Secure Intelligent Edge: A Lightweight Privacy-Preserving Scheme for Edge Computing Using Query-Adaptive Inner Product Functional Encryption

S Thejaswini[1]*, R Aparna[2]
Research Scholar, Department of Computer Science and Engineering[1]
Professor, Department of Information Science and Engineering[2]
Siddaganga Institute of Technology, Tumkur, Karnataka, India[1, 2]
Affiliated to Visveswaraya Technological University, Belagavi, Karnataka, India[1, 2]

***Abstract*—Recently, the digitalization era has led to a demand for secure digitalization transformation due to the *dataplexity* and *secureplexity* in modern applications; protecting sensitive data against unauthorized exploitation while extracting actionable intelligence has become a critical security necessity. In this context, Functional Encryption (FE) is used for privacy-preserving data analytics, and handling this at the edge exposes efficiency challenges and serious privacy risks. Existing FE schemes based on ECC, bilinear pairing, and IPFE incur high computational and communication complexities, which affect the overall efficiency of privacy-preserving data analytics in edge environments. To address these challenges, this study proposes a novel lightweight IPFE based scheme for *secure statistical* query processing over encrypted edge data, employs modular arithmetic to enable computation of query-specific inner product without revealing the underlying data, thereby supporting *selective computation* while preserving confidentiality. Query-dependent *feature selection* and *vector construction* are employed to reduce the dimensionality of the data involved in each query. The experimental results of the proposed scheme exhibit a computation efficiency of 99.98% and reduce the communication time by 99.06 % compared to existing approach while latency and throughput are analyzed under varying workloads. Thus, the overall analysis proved that the scheme is effective, efficient, scalable and suitable for high-performance edge server or intelligent edge server.**

Keywords*—Functional encryption; inner product functional encryption; lightweight; modular arithmetic; communication-computation efficiency; edge computing; privacy preserving

I. INTRODUCTION

The growing ubiquity of Internet of Things (IoT) infrastructures and ongoing digitalization services in modern applications face two main challenges such as dataplexity (data growth and analytical complexity) and secureplexity (computational demands of privacy preserving operations on sensitive data). In addition, in recent decades, many industries have externalized convoluted data to cloud-edge for storage and/or computation purpose to reduce maintenance cost. This digital transformation not only increases computational demands, but also intensifies security, privacy, and efficiency concerns. So, data must be encrypted prior to being externalized to maintain confidentiality [1]. Handling these concerns specifically in a resource-constrained environment, namely IoT environment and edge computing becomes challenging and leads to critical privacy risks [2-7]. These necessitate secure, efficient, and lightweight cryptographic mechanisms, particularly for edge computing ecosystem. In traditional cryptographic mechanisms (RSA, ECC, AES), confidentiality is provided strongly by encrypting and decrypting complete data. This follows an *all-or-nothing* access approach, which means that any authorized/unauthorized user who decrypts the data can gain access to complete original/raw information. This has led to several problems in modern applications such as privacy violations, exposure of sensitive personal information, possibility of insider attacks, increased risk of data misuse, regulatory non-compliance (GDPR, data protection laws) etc. However, in modern digital application systems, namely, analysis and monitoring services in banking, finance, healthcare systems, crime sectors, IoT etc., any authorized entities need only selectively access to data to compute the desired result, but not the complete original data. For example, A hospital stores all the comprehensive details of patients, including personal information such as Name, Address, Aadhaar number, Insurance amount, Salary details etc., for insurance claim purposes. However, an analyst/researcher only needs to know the average age of cancer patients, in which complete decryption of all patients' details is not required. Alternatively, a simple decision on the patient's disease and age is needed to determine the average age of cancer patients. To address the problems in traditional cryptographic mechanisms, there is a need for a cryptographic mechanism in which computation must be performed on encrypted data. Homomorphic encryption (HE), multi-party computation (MPC), and functional encryption (FE) are the three core mechanisms that support the computation of encrypted data. However, *HE* and *MPC* exhibit heavy computation and communication overhead on completely encrypted data and hence affect the overall performance of the privacy preserving mechanism in resource constrained environments, specifically for edge computing. Thus, motivated by these security and privacy requirements, this study focuses on an FE based scheme for edge computing ecosystem.

Functional Encryption [8-11] is an advanced public-key cryptographic primitive that permits a user possessing a functional decryption key to delve into only a specific function

* Corresponding author
Email IDs: thejaswinis@sit.ac.in, raparna@sit.ac.in

of encrypted data, without revealing the plaintext. Thus, FE provides fine grained access control for data access compared to HE and MPC mechanisms that support generalized data access control. For example, revisiting the aforementioned example of the average age of cancer patients' data, if an analyst/researcher wants to know the average age of cancer patients, sharing complete patient data is not feasible as it gives access to all the details of patients; instead, in such situations, a function can be effectuated for the analyst/researcher to only look at the patient's disease and age, giving the analyst/researcher the pertinent key and confine their access only to the desired information. Many studies have highlighted FE privacy preserving schemes, which are computational and communication intensive. Therefore, to provide a lightweight FE based privacy preserving scheme for edge computing, this study focuses on modular arithmetic based inner product functional encryption operations. The proposed scheme is lightweight, scalable, efficient, and suitable for edge computing ecosystems.

The main contributions of the proposed work are as follows:

- This study presents a query adaptive, modular arithmetic based IPFE privacy preserving approach in edge computing environment.
- Design of a novel Lightweight Query Adaptive Inner Product Functional Encryption (QA-IPFE) scheme using Modular Arithmetic. It implements secure statistical query processing over encrypted edge data using modular arithmetic to enable computation of query-specific inner products without revealing the underlying data. Query-dependent feature selection and vector construction are employed to reduce the dimensionality of the data involved in each query.
- The designed scheme enhances the system's performance by reducing computation complexity and communication time. Also, latency and throughput analysis was conducted across varying workloads to access the scalability and capability of proposed scheme.
- The scheme is tested using a large dataset, and the evaluation results validate that the proposed scheme is computation and communication efficient, making the scheme suitable for edge computing ecosystems.

The subsequent information is structured accordingly: Section II discusses the literature survey, Section III presents a novel scheme - the proposed work, Section IV presents the results and discussions by considering a large dataset and performance metrics, and finally, Section V highlights the conclusion.

II. LITERATURE SURVEY

The IPFE literature has explored a several approaches for encrypted functional evaluation, with emphasis on cryptographic construction, security, privacy, functionality, and practical implementation. Early studies mainly focused on developing simplified IPFE constructions, strengthening practical security, and providing additional privacy properties such as function privacy [13-15]. Subsequent studies explored practical improvements through updatable ciphertexts, optimized implementations, and function-hiding mechanisms [12], [17], [22]. Other studies extended IPFE toward data evaluation, decentralized architectures, and privacy-preserving federated learning [18], [19], [23]. These approaches provide different capabilities for performing computations over protected data, but the underlying cryptographic mechanisms can also introduce computational, communication, or representation overhead. Table I reveals the comparative review of existing IPFE based privacy preserving mechanisms across different computing environments.

Beyond these broader developments, attention has also extended to resource-constrained and edge computing environments. FE based privacy-preserving approach has been investigated for resource constrained IoT devices [16], with emphasis on improving the speed of computation. In edge computing, existing studies have investigated privacy-preserving approaches for smart healthcare on the Internet of Medical Things (IoMT) [20] and medical diagnosis [24], while [21] specifically addresses pairing-based IPFE scheme. These studies demonstrate the feasibility of protecting sensitive edge data, but their cryptographic operations can introduce computational and communication overhead, affecting latency and resource efficiency. Table II reveals the comparative review of existing edge-oriented studies.

Thus, the IPFE literature review systematically organizes and analysis existing Inner Product Functional Encryption (IPFE) schemes according to their cryptographic approaches, underlying assumptions, deployment environments, and reported findings.

1) Research gap: The state-of-the-art literature review reveals that majority of the security solutions discussed in the existing approaches rely on public key crypto mechanisms for function computationally intensive cryptographic constructions, such as bilinear pairing, fully homomorphic and lattice based IPFE schemes dominantly provide robust security services but often leads to increased computational complexity, larger ciphertext and key sizes, and higher communication overhead which affects the computational and communication efficiency of privacy-preserving data analytics in edge computing environments. Moreover, the prior works neither focus on system efficiency nor on security, neglecting the trade-off between the efficiency and security of the system. Limited research has explored Functional Encryption (FE) for edge-based data processing. In particular, the use of Inner Product Functional Encryption (IPFE) for privacy-preserving data processing in edge environments remains largely unexplored. This bottleneck and research gap highlights the need for a lightweight scheme and thus motivates the development of a lightweight, privacy-preserving, query-adaptive IPFE scheme that employs efficient modular arithmetic and query-driven features encoding to achieve function specific computation with computation efficiency of 99.98% and reduced communication time by 99.06% in edge computing system.

Thus, this study investigates a query adaptive, modular arithmetic IPFE-based light weight approach specifically for privacy-preserving computation in a resource-constrained edge environment.

TABLE I. COMPARATIVE REVIEW OF IPFE SCHEMES ACROSS DIFFERENT COMPUTING ENVIRONMENTS

Cryptographic Family Approach	Underlying Scheme	Ref	Deployment Environment	Key Findings & Observations	Key Limitations	Impact on Edge Computing
Lattice Based	LWE	[12]	Generic	Presents a scheme CUFE and generic io based construction for any function. CUFE allows data brokers securely update ciphertext access tags via one-way tokens without exposing plaintexts or requiring an always-online authority and provides post quantum resilience, dynamic access modification and mitigates key leakage.	lattice-based variant is only secure against adversaries bounded to a constant number of update token request, while the generic variant relies on indistinguishability obfuscation (iO), making it a theoretical feasibility result.	Large matrices/ vectors & computationally intensive lattice operations impose high computational, memory & communication overhead, particularly for resource-constrained edge nodes.
	RLWE	[17]	Multicore/ AVX2	Presents RLWE IPFE cryptographic library, accelerated via multi-core architectures, OpenMP & AVX2 vector instructions, and demonstrated its utility through a privacy-preserving biometric authentication protocol.	The polynomial multiplication bottleneck is addressed through implementation optimization, leaving the RLWE construction unchanged.	
Pairing Based	SXDH/ Asymmetric bilinear groups	[15]	Generic	Develops a nongeneric IPFE scheme providing full function privacy, so both message and function privacy are addressed efficiently.	Stronger privacy cryptographic construction involves more complex cryptographic operations than simpler inner-product FE constructions, performance reduced by 2*efficiency and storage loss.	Demands intensive scalar multiplication, mathematically large ciphertext impose high latency and cryptographic overhead particularly for resource-constrained edge nodes.
	SXDH	[18]	Cloud	Addressing security flaws in primal, privacy-preserving FHIPE framework reduces computation by 10× and communication costs by 2× compared to existing approaches.	System forces trade-off that increases ciphertext sizes (communication cost) and encryption latency (computation workload).	
		[19]	Decentralized	Dual-Mode Decentralized Multi-Client Functional Encryption (2DMCFE) provides strong security, high efficiency and with no accuracy trade off.	scheme requires a selective security model for its mathematical safety proof and users face a 4x slower key-generation setup due to heavy interactive verification steps.	
	asymmetric bilinear maps	[22]	Desktop/ Centralized	optimized secret-key FHIPE scheme that scales dual bases to cut key and ciphertext sizes in half down to n+1 group elements, providing a GGM security proof and a blueprint for single-key two-input functional encryption used in biometric iris-scanning and encrypted document searches.	Proving security in the Generic Group Model (GGM) means the system is only guaranteed safe against general algebraic attacks, leaving a potential vulnerability to non-generic attacks that exploit specific geometric flaws in an elliptic curve.	

	asymmetric Type-III pairing / DBDH	[23]	Cloud-Federated Learning	Dual-Mode Decentralized Multi-Client Functional Encryption (2DMCFE) provides strong security, high efficiency and with no accuracy trade off.	Setup Latency: Interactive zero-knowledge proofs and two-party computations with each authority make key generation four times slower, with costs scaling linearly with authority count and vector sizes.	
Discrete Logarithmic Based	ECC /ECDLP	[14]		IPFE scheme built on prime-order elliptic curve groups (ECC) that provides both data privacy and function privacy at the same time bypassing old theoretical limits to block all real-world generic attacks	The security proof only covers generic attacks (algebraic exploits), leaving it unproven against curve-specific math hacks	Computationally intensive modular exponentiation and group operations impose higher computation cost and latency specifically in edge network.
Multiple/ Alternative Assumption	DDH; LWE instantiation	[13]	Generic	Flexible framework for IPFE based on DDH and a generic construction that can alternatively be instantiated under LWE. They can be used to build future proof, quantum safe security system.	Security proof limited to selective model, restricted inner-product functionality to improve practicality, with different constructions offering various trade-offs between expressiveness and efficiency.	Computationally expensive modular exponentiation, pairing and group operations impose increased encryption, key-generation, and decryption latency; limited suitability for lightweight edge device

TABLE II. COMPARATIVE REVIEW OF PRIVACY PRESERVING MECHANISMS IN RESOURCE CONSTRAINED AND EDGE COMPUTING

Cryptographic Approach	Ref	Key Findings	Key Limitations	Impact on Edge Computing
Discrete Logarithmic Based - ECC /ECDLP	[16] IoT/ WSN	ECC-based lightweight IPFE scheme explicitly designed for IoT devices, proving that resource-starved devices can compute functional properties without excessive overhead.	System only works for **simple addition**, encryption is very slow on small chips (taking up to **12.3 seconds**), and the device crashes if it tries to handle more than 20 items at once due to low memory	Computationally intensive modular exponentiation and group operations impose higher computation cost and latency compared with lightweight modular arithmetic operations.
Pairing-based ECC (BLS12-381) - IPFE	[21]	Developed an optimized algorithm and tested on applications such as biomedical sensor data classification and wireless fingerprint-based indoor localization exhibit improvements in the performance of the system- achieved a 2.6× encryption and 3.4× decryption speedup on low-power devices.	The scheme's decryption is restricted to a small arithmetic output range because solving the underlying discrete logarithm requires a brute-force constant-time search that incurs high memory overhead as the system scales.	Demands intensive scalar multiplication, mathematically large ciphertext impose high latency and cryptographic overhead particularly for resource-constrained edge nodes.
Homomorphic Encryption	[24]	Scheme provides security with timely diagnosis of medical data and exhibits improvements in terms of efficiency and effective analysis of the proposed system.	The system only handles whole numbers, forcing it to round off decimals using a fixed multiplier which causes calculation errors,	High computation, memory & communication overhead, particularly for resource-constrained edge nodes.
	[20] Baseline Approach	Edge-Based Privacy Preserving Approach (EBPPA) focuses on integrity and privacy of IoMT data and performance comparison of the scheme shows significant improvements in terms of computation efficiency, communication overhead and less privacy leakage level.	Paillier homomorphic cryptosystem forces cause high computational latency, while Secret Sharing across nodes increases bandwidth overhead and creates a strict dependency on cloud resources.	
Modular Arithmetic - Modular matrix/vector operations	**Proposed QA-IPFE**	Lightweight query-based approach exhibits significant improvement in terms of computation and communication overhead. Also, latency and throughput analysis demonstrate favourable performance behaviour: capability and scalability.	Peak processing throughput is ultimately limited by computational overhead of large-integer modular arithmetic and the memory delays caused when moving these massive numbers through a small edge processor.	Lightweight design with reduced computation and communication overhead makes suitable for edge computing.

III. PROPOSED WORK

This study proposes a data privacy preserving scheme called QA-IPFE: Query Adaptive Inner Product Functional Encryption is a resource efficient, lightweight, and secure scheme that is tailored for the resource constrained edge computing environment. QA-IPFE is one of the variants of the IPFE scheme and is primarily based on modular arithmetic over a large prime field of length 128-bit. This scheme enables privacy-preserving computation of encrypted data.

A. System Model

The proposed work mainly consists of three components, and their significance is discussed in detail below:

1) Key Generation Authority (KGA): is a trusted setup authority responsible for performing the following operations:

- Define the security level parameter K of 128-bit.
- Generates a large prime P.
- Generates Public Key, which is a random invertible matrix A.
- Generates a Private Key, which is the inverse matrix of A, used to generate a function-specific key for a given query vector Y (FSK_Y).
- Generates the FSK_Y to securely compute the specific operations.
- Shares A with the user and FSK_Y to the Client.

2) Data Owner (DO): is an entity that owns the source of data. The responsibilities of the DO are as follows:

- Possesses and has full access to original data set/data source.
- Receives requests from the client via a query.
- Selects the appropriate dataset based on the query.
- Preprocessing the data set includes data cleaning and data transformation.
- Performs feature selection based on the query: Selection of required features of length *n* from the original data set to further process the query.
- Generate Encoded Feature vector: Represent the extracted features in encoded form.

$$X_i \in Z_P^n \quad (1)$$

where, $X_i = \{x_{i,1}, x_{i,2}, x_{i,3} \ \dots x_{i,n}\}$

N denotes the total number of records.

n denotes the number of selected attributes.

$x_{i,j}$ denotes the value of attribute j in record i.

- Defines a linear function/query vector based on query.

$$Y \in Z_P^n \text{ where } Y = \{y_1 y_2 y_3 \ \dots y_n\} \quad (2)$$

- Requests KGA (n, Y, client_id) to generate the security parameters.
- Receives a Public Key A and a large Prime P from KGA.
- Encrypts the encoded vector and Public Key to obtain a ciphertext (CT).

$$CT_i = Enc(A, X_i), \quad \{i = i \dots N \text{ records}\} \quad (3)$$

- Sends the $CT = \{CT_1, \dots \dots \ CT_N$ to the requested client.

3) Client: is an authorized data consumer whose responsibilities are as follows:

- Doesn't have access to the original data source.
- Specifies what computation is required in the form of a query to DA.
- Receives CT from DO.
- Requests and receives FSK_Y, from KGA.
- Performs function-specific computation on *CT* using FSK_Y to obtain Result.

The intended inner-product result for record is:

$$ri = Dec(FSK_Y, CT_i), \quad \{i = 1 \dots N\} \quad (4)$$

The aggregate result over all *N* records is defined as

$$Result = \sum_{j=1}^{N} r_i (mod\ P) \quad (5)$$

B. Proposed Architecture

In the QA-IPFE scheme initially, the query submitted by the client comprises the analytical requirements, which are subsequently processed and analyzed by the Data Owner (DO). Based on the input query, firstly, relevant features required for a specific computation on encrypted data are determined dynamically from the query. These features are encoded into numeric form and are called as query/function vector. Secondly, relevant features required for specific computation are also dynamically extracted from the dataset. Hence, the dataset of size *m* is reduced to size *n (n< m),* and *n* is called feature vector length. Since the feature vector and the corresponding query vector are dynamically determined based on the client's query, the scheme is named QA-IPFE. Then, the feature vector length is communicated to the Key Generation Authority (KGA), which generates the corresponding security parameters, namely the public key, private key, function specific key, and prime number. The DO generates the ciphertext by encrypting the encoded feature vector with the public key shared by the KGA and sending it to the relevant client. Upon receiving the ciphertext from the DO and the function-specific key from the KGA, function-specific computation is performed to obtain the required query result without revealing the actual data. Thus, in the decryption function, the computation is the inner product of the ciphertext and the function-specific key. The architecture of the proposed scheme is illustrated in Fig. 1.

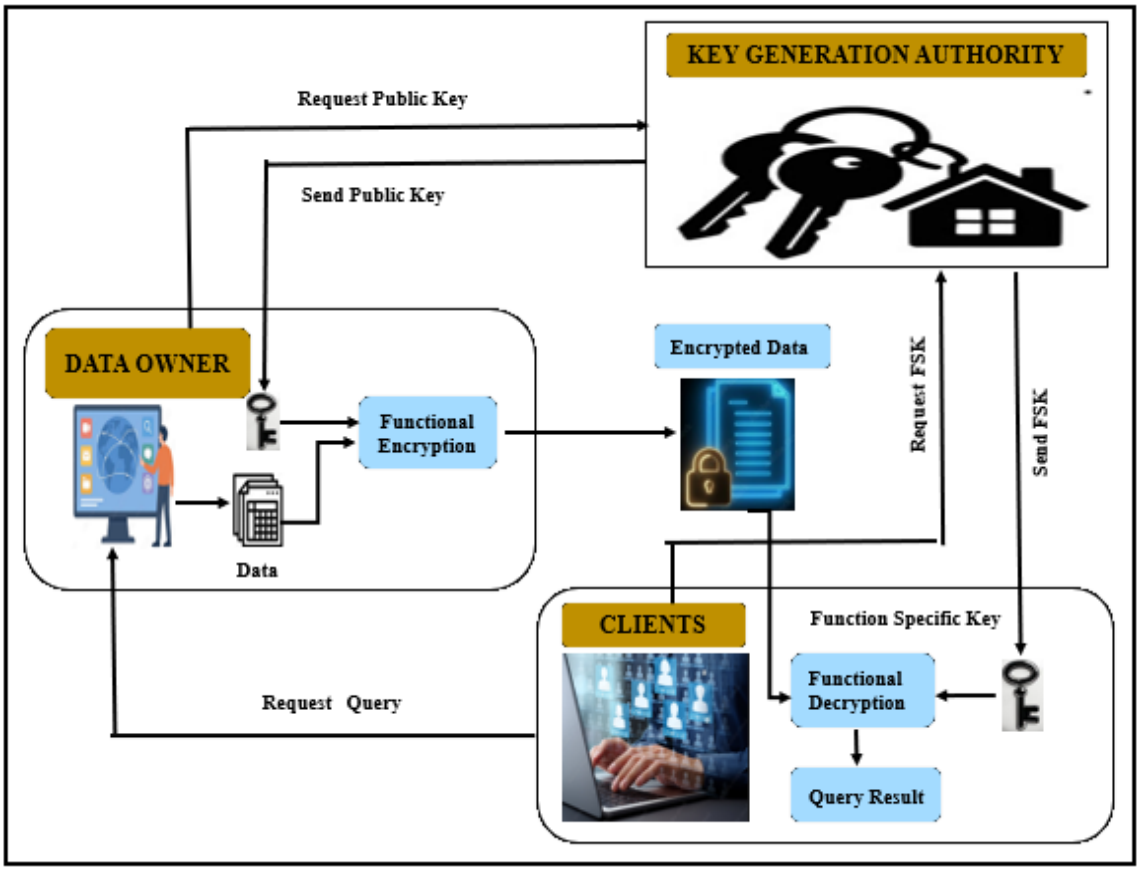

Fig. 1. Proposed architecture of QA-IPFE scheme.

Although the proposed approach avoids directly sharing the original plaintext data, access to the public key and prime number will not reveal the original records in plaintext form. Instead, the data is represented in encoded vector form. Thus, the plaintext is not available.

C. Algorithms

Initially, all the entities must register with the KGA and have secure communication channels to share the security parameters. For the proposed system, the security level parameter was set to 128 bits. The proposed architecture assumes that the Data Owner (DO), Key Generation Authority (KGA), and clients are authenticated and authorized to participate in the system. The KGA is considered a trusted authority. The overall workflow of the scheme consists of four phases:

PHASE 1: SETUP PHASE: (By KGA side)

KGA (Feature Vector Length n, Query Vector Y, Client ID)

Step 1: Define K=128-bit

Step 2: Generate large prime P = Gen_Prime(K) //Robin Miller algorithm.

Step 3: Compute Public Key A:

$$A \epsilon Z_P^{n*n} \; mod \; P \tag{6}$$

Step 4: Compute Private key A^{-1}:

$$A^{-1} mod \; P \tag{7}$$

return (A, P)

PHASE 2: DATA ENCRYPTION PHASE (By DO side)

Encrypt (Public Key A, Feature Vector X, Prime number P)

Compute Ciphertext: $CT = A \cdot X \; mod \; P$ (8)

return (CT)

PHASE 3: KEYGENERATION PHASE: (By KGA side)

FSK (Private Key A -1, Prime number P, Query Vector Y)

Compute Function Specific Key (FSK):

$$FSK = (A^{-1})^T \cdot Y \; mod \; P \tag{9}$$

return (FSK)

PHASE 4: DECRYPTION PHASE: (By Client side)

Decrypt (Function Specific Key FSK, Ciphertext CT Prime number P)

$$Result = CT^T \cdot FSK \; mod \; P \tag{10}$$

return (Result)

IV. Results and Discussions

This section presents the key findings of the proposed work and interprets its impact in the context of resource provisioning studies. As an important outcome of the proposed study, a summary of the data and insights that have emanated from the analysis is presented. In addition, the analysis paves the way for a comprehensive comparison of the proposed work with the baseline approach [20].

A. Experimental Setup and Key Findings

To date, this is the first work to apply query-adaptive, modular arithmetic based Inner Product Functional Encryption (IPFE) to this specific edge architecture. Therefore, the proposed approach is evaluated against a state-of-the-art Homomorphic Encryption (HE) baseline commonly deployed in privacy-preserving edge systems [20].

In the experimental setup, to validate the practical scalability of the proposed scheme, the experimental testbed was configured to simulate an intelligent edge server. Hardware execution was evaluated on an NVIDIA RTX 3080 GPU equipped with 32 GB of RAM, running on an Ubuntu 24.04LTS environment, and QA-IPFE is implemented using the Python programming language (version 3.9). The experimental test was conducted on a CRIME dataset of size 177 MB consisting of 743818 records and 28 features. The dataset was pre-processed, and executions were performed by considering several queries; based on the query, the required features were extracted to process the query with a query vector length ($l = 1,2,3$) and a security level of 128 bits.

B. Data Preprocessing

Data preprocessing is essential in the proposed IPFE scheme to ensure that the input data is consistent, valid, and represented in a form suitable for encrypted inner-product computation. Since IPFE operates on numerical vectors over a finite modular domain, missing, inconsistent, and non-standard data must be appropriately handled before vector construction. Hence, it enables the construction of valid feature vectors for mapping into the modular domain, which is essential for accurate and efficient IPFE computation. By retaining only relevant features, preprocessing also helps reduce computational and communication overhead. Therefore, the data preprocessing mechanism has been handled in the proposed scheme by dividing it into the following stages, as shown in Fig. 2:

a) Data cleaning: The columns with missing values were removed

b) Data transformation: Standardizes inconsistent and non-standard data representation into uniform formats. For Example:

- Date column values were inconsistent, with different date formats in the same column, such as 03-10-2024 and 05/10/2026. So, to maintain consistency, date formats were converted to one common date format.
- Gender of Victim: should contain F or M, but it also contains blank spaces and other characters, which are treated as non-standard values. To handles this, such column values were treated as Unknown(U).
- Since the dataset used in this study does not contain negative or fractional values, their encoding is not considered in the current implementation.

c) Feature selection: Query-dependent feature selection is adopted to retain only the attributes relevant to the intended queries, thereby reducing the feature-vector dimension, computational and communication overhead of the QA-IPFE scheme. In this step, the application-level query consists of words that represent the query condition and are mapped to the column names in the dataset. This tells the DO which features to use, and which records to select before encoding and encrypting.

For example: For a given query to obtain the "female crimes in Tumkur city during 2024", the CRIME dataset consists of several features. Based on the query conditions (Female, Tumkur, and 2024), query words are mapped to column names in the dataset (Gender, Area, Date of Occurrence) and are selected, neglecting the other irrelevant features. The dataset with selected features is used to filter the records based on the matching condition.

d) Feature vector construction: The query-relevant features are arranged in a predefined order to form a numerical vector suitable for modular representation and QA-IPFE computation.

For example: Reconsidering and continuing the workflow of the aforementioned example in the Feature Selection process, the dataset with selected features is used to filter the records based on matching the query condition.

For the above considered query, each record in the dataset is represented with three selected features in vector form: [Gender, Area, Year]. For a record with Gender=Female, Area=Tumkur and Year=2024, the corresponding query encoded vector is [1,1,1]. In case of mismatch, the corresponding feature(s) are indicated with zero. Suppose a record with Gender=Female, Area=Bangalore, and Year=2024; then the corresponding query-encoded vector will be [1,0,1].

Thus, the selected crime-data features are represented in binary numerical representations before encryption and after decryption to support computation over a finite field.

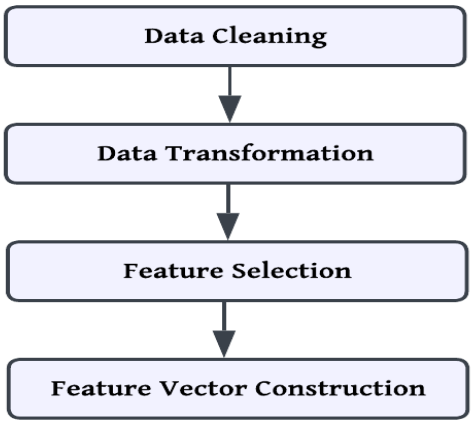

Fig. 2. Data processing steps of the proposed system.

C. Performance Analysis

The performance of the proposed work is analysed by considering two performance metrics: 1) Computation Efficiency 2) Communication Overhead 3) Latency Analysis and 4) Throughput Analysis.

Since comparable implementations of query-adaptive IPFE for edge computing were not identified in our literature review, an existing privacy-preserving scheme designed for an edge/IoMT environment [20] is selected as a reference for contextual performance comparison. Although the reference scheme employs homomorphic encryption rather than IPFE, both the approaches address privacy-preserving data processing in resource-constrained edge environments. The comparison provides context for discussing reported performance characteristics; however, differences in cryptographic construction and experimental settings limit direct quantitative conclusions.

a) Computational analysis: Computational efficiency is the number of computational resources required by the system to complete the task. Hence, to assess the computational efficiency of the proposed system, computational analysis was performed to quantify the execution time of the cryptographic operations, namely key generation, encryption, and decryption. Thus, the computational efficiency of the proposed system is computed as follows: Firstly, since all these operations directly contribute to secure IPFE computation, the time spent on QA-IPFE computation can be defined as:

$$T_{QA-IPFE} = T_{KG} + T_{Enc} + T_{Dec} \quad (11)$$

where,

$\boldsymbol{T_{KG}}$: Time required for different Key generation operations

$\boldsymbol{T_{Enc}}$: Time required for Encryption operation

$\boldsymbol{T_{Dec}}$: Time required for Decryption Operation

$\boldsymbol{T_{QA-IPFE}}$: Total time spent on cryptographic operations.

Secondly, the proposed scheme also involves communication between the entities that support the QA-IPFE computation but are not part of core cryptographic operations. The time spent on communication between the entities such as DO, KGA and Client to exchange data can be defined as:

$$T_{comm} = T_{send} + T_{receive} \quad (12)$$

where,

$\boldsymbol{T_{send}}$: time required to send data between entities

$\boldsymbol{T_{receive}}$: time required to receive data between entities

$\boldsymbol{T_{comm}}$: communication time required to send and receive data

between entities.

Finally, the computational efficiency (CE) of the proposed QA_IPFE scheme is given by:

$$CE(\%) = \frac{T_{QA-IPFE}}{T_{QA-IPFE} + T_{comm}} * 100 \quad (13)$$

A graph representing a computational analysis comparison between the proposed scheme and existing study is shown in Fig. 3. The proposed scheme exhibits a computational efficiency of 99.98% on average, comparable to the performance reported in [20]. This provides strong evidence that the proposed scheme is suitable for resource-constrained computing environments.

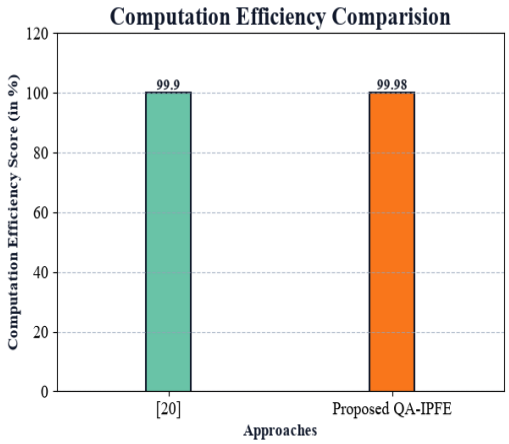

Fig. 3. Computation efficiency comparison.

b) Communication time analysis: In the proposed scheme, communication overhead refers to the amount of time involved in managing, coordinating, and transmitting information among participating entities, beyond the actual data payload. This can be a significant disadvantage when relying on continuous communication with the cloud or other network entities, thereby adding an extra burden to the network. Hence, reducing the communication overhead is essential for increasing the network efficiency and overall system performance. To address this issue, the proposed work employs a modular arithmetic-based IPFE scheme that offers significant benefits by minimizing communication overhead and exhibits a more efficient solution for addressing communication challenges compared to the baseline approach [20].

The communication overhead (TCO) in terms of time is the difference between the overall computation time and the communication time between the entities. Then overall computation time (T_{comp}) is defined as:

$$T_{comp} = T_{QA-IPFE} + T_{comm} \quad (14)$$

Thus, the proposed scheme incurs communication overhead (TCO), representing the time spent on communication between the participating entities and is defined as the difference between the overall computation time and the communication time between the entities.

$$T_{co} = T_{comp} - T_{QA-IPFE} \quad (15)$$

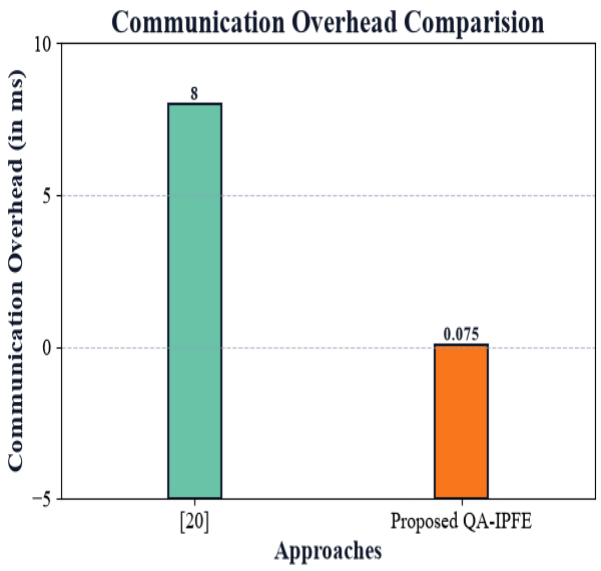

Fig. 4. Communication overhead comparison.

Fig. 4 shows reduced communication overhead comparison of the proposed scheme with the existing scheme. The communication overhead reduction (COR) is calculated as follows:

$$COR(\%) = \frac{T_{co_Baseline} - T_{co_QA-IPFE}}{T_{co_Baseline}} * 100 \qquad (16)$$

Hence, the communication overhead time is reduced from 8ms in the existing approach to 0.075ms in the proposed approach, resulting in a 99.06% reduction in communication overhead. This provides evidence that the proposed scheme is suitable for resource-constrained edge computing environments.

c) Latency analysis: The latency of the proposed IPFE scheme is measured as the end-to-end time between query submission and receipt of the computed result at the client. It therefore includes the computational and communication delays involved in query processing, IPFE computation, and result delivery and is given by:

$$Latency = T_{\boldsymbol{Result_Received_Client}} - T_{\boldsymbol{Querysubmission}} \qquad (17)$$

It was evaluated across different dataset sizes (records of sizes 1Lakh, 2Lakh, 4Lakh and 6Lakh,7.4Lakh) and vary query vector length while keeping fixed data size to assess the scalability of the proposed scheme. Since comparable latency measurements were not available for existing IPFE schemes in the considered setting, the evaluation focuses on the latency behavior of the proposed scheme under varying workloads. Thus, the experimental results shown in Fig. 5 and 6 indicate that the latency increases with the increase in the number of records and different query vector length respectively, as expected due to the higher processing and communication workload. However, the consistent upward trends indicate that the proposed scheme scales with increasing data volume in an edge environment. It also proves that QA_IPFE is computationally stable, handles data transmission efficiently without memory bottlenecks, and can reliably scale up to process high-density data profiles without unexpected performance drops.

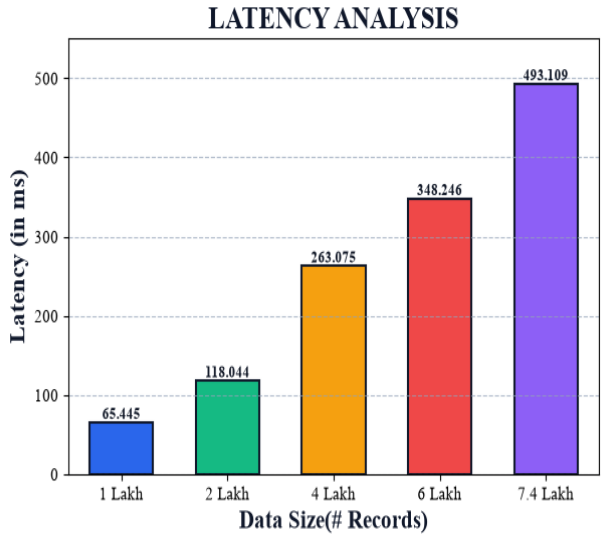

Fig. 5. Scalability of edge processing latency under varying data sizes.

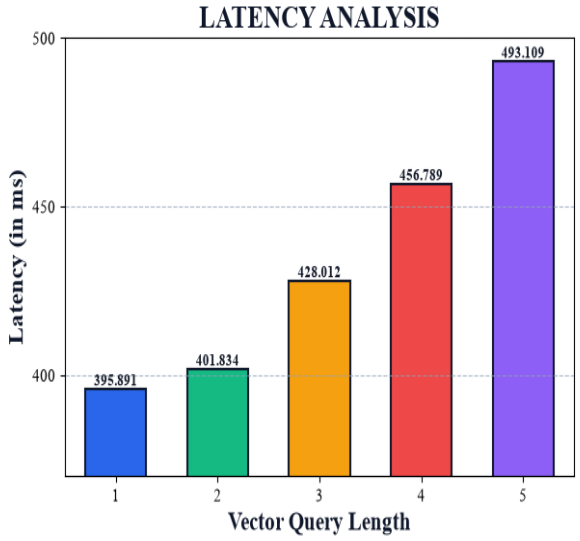

Fig. 6. Scalability of edge processing latency under varying query vector length.

D. Throughput Analysis

Throughput evaluates the amount of workload processed by the proposed QA-IPFE scheme within a given time. In this work, throughput is measured in terms of a) records per second (records/s) and b) bytes per second (bytes/s).

a) Record throughput (TpR) represents the number of records processed per second, and it is given by:

$$TpR = \frac{Number\ of\ matching\ records\ processed}{Total\ query\ processing\ time(s)} \qquad (18)$$

b) Byte throughput (TpB) represents the amount of data processed or communicated per second and is given by:

$$TpB = \frac{Number\ of\ bytes\ communicated}{Total\ query\ processing\ time\ (s)} \quad (19)$$

It was evaluated across different dataset sizes (records of sizes 1Lakh, 2Lakh, 4Lakh, 6 Lakh, and 7.4Lakh) to assess the ability of the proposed scheme. Thus, the experimental results are tabulated in Table III. Indicates that throughput values vary with data size due to changes in workload and corresponding processing or communication time. The record-based throughput reflects the data-processing capability, while the byte-based throughput represents the data-volume processing or communication capability. Overall, the results demonstrate the ability of the proposed scheme to process varying workloads within the measured query processing time.

TABLE III. SUMMARY OF THROUGHPUT ANALYSIS

Number of Records / Data size	Throughput	
	Records/s	bytes/s
7.4Lakh	3303.59	1554.87
6Lakh	1586.14	264.40
4Lakh	1228.12	350.15
2Lakh	986.34	780.69
1Lakh	0	1411.30

[a.] Note: If there are no matching records, then the Throughput record/s result will be zero.

1) Application scenario: The inner-product functionality has several practical applications [25]. For example, in descriptive statistics, the discipline of quantitatively describing the main features of a collection of information, such as calculating arithmetic means, moving averages, sums, and linear regression weights. This functionality is suited for applications where revealing the underlying data is unnecessary to achieve analytical utility; instead, the stakeholder is required to know only the result. Hence, the IPFE scheme can be directly applied to compute descriptive statistics in prominent sectors such as healthcare analytics, Crime analytics, financial analytics, traffic-urban analytics, smart grid utility telemetry, etc.

To demonstrate the practical efficiency of the proposed scheme QA-IPFE, a concrete and simple application scenario relevant to the modern-day use case of edge computing in the crime sector is discussed. The scheme focuses on executing descriptive statistical operations to compute the arithmetic sum over the encrypted crime data. Central Security Office (CSO) acts as a trusted authority that serves as the independent administrative core of the city's data network. The CSO is responsible for managing secret keys. Cyber Crime & Technical Division (CCTD) acts as a data owner who maintains the crime logs in the local edge server instead of uploading it to conventional centralized cloud repositories and Urban Planning Authority (UPA) acts as a client, wants to know the statistical analysis of crime happened in city/certain region (Example: total female crimes near central bus stop during the year 2024). This statistical data is used by UPA to physically alter the city layouts through CPTED (Crime Prevention Through Environmental Design), a strategy that uses architectural design to reduce the opportunity for crime. By analysing the statistical results, planners can successfully enforce: Design Safer Public Spaces, Optimize Street Lighting, and Place Transit Hubs Away from High-Risk Zones to prevent crimes. Upon request, the CCTD generates and distributes the secret keys. The UPA requests CCTD. In response, CCTD encrypts the relevant crime log and securely transfers it to UPA if UPA is an authorized user. UPA uses the function-specific key and the encrypted crime log to decrypt only the statistical insights, guaranteeing that sensitive underlying data vectors such as victim identities and active investigation records remain completely hidden from the UPA.

V. CONCLUSION

This research developed a lightweight inner product functional encryption scheme for privacy-preserving statistical query processing in an edge environment. The modular approach enables computation over encrypted data, thereby preserving complete data confidentiality even after decryption. Unlike conventional FE constructions that rely on elliptic curve cryptography, lattice based, bilinear pairing, the proposed scheme operates entirely over a prime field, thereby significantly reducing cryptographic complexity while preserving the core security guarantees of functional encryption. The experimental results and comparative analysis with existing studies, particularly in edge environments, confirm the efficiency and effectiveness of the proposed scheme by exhibiting a computation efficiency of 99.98% and a 99.06% reduction in communication overhead. Thus, the lightweight secure design of the proposed scheme inherently enforces suitability for edge computing ecosystems. Latency and throughput were further analysed across increasing workloads to assess the proposed system's behaviour and scalability. Also, the proposed system can be used in edge-based applications that require complete privacy preserving analytics over sensitive data, including finance, healthcare, crime investigation, law enforcement, auditing, and many more. As a future extension, the scheme can be extended with an increased security level, enabling real-time analytics and a wide range of applications such as operating queries on encrypted databases.

ABBREVIATIONS

Table IV gives the list of abbreviations used in this study.

TABLE IV. LIST OF ABBREVIATIONS

Abbreviations	Full Term
LWE	Learning With Errors
RLWE	Ring Learning With Errors
SXDH	Symmetric External Diffie-Hellman
DBDH	Decisional Bilinear Diffie-Hellman
ECC ECDLP	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
DDH	Decisional Diffie-Hellman

ACKNOWLEDGMENT

The authors acknowledge the support of institutional research facilities used for conducting this study.

REFERENCES

[1] J. Bacon, D. Eyers, T. F. J., M. Pasquier, J. Singh, I. Papagiannis and P. Pietzuch., "Information flow control for secure cloud computing," IEEE Transaction on Network Service Management, vol. 11, no. 1, pp. 76–89, Mar. 2014.

[2] Merlo, Tereza Raquel, Fariba Fard, and Suliman Hawamdeh, "Cloud Computing's Impact on the Digital Transformation of the Enterprise: A Mixed-Methods Approach", Sustainability 17, no. 13: 5755. https://doi.org/10.3390/su17135755

[3] Lavanya Jacintha Victor, "Cloud Computing: Revolutionizing Digital Transformation in the Modern Era", European Journal of Computer Science and Information Technology,13(18),12-23, 2025 Print ISSN: 2054-0957 (Print) Online ISSN: 2054-0965 (Online) Website: https://www.eajournals.org/

[4] G. Amudha, P. Gopika, S. G, V. R, M. G. Dinesh and S. S. R, "Cloud Computing: Transformations, Opportunities, and Challenges," 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2025, pp. 897-903, Doi: 10.1109/ICSSAS66150.2025.11080697.

[5] Karami, Amin & Karami, Mehdi, "Edge Computing in Big Data: Challenges and Benefits", International Journal of Data Science and Analytics, Volume 20, page no 6183-6226, 2025. Doi: 10.1007/s41060-025-00855-3.

[6] Thejaswini. S and R. Aparna, "Secure Edge: A Survey on Security Risks and Challenges in Edge Computing", 2025 International Conference on Emerging Computation and Information Technologies (ICECIT), Tumkur, India, 2025, pp. 1-8, doi: 10.1109/ICECIT67774.2025.11450956.

[7] Thejaswini. S and R. Aparna, "Edge Computing Unveiled: A Deep Dive into Latest Technologies, Challenges and Opportunities", 2025 Second International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS), Coimbatore, India, 2025, pp. 529-536, doi: 10.1109/ICC-ROBINS64345.2025.11085155.

[8] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: A new vision for public-key cryptography," Commun. ACM, vol. 55, no. 11, pp. 56–64, 2012.

[9] H. Wee, "Functional Encryption and Its Impact on Cryptography", Security and Cryptography for Networks: Lecture Notes in Computer Science, Vol 8642, Page No 318–323, Springer, 2014. https://doi.org/10.1007/978-3-319-10879-7_18

[10] Dupont, Pierre-Alain and Pointcheval, David, "Functional encryption with oblivious helper," in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Page no 205–214, 2017. https://doi.org/10.1145/3052973.3052996.

[11] D Boneh, A, Sahai, B Waters, "Functional Encryption: Definitions and Challenges", in Theory of Cryptography: Lecture Notes in Computer Science, vol 6597, Page no 253–273, Springer, Berlin, Heidelberg, 2011. https://doi.org/10.1007/978-3-642-19571-6_16.

[12] Valerio Cini, Sebastian Ramacher, Daniel Slamanig, Christoph Striecks, Erkan Tairi, "Inner-Product Functional Encryption with Updatable Ciphertexts", Journal of Cryptology, 2024, 37(1):8. https://doi.org/10.1007/s00145-023-09486-y.

[13] Abdalla, M., Bourse, F., De Caro, A., Pointcheval D, "Simple Functional Encryption Schemes for Inner Products", in Public-Key Cryptography - PKC 2015: Lecture Notes in Computer Science, vol 9020. Springer, Berlin, Heidelberg, 2015. https://doi.org/10.1007/978-3-662-46447-2_33.

[14] Agrawal, S., Agrawal, S., Badrinarayanan, S., Kumara Subramanian, A., Prabhakaran, M., Sahai, A, "On the Practical Security of Inner Product Functional Encryption. In: Katz, J. (eds) Public-Key Cryptography -- PKC 2015: Lecture Notes in Computer Science, vol 9020. Springer, Berlin, Heidelberg., 2015. https://doi.org/10.1007/978-3-662-46447-2_35.

[15] Datta, P., Dutta, R., Mukhopadhyay, S, "Functional Encryption for Inner Product with Full Function Privacy", in Public-Key Cryptography – PKC 2016: Lecture Notes in Computer Science, vol 9614. Springer, Berlin, Heidelberg, 2016. https://doi.org/10.1007/978-3-662-49384-7_7.

[16] Eugene Frimpong, Alexandros Bakas, Camille Foucault, Antonis Michalas, "Need for Speed: Leveraging the Power of Functional Encryption for Resource-Constrained Devices", Proceedings of the 9th International Conference on Internet of Things, Big Data and Security, (IoTBDS 2024), pages 62-74, ISSN 978-989-758-699-6, ISSN 2184-4976, France, 2024.

[17] S. Adhikary and A. Karmakar, "A Fast RLWE-Based IPFE Library and its Application to Privacy-Preserving Biometric Authentication," in IEEE Transactions on Emerging Topics in Computing, vol. 12, no. 1, pp. 344-356, Jan.-March 2024, doi: 10.1109/TETC.2023.3268003.

[18] X. Qian et al., "Privacy-Preserving Data Evaluation via Functional Encryption, Revisited", IEEE INFOCOM 2024 - IEEE Conference on Computer Communications, Vancouver, BC, Canada, 2024, pp. 11-20. Doi: 10.1109/INFOCOM52122.2024.10621262.

[19] J. Han, L. Chen, A. Hu, L. Chen and J. Li, "Privacy-Preserving Decentralized Functional Encryption for Inner Product," in IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 4, pp. 1680-1694, July-Aug. 2024. Doi: 10.1109/TDSC.2023.3288109

[20] Meng, L., Li, D, "Novel Edge Computing-Based Privacy-Preserving Approach for Smart Healthcare Systems on the Internet of Medical Things", Journal of Grid Computing 21, 66, 2023. https://doi.org/10.1007/s10723-023-09695-6

[21] Utsav Banerjee, "Privacy-Preserving Edge Computing from Pairing-Based Inner Product Functional Encryption", IEEE Global Communications Conference: Selected Areas in Communications: Cloud/edge Computing, Networking, and Data Storage, DOI: 10.1109/GLOBECOM54140.2023.10436785, 2023 IEEE Global Communications Conference | 979-8-3503-1090-0/23.

[22] S. Kim et al., "Function-Hiding Inner Product Encryption Is Practical," in Security and Cryptography for Networks (SCN), 2018, pp. 544–562.

[23] Y. Chang, K. Zhang, J. Gong and H. Qian, "Privacy-Preserving Federated Learning via Functional Encryption, Revisited", in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 1855-1869, 2023. Doi: 10.1109/TIFS.2023.3255171.

[24] Zhuo ran Ma, Jianfeng Ma et al., "Lightweight Privacy-Preserving Medical Diagnosis in Edge Computing", in IEEE Transactions on Services Computing, vol. 15, no. 3, pp. 1606-1618, 1 May-June 2022, Doi: 10.1109/TSC.2020.3004627.

[25] J. Lee, D. Kim, D. Kim,Y. Song, J. Shin, J.H. Cheon, Instant Privacy-preserving biometric authentication for hamming distance. Cryptology ePrint Archive, Paper 2018/1214 (2018). https://eprint.iacr.org/2018/.