transit but fall short when processing is required. Even with modern standardized post-quantum cryptographic (PQC) schemes like the National Institute of Standards and Technology (NIST)-standardized module-lattice-based key encapsulation mechanisms and digital signatures (*4*, *5*), secure computation remains limited as ciphertexts must be decrypted before analysis, thereby exposing raw data to external systems. This leads to potential misuse, such as unauthorized driver profiling or location-based tracking by third-party cloud or edge services and eavesdropper.

Homomorphic Encryption (HE) addresses this gap by enabling computation directly on encrypted data (*6*). Using HE, vehicles can encrypt their data before transmission and RSUs or cloud servers in turn perform computations on the ciphertext without ever accessing the underlying plaintext. Only a trusted party holding the decryption key, typically the vehicle itself or a backend controller, can decrypt the final output. Thus, HE allows for secure, privacy-preserving data processing, making it a promising cryptographic primitive for privacy-critical ITS applications.

In this study, we experimentally evaluate the performance and feasibility of deploying post-quantum HE in real-world V2X communication environments. Specifically, we implement and test two representative ITS scenarios involving privacy-preserving data aggregation. In the first scenario (addition-only), vehicles transmit encrypted binary presence indicators to a RSU, enabling encrypted vehicle count computation for congestion warning. In the second scenario (addition + multiplication), vehicles send encrypted real-valued speed data to the RSU, which homomorphically computes the average speed without accessing any individual inputs. Both scenarios are executed across two communication media: Wi-Fi (wireless) and Ethernet (wired), to capture realistic transmission behavior, latency (delay) characteristics, and system performance under increasing vehicle counts.

Unlike previous ITS-focused HE studies that rely primarily on simulations or theoretical models (*7–11*), this work conducts end-to-end experimentation using actual hardware and real network links, enabling an assessment of how HE impacts computational latency, communication overhead, ciphertext size, and throughput under realistic network conditions. To this end, we benchmark three leading lattice-based HE schemes, i.e., Brakerski-Fan-Vercauteren (BFV) (*12*), Brakerski-Gentry-Vaikuntanathan (BGV) (*13*, *14*), and Cheon-Kim-Kim-Song (CKKS) (*15*), across both wireless and wired V2X communication channels. These schemes were selected because they are widely adopted, support both exact (BFV, BGV) and approximate (CKKS) arithmetic, and are based on the Ring Learning With Errors (RLWE) problem (*16*), which is believed to be secure against quantum adversaries. In contrast, earlier ITS applications commonly employed partially homomorphic encryption (PHE) schemes such as Paillier (*17–23*), which support only homomorphic addition between ciphertexts and scalar multiplication by a constant, but lack homomorphic multiplication between encrypted messages. Furthermore, their security relies on the integer factorization problem, making them vulnerable to quantum attacks (*24*). Our work thus presents the first simulation-free benchmarking of post-quantum secure HE schemes for ITS, combining real-world network transmission with representative ITS computations to evaluate practical feasibility in latency-sensitive vehicular environments.

BACKGROUND AND LITERATURE REVIEW
This section presents the foundations for the HE schemes used in our experiments and reviews prior work that applied HE to ITS domains. It also outlines the mathematical structure and parameter choices essential for understanding the implementation and security guarantees of the selected schemes.

Overview of Homomorphic Encryption Schemes
A concise, high-level overview of the BGV, BFV, and CKKS homomorphic encryption schemes is provided to establish the foundational concepts for this study. BGV and BFV support encrypted computation for exact arithmetic (e.g., finite field arithmetic), while CKKS supports arithmetic for floating-point numbers to some precision accuracy. A HE scheme consists of a collection of the following algorithms:

- $KeyGen$ generates the secret key sk, the public key pk, and an evaluation key ek.
- $Encrypt$ encrypts a message $\boldsymbol{m}$ as a ciphertext ct using the public key pk.
- $Decrypt$ decrypts a ciphertext ct to a message $\boldsymbol{m}$ using the secret key sk.
- Add performs an addition between two ciphertexts ct_1 and ct_2.
- $Multiply$ performs a multiplication between two ciphertexts ct_1 and ct_2 using the evaluation key ek.
- $Modswitch$ performs a scaling on a ciphertext to control encryption noise or bit expansion. This is typically performed immediately after a multiplication.
- In addition to the above algorithms, CKKS also includes $Encode$ and $Decode$ steps, which maps vectors of floating-point numbers to integer polynomials using a scaling factor Δ to preserve some precision accuracy.

The attractive feature of these HE schemes is that performing Add or $Multiply$ on ciphertexts results in the same value as performing the same operations on the original messages and then encrypting. For instance, if ct_1 is an encryption of message $\boldsymbol{m_1}$ and ct_2 is an encryption of message $\boldsymbol{m_2}$, homomorphically adding ct_1 and ct_2 results in an encryption of $\boldsymbol{m_1} + \boldsymbol{m_2}$ (or $\boldsymbol{m_1} \times \boldsymbol{m_2}$ for $Multiply$). Furthermore, these operations can be performed by anyone with access to the public key and ciphertexts without needing or leaking any private information. These operations directly on ciphertexts are called homomorphic operations.

We omit the finer mathematical details of the described schemes and refer the reader to (*12–15, 25–27*) for a more comprehensive view of HE foundations. However, we briefly present the mathematical structure of ciphertexts used in BGV, BFV, and CKKS. Encrypted messages in these schemes take the form of polynomial pairs $(\boldsymbol{a}, \boldsymbol{b})$ over the ring:

$$R_q = \mathbb{Z}_q[x]/(x^n + 1)$$

where n is a power of two and q is the ciphertext modulus. Let $\boldsymbol{s} \in R_q$ be a small secret key and $\boldsymbol{a}$ be a uniformly random polynomial in R_q. The ciphertext component $\boldsymbol{b}$ is computed differently depending on the scheme:

- BGV: $\boldsymbol{b} = -\boldsymbol{as} + \boldsymbol{m} + t\boldsymbol{e} \ \ mod(x^n + 1, q)$
- BFV: $\boldsymbol{b} = -\boldsymbol{as} + \lfloor q/t \rfloor \boldsymbol{m} + \boldsymbol{e} \ \ mod(x^n + 1, q)$
- CKKS: $\boldsymbol{b} = -\boldsymbol{as} + \boldsymbol{m} + \boldsymbol{e} \ \ mod(x^n + 1, q)$

Here, $\boldsymbol{m}$ is the plaintext message (or its encoded polynomial), $\boldsymbol{e}$ is a small noise polynomial with integer coefficients, and t is the plaintext modulus used in BFV and BGV. For CKKS, the plaintext $\boldsymbol{m}$ is typically a polynomial obtained from encoding a vector of real-valued messages, and the scheme tolerates approximate arithmetic. Choosing the parameters n, q, and t is important for determining the appropriate security level and ciphertext size, which are discussed in the following subsection.

Parameter Selection and Security
We use the OpenFHE implementation of BFV, BGV, and CKKS (*28*), following the 2024 security guidelines proposed in (*29*). Parameters are chosen to ensure 128-bit post-quantum security and

efficient real-time operation on ITS edge hardware. **Table 1** summarizes the configuration of each scheme used in our experiments. The "Levels" column indicates the multiplicative depth of the homomorphic arithmetic circuit: a level of 1 permits only homomorphic additions (no ciphertext-ciphertext multiplications), while a level of 2 supports one such multiplication.

TABLE 1 HE Parameters (128-bit Secure) Used in This Study (Selected Based on (*29*))

Scheme	n	$\log_2(q)$	t	$\log_2(\Delta)$	Levels (Multiplicative depth + 1)	Public Key Size (bytes)	Ciphertext Size (bytes)
BFV	4096	106	65537	N/A	1	131895	131939
BGV	8192	106	65537	N/A	1	656789	394573
CKKS	16384	106	N/A	38	1	1312151	787791
CKKS	16384	106	N/A	38	2	1574473	1050129

Note- N/A: Not Applicable

Current use of Homomorphic Encryption in ITS

As introduced earlier, previous ITS-focused HE studies primarily used PHE such as Paillier and relied on simulations or theoretical models. While these efforts demonstrated the conceptual feasibility of privacy-preserving computations, they lacked real-world validation and relied on cryptosystems vulnerable to quantum attacks.

In contrast, somewhat homomorphic encryption (SHE) and fully homomorphic encryption (FHE) schemes, such as BGV, BFV, and CKKS, offer broader functionality. SHE supports a limited number of additions and/or multiplications on ciphertexts, while FHE supports an unlimited number of operations through bootstrapping (*12*, *30*). These schemes are based on hard lattice problems, making them resistant to quantum attacks. All practical SHE/FHE schemes to date are considered post-quantum secure (*30*). A growing body of research (*8–11*, *31–36*) has investigated SHE and FHE for ITS, including implementations using HE libraries like SEAL (*31–33*), Pyfhel (*8*, *34*), Helib (*32*), and PALISADE (*10*). Other studies leverage the TFHE scheme (*37*) for ITS (*11*), propose novel constructions (*9*), or survey HE-based methods (*36*). Related approaches, such as secure multi-party computation and federated learning, have also been explored for ITS applications (*38–40*).

As discussed in the Introduction section, our study is the first to experimentally evaluate SHE for ITS in a real communication environment, rather than through simulation or theoretical design. It is also the first to conduct such experiments using the OpenFHE library.

EXPERIMENTAL SETUP

This section details the experimental framework for benchmarking post-quantum secure HE schemes in V2X communication. We evaluated two privacy-preserving data aggregation scenarios, including encrypted summation and averaging, over real wireless (Wi-Fi) and wired (Ethernet) links. Experiments were conducted in a controlled lab setting using real hardware and OpenFHE implementations of the three described lattice-based HE schemes. In the following subsections, we demonstrate how HE can be leveraged in practical ITS use cases across both wireless and wired communications.

Rationale for Applying HE in ITS Scenarios

The growing integration of connected and autonomous vehicles with ITS infrastructure necessitates secure and privacy-preserving data sharing mechanisms. HE schemes offer a privacy-preserving approach by enabling computation on encrypted data, thereby preventing leakage of sensitive information such as location, presence, speed, and personal information even while data is being processed.

Wireless Communication

Wireless vehicle-to-infrastructure (V2I) communication scenarios, such as congestion detection, cooperative maneuvering, or signal advisory systems, often require vehicles to transmit data that reflects their spatial or behavioral state. As depicted in **Figure 1**, HE ensures that RSUs can compute congestion levels or queue lengths without accessing any vehicle's raw input, preserving anonymity. The applicability and advantages of HE for privacy preservation can be clearly demonstrated through the following real-world scenarios:
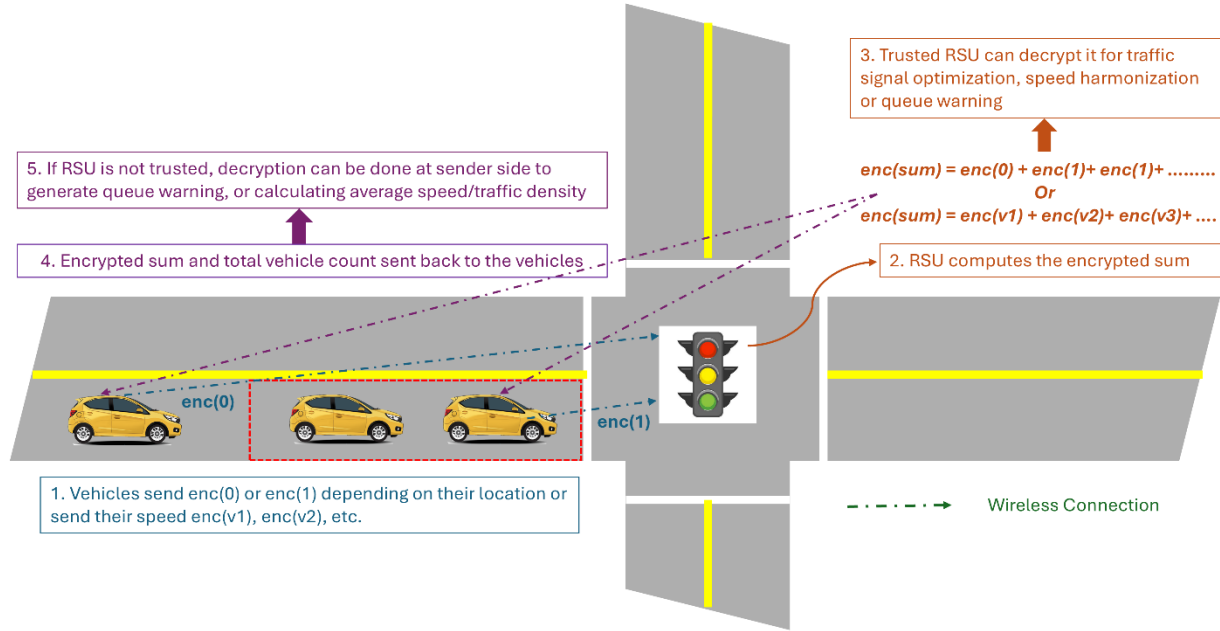

Figure 1 Privacy-preserving wireless vehicle-to-infrastructure (V2I) communication using homomorphic encryption

Example 1: Intersection Congestion Monitoring

In urban environments, RSUs deployed at intersections often assess congestion levels to trigger timely warnings. Vehicles transmit binary indicators - "1" if they are within a congestion detection zone, "0" otherwise. When unencrypted, these signals can be exploited to infer individual movement patterns. With HE, each vehicle encrypts its status, allowing the RSU to aggregate encrypted values and determine whether congestion thresholds are met without accessing individual data. The encrypted result then can be sent to an authorized controller or returned to vehicles for decryption, enabling privacy-preserving congestion warnings.

Example 2: Cooperative Speed Advisory Systems

Many V2I systems deliver Green Light Optimal Speed Advisory (GLOSA) messages by collecting speed data from approaching vehicles (*41*). Sharing raw speeds can enable long-term driver profiling. With HE, vehicles encrypt their speeds before transmission. The RSU computes an encrypted average without learning individual values, and only authorized parties with the decryption key can access the result, preserving driver privacy.

These examples underscore the importance of HE in enabling secure, privacy-preserving V2I data exchanges. Beyond congestion monitoring and speed harmonization, HE can be applied to other wireless V2X scenarios such as harsh braking detection and eco-driving assistance.

Wired Communication

While wireless communication supports direct V2I data exchange, backend communication between RSUs and centralized systems like Traffic Management Centers (TMCs) typically occurs over secure wired links (e.g., Ethernet). These connections aggregate local RSU data, coordinate infrastructure across regions, and disseminate control decisions. Despite their reliability, wired links remain vulnerable to insider threats and protocol-level leaks. Exposed backend data can reveal sensitive regional traffic patterns or fleet behaviors.

As shown in **Figure 2**, HE allows RSUs to compute over encrypted inputs from vehicles and forward encrypted aggregations, such as total vehicle count or average speed, to the TMC. In our setup, RSUs perform aggregation and act as senders, while the TMC simulates receiving encrypted data from multiple RSUs. Though decryption is performed only at the RSU in our experiment, the framework also supports TMC-side decryption or further encrypted processing, preserving privacy while enabling centralized coordination. We next describe two wired communication scenarios that align with our experimental design and highlight real-world applicability.
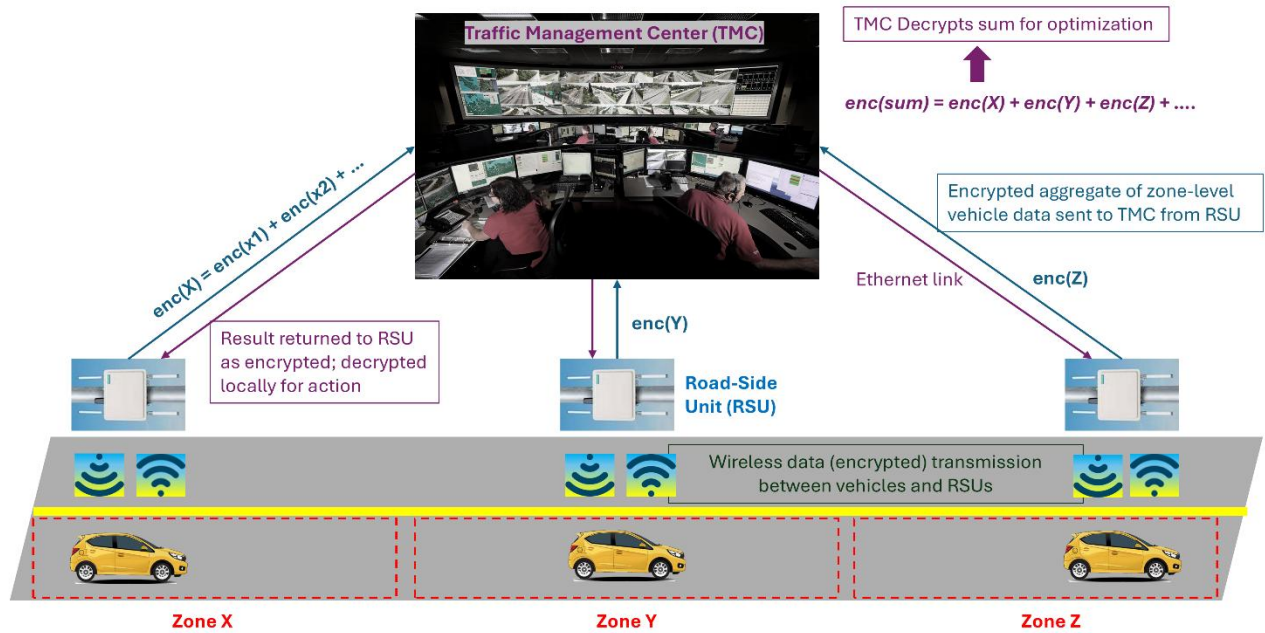

Figure 2 Privacy-preserving wired infrastructure-to-infrastructure (I2I) communication using homomorphic encryption

Example 1: City-Wide Congestion Heatmap Generation

In urban areas, RSUs collect binary presence indicators (enc(0)/enc(1)) from vehicles to assess congestion. Without encryption, such data can expose vehicle trajectories. With HE, vehicles send encrypted indicators, which RSUs aggregate and transmit to the TMC over Ethernet. The TMC decrypts aggregated values from multiple RSUs to generate a city-wide congestion heatmap. In our experiment, RSUs simulate this process by performing both aggregation and decryption, demonstrating feasibility for edge-level awareness and future TMC-level coordination.

Example 2: Regional Speed Pattern Analysis for Adaptive Traffic Control

In dynamic traffic management, RSUs may compute average travel speeds across road segments and relay this information upstream for re-routing decisions. Using HE, vehicles can send encrypted speed values to the RSU. The RSU performs homomorphic addition and then computes the average speed, all in ciphertext space. In our implementation, the RSU acts as both aggregator and decryptor and

may use the results to issue localized advisories. Alternatively, in a system-level design, the RSU could forward the encrypted result to the TMC for centralized decryption and broader coordination.

These wired setups highlight the flexibility of privacy-preserving ITS architectures. RSUs or edge nodes can either decrypt locally or securely forward encrypted data to TMCs for centralized processing. In both cases, HE ensures that sensitive vehicular data remains protected throughout the communication chain.

Experimental Workflow and Processing Steps

To evaluate the applicability of HE in real-world ITS applications, our experimental framework encompasses two representative scenarios. The first scenario focuses exclusively on addition operations, simulating use cases such as congestion counting and encrypted speed aggregation. This scenario is executed using all three HE schemes, BFV, BGV, and CKKS, to benchmark their performance in lightweight addition-only workloads. It is to be noted that, in this scenario, the vehicle (sender) can decrypt the final sum and perform plaintext post-processing to compute averages (i.e., dividing by the number of additions + 1) or threshold-based decisions, enabling practical computation without leaking individual inputs. The second scenario incorporates both addition and multiplication, using only the CKKS scheme, which supports approximate arithmetic. In this case, the averaging is performed homomorphically at the receiver side, without requiring decryption. BFV and BGV were excluded from this scenario due to their lack of native support for efficient fixed-point multiplication and ciphertext rescaling, both of which are necessary for averaging operations in encrypted space. Both scenarios were evaluated under two network conditions, Wi-Fi and Ethernet, to compare the communication and computation performance of HE under wireless and wired infrastructure settings. The experiments were implemented using the OpenFHE C++ library, which provides modular APIs for configuring scheme-specific parameters, encryption pipelines, and homomorphic operations. The data processing workflow is presented in **Table 2**.

TABLE 2 Experiment Workflow

Step	Sender	Receiver
1. Context and Key Generation	Generate the OpenFHE context and a public/private key pair using selected parameters for BFV, BGV, or CKKS. Share the public key with the receiver while securely storing the secret key.	Receive the context and public key to enable encryption and homomorphic operations.
2. Message Preparation and Encryption	Prepare plaintext messages (e.g., binary 0/1 or speed values). Encrypt data using the public key and encode via OpenFHE APIs.	Encrypt simulated data for multiple vehicles (in RSUs) using the same public key.
3. Serialization	Serialize encrypted ciphertexts into binary format using OpenFHE's serialization tools.	-
4. Fragmentation	Fragment serialized ciphertexts into ≤1400 byte UDP packets with prepended sequence numbers, ensuring compatibility with Wi-Fi/Ethernet MTUs.	-
5. Transmission over Network	Transmit encrypted UDP packet fragments over the selected medium (Wi-Fi for wireless or Ethernet for wired).	Receive and buffer incoming UDP fragments for reassembly.

Step	Sender	Receiver
6. Reassembly and Deserialization	-	Reassemble fragments using sequence numbers and deserialize to reconstruct the original ciphertext.
7. Homomorphic Computation	-	Apply homomorphic addition for both scenarios. For the average-speed scenario, use CKKS to perform multiplication for approximate division (i.e., average speed).
8. Serialization and Fragmentation	-	Serialize and fragment the homomorphic computation results (ciphertext).
9. Return Transmission	Receive result (ciphertext) fragments.	Transmit encrypted result fragments (ciphertext) back to the sender over Wi-Fi or Ethernet.
10. Decryption and Output	Reassemble and deserialize received ciphertext and decrypt using private key.	-
11. Final Output and Metrics	Perform any final plaintext-side computations if needed and log performance metrics.	Log performance metrics.

Experimental Parameters and System Configuration

Beyond the cryptographic parameters detailed in **Table 1**, this section outlines the practical system-level and runtime configurations adopted to ensure stable, low-latency operation of HE-based communication in both wireless and wired network settings. **Table 3** presents the system-level parameters used in our experimental setup.

TABLE 3 Experimental Parameters and System Configuration

Category	Parameter	Value/Setting
Fragmentation	Maximum Transmission Unit (MTU)	1400 bytes
Transmission delay (delay between successive packet transmissions)	Wi-Fi delay	Approximately 1.2*RTT without any packet fragment loss (upto ~4,500 ms for BFV; ~13,000 ms for BGV; and ~30,000 ms for CKKS)
	Ethernet delay	100 ms
Networking	Socket buffer size	4 MB (sender and receiver)
	Receive timeout (between fragments)	5 s
Simulated Vehicles/Speeds	Number of vehicles	50, 100, 200
	Speed range	40 – 60 MPH (randomized)
Execution	Threading model	Single-threaded (sequential)
Number of trials (samples) for each scenario	BFV	20
	BGV	20
	CKKS	5
Sender Device	Processor	13th Gen Intel Core i9-13900HX @ 2.2 GHz (base)
	CPU cores	24 cores (8 performance + 16 efficiency)
	RAM	32 GB
	Wi-Fi adapter	Intel Wi-Fi 6 AX201 160MHz

Category	Parameter	Value/Setting
	Wi-Fi protocol	802.11n
	Ethernet adapter	Killer E3100G 2.5GbE
Receiver Device	Processor	Intel Core i7-5500U @ 2.40GHz (base)
	CPU cores	2 cores / 4 threads
	RAM	8 GB
	Wi-Fi adapter	Intel Dual Band Wireless-AC 7265
	Wi-Fi protocol	802.11n
	Ethernet adapter	Cisco AnyConnect Virtual Adapter
Network Environment	Wi-Fi transmission/receiving speed	300 Mbps
	Ethernet transmission/receiving speed	1 Gbps

The experimental parameters summarized in **Table 3** reflect practical constraints of ITS deployments and technical characteristics of V2X communication environments. A 1400-byte packet fragment size was chosen to remain below typical MTU limits (e.g., 1500 bytes for Ethernet and Wi-Fi) (*42*), minimizing IP-layer fragmentation and reducing the likelihood of packet loss or reordering at the IP level. Although tests were conducted over standard Wi-Fi and Ethernet, the setup emulates critical aspects of cellular V2X (C-V2X) communication as defined by 3GPP LTE-V2X (Rel. 14/15) and 5G NR V2X (Rel. 16+) standards (*43*, *44*). These protocols support direct vehicle communication via sidelink (PC5) without built-in link-layer encryption or fragmentation handling, underscoring the need for size-constrained, delay-sensitive payloads (*45*).

While C-V2X peer-to-peer (P2P) was not directly tested, the connectionless UDP transmission over Wi-Fi serves as a suitable proxy for analyzing jitters, packet pacing, and reassembly reliability in similar vehicular wireless environments. To evaluate scalability, our experiments simulated 50, 100, and 200 vehicle inputs. For addition-only scenarios, encrypted binary presence values (enc(0)/enc(1)) were used and for the addition-plus-multiplication scenario, encrypted speeds between 40 - 60 mph were used to test the feasibility of CKKS-based homomorphic averaging.

Transmission delays, set to approximately 1.2 times of the observed round-trip time (RTT) in Wi-Fi) and 100 ms in Ethernet, were empirically chosen to avoid packet drops and ensure uninterrupted benchmarking under high message volumes. These delays were critical for controlled testing, as insufficient pacing in wireless transmission led to socket buffer overflows and halted execution. A receive timeout of 5 seconds (s) was configured to prevent indefinite blocking and to simulate realistic responsiveness requirements commonly expected in time-sensitive ITS applications. Finally, all operations were executed single-threadedly to reflect realistic computational limitations of edge vehicular devices. Each experiment was repeated to capture variability, with 20 trials conducted for BFV and BGV, and 5 trials for CKKS due to its higher latency and larger ciphertext size.

Performance Metrics

Performance was evaluated using several key metrics relevant to V2X deployments, including end-to-end computational latency, communication latency (= RTT – computational latency), jitter, number of UDP fragments, and times for fragmentation, reassembly, encryption, and decryption. These measurements enabled a comprehensive assessment of the feasibility of deploying HE schemes in real-time, near-real-time, and offline ITS applications.

RESULTS AND DISCUSSION

The experimental analysis reveals a fundamental constraint for privacy-preserving V2X systems: communication latency dominates performance bottlenecks across all tested HE schemes. Although the chosen parameters provide 128-bit post-quantum security with relatively compact ciphertexts (**Table 1**), the substantial increase in data size from encryption introduces significant transmission overhead, outweighing the computational differences across schemes. These challenges are particularly intensified

in vehicular settings, where real-time responsiveness is critical but constrained by network bandwidth, packet fragmentation, and jitter. To systematically evaluate these limitations and trade-offs, we present the results in three stages: (i) computation latency analysis (**Table 4**), (ii) network-level communication characteristics (**Table 5**), and visual insights (**Figures 3 and 4**) comparing communication delay and homomorphic operation time across HE schemes.

Computation Latency

        Table 4 summarizes the total end-to-end computational latency across three HE schemes, BFV, BGV, and CKKS, under Wi-Fi and Ethernet transmission scenarios. Each row represents a distinct combination of vehicle count, HE scheme, and communication medium, with measured latencies for encryption, homomorphic operations, decryption, and overall processing time.

TABLE 4 End-to-End Computational Latency of BFV, BGV, and CKKS Schemes Across Varying Vehicle Counts and Communication Media

Scheme	Vehicle count	Medium	Encryption time at sender (ms)	Time to perform homomorphic operations at receiver (ms)	Decryption time at sender (ms)	Mean computational end-to-end latency (ms)
BFV (addition only)	50 (49 additions)	Wi-Fi	6.40 ± 2.04	189.72 ± 14.87	4.60 ± 0.88	200.72
		Ethernet	5.84 ± 2.12	183.56 ± 4.19	4.36 ± 0.85	193.76
	100 (99 additions)	Wi-Fi	5.44 ± 2.14	347.61 ± 6.80	4.73 ± 1.08	357.78
		Ethernet	5.99 ± 2.14	343.00 ± 7.57	4.78 ± 0.96	353.77
	200 (199 additions)	Wi-Fi	6.54 ± 1.96	680.98 ± 37.33	4.94 ± 0.87	692.46
		Ethernet	6.40 ± 1.34	675.37 ± 5.95	4.97 ± 0.63	686.73
BGV (addition only)	50	Wi-Fi	10.19 ± 2.24	346.93 ± 12.24	8.16 ± 2.29	365.27
		Ethernet	9.70 ± 2.07	344.17 ± 14.31	7.36 ± 1.26	361.23
	100	Wi-Fi	7.42 ± 2.64	660.94 ± 13.12	5.87 ± 1.93	674.22
		Ethernet	9.19 ± 2.74	657.11 ± 18.58	7.38 ± 1.91	673.67
	200	Wi-Fi	10.04 ± 1.35	1316.11 ± 101.74	7.56 ± 1.10	1333.71
		Ethernet	9.63 ± 3.65	1290.67 ± 21.73	8.39 ± 3.74	1308.68
CKKS (addition only)	50	Wi-Fi	15.84 ± 0.50	685.10 ± 13.69	15.45 ± 2.70	716.39
		Ethernet	12.84 ± 4.60	684.06 ± 18.85	12.93 ± 3.84	709.83
	100	Wi-Fi	14.51 ± 4.69	1406.52 ± 19.63	14.47 ± 2.44	1435.50
		Ethernet	14.77 ± 7.04	1338.38 ± 20.42	22.30 ± 15.19	1375.45
	200	Wi-Fi	16.55 ± 2.30	2813.11 ± 122.54	28.88 ± 10.43	2858.54
		Ethernet	13.17 ± 2.74	2656.35 ± 22.52	18.91 ± 6.00	2688.43
CKKS (addition and multiplic-ation)	50 (49 additions + 1 multiplication)	Wi-Fi	16.15 ± 5.10	799.89 ± 13.60	24.44 ± 6.15	840.48
		Ethernet	12.09 ± 3.31	796.48 ± 22.05	15.97 ± 0.72	824.54
	100	Wi-Fi	14.75 ± 1.75	1555.08 ± 52.37	15.10 ± 0.81	1584.93
		Ethernet	14.34 ± 1.53	1580.64 ± 56.08	15.30 ± 2.02	1610.28
	200	Wi-Fi	20.17 ± 4.87	3210.93 ± 327.89	20.94 ± 6.15	3252.04
		Ethernet	14.44 ± 2.87	3069.82 ± 14.41	16.47 ± 1.93	3100.72

Note: 6.40 ± 2.04 (mean ± standard deviation)

        The results in **Table 4** indicate that while encryption and decryption times are relatively modest across all schemes (ranging ~5 to 25 milliseconds [ms]), the dominant cost lies in homomorphic operation times, which grow approximately linearly with the number of vehicles for both BFV and BGV. For instance, in the BFV scheme (Wi-Fi), increasing the vehicle count from 50 to 200 leads to a ~3.5 times increase in computation latency (189.72 ms to 680.98 ms), underscoring the linear aggregation cost of addition-based operations.

CKKS exhibits the highest computational overhead across all configurations. Even without multiplicative operations, CKKS latency grows steeply, from ~685 ms at 50 vehicles to over 2800 ms at 200 vehicles over Wi-Fi. When a single multiplication is added (e.g., for homomorphic averaging), the overhead further escalates, reaching 3210.93 ms on Wi-Fi and 3069.82 ms on Ethernet at 200 vehicles. This trend reflects CKKS's cost of rescaling and approximate arithmetic, confirming that multiplicative depth significantly increases processing complexity. Additionally, part of this increased latency can be attributed to the larger parameter size used in CKKS ($n = 16384$), which, while enabling higher precision and security, encounters more computational overhead than the smaller ring sizes used for BGV ($n = 8192$) and BFV ($n = 4096$).

While Wi-Fi and Ethernet configurations demonstrate nearly identical latency values across all trials, these values strictly reflect computation time and exclude network transmission and fragmentation delays, which are discussed next.

Communication Latency and Jitter

Table 5 highlights the communication-level latency, round-trip time (RTT), jitter, number of fragments, and associated processing delays (fragmentation and reassembly) for each HE schemes and configurations. These metrics provide insight into how encryption-induced message expansion affects the transmission performance of privacy-preserving V2X systems across different network environments.

TABLE 5 Communication-Level Latency, Fragmentation, and Jitter Characteristics of HE Schemes over Wi-Fi and Ethernet

Scheme	Vehicle count	Medium	Mean RTT (ms)	Jitter (ms)	Mean communication latency (ms)	Total number of fragments	Fragmentation time at both sender and receiver (ms)	Reassembly time at both sender and receiver (ms)
Baseline (without HE: Message sent: 4 bytes; received: 6 bytes)	200	Wi-Fi	4.14	3.95	-	1	-	-
		Ethernet	3.81	1.12	-	1	-	-
BFV (addition only)	50	Wi-Fi	3289.62	66.40	3088.90	95	0.07 ± 0.02	0.46 ± 0.35
		Ethernet	3191.32	31.99	2997.56		0.06 ± 0.02	0.43 ± 0.35
	100	Wi-Fi	3430.66	13.10	3072.89		0.06 ± 0.02	0.44 ± 0.38
		Ethernet	3420.37	19.57	3066.60		0.07 ± 0.02	0.49 ± 0.40
	200	Wi-Fi	3769.45	43.13	3076.99		0.07 ± 0.02	0.50 ± 0.37
		Ethernet	3767.48	25.02	3080.75		0.07 ± 0.02	0.40 ± 0.27
BGV (addition only)	50	Wi-Fi	9590.36	220.96	9225.09	284	0.24 ± 0.09	1.20 ± 0.86
		Ethernet	9642.65	25.19	9281.42		0.20 ± 0.07	1.15 ± 0.84
	100	Wi-Fi	10003.95	262.85	9329.74		0.20 ± 0.06	1.15 ± 0.93
		Ethernet	9924.82	164.72	9251.15		0.21 ± 0.08	1.17 ± 0.82
	200	Wi-Fi	10618.15	131.54	9284.44		0.22 ± 0.06	1.24 ± 0.92
		Ethernet	10549.65	166.65	9240.97		0.22 ± 0.10	1.12 ± 0.76
CKKS (addition only)	50	Wi-Fi	19269.65	110.50	18553.26	566	0.43 ± 0.14	3.03 ± 2.72
		Ethernet	19176.56	84.87	18466.73		0.41 ± 0.12	2.73 ± 2.74
	100	Wi-Fi	19985.50	55.49	18550.00		0.46 ± 0.20	3.00 ± 2.72
		Ethernet	19764.37	130.11	18388.91		0.57 ± 0.38	2.44 ± 2.21
	200	Wi-Fi	21569.25	687.71	18710.71		0.50 ± 0.16	3.28 ± 2.96
		Ethernet	21179.45	119.41	18491.02		0.43 ± 0.13	2.83 ± 2.49

Scheme	Vehicle count	Medium	Mean RTT (ms)	Jitter (ms)	Mean communication latency (ms)	Total number of fragments	Fragmentation time at both sender and receiver (ms)	Reassembly time at both sender and receiver (ms)
CKKS (addition and multiplication)	50	Wi-Fi	22253.20	59.90	21412.71	754	0.57 ± 0.20	4.38 ± 5.99
		Ethernet	22368.36	17.70	21543.82		0.40 ± 0.10	4.90 ± 6.05
	100	Wi-Fi	23122.92	85.62	21537.98		0.49 ± 0.21	4.75 ± 6.12
		Ethernet	23171.26	73.18	21560.98		0.44 ± 0.09	5.78 ± 8.46
	200	Wi-Fi	24972.31	477.79	21720.27		0.51 ± 0.09	4.97 ± 6.92
		Ethernet	24630.92	32.56	21530.20		0.39 ± 0.12	4.79 ± 6.47

Note: 0.07 ± 0.02 (mean ± standard deviation)

The results reveal several clear patterns. Baseline communication, which involves unencrypted messages as small as 4-6 bytes, achieves RTTs of just 3.8-4.1 ms across both Wi-Fi and Ethernet, with zero fragmentation overhead. This reflects the negligible footprint and near-instantaneous delivery of plaintext messages under normal conditions.

In contrast, HE schemes suffer from extreme fragmentation due to ciphertext expansion. For instance, BFV with ~132 KB ciphertext generates 95 fragments for a modest load of 50 vehicles, while CKKS, when configured for both addition and multiplication, requires 754 fragments to facilitate a ciphertext of ~1 MB. This level of fragmentation increases RTT values beyond 24,000 ms (24 s) and results in total communication latencies exceeding 21,700 ms (21.7 s) for some configurations. These values underscore the critical cost of ciphertext size in real-time V2X systems.

Notably, despite the fundamental differences between Wi-Fi and Ethernet, such as Wi-Fi's shared, interference-prone wireless spectrum versus Ethernet's dedicated, full-duplex wired channel, and their respective disparities in bandwidth, jitter, and error rate, the total communication latency across schemes remains nearly identical in both mediums. This counterintuitive outcome can be attributed to the fact that fragmentation, serialization, UDP socket buffering, and inter-packet pacing dominate the delay pipeline, thereby neutralizing the practical benefits of Ethernet's higher bandwidth.

However, one key difference lies in jitter behavior. Wi-Fi demonstrates significantly higher variability in packet delivery timing, with jitter reaching as high as 687.71 ms in the CKKS scenario with 200 vehicles. By contrast, Ethernet consistently maintains jitters below 170 ms across all tested configurations. This disparity in stability has direct implications for packet pacing strategies. In the Wi-Fi setup, packet loss and program halts were observed unless packets were sent with a delay of at least 1.2 times the RTT, leading to inter-packet (inter-trial) spacing as long as 30 seconds for high-load CKKS tests. On the other hand, Ethernet's deterministic environment enabled fixed 100-ms inter-packet delays without any packet loss, supporting continuous data transmission under high volume conditions. Therefore, from an ITS deployment perspective, Ethernet supports continuous encrypted updates every 100 ms, making it well-suited for real-time backend analytics, adaptive signal control, and fleet routing. In contrast, Wi-Fi, constrained by inter-packet pacing and jitter, supports intermittent data updates (e.g., every 20-30 s), better aligned with non-safety-critical applications such as eco-driving feedback or queue estimation.

Latency and Overhead Analysis Across HE Schemes

Figure 3 compares the total communication latency (Wi-Fi) across HE schemes. The bar heights correlate strongly with ciphertext size and fragment count. BFV exhibits the lowest latency (~3 s), followed by BGV (~9.2 s), while CKKS (with multiplication) peaks at over 21 s. This reflects the direct relationship between ciphertext size and transmission delay. Notably, these ciphertext sizes are a function of the polynomial ring (n) used in each scheme, i.e., 4096 for BFV, 8192 for BGV, and 16384 for CKKS, which significantly impacts serialization size and the resulting number of fragments. This figure reinforces a key insight- communication latency, driven by fragmentation, is the dominant bottleneck in

post-quantum HE-based V2X systems. Even relatively small ciphertexts like BFV (132 KB) generate ~100 fragments, stressing Wi-Fi under tight pacing intervals.
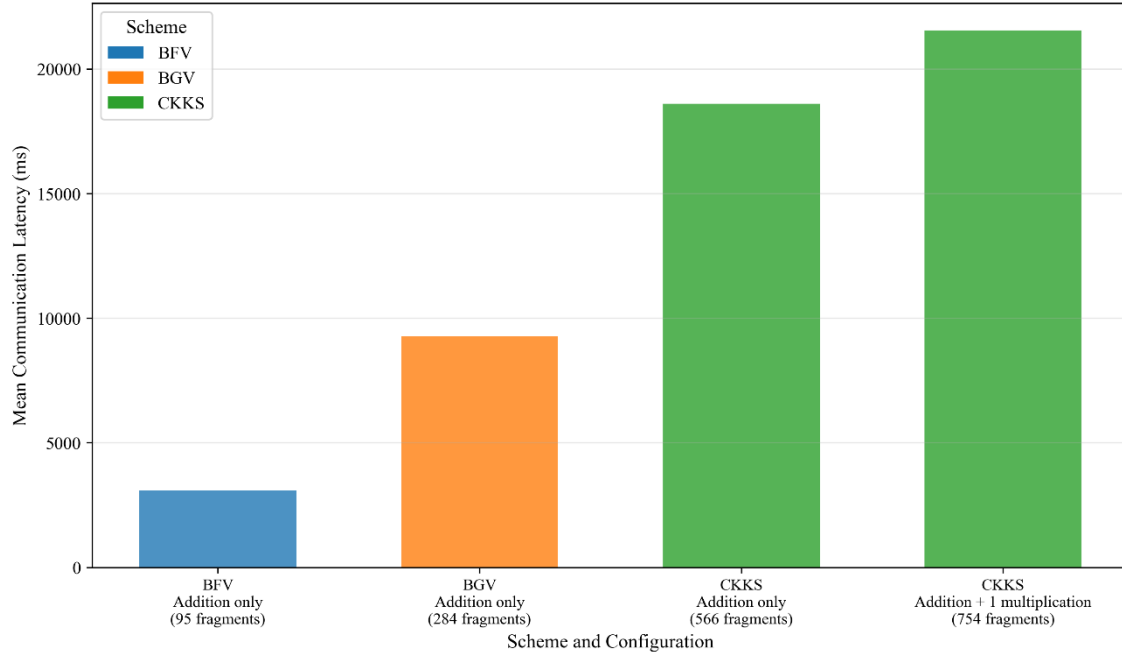

Figure 3 Mean communication latency by HE schemes and parameters (Wi-Fi)

Figure 4 presents the mean HE computation times at the receiver across different workloads in Wi-Fi scenario. As the number of additions increases, computation time gradually rises for all schemes. When a single multiplication is added to the CKKS workload, the total processing time increases further, e.g., from approximately 2813 ms to 3211 ms for 200 vehicles, highlighting the additional computational cost associated with multiplicative operations in CKKS.

Figure 4 Mean homomorphic computation time by number of additions and multiplicative depth (Wi-Fi)

Feasibility and Suitability of Post-Quantum HE Schemes for ITS Use Cases

The combined experimental results suggest that HE can support a range of privacy-preserving ITS applications, if use case latency requirements align with observed performance. For congestion monitoring, the BFV scheme, with a total latency of approximately 3 seconds, can accommodate sub-5-second updates, which is sufficient for most intersection management systems. Its low encryption and decryption overhead also minimize computational demands on edge devices, making it practical for deployment on in-vehicle or RSUs.

BGV demonstrates approximately 9 seconds of latency under workloads representative of regional congestion analysis, making it suitable for generating traffic heatmaps or aggregated flow summaries on 10-30 second cycles. This timing allows for a balance between preserving user privacy and delivering timely system-wide insights. In contrast, the CKKS scheme, which suffers 21 to 32 seconds of end-to-end latency depending on vehicle count and operation type, is less suitable for near-real-time applications such as GLOSA. While it is theoretically possible to reduce this latency by selecting smaller parameters to reduce ciphertext size, such adjustments would compromise the 128-bit post-quantum security target. Therefore, CKKS remains largely impractical for latency-critical environments but is still feasible for less time-sensitive use cases such as minute-scale eco-driving feedback or historical average speed analytics, where encrypted aggregation of floating-point values can enhance both privacy and insight.

However, it is important to note that none of the tested HE schemes, under current parameterizations of at least 128-bit security and network constraints, meet the stringent end-to-end latency requirements of real-time, safety-critical applications, such as forward collision warning or emergency braking systems, which typically demand round-trip latencies below 100 ms (*46*). The experimental latency values observed, ranging from 3 to over 30 seconds depending on scheme and workload, fall well outside this threshold. As such, the direct use of HE in latency-critical vehicular safety systems remains impractical without significant advances in ciphertext compression, high-throughput networking, multi-threaded data transmission, and specialized hardware acceleration.

CONCLUSION AND FUTURE WORK

This study presents the first real-world experimental benchmarking of post-quantum secure SHE schemes (BFV, BGV, and CKKS) for privacy-preserving data aggregation in ITS. Unlike prior works that relied primarily on simulation or theoretical constructions, we implemented and evaluated HE-based computations across realistic V2X communication setups using both wireless (Wi-Fi) and wired (Ethernet) networks. Two representative ITS use cases were tested: congestion detection using homomorphic addition, and average speed computation using both addition and multiplication (CKKS only). All schemes were implemented using the OpenFHE library and executed on real hardware with application-level measurements of end-to-end computational and communication delay.

Our results demonstrate that even with conservative parameters ensuring 128-bit post-quantum security, the tested HE schemes are practically feasible for latency-tolerant ITS applications. BFV supports sub-5-second updates and is well-suited for intersection-level applications like congestion monitoring. BGV, with moderate latency, enables regional aggregation tasks such as traffic density estimation on 10-30 second intervals. Although CKKS encounters higher overhead, it remains viable for minute-level analytics involving encrypted floating-point computations, such as eco-driving insights. However, even with 128-bit post-quantum security, the observed total end-to-end latencies (3 to over 30 seconds) indicate that these HE schemes are not currently viable for real-time safety-critical applications like collision avoidance, which require sub-second response times. These findings affirm the potential of HE to enable secure and privacy-preserving V2X data processing in future ITS deployments, provided the application latency requirements are aligned with the performance characteristics of each scheme.

Future work will explore optimizing HE parameter configurations and hardware acceleration (e.g., GPU or FPGA support) to reduce latency and improve scalability for larger vehicle sets. Real-world deployment trials using C-V2X radios and mobile edge platforms will help validate feasibility beyond lab conditions. Additionally, integrating HE with secure multi-party computation and differential privacy

may offer stronger end-to-end guarantees for collaborative ITS services without compromising performance or privacy.

ACKNOWLEDGMENTS

Note that ChatGPT was used solely for checking grammar and paraphrasing texts. No information, text, figure, or table has been generated, nor has any kind of analysis been conducted using any Large Language Model or Generative Artificial Intelligence.

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper: A. Mamun, K. Yates, A. Rakotondrafara, M. Chowdhury, R. Cartor, and S. Gao. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. Yoshizawa, T., D. Singelée, J. T. Muehlberg, S. Delbruel, A. Taherkordi, D. Hughes, and B. Preneel. A Survey of Security and Privacy Issues in V2X Communication Systems. *ACM Computing Surveys*, Vol. 55, No. 9, 2023, pp. 1–36. https://doi.org/10.1145/3558052.
2. 5G Automotive Association (5GAA). *Privacy by Design in V2X Communication Systems: Recommendation, Assessment and Implications*. 5GAA, 2021.
3. National Institute of Standards and Technology (US). *Advanced Encryption Standard (AES)*. Publication error: 197. National Institute of Standards and Technology (U.S.), Washington, D.C., 2001, p. error: 197.
4. National Institute of Standards and Technology. *Module-Lattice-Based Digital Signature Standard*. Publication FIPS 204. National Institute of Standards and Technology, Gaithersburg, MD, 2024, p. FIPS 204.
5. National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. Publication FIPS 203. National Institute of Standards and Technology, Gaithersburg, MD, 2024, p. FIPS 203.
6. Marcolla, C., V. Sucasas, M. Manzano, R. Bassoli, F. H. Fitzek, and N. Aaraj. Survey on Fully Homomorphic Encryption, Theory, and Applications. *Proceedings of the IEEE*, Vol. 110, No. 10, 2022, pp. 1572–1609.
7. Wang, G., Q. Zeng, L. Shen, S. Ding, X. He, Z. Zhai, H. Li, and Z. Shi. Towards Efficient Privacy-Preserving Keyword Search for Outsourced Data in Intelligent Transportation Systems. https://www.ssrn.com/abstract=5180330. Accessed July 23, 2025.
8. Palma, D., P. L. Montessoro, M. Loghi, and D. Casagrande. A Privacy-Preserving System for Confidential Carpooling Services Using Homomorphic Encryption. *Advanced Intelligent Systems*, Vol. 7, No. 5, 2025, p. 2400507. https://doi.org/10.1002/aisy.202400507.
9. Mi, B., J. Zhou, D. Huang, and Y. Weng. Privacy-Preserving Data Processing Method for IoV Based on Homomorphic Conjugacy Search Problem. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 25, No. 7, 2024, pp. 7374–7387. https://doi.org/10.1109/tits.2024.3351837.
10. Karim, H., and D. B. Rawat. TollsOnly Please—Homomorphic Encryption for Toll Transponder Privacy in Internet of Vehicles. *IEEE Internet of Things Journal*, Vol. 9, No. 4, 2022, pp. 2627–2636. https://doi.org/10.1109/jiot.2021.3056240.
11. Ameur, Y., and S. Bouzefrane. Enhancing Privacy in VANETs through Homomorphic Encryption in Machine Learning Applications. *Procedia Computer Science*, Vol. 238, 2024, pp. 151–158. https://doi.org/10.1016/j.procs.2024.06.010.
12. Fan, J., and F. Vercauteren. Somewhat Practical Fully Homomorphic Encryption. 2012.
13. Brakerski, Z., C. Gentry, and V. Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without Bootstrapping. *ACM Transactions on Computation Theory*, Vol. 6, No. 3, 2014, pp. 1–36. https://doi.org/10.1145/2633600.
14. Brakerski, Z., C. Gentry, and V. Vaikuntanathan. Fully Homomorphic Encryption without Bootstrapping. , 2011.
15. Cheon, J. H., A. Kim, M. Kim, and Y. Song. Homomorphic Encryption for Arithmetic of Approximate Numbers. 2016.
16. Lyubashevsky, V., C. Peikert, and O. Regev. On Ideal Lattices and Learning with Errors over Rings. *Journal of the ACM*, Vol. 60, No. 6, 2013, pp. 1–35. https://doi.org/10.1145/2535925.
17. Paillier, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology — EUROCRYPT '99* (J. Stern, ed.), Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 223–238.
18. Ogundoyin, S. O. An Anonymous and Privacy-preserving Scheme for Efficient Traffic Movement Analysis in Intelligent Transportation System. *SECURITY AND PRIVACY*, Vol. 1, No. 6, 2018. https://doi.org/10.1002/spy2.50.

19. Farokhi, F., I. Shames, and K. H. Johansson. Private Routing and Ride-sharing Using Homomorphic Encryption. *IET Cyber-Physical Systems: Theory & Applications*, Vol. 5, No. 4, 2020, pp. 311–320. https://doi.org/10.1049/iet-cps.2019.0042.

20. Ren, W., X. Tong, J. Du, N. Wang, S. C. Li, G. Min, Z. Zhao, and A. K. Bashir. Privacy-Preserving Using Homomorphic Encryption in Mobile IoT Systems. *Computer Communications*, Vol. 165, 2021, pp. 105–111. https://doi.org/10.1016/j.comcom.2020.10.022.

21. Sultan, A., S. Tahir, H. Tahir, T. Anwer, F. Khan, M. Rajarajan, and O. Rana. A Novel Image-Based Homomorphic Approach for Preserving the Privacy of Autonomous Vehicles Connected to the Cloud. *IEEE Transactions on Intelligent Transportation Systems*, 2022, pp. 1–13. https://doi.org/10.1109/TITS.2022.3219591.

22. Liu, Y., X. Xiao, F. Kong, H. Zhang, and J. Yu. Towards Efficient Privacy-Preserving Conjunctive Keywords Search over Encrypted Cloud Data. *Future Generation Computer Systems*, Vol. 166, 2025, p. 107716. https://doi.org/10.1016/j.future.2025.107716.

23. Lakhan, A., T.-M. Groenli, H. Wu, M. Younas, and G. Ghinea. A Novel Homomorphic Blockchain Scheme for Intelligent Transport Services in Fog/Cloud and IoT Networks. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 26, No. 2, 2025, pp. 1914–1929. https://doi.org/10.1109/TITS.2024.3493452.

24. Shor, P. W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. Presented at the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 1994.

25. Gao, S., and K. Yates. Leveled Homomorphic Encryption Schemes for Homomorphic Encryption Standard. , 2024.

26. Gao, S. Efficient Fully Homomorphic Encryption Scheme. , 2018.

27. Case, B. M., S. Gao, G. Hu, and Q. Xu. Fully Homomorphic Encryption with K-Bit Arithmetic Operations. , 2019.

28. Openfheorg/Openfhe-Development. OpenFHE, July 31, 2025.

29. Bossuat, J.-P., R. Cammarota, I. Chillotti, B. R. Curtis, W. Dai, H. Gong, E. Hales, D. Kim, B. Kumara, C. Lee, X. Lu, C. Maple, A. Pedrouzo-Ulloa, R. Player, Y. Polyakov, L. A. R. Lopez, Y. Song, and D. Yhee. Security Guidelines for Implementing Homomorphic Encryption. , 2024.

30. Creeger, M. The Rise of Fully Homomorphic Encryption: Often Called the Holy Grail of Cryptography, Commercial FHE Is Near. *Queue*, Vol. 20, No. 4, 2022, pp. 39–60. https://doi.org/10.1145/3561800.

31. Hannemann, A., and E. Buchmann. Is Homomorphic Encryption Feasible for Smart Mobility? Presented at the 18th Conference on Computer Science and Intelligence Systems, 2023.

32. Akkaya, K., V. Baboolal, N. Saputro, S. Uluagac, and H. Menouar. Privacy-Preserving Control of Video Transmissions for Drone-Based Intelligent Transportation Systems. Presented at the 2019 IEEE Conference on Communications and Network Security (CNS), Washington DC, DC, USA, 2019.

33. Boudguiga, A., O. Stan, A. Fazzat, H. Labiod, and P.-E. Clet. Privacy Preserving Services for Intelligent Transportation Systems with Homomorphic Encryption: Presented at the 7th International Conference on Information Systems Security and Privacy, Online Streaming, --- Select a Country ---, 2021.

34. Costantino, G., M. De Vincenzi, F. Martinelli, and I. Matteucci. A Privacy-Preserving Solution for Intelligent Transportation Systems: Private Driver DNA. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 1, 2023, pp. 258–273. https://doi.org/10.1109/TITS.2022.3217358.

35. Chen, J., K. Li, and P. S. Yu. Privacy-Preserving Deep Learning Model for Decentralized VANETs Using Fully Homomorphic Encryption and Blockchain. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 8, 2022, pp. 11633–11642. https://doi.org/10.1109/tits.2021.3105682.

36. Sun, X., F. R. Yu, P. Zhang, W. Xie, and X. Peng. A Survey on Secure Computation Based on Homomorphic Encryption in Vehicular Ad Hoc Networks. *Sensors*, Vol. 20, No. 15, 2020, p. 4253. https://doi.org/10.3390/s20154253.
37. Chillotti, I., N. Gama, M. Georgieva, and M. Izabachène. TFHE: Fast Fully Homomorphic Encryption over the Torus. , 2018.
38. Mohammed, {Mazin Abed}, A. Lakhan, {Karrar Hameed} Abdulkareem, {Dilovan Asaad} Zebari, J. Nedoma, R. Martinek, S. Kadry, and B. Garcia-Zapirain. Homomorphic Federated Learning Schemes Enabled Pedestrian and Vehicle Detection System. *Internet of Things (Netherlands)*, Vol. 23, 2023. https://doi.org/10.1016/j.iot.2023.100903.
39. Ying, Z., S. Cao, X. Liu, Z. Ma, J. Ma, and R. H. Deng. PrivacySignal: Privacy-Preserving Traffic Signal Control for Intelligent Transportation System. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 9, 2022, pp. 16290–16303. https://doi.org/10.1109/TITS.2022.3149600.
40. Wang, K., C.-M. Chen, M. Shojafar, Z. Tie, M. Alazab, and S. Kumari. AFFIRM: Provably Forward Privacy for Searchable Encryption in Cooperative Intelligent Transportation System. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 11, 2022, pp. 22607–22618. https://doi.org/10.1109/TITS.2022.3177899.
41. Pariota, L., L. D. Costanzo, A. Coppola, C. D'Aniello, and G. N. Bifulco. Green Light Optimal Speed Advisory: A C-ITS to Improve Mobility and Pollution. Presented at the 2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Genova, Italy, 2019.
42. Lonc, B., A. Aubry, H. Bakhti, M. Christofi, and H. A. Mehrez. Feasibility and Benchmarking of Post-Quantum Cryptography in the Cooperative ITS Ecosystem. Presented at the 2023 IEEE Vehicular Networking Conference (VNC), Istanbul, Turkiye, 2023.
43. Harounabadi, M., D. M. Soleymani, S. Bhadauria, M. Leyh, and E. Roth-Mandutz. V2X in 3GPP Standardization: NR Sidelink in Release-16 and Beyond. *IEEE Communications Standards Magazine*, Vol. 5, No. 1, 2021, pp. 12–21. https://doi.org/10.1109/MCOMSTD.001.2000070.
44. Wang, X., S. Mao, and M. X. Gong. An Overview of 3GPP Cellular Vehicle-to-Everything Standards. *GetMobile: Mobile Computing and Communications*, Vol. 21, No. 3, 2017, pp. 19–25. https://doi.org/10.1145/3161587.3161593.
45. Qualcomm Technologies, Inc. *Cellular-V2X Technology Overview, 80-PE732-63 Rev B*. Publication 80-PE732-63 Rev B. Qualcomm Technologies, Inc., 2019.
46. Karagiannis, G., O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions. *IEEE Communications Surveys & Tutorials*, Vol. 13, No. 4, 2011, pp. 584–616. https://doi.org/10.1109/SURV.2011.061411.00019.